DDOS ATTACK DETECTION USING HYBRID (CCN AND LSTM) ML MODEL

Thura Jabbar Khaleel [1]
[1] Informatics Institute for Postgraduate Studies
Iraqi Commission for Computers and Information
Baghdad , Iraq
Thurajk@yahoo.com

Nadia Adnan Shiltagh
[2]College of Engineering
University of Baghdad
Baghdad, Iraq
nadia.aljamali@coeng.uobaghdad.edu.iq

Abstract **- LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Networks) are two types of deep learning algorithms; by combining the strengths of LSTM and CNN, researchers have developed deep learning models that can effectively detect SDN (Software-Defined Network) attacks including Distributed Denial of Service. These models effectively analyze network traffic, encompassing temporal and spatial characteristics, resulting in precise identification of malicious traffic.In this research, a hybrid model composed of CNN and LSTM is used to detect the DDoS attack in SDN network. Where the CNN component of the model can identify spatial patterns in network traffic, such as the characteristics of individual packets, while the LSTM component can capture temporal patterns in traffic over time, such as the timing and frequency of traffic bursts. The proposed model has been trained on a labeled network traffic dataset, with one class representing normal traffic and another class representing DDoS attack traffic. During the training process, the model adjusts its weights and biases to minimize the difference between its predicted output and the actual output for each input sample. Once trained, the hybrid model classifies incoming network traffic in the dataset as either normal or malicious with an initial accuracy of (78.18%) and losses of (39.77%) at the 1st epoch till it reaches an accuracy of (99.99%) with losses of (9.29×10-5) at the epoch number 500. It should be mentioned that the hybrid model of CNN and LSTM for DDoS detection is implemented using Python Anaconda platform with an ETA 28ms/step.**

Index Terms - DDoS, Cyber-attack, ML, SDN, LSTM, Python.

I. INTRODUCTION

In the present era, the world has transformed into a closely connected global village. This shift can be attributed to the seamless flow of communication, ease of data sharing, and the pervasive computerization of various aspects of life. As a result, information is abundantly available all around us. In this context, information systems have emerged as a central component in enterprises, regardless of their size or operational domain. Malware attacks have increased recently that used to attack information systems; the nature of malware attacks has also dramatically changed as sophisticated attacks have become ubiquitous. The sophistication and complexity of malware have manifested in miscellaneous ways; Different enrollments are subject to malicious attacks using a variety of techniques, including fuzzing, denial of service, Distributed Port scanning, probing, and Denial-of-Service (DDoS). These assaults may pose a threat to the transport and application layers or to other protocols. like file transfer protocol, internet control message protocol, user datagram protocol, transmission control protocol, simple mail transfer protocol, hypertext transfer protocol, etc. To treat with such attacks, network-based intrusion detection systems could be used through network scanning and detecting them. In this situation, The difficulty of cyber security has grown, and research in this field is expanding. Different procedures and technologies are created to provide a level of safety that can detect and prevent these threats [1, 2]

For malware classification, the machine learning field has been promised. When there is a lot of data available, deep learning, a branch of machine learning, enables achieving artificial neural network (ANN) models produce better outcomes than conventional methods. Deep learning (DL), a subfield of artificial intelligence, is currently widely used in the classification and pattern recognition industries. Initially, deep learning networks were primarily utilized for image classification. However, over time, the application of deep learning has extended to various other domains and disciplines, encompassing a wide range of areas. Nevertheless, to achieve this purpose, dealing with different characteristics present in each domain is needed. Deep learning often uses several information-processing layers inside a hierarchical structure. Additionally, deep learning has emerged as a prominent area of research, leading to the exploration and publication of numerous architectures of deep neural networks (DNNs). These include Deep belief nets (DBNs), recurrent neural networks (RNNs), convolutional neural networks (CNNs), and various types of autoencoders such as Autoencoders that are variational, contractive, sparse, and denoising. Among the different deep learning algorithms, CNN and LSTMs (long short-term memory neural networks, which is a type of RNN), is efficient at doing complicated tasks like running self-driving cars and recognizing faces and objects. As a result, several academics have tried to use CNN and LSTM to identify various forms of assaults. [2]

SDN, or software-defined networking is a promised network model it provides excellent scalability, controllability, adaptability, and manageability. Despite having tremendous potential characteristics, intrinsically, SDN lacks security. Since it has been suffering from DDoS attacks so far, it is one of the primary dangers to Internet accessibility [2].

According to the statistics, cyber-attacks had been rated to be the fifth top risks in 2020; the global yearly cost of cybercrimes is valued to be six trillion per a year; while Cisco statistics estimates that in 2023 the DDoS (Distributed Denial of Service) assaults will raise to 15.4 million. Identifying malware and cyber-attacks has always been a difficult task. Companies and governments have put a lot of effort and money towards reducing the effect of these dangers. Anti-malware tools are frequently unsuccessful since new malware does not have a signature in the anti-malware database [11]. This research has proposed a DDoS cyber-attacks detection from scratch using deep learning techniques in order to establish a solution for the previously mentioned problems. It depends on the architecture of CNN and LSTM to detect the SDN network under DDoS assault. The remainder of the essay is structured as follows: Section II demonstrates the literature survey of related works. Presented in Section III the structure of the hybrid CNN and LSTM model. Section IV presents the deep learning frameworks. The proposed system modelling is discussed in Section V along with its subsections. The outcomes are covered in Section VI. Finally, Section VII shows the study's conclusions.

II. LITERATURE SURVEY

difficulties with detecting threat from within are studied, and other machine learning-based solutions have been put out. The previous methods of detection, such depend on Because of the features of the data, such as the difficulty in labeling insider threat, complexity, and dealing with a diversity of data in a single feature, feature engineering is time-consuming and difficult to distinguish between normal and abnormal behavior. [11]. To set a new norm for such data, deep learning is presented. for deciphering comprehensive models from extremely complicated data. DL is a powerful tool for analyzing user activity and spotting harmful conduct. Newly, in-depth learning techniques like recurrent nervous system (RNN), convolutional nervous system (CNN), and graph nervous system (GNN), are techniques which have been suggested for detecting danger from inside. Similarly, many investigation of Deep Feed-forward Network of Neurons

are completed, like Deep Belief Network (DBN), Deep Boltzmann Machine (DBM), and deep auto-encoders. The deep auto-encoders include both of encoder and decoder. Encoder is used to encode the input data to stash what was presented in the process of reconstructing the input data by the decoder. The essential goal of the deep encoders is to make reconstructed data close to original data as possible.

In [1], authors had discussed the cybersecurity challenges and illustrated the awareness about cybersecurity events that were successfully used in industrial environments. They quickly outline cybersecurity issues and provide a cutting-edge infrastructure that enables these events to be held online. The authors offer Sifu, a platform for cybersecurity awareness that automatically evaluates problems in accordance with secure coding rules and utilizes artificial intelligence to give players solution-guiding cues. The Sifu platform also enables remote (online) learning in times of social estrangement as a result of its qualities. Four online real-world Cybersecurity Challenges events allowed for the evaluation of the Sifu platform-based Cybersecurity Challenges events. They provide the results of three polls demonstrating that the CSC events on the Sifu platform are sufficient to increase industrial software engineers' knowledge of safe coding.

In [2], According to the writers a hybrid-unsupervised deep learning technique using the stack auto-encoder and support vector machine for one named (SAE-1SVM) for detection the (DDoS) attack. Their empirical results showed that their proposed algorithm could achieve 99.35% of average accuracy with a minor set of flow features. The (SAE-1SVM) reveals a significant reduction in processing time without compromising on the high detection rate.

Abdurrahman Pektaş and Tankut Acarman had proposed a in-depth learning-based approach to identifying network intrusions using characteristics based on flow. Their work is addressed in [3]; they examine the impact of flow status interval, convolution filter size, flow window size, and long short-term memory units for the detection performance regarding level in statistical metric values. The proposed flow-based intrusion approach outperforms other available methods in terms of performance. It detected abnormal traffic with an accuracy of 99.09%, and had a false alarm rate of 0.0227.

In [3], the researchers reviewed the different kinds and purposes of the cyber-attacks, and they discussed the methods to prevent these attacks and reduce their damages. Its objective of their investigation was to study the difficulties, shortcomings, and strengths of the suggested approaches as well as to scan and thoroughly analyze the standard advancements made in the field of cyber security. The several new descendant assaults are examined in depth. The history of early-generation cyber-security techniques are explored together with standard security frameworks. Additionally, new breakthroughs, security concerns, dangers, and emerging trends in cyber security are discussed. It is anticipated that the thorough review study offered to information technology and cyber security research would be helpful.

In [4], researchers used deep learning for identifying sluggish DDoS assaults in networks with SDN. They proposed their system using an amalgamated "Convolutional Network of Neurons-Long-Short Term Memory" (CNN-LSTM) approach to recognize sluggish DDoS assaults in networks due to SDN-based networks. The effectiveness of this approach was assessed using standard datasets. They obtained an impressive result quietly when all measured metrics for performance over 99%. Their hybrid CNN-LSTM approach also exceeds, in terms of performance, other deep learning approaches, such as Multi-Layer Perceptron (MLP) and typical machine learning models like 1-Class Support Vector Machines (1-Class SVM).

In [5], researchers presented an investigation about in-depth learning applications when a DDoS assault is discovered on SDN controllers and their work is addressed as [8]. Their paper evinces it is (RNN LSTM) an applicable algorithm for in-depth learning could be used to detect DDoS and its mitigation in the SDN controller. According to their obtained results, it was clear changing the split ratio between the training and test datasets might have varied outcomes for the performance of the deep learning algorithm. So, Their concluded remarks stated that RNN-LSTM is also a good model for the mitigation and identification of DDoS attacks in SDN networks.

III. Hybrid (CCN and LSTM) Model for DDoS Detection

Generally, denial-of-service attacks are launched using homebrewed scripts or DoS tools like Low Orbit Ion Canon, while DDoS attacks are launched from botnets to large clusters of connected devices like cellphones, PCs, or routers. These compromised devices can be infected with malware, enabling attackers to remotely control them and initiate large-scale attacks. The main security issues within the SDN are problems illegal access to the controller, man-in-the-middle assault, intrusions, and a packet-modifying update to the flow rule. A similar issue is the controller being taken over by malicious transmissions., controller communication flood, switch denying service, and difficulty in setup. DDoS is among the dreadful most frequent dangers that are successfully combatted prevent the controller from receiving regular traffic. The attacker makes the controller unusable by bombarding it with more malicious packets than it can handle.. By leveraging several hacked switches (bot) to create malicious packets, the assault is made possible. As shown in Fig. (1), the attacker creates a botnet, or a group of bots, from the switches linked to the controller and then seizes control of the whole network to continue operating after rendering the controller unusable. [7

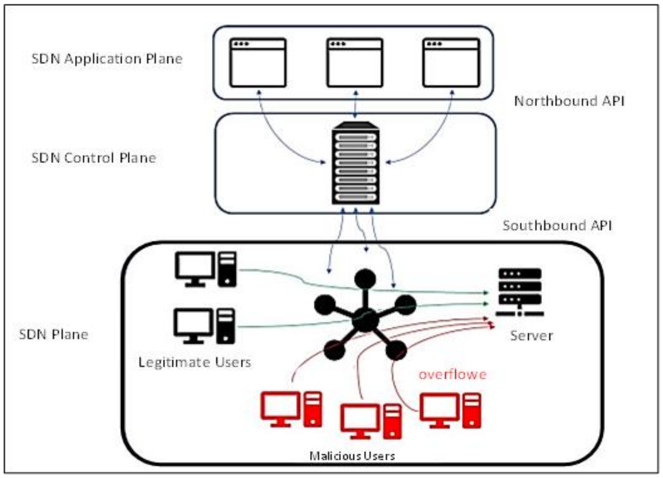

Fig 1: DDoS attack on SDN network [4].

Generally, the CNN-LSTM is used for image labeling, video labeling, and activity recognition. Their common features have been developed to apply visual time series prediction problems and generate text annotations from image sequences. Fig. (2) shows the basic structure of CNN-LSTM with the input layer, visual feature extraction, sequence learning, and output layer, respectively [6]

Pooling layers are inserted between layers to speed up computation and gradually increase compositional and spatial invariance. Long-term temporary dependence traits are detectable by LSTM. The hybrid model eventually learns a superior regression model, and the fully connected neural network excels at mapping for DDoS attack prediction. [8]

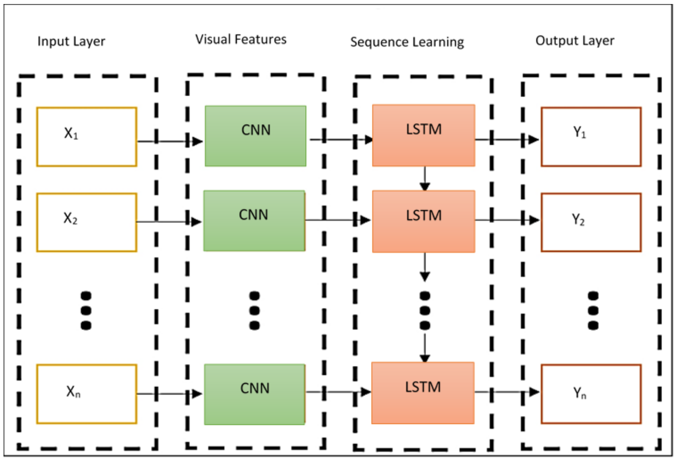

Fig 2: Hybrid CNN-LSTM Model.

IV. DEEP LEARNING METHODS AND FRAMEWORKS

There are several powerful techniques that could be applied to the algorithms of deep learning to optimize the model and reduce the training time. Some of these techniques are listed below and Table (1) shows the merits and demerits of each technique.

- ❖ Backpropagation
- ❖ Stochastic Gradient Descent
- ❖ Max Pooling
- ❖ Learning Rate Decay
- ❖ Dropout
- ❖ Transfer Lea

TABLE I

COMPARISON OF DEEP LEARNING METHODS [13].

Method	**Description**	**Merits**	**Demerits**
Backpropagation	Used in optimization problem	For the calculation of gradient	Sensitive to noisy data
Stochastic Gradient Descent	To find optimal minimum in optimization problems	Avoids trapping in local minimum	Longer convergence time, computationally expensive
Max Pooling	Applies a max filter	Reduces dimension and computational cost	Considers only the maximum element which may lead to unacceptable result in some cases
Learning Rate Decay	Reduce learning rate gradually	Increases performance and reduces training time	Computationally expensive
Dropout	Drops out units/ connection during training	Avoids overfitting	Increases number of iterations required to converge
Transfer Learning	Knowledge of first model is transferred to second problem	Enhances performance, rapid progress in training of second problem	Works with similar problems only

A deep learning framework helps model a network more quickly without going into the details of the underlying algorithms.

TABLE II

COMPARISON OF DEEP LEARNING FRAMEWORKS [9].

Deep learning framework	Release year	Language written in	CUDA supported	Pre-trained models
TensorFlow	2015	C++ and Python	yes	Yes
Keras	2015	Python	yes	Yes
PyTorch	2016	Python and C	yes	Yes
Caffe	2013	C++	yes	Yes
Deeplearning4j	2014	C++ and Java	yes	Yes

V. PROPOSED SYSTEM MODELLING

The proposed DDoS detection system deals with a meta-data dataset of the randomly spoofed DoS attacks deduced from the backscatter packets gathered by the UCSD Network Telescope between March 1, 2015, and February 28, 2017. It is an SDN dataset and is used for traffic classification by deep learning. The proposed system was designed and implemented on a SDN network, where the data inside the dataset perform features of the SDN network. Fig. (3) demonstrates the system structure. The deep learning technique is the core of this system, as it is used to classify the SDN features as legitimate traffic or DDoS attack, since the dataset includes both DDoS attack and normal traffic. Hybrid CNN-LSTM network model is used which performs the artificial neural network on which the deep learning is implemented. It should be mentioned that the system is programmed using Anaconda Python with Jupiter Notebook

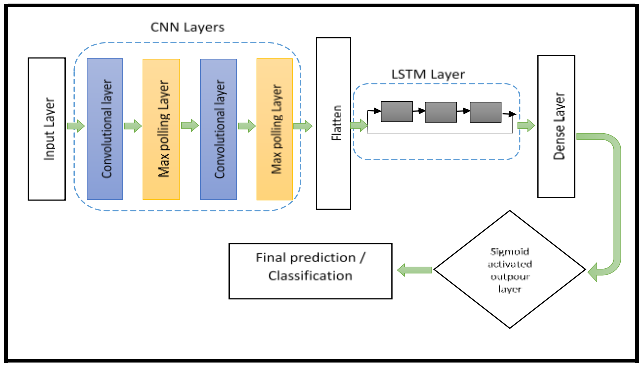

Fig 3: Proposed System Model.

I. Data Pre-processing and Feature Extraction

The dataset used in this system is taken from Mendeley datasets available on https://data.mendeley.com/datasets/jxpfjc64kr/1 website [15]. It is generated by the mininet emulator as a SDN traffic dataset containing both normal and malicious attack traffics. SDN network simulation was run for 250 minutes, and 104345 rows of data were acquired. In this system, 10000 rows were selected from the dataset for the deep learning stage.

Number of Packet_ins messages

In this proposed system, the dataset must be prepared first before performing deep learning processes. Logistic regression is the core of the pre-processing. Before applying the logistic regression, the dataset was checked if it contained any NULL values; this is done using the 'dropna' and 'isna' methods which are called from pandas library. Then, all dataset's rows with any NULL value and any missing values will be dropped, and the data is ready for further analysis and modeling. Logistic regression comes then. For the dataset, logistic regression predicts the probability of dataset's values and the results were It calculates the optimal coefficients for each feature to maximize the likelihood of the observed target values.

Logistic regression is performed using LogisticRegression method from sklearn library using the ‘liblinear’ as a ‘solver’ parameter which specifies the algorithm to be used for optimization. ‘Liblinear’ solver is suitable for small datasets and can handle L1 and L2 regularization. The ‘multi-class’ parameter is set to ‘ovr’ which stands for “one-vs-rest” this means that the logistic regression model will be trained using the “one-vs-rest” approach where each class is treated as a binary classification problem. Relating to the logistic regression, the method code value counts() is called for the pandas DataFrame ‘df’, which returns the unique value count in the ‘port_no’ column. It gives a series of objects containing counts of unique values. The output will show the number of times each unique value appears in the ‘port_no’ column. It resulted in 5 unique values for ‘port_no’ -listed in Table (2) below, with their corresponding counts- which give an insight into the distribution of network traffic across different ports in the dataset.

TABLE III
PORT NUMBERS OF PANDAS DATA FRAME.

Port number	Count of appearance in data frame
2	29148
1	29139
3	28413
4	15637
5	1502

For deep learning process, the pandas data frame drops some columns (‘dt’, ‘switch’, ‘src’, ‘dst’, ‘port_no) using the drop() method. Then, it uses one-hot encoding to convert the categorical features into binary features using the get_dummies() method. Later, the data is scaled using the MinMaxScaler() method and assigns the scaled data to a new data frame ‘new_df’. The label column is then renamed to ‘class’ and the original ‘label’ column is deleted; and the data is separated into input features X and target variable Y.

Then splitting stage is performed where the data is separated into training and testing sets using the train_test_split() method; the training rate is 80% and the testing rate is 20% of the total data. The stratify parameter is used to ensure that the distribution of the target variable is approximately the same in the training and testing sets. The overall dataset pre-processing is illustrated in a flowchart in Fig. (4). The split data of 80% for training and 20% for test (x_train and x_test) variables with ((83468, 18), (20868, 18)); this means that x_train has 83468 rows and 18 columns, while x_test has 20868 rows and 18 columns

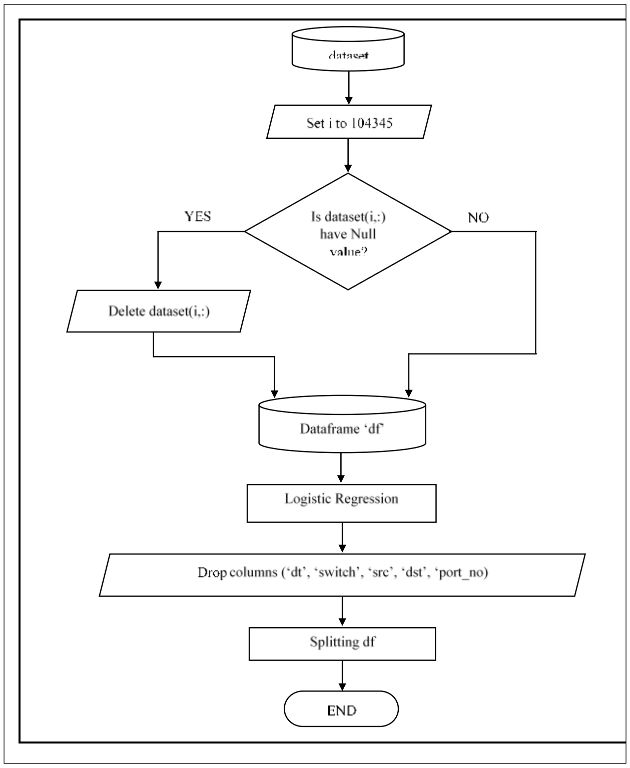

Fig 4: Dataset pre-processing.

II. Data Training

The proposed model is based on utilizing two deep neural networks (DNN) algorithms. The first is the Convolutional Neural network (CNN) that consists of multilayers, including: dimensional convolutional layer, max pooling layer, two LSTM layer, and the output layer where dense layer was. While the second algorithm is the LSTM layers of this model. Particularly, our approach specifically utilizes a hybrid combination of two deep learning models, namely (CNN) and (LSTM). In addition, we incorporate the checkpoint network, a widely popular technique currently gaining momentum in the deep learning domain. The model consists of 1 input layer, 1 convolutional layer (Conv 1D), doble LSTM layers, one max poling layer and one dense layer (Dense) in output. Table (3) shows the training model.

TABLE IV
LSTM AND CNN MODEL’S PARAMETERS.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 18, 1)]	0
conv1d (Conv1D)	(None, 18, 64)	256
max_pooling1d (MaxPooling1D)	(None, 9, 64)	0
lstm_1 (LSTM)	(None, 9, 128)	98816

lstm_2 (LSTM)	(None, 128)	131584
dense_1 (Dense)	(None, 1)	129

Two types of activation functions were used in DNN model, rectified linear 'ReLU' for the CNN and 'sigmoid' for the dense layer. Equations (1) and (2) show the mathematical expression of 'ReLU' and 'sigmoid' activation functions, respectively.

$$\boldsymbol{f(x) = \max(0, x)} \text{ ………… Eq. (1)}$$

$$\boldsymbol{f(x) = \frac{1}{1+e^{x}}} \text{ ………… Eq. (2)}$$

The setting of the DNN (also shown in auto-encoder setting later) is as follows:

1. The input is a feature which is a tensor of (18, 1).
2. The first layer is a 1D convolutional layer with 64 filters, kernel size of 3, a 'same' padding, and a 'ReLU' activation function.
3. The second layer is a 1D max pooling layer with a pool size of 2 and a 'same' padding.
4. The third layer is the LSTM layer with 128 units and returns sequences.
5. The fourth layer is another LSTM layer with 128 units and no return sequences.
6. The output layer is a dense layer with one unit and a 'sigmoid' activation function.

It is important to note that the CNN and LSTM models were interconnected using the auto-encoder process. This unsupervised neural network technique is employed to train the network in such a way that the outputs are closely match the input vectors. This approach allows for enhanced learning and representation of the data. It might be used to create representations of input data in higher or lower dimensions. Neural networks are incredibly adaptable due to the utilization of unsupervised learning of compressed data encoding. The amount of computing resources required to create an efficient model is reduced further by the fact that such networks may be trained one layer at a time.

As illustrated in Fig. (5), if the hidden levels are less dimensional than the input and output layers, the network will be employed for data encoding since it supports compression.

A stacked auto-encoder is produced by training multilayered auto-encoders in sequence, which enables incremental information reduction.

model is reduced further by the fact that such networks may be trained one layer at a time.
As illustrated in Fig. (5), if the hidden levels are less dimensional than the input and output layers, the network will be employed for data encoding since it supports compression.
A stacked auto-encoder is produced by training multilayered auto-encoders in sequence, which enables incremental information reduction.

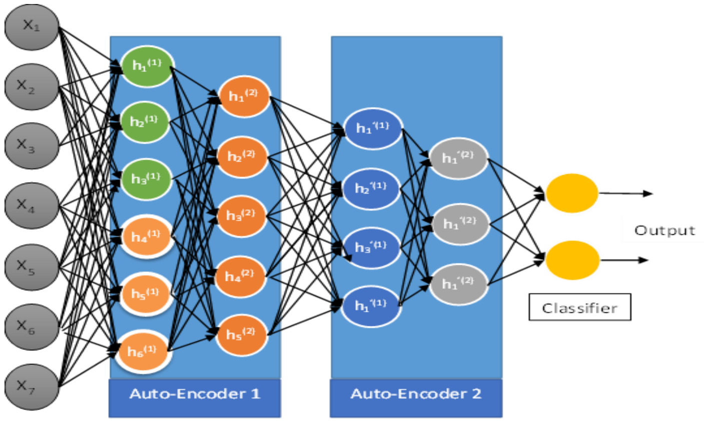

Fig. 5: Deep auto-encoder structure.

The LSTM auto-encoder model has the following description:

- Layer 1, Input Layer, reads the input data and outputs (18, 1) features for each.
- Layer 2, Convolutional Layer conv1d, reads the input data and outputs (18, 64) features for each.
- Layer 3, max pooling1d, max-pooling is an operation used to reduce the spatial dimension of the output from the conv1d, if it is not used, and replace it with Convolution to extract the most important features. It takes the inputs from Layer 2. It outputs a feature vector. The output of this layer is the feature vector of the input data. It outputs (9, 64) features for each.
- Layer 4, LSTM, reads the input data layer 3 and outputs (9, 128) features for each.
- Layer 5, LSTM, reads the input data layer 4 and outputs (128) features for each.

Dropout, takes the inputs from Layer 5. The output of this layer is the encoded feature vector of the input data.

III. DDoS Detection

To create a python deep learning model using a combination of CNN and LSTM in an auto-encoder with checkpoint, these steps should be followed:

1. Define the architecture of the auto-encoder, which includes the encoder and decoder parts. The encoder part should consist of a stack of CNN layers followed by an LSTM layer, while the decoder part can have a LSTM layer followed by a stack of transposed CNN layers.
2. Prepare and preprocess the training data as needed for the specific task, such as normalizing the input data.
3. Set up the training process, including the optimizer, loss function, and other parameters.
4. Set up checkpoint by specifying the directory to save the checkpoints and the frequency at which to save them.
5. Train the model by passing the training data through the auto-encoder and updating the weights using the optimizer and loss function.
6. Evaluate the model on a separate test dataset to assess its performance.
7. Finally, use the trained model with the encoder part of the auto-encoder to encode new data, and the decoder part to decode the encoded data. The overall planned system's structure is shown in Figure. (6), and the flowchart of the detection stage is illustrated in Figure. (7)

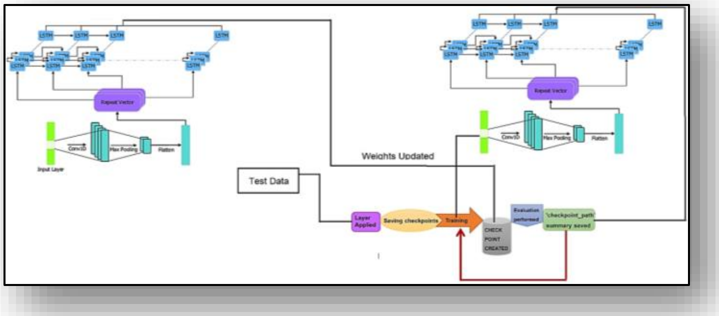

Figure (6): Structure of the DDoS detection system.

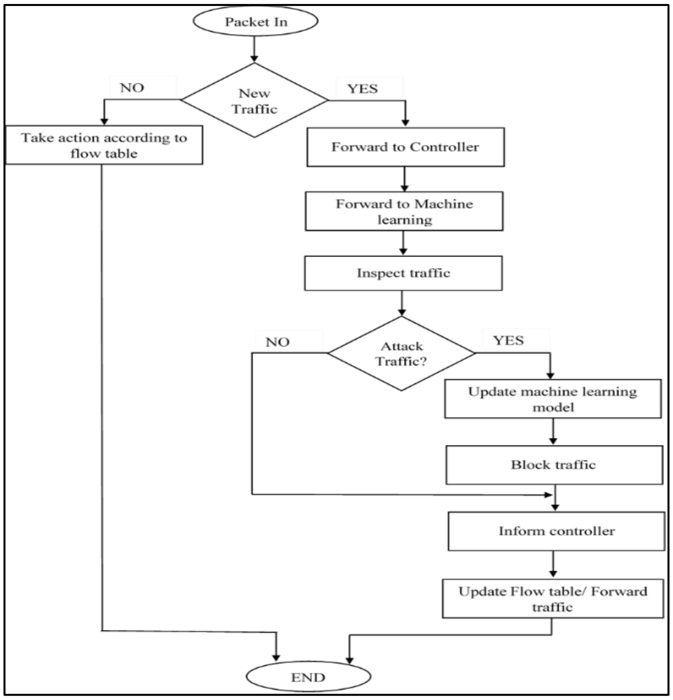
Packet In
New Traffic
NO
YES
Take action according to flow table
Forward to Controller
Forward to Machine learning
Inspect traffic
Attack Traffic?
NO
YES
Update machine learning model
Block traffic
Inform controller
Update Flow table/ Forward traffic
END

Fig 7: Flowchart of DDoS detection.

IV. RESULTS

The training process were implemented over 500 epochs. In each epoch, there is ETA, loss, and accuracy. The ETA refers to the Estimated Time of Arrival for the completion of a certain number of epochs during the training of a machine learning model. The loss refers to a measure of how well a model is able to perform on a given task. During training, the goal is to minimize the loss function by adjusting the model's weight. The accuracy refers to the ratio of the number of correctly predicted labels to the total number of labels in the dataset. For the first ten epochs, the results were as shown in Table (4).

TABLE V
TRAINING RESULTS.

Epoch No.	ETA	Loss	Accuracy
1	27 sec.	31.98%	92.450%
2	24 sec.	17.65%	92.450%
3	25 sec.	17.17%	93.268%
4	25 sec.	14.21%	94.337%
5	25 sec.	12.52%	94.809%
6	25 sec	11.24%	96.268%
7	25 sec	10.76%	96.268%
8	26 sec.	10.41%	96.268%
9	26 sec.	9.42%	96.321%
10	27 sec.	9.14%	96.321%

The epoch number 500 achieved an accuracy of 99.9%, an ETA of 32 sec., and a loss of 0.33%. Fig. (8) illustrates the results as a plot.

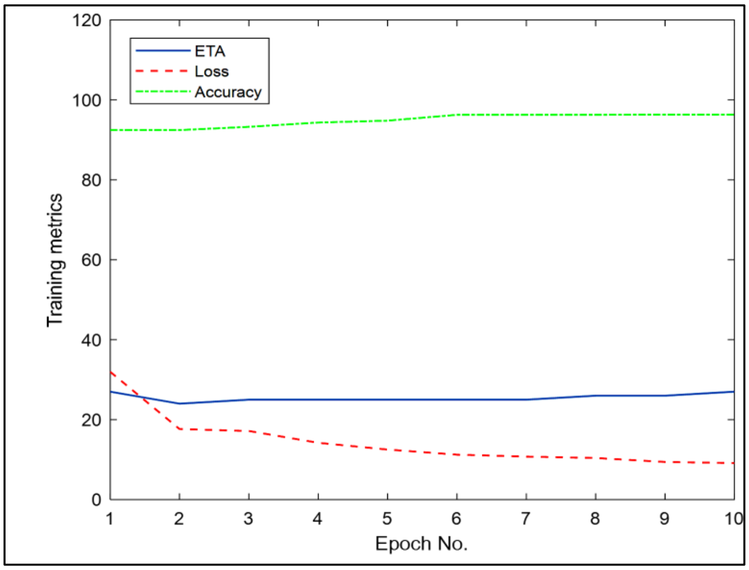

Fig:8 Training metrics.

The accuracy and the loss of the proposed LSTM-CNN model is illustrated in Figure. (9) and Figure. (10), respectively, were the trained and valid metrics are shown.

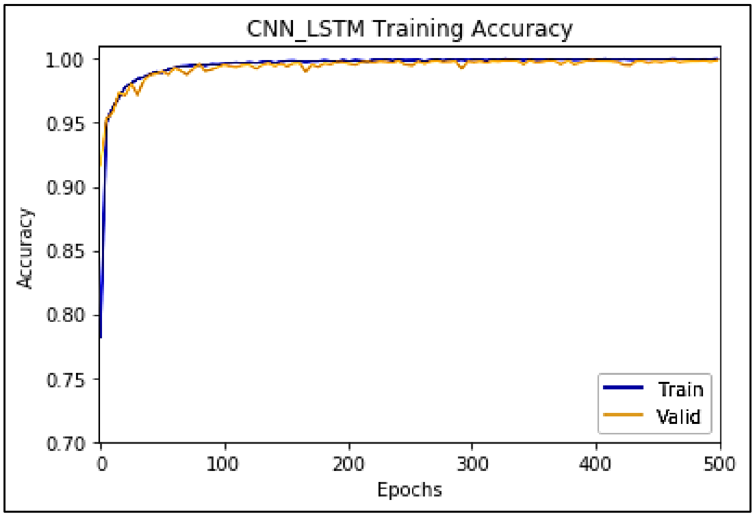

Fig 9: Accuracy of LSTM-CNN model.

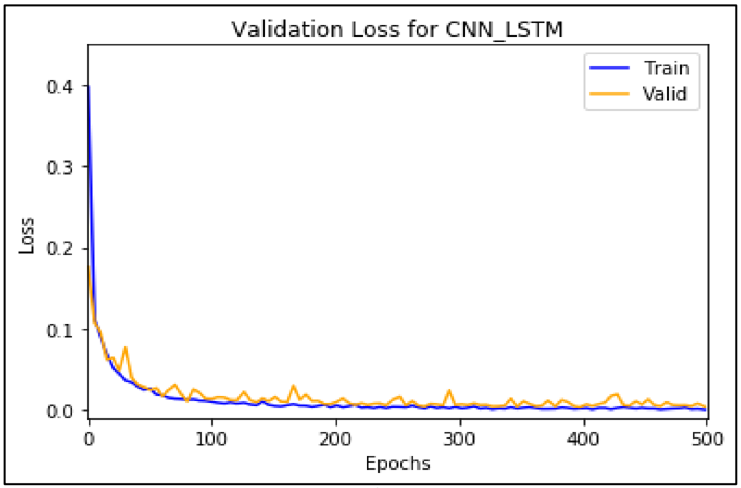

Fig.10: Validation loss of LSTM-CNN model

In Fig. 11, the confusion matrix depicts the ratios of Normal-Normal which performs the true positive (TP). The (TP) indicates that the LSTM-CNN model accurately predicted the normal traffic with a 100% rate. Similarly, the True Negative (TN) shows that the LSTM-CNN model correctly predicted the DDoS traffic with 100% accuracy. The False Positive (FP), on the other hand, represents instances where the model predicted a normal traffic as an attack, with a rate of 0.074%. Finally, the False Negative (FN) denotes cases where the model correctly predicted an attack traffic when it was normal, with a rate of 0.087%.

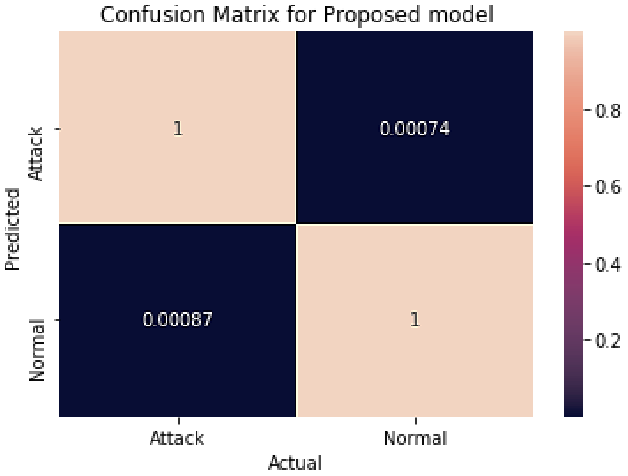

Fig 11: Confusion Matrix of the Proposed Model.

V. CONCLUSION

Python, specifically the Python Anaconda distribution, is a popular language for developing and implementing machine learning models, including the combined CNN-LSTM model for DDoS detection in SDN networks. Monitoring the ETA, accuracy, and loss during the training process can provide insight into the model's progress and performance. ETA can be used to estimate the time required to complete the training process. At the same time, accuracy measures the model's ability to classify inputs correctly, and loss measures what distinguishes them the model's predicted for each input sample, output and the real output. Overall, the hybrid model of CNN and LSTM implemented using Python and trained on labeled network traffic data has the potential to provide a high level of accuracy in detecting DDoS attacks in SDN networks, making it a valuable tool for network security. Based on the findings, the LSTM-CNN model demonstrated exceptional performance to identify DDoS assaults, achieving an accuracy rate of 99.9%. The results are further supported by the confusion matrix, which indicates that the system accurately distinguishes between normal traffic and DDoS traffic with a precision of 100%.

REFERENCES

[1] Tiago Espinha Gasiba , Ulrike Lechner, and Maria Pinto-Albuquerque**, "Sifu - a cybersecurity awareness platform with challenge assessment and intelligent coach"**, Cybersecurity (ISSN:2523-3246), 2020.

[2] Lotfi Mhamdi, Desmond McLernon1, Fadi El-moussa, Syed Ali Raza Zaidi, Mounir Ghogho, and Tuan Tang, "A Deep Learning Approach Combining Autoencoder with One-class SVM for DDoS Attack Detection in SDNs", International Conference on Communications and Networking (ComNet), 2020.

[3] Yuchong Li and Qinghui Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments", Elsevier Energy Report (ISSN: 8176–8186), 2021.

[4] Beny Nugraha and Rathan Narasimha Murthy, "Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks", IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2020.

[5] James Dzisi Gadze, Akua Acheampomaa Bamfo-Asante, Justice Owusu Agyemang , Henry Nunoo-Mensah, and Kwasi Adu-Boahen Opare, "An Investigation into the Application of Deep Learning in the Detection and Mitigation of DDOS Attack on SDN Controllers", Technologies, Vol. 9, No. 14, 2021.

[6] AbdulkadirTasdelen and Baha Sen, "A hybrid CNN-LSTM model for pre-miRNA classification", Scientifc Reports, 2021.

[7] Inam Abdullah Abdulmajeed and Idress Mohammed Husien "MLIDS22- IDS Design by Applying Hybrid CNN-LSTM Model on Mixed-Datasets " , Informatica ‹vol. 46, p.p 121-134, 2022.Inam Abdullah Abdulmajeed and Idress Mohammed Husien "MLIDS22- IDS Design by Applying Hybrid CNN-LSTM Model on Mixed-Datasets " ,

[8] Kong Z, Cui Y, Xia Z, Lv H., "Convolution and Long Short-Term Memory Hybrid Deep Neural Networks for Remaining Useful Life Prognostics", Applied Sciences, Vol. 9, No. 19, 2019.

[9] Amitha Mathew, P. Amudha, and S. Sivakumari, "Deep Learning Techniques: An Overview", International Conference on Advanced Machine Learning Technologies and Applications, © Springer Nature Singapore Pte Ltd. 2021. https:// doi.org/ 10.1007/978-981-15-3383-9_54

[10] Mnih V., Badia A.P., Mirza M., Graves A., Lillicrap T., Harley T., Silver D., and Kavukcuoglu K., "Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning", pp. 1928–1937, 2016.

[11] Northport, N.Y. ‹"Top 10 Cybersecurity Predictions And Statistics For 2023" ‹Cybercrime Magazine ‹2022.