ANALYSIS OF DEEP NEURAL NETWORK ALGORITHMS FOR MITIGATING DISTRIBUTED DENIAL OF SERVICE ATTACKS IN SOFTWARE DEFINED NETWORKS.

Supported by tetfund

T. O. Oladele[1] and E. R Jimoh[2]

[1]Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria.

[2] Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria.

*Corresponding Author's E-mail: jimohebunayo2019@gmail.com;

Received: September 14, 2023 **Accepted:** November 28, 2023

Abstract: Over the years DDoS attack are not easily noticed and preventing them has been uneasy due to the fact that the sources addresses are initiated and other method are used to hide attack sources, Physical gadgets only cannot solve DDoS attacks. Also, because of the necessity to enhance global view of the network Data Centre, network operators have also evolved from the traditional based network to Software Defined Network (SDN) because SDN gives more reliability, flexibility and a secure network environment . Even with the enormous capabilities of SDN, it is still faced with several security challenges due to its architecture complexity. SDN architecture is very vulnerable to DDoS attack. In this research work, six (6) different Deep Neural Network (DNN) models has been analyzed and from the analysis it has been seen that The Long Shortterm Memory (LSTM) out performs other five (5) algorithms with accuracy of 100%, loss of 0.000287211 which is very low compared to other algorithms considered. LSTM takes 915,002 milliseconds per Iteration which is quite high, the Convolutional Neural Networks (CNN) takes 300,000.635 milliseconds per iteration and it has 99.99% accuracy. Therefore, fusing the LSTM and CNN models into DDoS mitigation system in software defined networks will be beneficial. This study is limited to User Datagram protocol (UDP) and Transfer Control Protocol (TCP) attacks.

Keywords: Algorithms, Convolutional Neural Network (CNN), Deep Neural Networks (DNN), Distribution Denial of Service (DDoS), Long Short Term Memory (LSTM) and Software Defined Networks (SDN)

Introduction

Important information was stored into files before computers came into existence. These important documents were stored physically and could be subsequently taken away by unauthorized users. To ensure that these documents were safe they were kept in cabinets which had locks and key to lock and unlock them when the authorized users need access. That was the approach that was used to keep the information safe for legitimate users before the era of computerization. But in this computerized era, most critical information, if not all, are stored in the computer as files, but that has not changed the basic requirements in securing information which are: Confidentiality, Accessibility and Integrity (Zargar, Joshi & Tipper, 2013). Information has to be in a reliable form because information is power. Information that is not reliable is useless. It is known that data has value and that anything that has worth is an asset. Data should always be consistent. Only trained authorized individuals should modify data in order to ensure Data integrity and Data consistency. Information that lacks integrity is worthless (Zargar, Joshi &Tipper, 2013). The concept of confidentiality describes how information should be made available to reliable individuals. When users access confidential information about a system, person, or organization, a high level of security must be put in place. For example, the bank credit information of a customer is also an asset and like all other assets it needs to be secured. Confidentiality implies ensuring that the secret code of an individual is not accessible to another person. Accessibility means that the users must be able to access data when needed. Refusal or a Denial-of-Service Attack occurs when the required data is not accessible at the required moment (Gond & Nath, 2015).

With millions of computers connected, the Internet is the world's biggest computer network. People use the internet simply to obtain information and conduct online searches in earlier times (GCF, 2013). The accessibility of the internet is very crucial. The world we are presently is totally relying on the internet (Mahajan & Sachdeva, 2013). Today's internet not only contains a lot of varieties of information, but it is an evolving tool, which provides a platform to connect with people, interact with these people and also access and share information content with them. Internet users find the internet as a point of gathering useful information. It is no more news that as internet is expanding, it is increasingly being included into our daily lives. Also, because of the necessity to enhance global view of the network data centre, network operators have also evolved from the traditional based network to Software Defined Network (SDN) because SDN gives more reliability, flexibility and a secure network environment (Zhang *et al.*, 2018; Visu *et al;* 2019, Molina *et al.*, 2018).

In this new era that data has ended up in electronic form and our daily activities have been digitalized, network security is of utmost significance. In today's Internet, one of the most potent threats to internet security is Denial of Service (DoS) (Douligeris & Mitrokotsa, 2004). DoS is a very common form of Cyber-attack, because it is easy and cheap for intruders to penetrate victims using this

technique. Due to the severity of the DoS attacks, many mechanisms have been developed to curb the act. An example of a cyber-attack is a Distributed Denial of Service (DDoS) attack, which blocks access to networks by overloading targets with excessive amounts of malicious traffic and stealing their bandwidth, and since most servers have limited traffic packets that can process successfully and competently, once the limit is exceeded there will be degradation in the system's performance (Gond &Nath, 2015). Due to the simultaneous deployment of the attack on several machines, DDoS is a distributed denial of service attack (Mahajan & Sachdeva, 2013). Distributed Denial of Service (DDoS) is one of the most critical attacks for networks (Brutag, 2000). DDoS attack is not new in technology; it was first experienced in the technology environment in 1999. The severity of the problems has been steadily getting worse.

DDoS attacks are not easily noticed and preventing them is not usually easy due to the fact that the source addresses are initiated and other methods are used to hide attack sources. There are two major reasons which make protection against DDoS attack difficult (Peng, Leekie & Ramamohanarao, 2007). The first reason is that the zombies that act in DDoS attack are very numerous and these zombies cover huge number of geographical areas in any place in the world. Therefore, the size of traffic that a zombie sends might actually be small, but the victims' host receives a massive size of aggregated traffic. The second reason that makes protection against these attacks an uneasy one is that it is very hard to detect the intrusion back to zombies because they often spoof their Internet Protocol (IP) addresses and are under the attacker's control, making it hard to track the actual system.

A DDoS attack increases the amount of effort required to detect intrusion and develop effective countermeasures, as well as causes lengthy system timeouts, lost revenue, and greater workloads. According to Sachdera *et al.,* (2010) Syn flood, teardrop, Snurf, and ping of death are examples of DoS attacks. Packet flooding is the most prevalent type of DDoS attack, in which numerous, purportedly authenticated Internet Control Message Protocol (ICMP), Transfer Control Protocol (TCP), and User Datagram Protocol (UDP) packets are sent to targeted locations.

The network perspective known as SDN allows network operators to use open interface such as open flow control to programmatically, track, setup, control and change network (Zeebaree *et al.,* 2020). SDN which proposes to concentrate network intelligence on a single network component by separating the data packet forwarding mechanism (Data plane/layer) from the routing process control plane/layer, has made networks easier for network managers to maintain (Haji, *et al.*, 2021). SDN is more versatile since it is software-based, allowing users to manage resources more easily when they are remote on the control plane (Mousa *et al.*, 2016; Xu *et al.*, 2018). One of the most well-known aspects of SDN is the separation of the control plane from the data plane (Lawal, 2018). The decision of where to send packets is carried out by the control plane while the data plane implements the decisions and forwards the packets (Murgaa, *et al.*, 2021). The openflow can be referred to as the communication protocol between physical switches and SDN controller.

Even with the enormous capabilities of SDN, it is still faced with several security challenges due to its architecture's complexity. SDN architecture is very vulnerable to DDoS attack (Dong *et al.,* 2018; Jose, Nair & Paul, 2021). Due to SDN's centralized architecture, which affects the entire network, DDoS attacks are common (Nadeem *et al.*, 2022). Security of the SDN controller is a serious concern because it serves as the operating system that controls execution of various network applications and its functionalities. When the SDN controller is undergoing DDoS attack, it loses its centralized control since the controller and the rest of the network are separate (Nadeem *et al.*, 2022). For SDN controller to detect DDoS attacks, it has to repeatedly gather network traffic data from the switches in order to pinpoint when the DDoS attack occurred.

Cyber-attack has negative impact on the security and defence of any Nation (Amaral, 2014). Authentication, confidentiality, availability, integrity, and lack of reputation are issues with internet security (Mahajan & Sachdeva, 2013). In SDN the case of Cyber-attacks are prevalent and pervasive, without security measures in place our data might be subjected to attacks (Naoum & Ross, 2016). Many users have been victims of Cyber-attacks and have been denied the service that should be offered legitimately which leads to distortion of processes due to unavailability of resources (Yasar, 2020). Physical gadgets only cannot solve DDoS attacks in the SDN. Hence, the need to build an intelligent technique to curb the ill act. Businesses have found it to be extremely difficult and time-consuming to manually configure each unique network software switch. Network needs are also steadily increasing, making it difficult to navigate hardware switches. The need for Software Defined Networks is highly required (Karmakar *et al*., 2017). SDN is projected as the main component of the next generation network (Boukria & Guerroumi, 2019). SDN gives flexibility, but has become a valuable target for intruders due to its significant role; there is a need for a technique to mitigate invaders since, in this era, DDoS attack mitigation in a SDN environment is a major concern (Juan *et al.*, 2021; Boukria & Guerroumi, 2019).

This research work analyzed the performance of six (6) different Deep Neural Network (DNN) learning techniques for mitigating Cyber-attacks (DDoS) attacks in SDN environment using time taken per iteration, loss function, accuracy, recall and precision.

Materials and Method

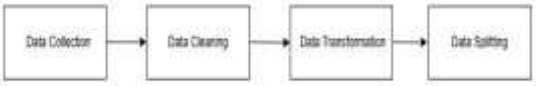
The analysis of six (6) different Deep Neural Learning algorithms has been examined to determine the best performance for the DDoS mitigation in Software Defined Networks. The following Deep Neural Learning algorithms performances were analyzed: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN)**,** Recurrent Neural Network (RNN), Long Short- Term Memory Networks (LSTM), Gated Recurrent Networks (GRN) and Deep Residual Network (DRN). The analysis of the Six (6) Deep Learning Algorithms was evaluated to effectively

determine their performance, the best performed in terms of accuracy, loss and compilation time of the six (6) selected Deep Learning Algorithms, the analysis of this algorithms was done with respect to ISCXIDS2018 DDoS dataset. The original DDoS attack dataset was extracted, and the performance of several Deep Neural Learning classifiers for detecting DDoS attacks in SDN was analyzed. The DDoS dataset is initially loaded as a pandas DataFrame. The train test split method from Scikit-learn was then used to divide the dataset into training and testing sets (70% and 30% respectively). The methods used in achieving the study's aim and objectives can be divided into the following phases.

Data Preparation

Data preparation follows the following procedure in Figure 2.1

Figure 2.1: Data Preparation Procedure



Data Collection

The dataset used to evaluate our model is the ISCXIDS2018 dataset. The rationale behind selecting the ISCXIDS2018 dataset was to have a reliable dataset that is considered to be a sufficient benchmark dataset. This Data Set has the following characteristics:

a. Realistic network and traffic
b. Labeled dataset
c. Total interaction capture
d. Complete capture
e. Diverse intrusion scenarios

Data Cleaning

Building an efficient Deep learning model depends on the preprocessing phase. To achieve that, we first needed to remove the redundant data from the dataset. A dataset of 676,319 records was used for classification.

Data Transformation

All the selected features were normalized into the range of 0-1 with min-max feature scaling before feeding them to the Deep Learning models. Min-Max scaling technique can be applied in Pandas using the min() and max() methods. Sklearn→ label encoder()- Fit-transform(). Min-Max is one of the widely used normalization technique in Deep learning.

Data Splitting

A method from the Scikit-learn library (sklearn.model-selection) Sklearn.model-selection (train-test spilt) was applied to split the dataset as follows: 70% of the records were used in training and 30% for testing.

Feature Selection process: Feature selection techniques are applied to reduce the dimensionality of your dataset. Feature selection is about choosing a subset of the most relevant features from the data set collected, using Sklearn.feature_selection, import StandardScaler.

DNN Models for DDoS attacks in SDN Algorithm

Step 1: Start

Step 2: Import necessary libraries and load data

Step 3: Import deep_ learning_ library as dnn

Step 4: Import data_ processing_ library as dp

Step 5: Define a function to analyze DNN models

Step 6: Function analyze_dnn_model (model, data)

Step 7: Evaluate Model performance

Step 8: Metrics = model.evaluate (data.test_features, data.test_labels)

Step 9: Display performance metrics

Step 10: plot training_test graph

Step 11: Perform hyperparameter tuning (best_hyperparameters = hyperparameter_tunning (model, data. Train_ features, data. train_labels)

Step 12: load preprocess data

Step 13: load pre- trained DNN models

Step 14: Analyze the selected models

Step 15: Export analysis results

Step 16: End

Figure 2.1: Analyzing DNN Models for DDoS attacks in SDN Algorithm

Results and Discussion

Analysis of Multilayer Perceptron (MLP) Model

The MLP model which result specifies the time in 282s 595 us/step is the step it takes one step or iteration to be completed by the algorithm. Microseconds, or one millionth of a second, is what "us" stands for. Therefore, it takes one iteration 282,000.595 milliseconds to be completed. In applications where real-time performance is crucial, the amount of time required for each step can be a significant statistic for assessing the effectiveness and performance of an algorithm. Accuracy in Deep Neural learning refers to the percentage of correctly categorized examples in the dataset. It is determined by dividing the total number of predictions by the number of correct predictions. On the other hand, validation accuracy refers to the model's performance on a validation set, which is a subset of the dataset that is not used for training but is used to assess the model's effectiveness. The model's parameters are changed during training in order to reduce the loss function on the training set as shown in Table 3.1.

Table 3.1: Analysis of Multilayer Perceptron (MLP) Model

Epoch	Time taken per iteration	Loss	Accuracy	Precision	Recall	Val-loss	Val-Acc	Val-Precision	Val-Recall
1	180s 379us	0.8601	0.6585	0.6585	0.6585	0.6736	0.6586	0.6586	0.6586
2	325s 687us	0.6733	0.6588	0.6588	0.6588	0.6733	0.6586	0.6586	0.6586
3	362s 764us	0.6739	0.6588	0.6588	0.6588	0.6739	0.6586	0.6586	0.6586
4	342s 722us	0.6740	0.6588	0.6588	0.6588	0.6741	0.6586	0.6586	0.6586
5	292 616us	0.6741	0.6588	0.6588	0.6588	0.6750	0.6586	0.6586	0.6586
6	425s 898us	0.6743	0.6588	0.6588	0.6588	0.6754	0.6586	0.6586	0.6586
7	302s 638us	0.6745	0.6588	0.6588	0.6588	0.6755	0.6586	0.6586	0.6586
8	382s 808us	0.6747	0.6588	0.6588	0.6588	0.6740	0.6586	0.6586	0.6586
9	370s 782us	0.6748	0.6588	0.6588	0.6588	0.6746	0.6586	0.6586	0.6586
10	282s 595us	0.6750	0.6588	0.6588	0.6588	0.6753	0.6586	0.6586	0.6586

Analysis of Convolutionary Neural Network (CNN Model)

To match the CNN input format, which calls for a three-dimensional tensor of shape (batch size, sequence length, input dim), the training and testing sets are restructured. Using the Keras package, a CNN model was created with two fully connected layers, a Convolutional layer, a layer for maximum pooling, and two more layers utilizing the training set. The Adam optimizer and categorical cross-entropy loss was used to build the model. Lastly, a batch size of 32 was used to train the model for 10 epochs, and analyze the accuracy, precision and recall of the model on the training and testing set using accuracy, precision and recall function from scikit-learn. The Evaluation of Convolutionary Neural Network (CNN Model) with respect to ISCXIDS 2018 DDoS dataset is shown in Table 3.2.

The CNN mode whose result specifies the time in 300s 635 us/step is the step it takes one step or iteration to be completed by the algorithm. It takes the CNN 300,000.635 milliseconds to complete each step. Accuracy in Deep Neural learning refers to the percentage of correctly categorized examples in the dataset. It is determined by dividing the total number of predictions by the number of correct predictions. On the other hand, validation accuracy refers to the model's performance on a validation set, which is a subset of the dataset that is not used for training but is used to assess the model's effectiveness. The model's parameters are changed during training in order to reduce the loss function on the training set is shown in Table 3.2

Table 3.2: Analysis of Convolutionary Neural Network (CNN) Model

Epoch	Time taken per iteration	Loss	Accuracy	Precision	Recall	Val-Loss	Val-Acc	Val-Precision	Val-Recall
1	155s 328us	0.6106	0.9450	0.9450	0.9450	0.1319	0.9991	0.9991	0.9991
2	183s 386us	0.1204	0.9995	0.9995	0.9995	0.1127	0.9997	0.9997	0.9997
3	163s 344us	0.1097	0.9996	0.9996	0.9996	0.1108	0.9996	0.9996	0.9996
4	185s 391us	0.1065	0.9995	0.9995	0.9995	0.1036	0.9996	0.9996	0.9996
5	168s 355us	0.1054	0.9994	0.9994	0.9994	0.1080	0.9997	0.9997	0.9997
6	169s 358us	0.1047	0.9994	0.9994	0.9994	0.1040	0.9999	0.9999	0.9999
7	187s 395us	0.1042	0.9995	0.9995	0.9995	0.1017	0.9996	0.9996	0.9996
8	234s 495us	0.1037	0.9995	0.9995	0.9995	0.1015	0.9999	0.9999	0.9999
9	280s 592us	0.1032	0.9995	0.9995	0.9995	0.1006	0.9997	0.9977	0.9997
10	300s 635us	0.1032	0.9995	0.9995	0.9995	0.0995	0.9999	0.9999	0.9999

Analysis of Recurrent Neural Networks (RNN Model)

The DDoS dataset is initially loaded as a pandas DataFrame. The train test split method from Scikit-learn was then used to divide the dataset into training and testing sets (70%, 30 % respectively). To match the RNN input format, which calls for a three-dimensional tensor of shape (batch size, timesteps, input dim), restructuring of the training and testing sets was also done. Using the Keras library and the training set, an RNN model is created with an RNN layer and a fully connected layer. The Adam optimizer and categorical cross-entropy loss is used to build the model. Lastly, a batch size of 32 was used to train the model for 10 epochs and evaluate the accuracy, precision and recall of the model on the training and testing set using accuracy, precision and recall function from scikit-learn. Before training the model, it was ensured that the dataset was correctly prepared and preprocessed. The analysis of Recurrent Neural Networks (RNN Model) with respect to ISCXIDS 2018 DDoS dataset is shown in Table 3.3.

Table 3.3: Analysis of Recurrent Neural Network (RNN) Model

Epoch	Time taken per iteration	Loss	Accuracy	Precision	Recall	Val-Loss	Val-Acc	Val-Precision	Val-Recall
1	389s 823us	0.0386	0.9984	0.9984	0.9984	0.0063	0.9985	0.9985	0.9985
2	372s 786us	0.0184	0.9994	0.9994	0.9994	0.0072	1.0000	1.0000	1.0000
3	451s 952us	0.0154	0.9997	0.9997	0.9997	0.0045	1.0000	1.0000	1.0000
4	535s 1ms	0.0135	0.9997	0.9997	0.9997	0.0047	1.0000	1.0000	1.0000
5	743s 2ms	0.0179	0.9997	0.9997	0.9997	0.0091	0.9999	0.9999	0.9999
6	673s 1ms	0.0187	0.9998	0.9998	0.9998	0.0068	1.0000	1.0000	1.0000
7	771s 2ms	0.0133	0.9998	0.9998	0.9998	0.1456	0.9991	0.9991	0.9991
8	902s 2ms	0.0233	0.9998	0.9998	0.9998	0.0338	0.9999	0.9999	0.9999
9	566s 1ms	0.0147	0.9998	0.9998	0.9998	0.0067	1.0000	1.0000	1.0000
10	954s 2ms	0.0245	0.9997	0.9997	0.9997	0.0140	1.0000	1.0000	1.0000

954s 2 ms/step, which appears in the RNN model result, refers to the time required by the model to complete one step or iteration in a milliseconds which is 954,002milliseconds. The amount of time required for each step can be a crucial parameter for assessing an algorithm's effectiveness and performance, particularly in applications where real-time performance is essential. The percentage of correctly categorized examples in a dataset is measured by the Deep Neural Learning metric accuracy. In order to compute it, divide the total number of predictions by the number of correct predictions. On the other hand, validation accuracy is the model's performance on a validation set, which is a subset of the dataset that is not used for training but is used to assess the model's effectiveness. The parameters of the model are changed during training to reduce the loss function on the training set.

Analysis of Gated Recurrent Network (GRN Model)

In this code, the DDoS dataset was loaded as a panda DataFrame. Then split the dataset into training and testing set (70%, 30% respectively) using the train-test split function from scikit-learn. The training and testing set were reshaped to fit the GRN input format, which requires a three-dimensional tensor of shape (batch_size, timesteps, input_dim). Using the training set to define a GRN model with two GRN layers and a fully connected layer using the Keras library. The model complies with categorical cross-entropy loss and the Adam optimizer. Finally, the model was trained for 10 epochs using a batch size of 32 and evaluates the accuracy, precision and recall of the model on the training and testing set using accuracy, precision and recall function from Scikit-learn. The accurate results shows how much secure is the proposed model. Gated Recurrent Network (GRN) with respect to ISCXIDS 2018 DDoS dataset is shown in Table 3.4.

The time in milliseconds required by the model to complete one step or iteration as shown in the GRN model result by the value 1088s 2ms /step. The model in this instance completes one step in 1,088,002 milliseconds. In applications where real-time performance is essential, the amount of time required for each step can be a key parameter for assessing an algorithm's effectiveness and performance. The percentage of examples in a dataset that are correctly categorized is measured by the Deep Neural learning data accuracy. By dividing the total number of predictions by the number of correct predictions, it is determined. The model's accuracy when applied to a validation set, a subset of the dataset that is not used for training but is used to assess the performance of the model, which is known as validation accuracy. The parameters of the model are changed as it is being trained in order to reduce the loss function on the training set as shown in Table 3.4.

Table 3.4: Analysis of Gated Recurrent Network (GRN) Model

Epoch	Time taken per iteration	Loss	Accuracy	Precision	Recall	Val-Loss	Val-Acc	Val-Precision	Val-Recall
1	768s 2ms	0.0620	0.9929	0.9929	0.9929	0.0193	0.9999	0.9999	0.9999
2	871s 2ms	0.0197	0.9994	0.9994	0.9994	0.0070	0.9999	0.9999	0.9999
3	1004s 2ms	0.0214	0.9990	0.9990	0.9990	0.0070	0.9999	0.9999	0.9999
4	903s 2ms	0.0211	0.9994	0.9994	0.9994	0.0618	0.9999	0.9999	0.9999
5	1060s 2ms	0.0211	0.9995	0.9995	0.9995	0.0040	0.9999	0.9999	0.9999
6	1102s 2ms	0.0183	0.9996	0.9996	0.9996	0.0287	0.9997	0.9997	0.9997
7	964s 2ms	0.0180	0.9996	0.9996	0.9996	0.0062	0.9999	0.9999	0.9999
8	949s 2ms	0.0235	0.9994	0.9994	0.9994	0.0083	0.9999	0.9999	0.9999
9	1033s 2ms	0.0161	0.9997	0.9997	0.9997	0.0073	0.9999	0.9999	0.9999
10	1088s 2ms	0.0190	0.9997	0.9997	0.9997	0.0122	0.9999	0.9999	0.9999

Analysis of Long Short-Term Memory Networks (LSTM Model)

In this code, the DDoS dataset was loaded as a pandas DataFrame. Then split the dataset into training and testing set (70%, 30% respectively) using the train-test split function from scikit-learn. The training and testing set was reshaped to fit the LSTM input format, which requires a three-dimensional tensor of shape (batch_size, timesteps, input_dim). Using the training set to define a LSTM model with two LSTM layers and a fully connected layer using the Keras library. The model complies with categorical cross-entropy loss and the Adam optimizer. Finally, the models were trained for 10 epochs using a batch size of 32 and evaluate the accuracy, precision and recall of the model on the training and testing set using accuracy, precision and recall function from Scikit-learn. The accurate results shows how much secured is the proposed model. Long Short Tern Memory Network (LSTM) with respect to ISCXIDS 2018 DDoS dataset is shown in Table 3.5.

The time in milliseconds (ms) required by the model to complete one step or iteration as shown in the LSTM model result by the value 915s 2ms /step. The model in this instance completes one step in 915,002 milliseconds. The percentage of examples in a dataset that are correctly categorized is measured by the Deep Neural learning data accuracy. By dividing the total number of predictions by the number of correct predictions, it is determined. The model's accuracy when applied to a validation set, a subset of the dataset that is not used for training but is used to assess the performance of the model, is known as validation accuracy. The parameters of the model are changed as it is being trained in order to reduce the loss function on the training set is shown in Table 3.5.

Table 3.5: Analysis of Long Short-Term Memory Networks (LSTM Model)

Epoch	Time taken per iteration	Loss	Accuracy	Precision	recall	Val-loss	Val-acc	Val-precision	Val-recall
1	969s 2ms	0.0059	0.9977	0.9977	0.9977	4.2966e-04	0.9999	0.9999	0.9999
2	1019s 2ms	7.9556 e-04	0.9998	0.9998	0.9998	2.0539 e-04	0.9999	0.9999	0.9999
3	1134s 2ms	4.3953 e-04	0.9999	0.9999	0.9999	1.5037e-04	1.0000	1.0000	1.0000
4	1188s 3ms	3.8768 e-04	0.9999	0.9999	0.9999	8.1273e-05	1.0000	1.0000	1.0000
5	809s 2ms	3.6894 e-04	0.9999	0.9999	0.9999	7.7047e-05	1.0000	1.0000	1.0000
6	714s 2ms	4.3144 e-04	0.9999	0.9999	0.9999	1.2113e-04	1.0000	1.0000	1.0000
7	1157s 2ms	3.1490 e-04	0.9999	0.9999	0.9999	8.7659e-05	1.0000	1.0000	1.0000
8	1092s 2ms	3.3615e -04	0.9999	0.9999	0.9999	6.5740e-05	1.0000	1.0000	1.0000
9	4945s 10ms	3.5286e-04	0.9999	0.9999	0.9999	6.9000e -05	1.0000	1.0000	1.0000
10	915s 2ms	2.8711e -04	1.0000	1.0000	1.0000	6.6733e-05	1.0000	1.0000	1.0000

Analysis of Deep Residual Networks (DRN Model)

The time in milliseconds required by the model to complete one step or iteration as shown in the DRN model result by the value 695s 1ms /step. The model in this instance completes one step in 695,001milliseconds. The percentage of examples in a dataset that are correctly categorized are measured by the Deep Neural learning data accuracy. By dividing the total number of predictions by the number of correct predictions, it is determined. The model's accuracy when applied to a validation set, a subset of the dataset that is not used for training but is used to assess the performance of the model, is known as validation accuracy. The parameters of the model are changed as it is being trained in order to reduce the loss function on the training set is shown in Table 3.6.

Table 3.6: Analysis of Deep Residual Network Model

Epoch	Time taken per iteration	Loss	Accuracy	Precision	Recall	Val-Loss	Val-Acc	Val-Precision	Val-Recall
1	396s 837us	0.0907	0.9996	0.9978	0.9978	0.0272	1.0000	0.9996	0.9996
2	349s 737us	0.0220	1.0000	0.9999	0.9999	0.0170	1.0000	0.9999	0.9999
3	1049s 2ms	0.0156	1.0000	1.0000	1.0000	0.0131	1.0000	1.0000	1.0000
4	1314s 3ms	0.0127	1.0000	1.0000	1.0000	0.0114	1.0000	1.0000	1.0000
5	1257s 3ms	0.0111	1.0000	1.0000	1.0000	0.0099	1.0000	1.0000	1.0000
6	1751s 4ms	0.0101	1.0000	1.0000	1.0000	0.0254	1.0000	1.0000	1.0000
7	889s 2ms	0.0097	0.9999	1.0000	1.0000	0.0083	1.0000	1.0000	1.0000
8	470s 992us	0.0088	1.0000	1.0000	1.0000	0.0085	1.0000	1.0000	1.0000
9	844s 2ms	0.0084	1.0000	1.0000	1.0000	0.0074	1.0000	1.0000	1.0000
10	695s 1ms	0.0080	1.0000	1.0000	1.0000	0.0076	1.0000	1.0000	1.0000

From the various Deep Neural Network Analysis shown in the Table 4.1, the CNN model has the least time taken per iteration of 300s, 635us, loss of 0.1032 and accuracy of 0.9999, while LSTM out performs CNN in terms accuracy with a value of 1.0000 and also in loss with a lesser value

of 0.000287211 but takes more time to complete an iteration (915s, 2ms). Considering that the three metrics are crucial to the accurate and timely mitigation of DDoS attack in SDN environment. Fusing the LSTM and CNN algorithms would be beneficiary in the timely and accurate mitigation of DDoS attacks in SDN Environments.

Table 3.7: Summary of the Analyzed Deep Neural Network Algorithms

Deep Learning Algorithm	Time per iteration	Loss	Accuracy	Precision	Recall
MLP	282s, 595us	0.6750	0.6588	0.6588	0.6588
CNN	300s, 635us	0.1032	0.9995	0.9995	0.9995
RNN	954s, 2ms	0.0245	0.9997	0.9997	0.9997
LSTM	915s, 2ms	0.000287211	1.0000	1.0000	1.0000
GRN	1088s, 2ms	0.0190	0.9997	0.9997	0.9997
DRN	695s, 1ms	0.0080	1.0000	1.0000	1.0000

Table 3.7 shows analysis of six (6) Deep Neural Network Algorithms from the Table 3.7, it can be seen that the MLP model has the least value of Accuracy, Precision and Recall. The Long short Term Memory Network (LSTM) has 100% Accuracy with a Precision and Recall value of 100%, but its Time taken per iteration is higher than Convolutional Neural Network (CNN). It takes LSTM almost thrice the time it takes CNN for each iteration. Also, the loss function of LSTM is very low compared to the five (5) algorithms analyze. The MLP has the least Time taken per Iteration but MLP model Accuracy is poor compared to the models analyzed.

Table 3.8: Summary of the Analyzed Deep Neural Network Algorithms

Deep Learning Algorithm	Val/Test Loss	Val/Test Accuracy	Val/Test precision	Val/Test Recall
MLP	0.6753	0.6586	0.6586	0.6586
CNN	0.9995	0.9999	0.9999	0.9999
RNN	0.0140	1.0000	1.0000	1.0000
LSTM	0.000066733	1.0000	1.0000	1.0000
GRN	0.0122	0.9999	0.9999	0.9999
DRN	0.0076	1.0000	1.0000	1.0000

Table 3.8 shows analysis of six (6) Deep Neural Network Algorithms from the Table 3.8, it can be seen that the MLP model has the least value of validation Accuracy, Precision and Recall. The Long short Term Memory Network (LSTM) has 100% validation Accuracy with a Precision and Recall value of 100%. Therefore, the LSTM-CNN will be fused into the proposed Enhanced Intelligent LSTM-CNN System since this two algorithms has the best value for Accuracy, Time taken per iteration and Loss.

Conclusion

Over the years DDoS attack has negative impact on the internet security, Authentication, confidentiality, availability, integrity, and lack of reputation are issues with internet security. In SDN the case of DDoS attacks are prevalent and pervasive, without security measures in place our data might be subjected to attacks. Many users have been victims of DDoS attacks and have been denied the service that should be offered legitimately which leads to distortion of processes due to unavailability of resources. This research work has contributed to knowledge by examining the performance of six (6) different Deep Neural Network Models using four performance metrics: loss function, accuracy, precision and recall and from the analysis it has been seen that LSTM out performs other five (5) algorithms with accuracy of 100%, loss of 0.000287211 which is very low compared to other algorithms considered. The LSTM takes 915,002 milliseconds per Iteration which is quite high, the CNN takes 300,000.635 milliseconds per iteration and it has 99.99% accuracy. This suggests that fusing LSTM and CNN model can significantly mitigate DDoS attacks in SDN. Also combining other statistical methods with the fused LSTM-CNN model can be looked into in future works since mitigating DDoS is a complicated task which requires multi faceted approach.

References

Amaral A. C. (2014). The cybernetic threat for the security and defence of Brazil, *Vision ConjuntionNumero*. 10(1), 19-22.Boukira, S., and Guerroumi, M. (2019). International conference on theoretical and applicative aspects of computer science (ICTAACS). Doi:10.1109/ICTAACS 48474.2019.8988138.

Brutlag, J. D. (2000). Aberrant Behavior Detection in Time Series for Network Monitoring. In Proc. of 14th USENIX LISA conference on system administration pp.139-146.

Dong, L., Chang, Y., Qizhao, Z., and Junquing, Y. (2018). Using SVM to detect DDoS attack in SDN Network, material science and engineering. 2nd Annual international conference on cloud Technology and communication Engineering. Vol 466.

Douligeris C., and Mitrokotsa A. (2004). DDOS attacks and Defence Mechanism: Classification and State of the art, *Computer Networks* 44(5), 643-666. DOI: 10.1016/j.comnet.2003.10.003.

Gond S., and Nath A. (2015). A Mitigation Model for DDos attack in wireless sensor Network, Department of Computer Science and Engineering National Institute of technology, India.

Haji, S.H., Zeebaare, S. R. M., Saeed, R. H., Ameen, S. Y., Shukur, H. M. S., Omar, N., Sadeeq, M. A. M., Ageed, Z. S., Ibrahim, I. M., and Yasin, H. M. (2021). Comparison of Software Defined Networking. *Asian Journal of research in Computer Science*. 9(2), 1-18. Article no AJRCOS.68725 ISSN 2581-8260.

Juan, A., Cabrera, G., Frank, H.P., Fitzek, S. H., Sebastian, A. W., Itting, J. Z., Sandra, Z., Thorsten, S., Meryem, S., Christof, W. F., (2021). Intelligent Networks. https:/DOI: 10.1016/B978-0-12-821343-8.00017-4. KarlsruherInstitut Fur Technologie pp. 1-5.

Karmakar, K. Varadharajan, K., Tupakula, V. (2017). Networks (SDN). In Fourth international conference on Software Defined Systems. pp.112-117.

Lawal, B. H., and Nucy, A. T. (2018). Real –time detection and mitigation of distributed denial of service (DDoS) attacks in software defined network (SDN). 26th Signal processing and communication Applications conference (SLU). pp. 1-4.

Mahajan D., and Sachdeva M., (2013), DDOS Attack Prevention and Mitigation Techniques- A Review, *International Journal of Computer Applications*. 67: 0975-8887.DOI: 10.5120/11504-7221.

Monila, E., and Jacob, E., (2018). Software defined networking in cyber physical systems: A survey *computers and Electrical Enginering* 66 (11), 14-20.

Mousa, M., Bahaa-Eidin, A. M., and Sobh, M. (2016). Software defined networking concepts and challenges.11th International conference on comp engineering & Systems (ICCES) pp. 79-90.

Muragaa, W. H., Seman, K., and Marhusin, M. F. (2017). A POX controller module to prepare a list of flow Header information extracted from SDN traffic, World Academic of Science, Engineering and Technology. *International Journal of Computer and System Enginerring*. 11(12), 1305-1308.

Nadeem, M. W., Goh, H. G., Ponnusamy, V., and Aun, Y. (2022). DDoS Detection in SDN using Machine Learning Techniques. *Computers, Materials & Continua*. 71(1), 770-790

Naoum, N. and Ross K. (2016). Exploiting P2P Systems for DDoS Attacks, Department of Computer and Information Science Polytechnic University, Brooklyn, NY 1120.

Peng T., Leekie C., and Ramamohanorao K. (2007), Survey Network based Defence mechanism countering the DOS and DDOS problems. *Computer Journal of ACM Computing Surveys*, 39 (1), 123-128.

Sachedera M., Singh G., Kumar K., and Sigh K. (2010). DDOS incidents and their Impact: *A review International Arab Journal of Information Technology*. 7 (1). pp.14-20.

Visu, P., Lakshmanan, L., Muruganantha n, V., and Cruz, M. V. (2019). Software defined forensic framework for malware disaster management in internet of things devices for extreme surveillance. *Computer communications.* 147 (5). pp. 14-20.

Xu, H., Huang, H., Chen, S., Zhao, G., and Huang, L. (2018). Achieving high Scalability through hybrid switching in software defined networking. *IEEE/ACM transaction on networking* Vol 26. pp.618- 632.

Yasar S. H. (2020). Network intrusion detection for distributed Denial of Service (DDoS) Attacks using machine learning classification techniques. Thesis submitted in the Department of Electrical and Computer Engineering, University of Victoria. pp. 13-17.

Zadar, Joshi and Tipper (2013). A survey of defence mechanism against distributed Denial of service DDos flooding attacks. IEEE communication surveys and Tutorials ISSN 2249-8958. Vol 2.

Zeebarees, S., and Ameens, Sadeeq M. (2020). Social Media networks security threats, risks and recommendations. A case study in the Kurdistan region. *International Journal of innovation, Creativity and change*. 13(1), 349-365.

Zhang, Y., Cui, L., Wang, W., and Zhang, Y. (2018). A survey on software defined networking with multiple controllers. *Journal of network and computer applications* 103 (3), 101-118.