*International Journal of Mechanical Engineering*

# DDOS ATTACK DETECTION IN DISTRIBUTED SDN SYSTEM USING DEEP LEARNING

**Ayenew Kassie Adimas [1], Mulatu Gebeyaw Astarkie[2], Gezahegn Mulusew[3], Dr. Y.Nagesh[4]**

[1]Department of Information Technology, Debre Tabor University, Debre Tabor, Ethiopia

[2]Department of Information Technology, Debre Tabor University, Debre Tabor, Ethiopia

[3]Department of Information Technology, Debre Tabor University, Debre Tabor, Ethiopia

[4] Department of Information Technology, Debre Tabor University, Debre Tabor, Ethiopia

## ABSTRACT

Software Defined Networking (SDN) provides a promising networking architecture that provides a solution for traditional networks by decoupling the control plane from the data plane. With the advantage of this features, the controller can get global view of the entire network. Since, the controller acts as the brain of the network in SDN environment. However, SDN controller is mainly attacked by security problems, among such security problems the most common one is Distributed Denial of Service (DDoS) attacks which leads to exhaustion of the system resources and causes the availability of the services given by the controller. As a result, it is critical to design DDoS attack detection mechanism to mitigate the controller attack at the initial stage. In this case, one of the most promising methods to confirm SDN security is the use of deep learning technique to detect and classify the network traffic into normal and attack. We proposed a deep learning algorithm LSTM with autoencoder to detect and classify network traffic in SDN controller. To achieve this paper, we have trained our model against the recently published CICDDoS2019 dataset were used for model evaluation. Finally, by training and testing with our model Long short-term memory (LSTM) we have achieved the highest prediction accuracy rate 98.93% and lower false positive rate of 1.07%. As compared to other benchmarking Machine Learning classification approaches Support Vector machine (SVM) and Naïve Bayes classifiers (NB), our model performance is best accuracy in protection of these network attacks.

**Key words**: SDN, DDoS attacks, Deep Learning, LSTM

## 1. INTRODUCTION

Traditional network architecture is complex and rigid due to this managing the network to satisfy changing business requirements is difficult. Hence, networking principles have remained permanent over the past decade and networks are built using more or less sophisticated switches and routers [15]. These devices are being developed usually using proprietary operating system and interfaces. Configuration of different systems also increases the probability of configuration problems. This issue leads to the incompatibility of different versions of systems from one vendor makes different networks difficult and very expensive to manage. Now due to the drawback of traditional networks scholars created a new technology called Software Defined Networking (SDN) to make networks more flexible, scalable, dynamic and to allow easier management of network devices from different vendors.

SDN is designed to decouple the network data plane from the control plane and Networking devices could also be programmed directly [8]. Which is a newly emerging network technology i.e., dynamic, manageable and cost effective. It works based on the abstraction of forwarding plane from the control plane, this abstraction makes the network directly programmable and flexible, for configuring, managing, securing and optimizing the network resources dynamically and automatically.

In SDN decisions about how packets should flow through the network can be made in the forwarding plane and Packet handling rules are sent to the switches from a controller. The controller is a software application running on a server located remotely. It helps the switches to get support or guidance for packet handling. Hence, Switches and controller communicate via the controller's south-bound interface and Applications can talk to the controller via the controller's north-bound interface, this communication is achieved by the OpenFlow protocol [16].

This centralized SDN controller communication signifies a Single Point of Failure, which makes SDN architectures to be highly vulnerable to disruptions and attacks. The distribution of the SDN control plane has been recently addressed either with a hierarchical distributed controller organization or with a flat distributed controller organization. These approaches avoid having a single point of failure and enables to scale up sharing load among several controllers. However, these distributed SDN control planes have been designed for datacenters. Nevertheless, these core benefits that are the hype of SDN are also main causes of Security attack concern, such as, DDoS attack which can bring down the whole network connection and makes network resources unavailable to it users [14]. Distributed denial-of-service (DDoS) attacks have been a real threat for network, digital, and cyber infrastructure. These attacks are capable to cause massive disruption in any Information Communication Technology (ICT) infrastructure. There could be numerous reasons for launching DDoS attacks, these includes for financial gains, political gains and network disruption. Such types of attacks can interrupt service accessibility, degrade the network capacity, reduced performance and even bring down Internet access. Increasing reliance on Internet and data centers has aggravated this problem.

In a DDoS attack multiple compromised system are usually infected with an attack and are target to a single or multiple victims in the network. The attack traffic flooding the victim uses many different spoofed source IP addresses. This effectively makes it impossible to stop the attack by only blocking traffic based on source IP addresses [17]. It is also very difficult to distinguish legitimate and attack traffic. Therefore, effective deployment of DDoS attack detection and mitigation response method has always been a critical factor for the proper network operation. This issue is even more pronounced in software defined networking technologies.

Now a day’s a number of proprietary and open-source solutions exist for DDoS attack detection and mitigation. However, these attacks continue to grow in frequency, sophistication, and severity. Recently, rapid detection and mitigation of DDoS attacks has become severely challenging as attackers growing and continue to use novel techniques to launch DDoS attacks. Due to the increasing number of DDoS attacks and growing diversity in their attack types, causing disastrous impact now this has made DDoS attack detection and mitigation the most important obligatory task in network technology.

## 2. System Architecture of Proposed model

In this section in order to eliminate DDoS attack traffic deep learning-based models with LSTM which is a kind of Recurrent neural network (RNN) technique is proposed. The model controls the coding and dynamic nature of SDN and implemented on usual DDoS attack detection mechanism. After developing the architecture of the Deep Learning model system and building of the model, we have set the placement of the DDoS attack detection module. The modules required their interconnections and the place the modules reside shall be determined. After these tasks are performed the attackers or normal users have been sent the packets to the OpenFlow Switches.

When the packet arrives at the OpenFlow switch, the packet information checked such as the information on the packet header fields. The information of the incoming packets are checked against the flow entries, if a match is found then a specified action can be executed. otherwise, the packet is sent to the controller through the southbound API using a packet IN control message. That means the Attackers or normal users have been sent the packets to the OpenFlow Switches. The proposed controllers are connected as a cluster. When the traffics arrived at the proposed POX controller cluster, they forwarded through the northbound API to traffic classifier module of Deep Learning Long Short-Term Memory of application layer.

As shown below in figure 1 we are using distributed SDN controller therefore controllers are connected as a cluster. Then when the traffic arrived at the controller cluster, they forwarded through the northbound API to traffic classifier module of Deep Learning LSTM application layer attack classifier. Then the packet is classified as an attack traffic or a normal traffic. Our proposed model framework contains LSTM modules of traffic classifier which classifies the traffic as attack or normal by using Long Short-Term Memory trained module. Generally, our model traffic flow data classifier selects a particular feature needed for attack classification and give to the classifier then the classifier module classifies the traffic flow by using Deep Learning method module that determines whether a given traffic pattern belongs to DDoS attack or normal traffic.

The reason that we decided to use LSTM with autoencoder in our proposed SDN model for DDoS attack detection is that the autoencoder is trying to learn the best parameters to reconstruct the input at the output layer. Moreover, we adapted the LSTM algorithm for our model to solve the issues of vanishing and exploding gradient problems and LSTM have its own forget gate, cell states, input gate and output gate internally.

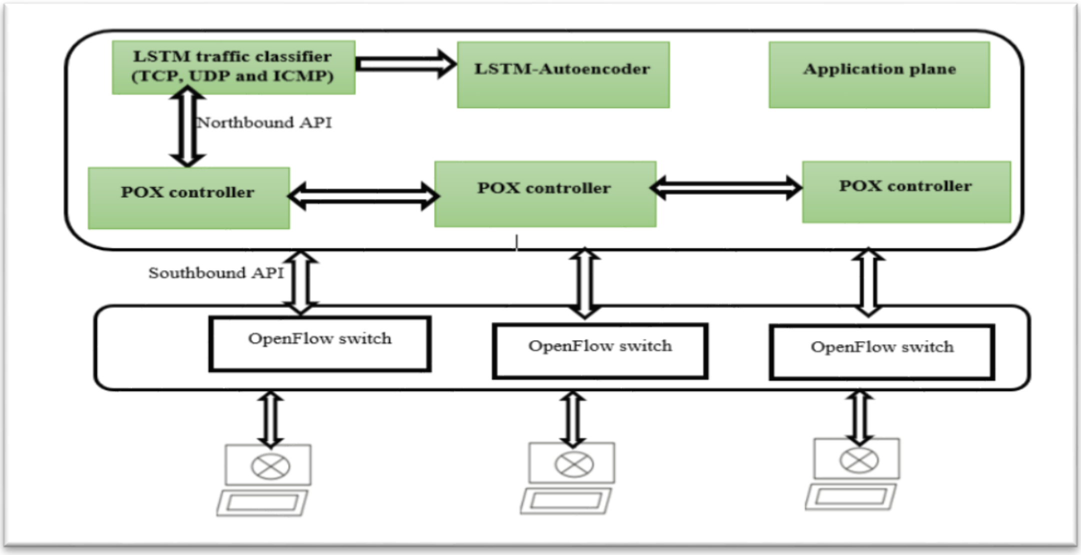

Figure 1: System architecture of proposed model

The step by step working of our proposed model system architecture is as follows: LSTM traffic classifier module selects particular features which are needed for attack detection and give to the classifier. Finally, the classifier module classifies the traffic flow by using Deep Learning method to determine whether the given traffic pattern belongs to DDoS flooding attack or normal traffic.

Table 1: **Algorithm**

<table>
<tr><td>1</td><td><b>As input:</b> It uses CICDDoS2019 dataset</td></tr>
<tr><td>2</td><td><b>As output:</b> LSTM classifier classifies the traffic (attack/normal)</td></tr>
<tr><td colspan="2">Follows the following steps:<br>A. Train CICDDoS2019 dataset with LSTM classifier then<br>B. Deploy the trained module on SDN application layer<br>C. The classifier module selects features for attack detection<br>D. Then our LSTM classifier detects features<br>E. If the classifier detects the connection as attack<br>➢ It displays the traffic as attack<br>F. Else<br>➢ If it is normal, it displays the traffic as normal<br>G. Repeat the above two steps for all incoming connection</td></tr>
</table>

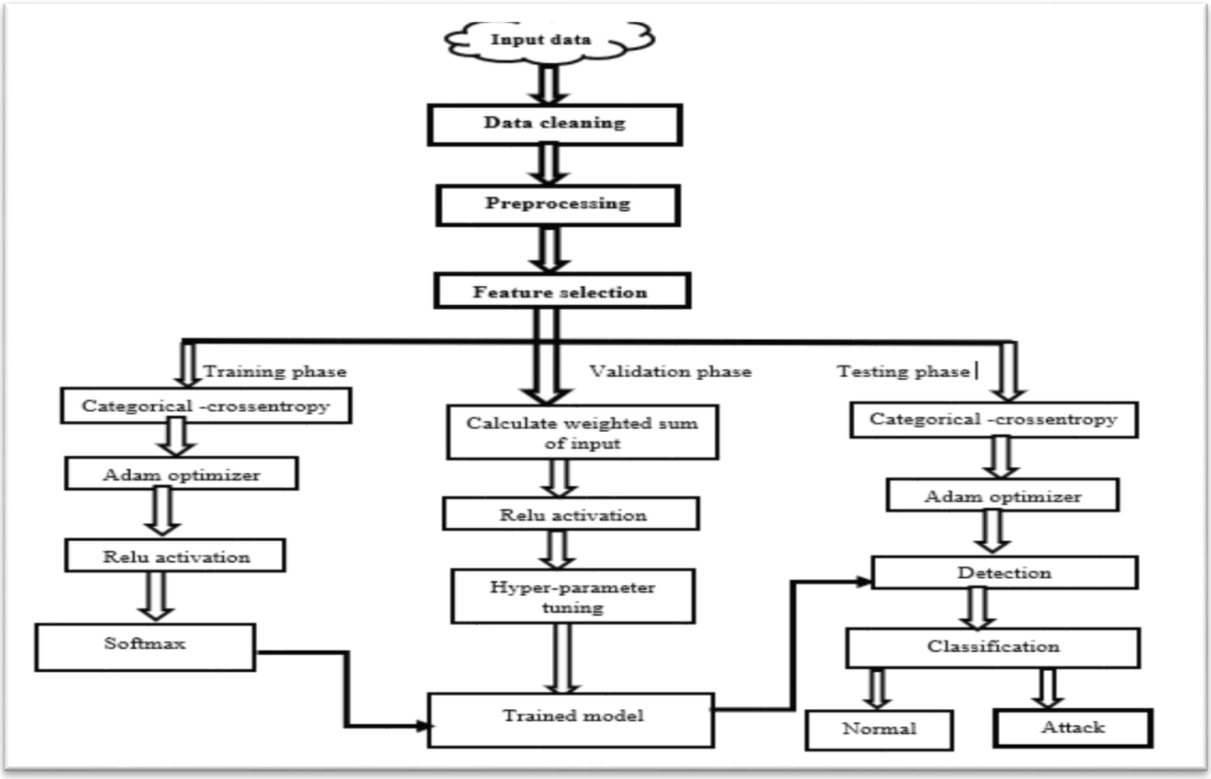

Figure 2: System training process flow

### 2.1. Proposed model Experimental result

In our study we have the normal traffic and attack dataset with total of 190,910 instances are captured as in Comma Separated Values (CSV) file format by passing several steps using python programming language. Then we make our dataset to fit for the model by shaping and reshaping, the collected dataset. In our study 28 relevant features and the total dataset was labeled as using Binary classification which is either normal class and attack class from this, we take 38183 datasets for testing and validation and 152,728 datasets for normal class.

Table 2:  Total datasets used in our model

| No | Datasets | Number of records |
|---|---|---|
| 1 | Training data | 152,728 |
| 2 | Testing data | 19091 |
| 3 | Validation data | 19091 |
| Total datasets records | 190910 | |

For our model experiment from the total dataset, we have taken 80% (152,728) of the dataset was used for training ,10% (19091) used for testing our model performance accuracy and the remaining 10% (19091) for validation to eliminate overfitting and underfit problem.

Table 3: Confusion matrix using selected features

| LSTM | | Predicted class | |
|---|---|---|---|
| | | Attack | Normal |
| Actual class | Attack | 18887 | 10 |
| | Normal | 0 | 204 |

Table 4: LSTM classification Performance Metrics

| Accuracy score | Precision score | Recall score | F1 score |
|---|---|---|---|
| 98.93% | 0.9893146 | 0.9893146 | 0.98931456 |

Table 5: Classification Accuracy of LSTM Model

| Our Model | Number test instance | Correctly classified | Incorrectly classified | Percent of correctly classified | percent of incorrectly classified |
|---|---|---|---|---|---|
| LSTM | 19091 | 18887 | 204 | 98.93% | 1.07% |

### 2.1.1. Naive Bayes model comparison result

We have compared our proposed model with Naïve Bayes Machine Learning algorithm with confusion matrix and based on performance measures with the same dataset we have tested the algorithm but it achieves less performance than our proposed approach LSTM. Short summary for Naïve Bayes performance metrics experiment.

Table 2: Naive Bayes classification Performance Metrics

| Accuracy score | Precision score | Recall score | F1 score |
|---|---|---|---|
| 0.6875049 | 0.33846 | 0.99 | 0.65476 |

**False Positive Rate**: 31.59539255924798

Table 3: Confusion matrix using selected features

| Naïve Bayes | | Predicted class | |
|---|---|---|---|
| | | Attack | Normal |
| Actual class | Attack | 25833 | 0 |
| | Normal | 418 | 11932 |

### 2.1.2. Support Vector machine (SVM)

In this experiment we have compared our proposed model with SVM Machine Learning algorithm for DDoS attack detection based on performance measures with the same dataset. Then we have tested the algorithm but it achieves less performance than our proposed Deep Learning approach LSTM but we have got best result than Naïve Bayes (NB) model. Experimental result short summary for SVM performance metrics is as follows:

Table 4 SVM classification Performance Metrics

| Accuracy score | Precision score | Recall score | F1 score |
|---|---|---|---|
| 93% | 0.88 | 0.88 | 0.93 |

Figure 17 below shows that performance evaluation of our proposed Deep Learning model with other classical benchmarking Machine Learning algorithms with the same dataset. Our proposed approach performs best, as compared to the other benchmarking algorithms.

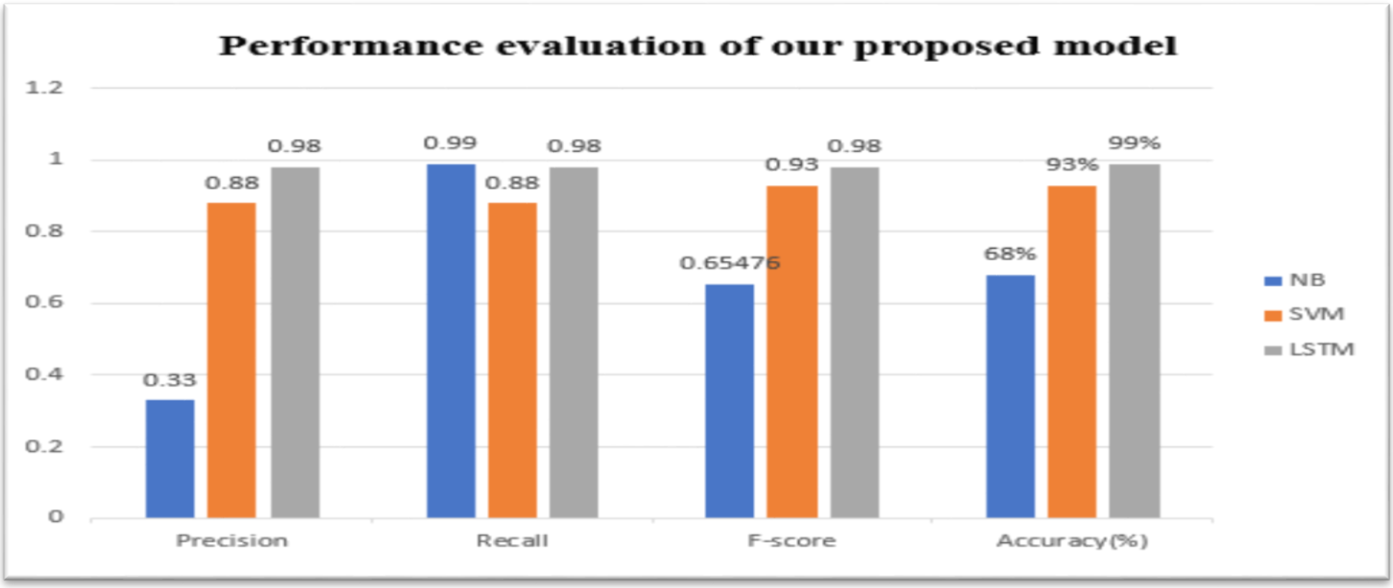

Figure 3: Our proposed model performance evaluation

In general, when we compare our proposed model LSTM with that of Support Vector Machine (SVM) and Naive Bayes (NB) model, we have achieved 98.93 % with best accuracy performance and with this model we can achieve low false positive and false negative rates in terms of precision, recall, F1-score and accuracy.

As we have shown below in figure 5 the validation loss increases at epoch 5 up to 6. This indicates the model overfitting but the model has been stopped early at epoch 8. Because the validation loss starts to goes up that indicates the degradation in model performance.

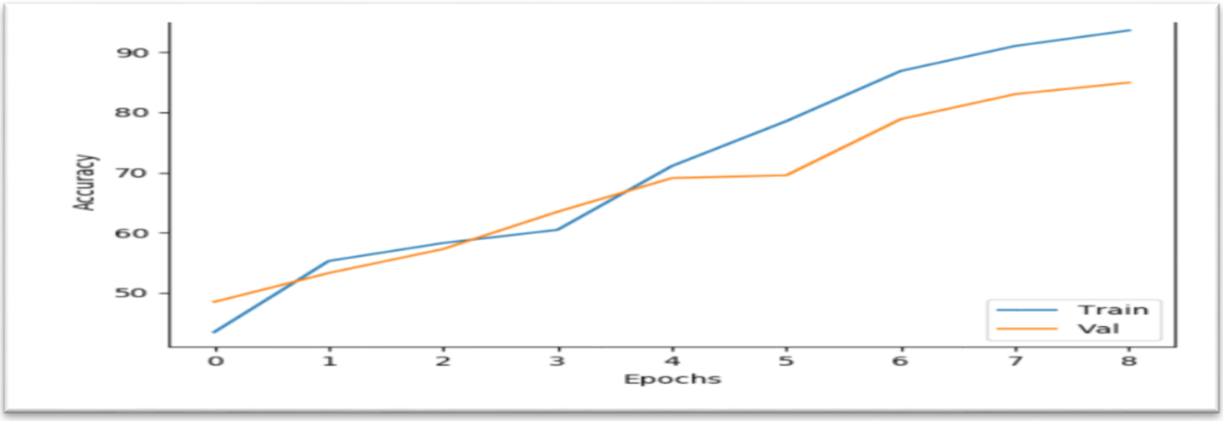

Figure 4: Our proposed model average training and validation Accuracy

As we have shown below from figure 21 the training and testing decreases starting from at epoch 2. This indicates our dataset fits the model but the model has been stopped early at epoch 8. Because the training and testing starts to goes up that indicates the degradation in model performance.

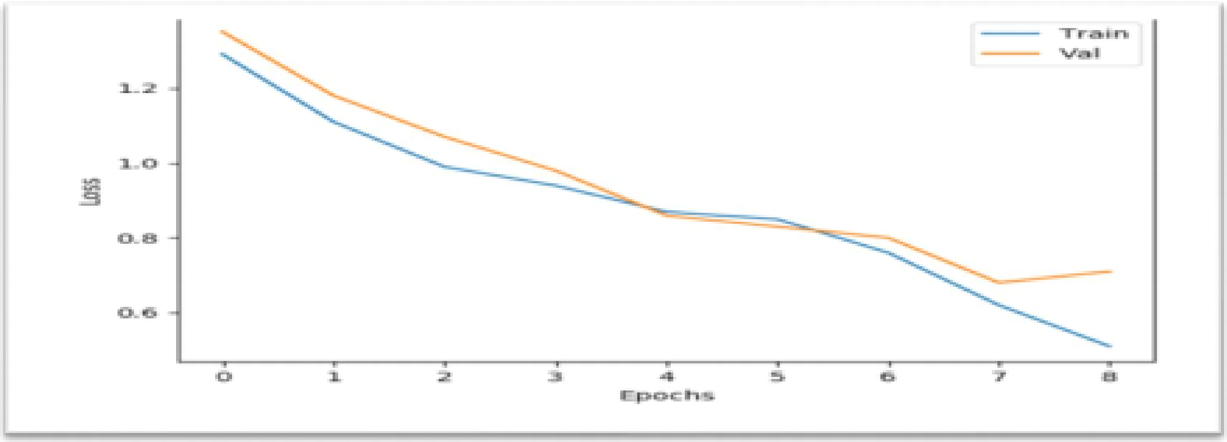

Figure 5: Proposed model average training and validation model Loss rate

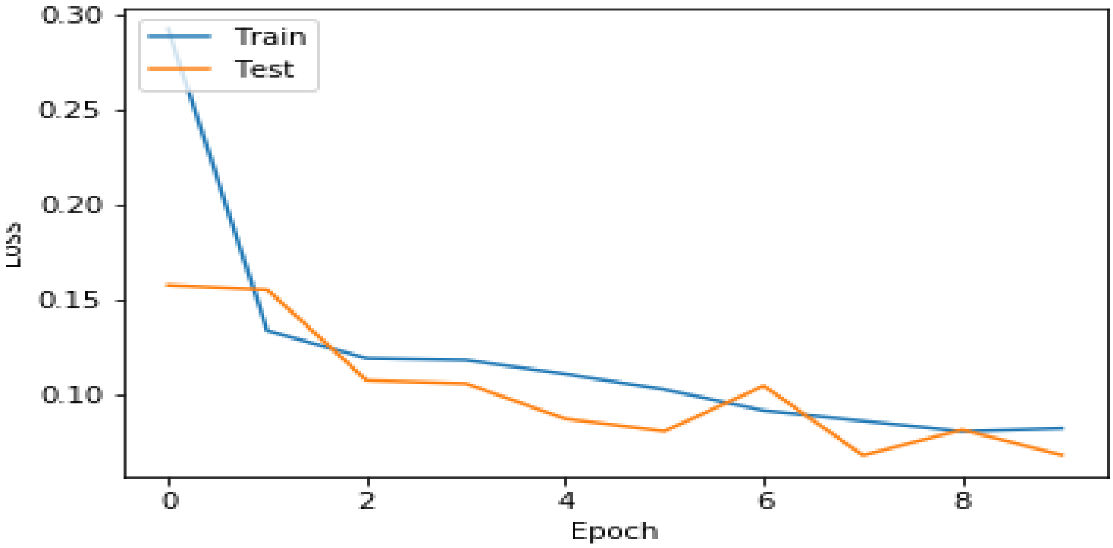

Figure 6: Proposed model training and testing model Loss rate

### 2.2. Experimentation Set up

The simulation is done on a Toshiba laptop computer with configuration Core™ i5-8250U CPU @1.8GHz, 8GB internal RAM and 4GB Graphical Processing Unit. The inner graphics card supports high-performance preference for Python and Microsoft Windows 10 operating System. The OS is Linux Ubuntu 18.04 and Mininet version 2.0.0 testbed is run in native on Ubuntu. Then we have got used Mininet 2.0.0 as a testbed which supports OpenFlow version 1.0. By using Mininet, a tree-type network of depth one with 3 clustered controllers, 3 OpenFlow switches and 24 hosts are created. OpenFlow is used for the network switches. Finally, our classification trained modules are integrated with our distributed SDN controller on the application layer using Python programming language and then we have identified and solve the problems occur on distributed SDN controllers using our classifier. Then we have identified weather the traffic is attack or normal on SDN controller and we have done mitigation of the attack. That means prevention mechanism is done to avoid the DDoS attack in its initial stage before damaging our network. When the traffic generated to our controller the classifier directly detects the packet as normal or malicious and we have also done mitigation when the network is under attack, mitigating flows are added then any host can ping each other. Therefore, in our work best performance with best detection rate with low false positive and false negative rate is achieved.

## 3. Conclusion

The proposed model distributed SDN controller for DDoS attack detection system is evaluated by applying Deep Learning method LSTM and as a benchmarking we have compare with Machine Learning models such as SVM and Naïve Bayes (NB) algorithms as anomaly classifier. Experiments was carried out for DDoS detection using LSTM, SVM and NB to classify the network traffic. Finally, we have trained and tested our classifiers by using the commonly used DDoS detection system evaluation dataset, which enables the model to classify the traffic. our experimentation results shows that accuracy of LSTM model test shows 98.93%, accuracy with a false positive rate of 1.07%, and SVM shows 93% accuracy and NB model test set shows 68%. From this we have concluded that our model has better performance than the benchmarking machine learning DDoS attack detection models in distributed software defined networking environment.

## REFERENCE

[1] Aggarwal, Y., & Kumari, U. (2019). Software Defined Networking : Basic Architecture &. April. https://doi.org/10.13140/RG.2.2.29261.69605

[2] Aladaileh, M. A., Anbar, M., Hasbullah, I. H., Chong, Y. W., & Sanjalawe, Y. K. (2020). Detection Techniques of Distributed Denial of Service Attacks on Software-Defined Networking Controller-A Review. IEEE Access, 8(August), 143985–

[3] Braun, W., & Menth, M. (2014). Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices. Future Internet, 6(2), 302–336. https://doi.org/10.3390/fi6020302

[4] Catak, F. O., & Mustacoglu, A. F. (2019). Distributed denial of service attack detection using autoencoder and deep neural networks. Journal of Intelligent and Fuzzy Systems, 37(3), 3969–3979. https://doi.org/10.3233/JIFS-190159

[5] Cui, J., He, J., Xu, Y., & Zhong, H. (2018). TDDAD: Time-based detection and defense scheme against DDoS attack on SDN controller. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10946 LNCS(August), 649–665. https://doi.org/10.1007/978-3-319-93638-3_37

[6] Deepa, V. (2019). Design of Ensemble Learning Methods for DDoS Detection in SDN Environment. 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 1–6.

[7] Elsayed, M. S., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2020). DDoSNet: A Deep-Learning Model for Detecting Network Attacks. Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020, 391–396. https://doi.org/10.1109/WoWMoM49955.2020.00072

[8] Polat, H., & Polat, O. (2020). Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models.

[9] Prasad, K. M., Reddy, A. R. M., & Rao, K. V. (2014). DoS and DDoS Attacks: Defense, Detection and TracebackMechanisms -A Survey. Global Journal of Computer Science and Technology, 14(7), 19.

[10] Rohith Gandhi. (2017). Introduction to Sequence Models — RNN, Bidirectional RNN, LSTM, GRU. Towardsdatascience, 1–8. https://towardsdatascience.com/introduction-to-sequence-models-rnn-bidirectional-rnn-lstm-gru-73927ec9df15

[11] Sahoo, K. S., Puthal, D., Tiwary, M., Rodrigues, J. J. P. C., Sahoo, B., & Dash, R. (2018). An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics. Future Generation Computer Systems, 89, 685–697. https://doi.org/10.1016/j.future.2018.07.017

[12] Salman, O., Elhajj, I. H., Kayssi, A., & Chehab, A. (2016). SDN controllers: A comparative study. Proceedings of the 18th Mediterranean Electrotechnical Conference: Intelligent and Efficient Technologies and Services for the Citizen, MELECON 2016, April. https://doi.org/10.1109/MELCON.2016.7495430

[13] Scott-Hayward, S., O'Callaghan, G., & Sezer, S. (2013). SDN security: A survey. SDN4FNS 2013 - 2013 Workshop on Software Defined Networks for Future Networks and Services. https://doi.org/10.1109/SDN4FNS.2013.6702553

[14] Singh, J., & Behal, S. (2020). Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. Computer Science Review, 37, 100279. https://doi.org/10.1016/j.cosrev.2020.100279

[15] Tayfour, O. E., & Marsono, M. N. (2020). Collaborative Detection and Mitigation of Distributed Denial-of-Service Attacks on Software-Defined Network. Mobile Networks and Applications, 25(4), 1338–1347. https://doi.org/10.1007/s11036-020-01552-0

[16] Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., & Liu, Y. (2018). A Survey of Machine Learning Techniques Applied to Software Defined Networking ( SDN ): Research Issues and Challenges. IEEE Communications Surveys & Tutorials, PP(c), 1. https://doi.org/10.1109/COMST.2018.2866942

[17] Ye, J., & Cheng, X. (2018). A DDoS Attack Detection Method Based on SVM in Software Defined Network. 2018.