International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Perceptron Based Deep Learning Technique to Enhance Quality of Service (QoS) and Security in Software Defined Network

Keerthy N.*[1], Deepa N. P.[2], Mahesh Kumar N.[3], Sapna P. J.[4], K. N. Pushpalatha[5]

Submitted: 27/11/2023 **Revised:** 07/01/2024 **Accepted:** 15/01/2024

Abstract: As network technology are always being improved, the Internet economy is quickly growing. Consequently, it is critical to pay attention to the reliability and safety of the network services offered by the ISP. A unified monitoring and control mechanism is available with state-of-the-art technologies such as Software-defined Network (SDN), however the SDN controller receives too much data to handle network traffic maintenance independently. Through the use of software-defined networking (SDN), networks are able to continuously monitor traffic, detect threats, adjust security policies, and include security services. Threats like man-in-the-middle attacks, DoS attacks, and saturation attacks are brought about by the SDN. So, the centralised controller can employ modern methods, like AI, to govern the flow of data across the network. Managing network congestion and detecting distributed denial-of-service (DDoS) assaults are the main concerns of this article. This study uses the Multilayer Perceptron (MLP) to detect DDoS attacks and connection congestion through packet loss using data acquired from the Open Flow Switch Table. Simulation results show that the proposed methodology out performs the status quo in terms of network performance.

Keywords: *Deep learning, Congestion detection, and avoidance. Software-Defined Network (SDN), Quality of Service, DDoS attack, Multilayer Perceptron (MLP)*

1. Introduction

It is becoming more difficult for internet service providers to ensure ubiquitous accessibility in this age of rapidly developing technologies in areas such as big data, cloud computing, and mobile communication (1G to 5G and beyond). In contrast, switches and routers are the backbone of a conventional network that relies on hand-crafted configurations. The introduction of new network topologies and the difficulty in diagnosing failures in physical equipment are two factors that increase the expense of maintaining traditional networks. As a centralised mechanism for the configuration of the network, Software-defined Network (SDN) [1] is utilised to circumvent this issue. Software-defined networking (SDN) is a method for dynamically and programmatically establishing systems and networks, which allows for efficient network setup. Network performance is improved and new configuration modifications to the system or network can be easily and cost-effectively implemented with SDN, in contrast to traditional network configuration. The three open flow switches are linked to the centralised monitor and controller (C0), as shown in Fig. 1, (C0: Controller, S1,S2, S3: OpenFlow Switches) which represents the architecture of this software-defined network [2]. From a central location, system administrators handle network traffic and configurations in the SDN.

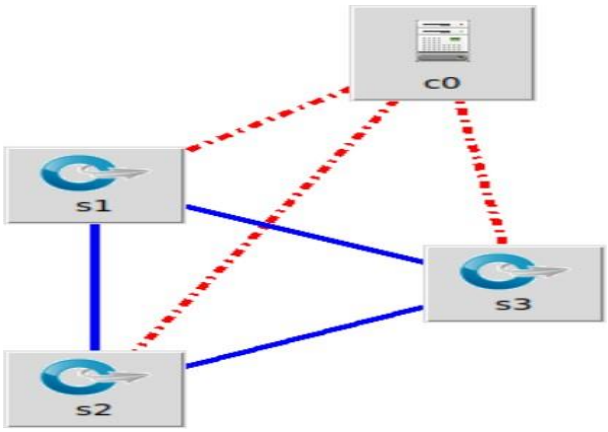


Fig. 1: The Architecture of the SDN.

Among 1he many facets of network management are load balancing, policy design for routing, fault tracking, identification of congestion, classification of packets as secure or unsecured, and provision of exceptional security to end users or designated users. An SDN centralised controller can manage the aforementioned types of traffic, but it can't always reroute massive amounts of data, identify network attacks, prioritise packet classification, adjust to changes in network configuration, etc. To get around this, you can use the Centralised Control (C0) that is based on AI to improve performance. Network congestion and distributed denial-of-service (DDoS) attack detection are the primary focus of this paper. Distributed denial of service (DDoS) attacks occur when an adversary node tries to overwhelm a target node or its surrounding area with an excessive amount of Internet traffic in an effort to disrupt

[1] *Department of Electronics and communication engineering, Global Academy of Technology.*
[2,3,4,5] *Department of Electronics and communication Engineering, Dayananda Sagar college of Engineering.*
**Email: keerthygowda18@gmail.com*

normal communication or network traffic. Attack on the SDN network depicted in Fig. 2. Using the NSLKDD training data set, Abubakar et al. [3] investigated multiple approaches and proposed a scalable system flow intrusion detection system.

A DL framework named RNN (Recurrent neural network) was proposed by authors M. S. Elsayed et al. [4] for recognising network assaults using the recently released data set CICDDoS2019, which comprises a variety of DDoS attacks. Despite using the NSLKDD and CICDDoS2019 data sets as a foundation, a comparable dataset is being used in the simulation after a pre-processing technique described in the study. For a framework that can find and prevent DDOS attacks on Internet of Things devices, Mahesh Kumar N et al. [5] advocated for an adaptive ML-based approach. By utilising a separate SDN controller and several ML approaches, this framework reconfigures system or network resources for lawful hosts, mitigating these types of assaults on process or Open Flow (OF) switches. Classifying network assaults using a model developed using the SVM approach was proposed by J. Ye et al. [6]. This simulation makes use of a Floodlight controller and an SDN environment. The algorithm employed here achieves an accuracy level of about 95%. Using SAE-MLP to categorise attacks in SDN according to network traffic was proposed by N. Ahuja et al. [7]. On the other hand, using a large data set to train a Multilayer Perceptron (MLP) model would yield better results in less time Mahesh Kumar N et al. [23]. In order to detect and eliminate attack flows from a software-defined network, this paper proposes a technique.

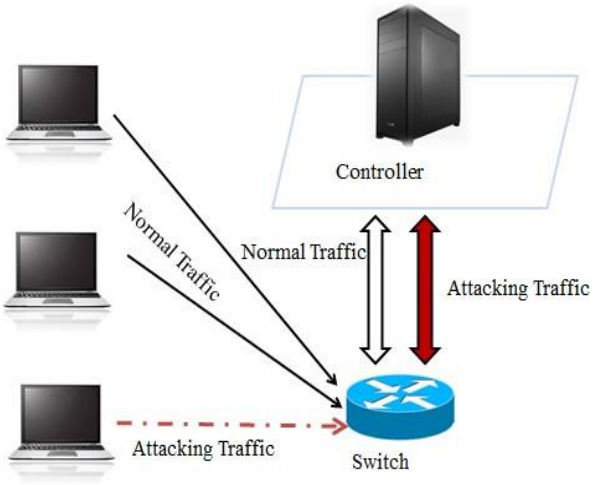


Fig. 2: The SDN infrastructure with DDOS attack.

When one connection prevents another from connecting, resulting in link failure and packet loss during data transfer, this phenomenon is known as network congestion, which reduces Quality of Service (QoS). S. M. Mousavi et al. [8] suggested a method to reduce SDN connection congestion using machine learning to forecast SDN congestion based on link status and OpenFlow switches. The suggested method makes use of UDP-based congestion detection, which relies on a link's packet loss and uses a 20% cutoff to identify congestion. When congestion is observed in the network, the next shortest route or path is determined using Multilayer Perceptrons using the Dijkstra algorithm..

The SDN network uses a Multilayer Perceptron (MLP) to detect and avoid congestion and DDoS attacks. As a subset of Machine Learning, ML Perceptrons are feed-forward ANN system models that use a non-linear approach to outperform linear perceptrons in prediction and classification analyses. A Multilayer MLP has three layers—an input(I/P) layer, a hidden layer, and an(O/P) output layer—as depicted in Fig. 3. When making predictions or classifying data, MLP employs a backpropagation approach.

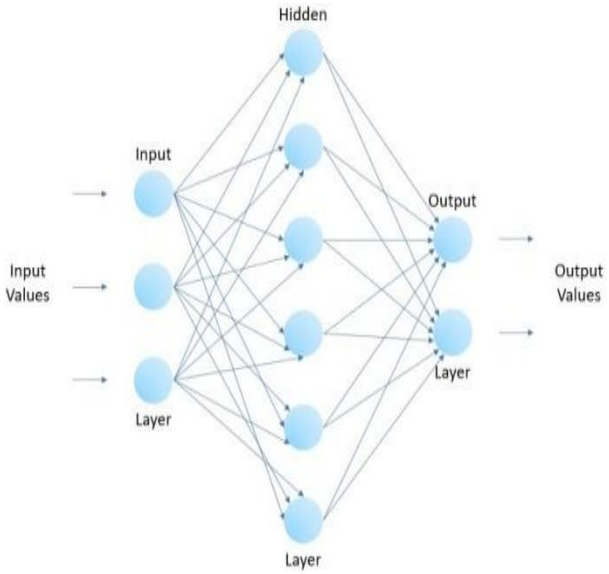


Fig. 3: The architecture of perceptron layer in an MLP.

The research article's abstract can be summarized as follows: The initial part covers project introduction, background information, and literature review, while subsequent sections delve into the proposed methodology for identifying and mitigating DDoS attacks and network congestion (Sections II and III). Section IV presents the findings from the comparative analysis, and the conclusion is provided in Section V

2. Discovering and preventing distributed denial of service attacks

Intrusion detection systems, whether they are software or hardware based, monitor a network for any signs of suspicious activity or rule violations. A DDoS assault against intrusion detection systems is the primary subject of this article. By following the steps outlined in this article, SDN networks can be protected from the DDoS attacks and spoofing off attack traffic can be stopped. For packet management, the SDN controller searches entries in the flow table for possible interfaces to send packets to. In Table 1 you can see the structure of the flow matrix. In the SDN OpenFlow switch, the process or flow ttable is the basic dala or information structture for managing the forwarding policies and quickly transmitting major network data. The rules for data forwarding are included in

the f1ow table, which is composed of many f1ow entries. This table governs the forwarding of packets by the switch. The flow table is structured with header fields, counters, and actions for each entry.

Table 1: Flow table structure [5].

Secure Channel
Flow Tabel

Header Fields	Counters	Actions

Ingress Port	Ether Source	Ether Dst	Ether Type	Vlan ID	Vlan Priority	IP Src	IP Dst	IP Port	IP TOS Bits	TCP/UDP Src Port	TCP/UDP Dst Port

Gathering process ttable slatus messages and information from the OpenFlow switch is known as flow state collection. When it comes to keeping an OpenFlow-based network environment running smoothly, this procedure is invaluable. The OpenFlow switch regularly replies to the Ryu controller's requests for flow statistics, and this network keeps track of all of the statistics for each switch. The network topology that is taken into account for DDoS assaults and mitigation in this article is shown in Fig. 4, C0: Controller, S1: OpenFlow Switches, H1 to H5: Hosts. There is a single open flow switch, five hosts, and a controller in it. We scrupulously document all traffic, whether it is malicious or innocuous. An overwhelming number of packets sent to a single target from a variety of fake IP addresses constitutes a distributed denial of service (DDoS) attack.

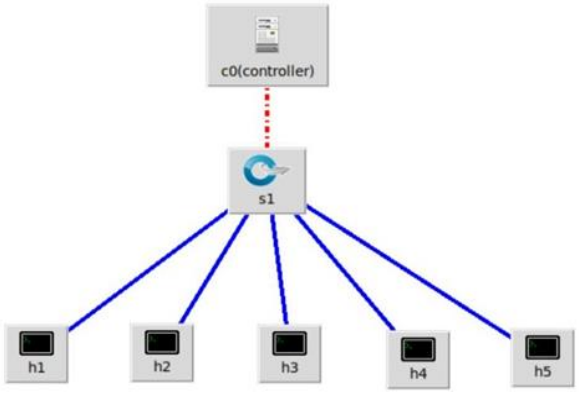


Fig. 4: The Network topology for a DDoS Detection Network.

By comparing the process table characteristic value information to typical attack flows in the system or network, one can identify malicious attack flows in the network by their dissimilarities and irregularities. The data from the flow table is used to consider the following five parameters [5].

i. The **speed of the source IP**, also known as SSIP, is defined as "the number of sources addresses (IP) per unit of T time" in equation 1, as figure 8(a) showed:$SSIP = (Sum\_IPsrc) \ / T$ (1)

*The **source IP** number is represented by Sum_IP, and the sampling interval is denoted by (T) time. A surge in the total number of source IP addresses and a deluge of strange data packets are both indicators of an assault on the system. Figure 8(a) shows an increase in the number of attacks caused by random packet forging, which interrupts the network's genuine data transfer.*

ii Fig. 8(b) displays the **Speed of Flow Entries** (SFE), which is determined using equation 2.

$$SFE = N/T \quad (2)$$

There are a total of N flow entries and an average of T packets. A packet attack is characterised by an abnormally high number of entries. Figure 8(b) shows that the average number of the f1ow entries per unit of the (T) time is significantly lower during a network attack, whereas the number grows greatly.

iii. The **standard deviation of flow packets**, or SDFP, is the total number of packets in a given time period (T), as shown in Figure 8(c) and computed using equation 3.

$$\text{SDFP}=\sqrt{1/N \sum_{i=1}(\text{packets}i - Mean\_Packets)^2} \quad (3)$$

"Where $\frac{1}{N}\sum \quad N\ i = 1\ ,\ i$ is the total avg no. of packets in a given time period" . A string of anamolus flow entries is sent to execute an attack. "The sum of all entries for each period of N." In order to create an attack effect, data packets used in general tend to be relatively short in size, and 1he SDFP is lower than the typical network traffic f1ow. As shown in figure 8(c) by a modest spike in the graph, the f1ow pieces are less bulky compared to the typical network traffic.

iv. The **deviation of the flow bytes (SDFB**) is defined in equation 4 as "the SD is the no. of bits in the T period," as illustrated in Figure 8 (d).

$$\text{SDFB}=\sqrt{1/N \sum_{i=1}^{N}(\text{bytes}i - Mean\_Packets)^2} \quad (4)$$

Where$\frac{1}{N}\sum_{i=1}^{N} \quad N\ byt$, has an average number of packets during the T-period. A malevolent node will provide fewer data packets than regular traffic in order to lessen packet burden, and the SD flow bits will be lower as well.

v. The **ratio flow pair**, which is displayed in Figure 8(e) and is computed using equation 5, indicates the fraction of total flow entries that are interactive.

$$RFIP = \frac{2*Pair\_Sum}{N} \quad (5)$$

This variable stores the total number of interactive f1ow entries. The amount of entries in the interactive flow each T period drops when an attacker uses a flood of bogus source addresses.

Using flow table statistics, multilayer perceptrons (MLPs) may distinguish between DDoS attack traffic and the network's original, non-attacking traffic; this is the goal of this study. This sample sequence is classified as either norma1 or abnormal traffic based on the five characteristics. The two states of the network are represented by 0 and 1, respectively. This method aids in the effective mitigation of attack traffic in the network. After recording both attack and normal traffic samples, we label all attack traffic as "1," and all normal traffic as "0." Then, we train the ML model with 80% of the dataset and test it with 20% using an MLP classifier to determine its parameters. It is common practice for DDoS attacks to employ MLP in a two-stage classification: Figure 5 shows a flow diagram of a DDoS attack mitigation process utilising MLP.

To begin, SDN's adaptive network data collecting system takes flow table features into account while extracting data; this allows the system to be trained to distinguish between legitimate and malicious traffic. Upon detecting a distributed denial of service (DDoS) assault, The controller will be notified by MLP to remove the attack flows from the network.

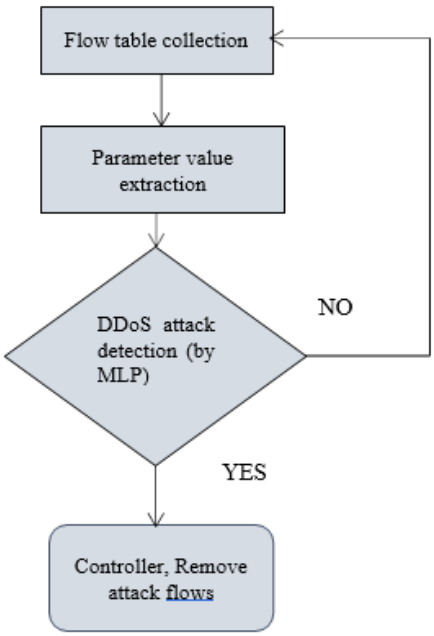


Fig. 5: The Deployment and Mitigation Flow Diagram for DDoS.

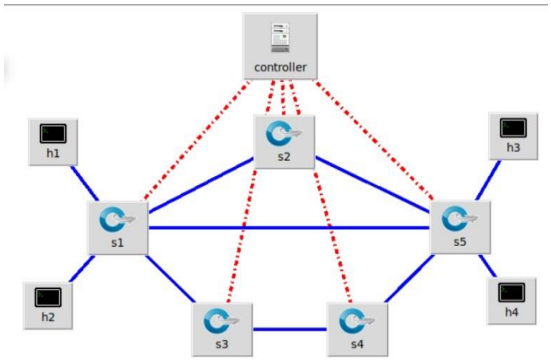
3. Quality of Service: Identifying and managing link congestion

The goal of quality of service (QoS) is to help organisations better understand their network's packet loss, latency, and jitter as well as to optimise the performance of various applications running on that network within the constraints of that network's capacity. When there is a lot of attack traffic or a network node is carrying more data than it can handle, this is called a bandwidth hog, and it affects quality of service. Because of this, it causes packet loss, impacts network queuing, and prevents the establishment of new connections between lines.

Software-defined networking (SDN) simplifies complex network tasks like traffic management and routing via programmability enabled by logically centralised administration. It needs to understand the whole SDN architecture's network design so it can set up data communication paths among any data plane origin pairs. This paper discusses how dynamic routing makes use of the Dijkstra algorithm, often known as the Shortest Path First (SPF) algorithm, to find the shorttest path between the active nodes in a system or network structure by considering the link weights. Before updating the controller to use the shortest route in the network, the algorithm called Dijkstra determines the shortest path for transmission from the sender node to the receiver node.

Congestion detection makes use of the network structure depicted in Figure 6, Controller: S1 to S5: Open Flow Switches. H1 to H4: Hosts. which includes four hosts, three variable pathways, and a single controller. There are three separate methods for pings to get from h1 to h3. In this case, the controller is set to route traffic by the shortest route found by the Dijkstra a1gorithm or the Shortest Path First (SPF) a1gorithm, regardless of whether the quickest path from h1 to h3 is via (h1, s1, s5, h3).

Fig. 6: A topology for networks that enables the detection and control of congestion.



The suggested methodology consists of three stages, and the primary emphasis of this paper is on forecasting link congestion in UDP services [10]:

- Gathering sufficient data to train the model.
- Setting up the multi-layer perceptron (MLP) architecture on the control p1ane by utilising the trainning the dataset.
- The usage of MLP for detecting link congestion and selecting paths, which use the Dijkstra a1gorithm to determine the shortest route.

The SDN network's congestion detection architecture is illustrated in Fig. 7. Finding the shortest path from any given source node to any given destination node is the first task of the application plane routing module. Three modules make it up: a topology or network discovery module, a time monitor, and an MLP model; the Ryu controller is located on the contro1 plane. The OpenFlow switches' connection state may be detected, the data plane's regular data exchange information can be monitored over

time, and a recommended technique can be used to intelligently prevent congestion.

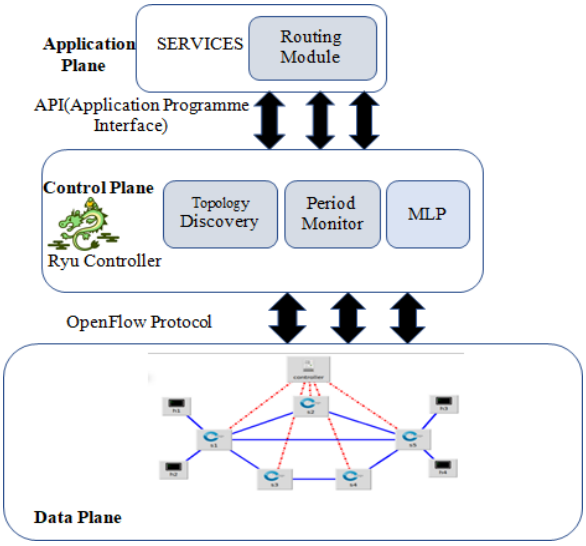


Fig. 7: Architecture of the SDN in congestion control.

Topology Discovery accomplishes this. Connection failure and packet loss happen on a specific link as a result of congestion, which happens when there is a lot of traffic flowing between the lines. Using the packet loss and loss rates as inputs, the MLP algorithm may identify connection congestion. In addition to reducing the burden of dynamic routing, the MLP algorithm, when used efficiently, provides the capacity to learn and make better routing decisions based on past experiences. Gathering training data based on link packet loss, using a 20% cutoff rule, is the initial stage. Using a generation rate of 1 for each allocated path and with the feature extraction and congestion flag set to 1, all data flows from source to destination are logged.

Once the training dataset is prepared, the MLP system or model is iteratively trained using 80% of the data set, while the remaining 20% is used for testing purposes. The routing algorithm learns from the trained model and can now identify connection congestion when the threshold exceeds 20%. Therefore, the network's controller may not only identify a suitable path to minimise congestion but also predict the related output from new input.

4. Results And Discussions

The foundation of a software-defined network (SDN) environment is a Mininet testbed equipped with Ryu Controller and OpenFlow Switch. The paper's system and network architecture for DDoS attack detection and congestion is shown in Figures 4 and 6. In a closed-loop system, five hosts, one switch, and one controller are taken into account for DDoS attack detection. H4 is seen as someone who could be a victim. In order to train models for attack detection, these devices gather data on both normal and attack traffic.

Various protocols, such as TCP, UDP, and ICMP flood, are employed to produce typical traffic. Attacking networks with suspicious traffic is made easy with Hping3. Making malicious traffic that looks like typical TCP, UDP, and ICMP floods is part of this. To teach the model to differentiate between malicious and benign network traffic, we employ both types of data. The samples of malicious and benign traffic are shown in Table 2 and Table 3, respectively. The results of these two types of traffic are displayed in Figure 8(a), 8(b), 8(c), 8(d), and 8(e). Every three seconds, the data is retrieved from the OpenFlow switch in the network.

Table 2: Standardized traffic samples.

SFE	SSIP	RFIP	SDFP	SDFB
0	0	1	524.3697	1121630
0	0	1	497.3681	1054327
0	0	1	497.375	1053553
2	0	1	472.4735	9945132
4	0	1	607.3261	1292278
2	0	1	573.8044	1198618
0	0	1	570.9191	1189725

Attack traffic commonly employs pseudo-random IPs and ports. Over time, origin IP and origin ports increase. Figures 8(a) and 8(b) exhibit similar growth patterns. Even when intruders deliver massive data packets under normal operating conditions, they are short and unmodified to maximise attack efficacy. As demonstrated in figures 8(c) and 8(d), the standard deviation of process or flow packets and stream bits in a T period is mild and fluctuates only slightly.

Table 3: Anomalous Traffic samples:

SFE	SSIP	RFIP	SDFP	SDFB
944	942	0.004242	0	0
649	649	0.002513	0	0
467	467	0.001943	0	0
330	330	0.001472	0	0
578	578	0.000873	0	0
644	644	0.000765	0	0
642	642	0.000682	0	0

The two defining features are huge and visibly changing during regular times, yet they are minute and subtly changing during abnormal periods. The target site is overwhelmed with requests at that time because malicious attackers frequently use virtual random source 1P addresses and source port numbers. Consequently, as illustrated in

Figure 8(e), the substantial amount of interaction flow drops sharply, and there are periods when no interacting process or flow entries at all. Under normal circumstances, the RFE is relatively high, and it changes with the normal alter or change.

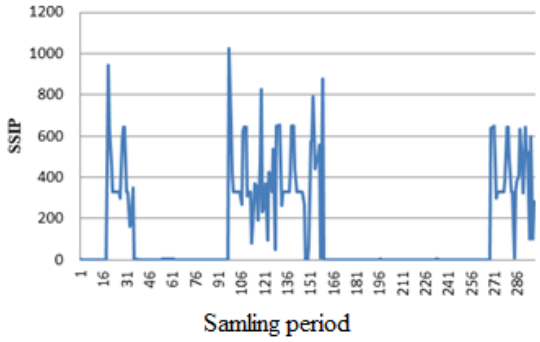


Fig. 8(a): Rate of SSIP

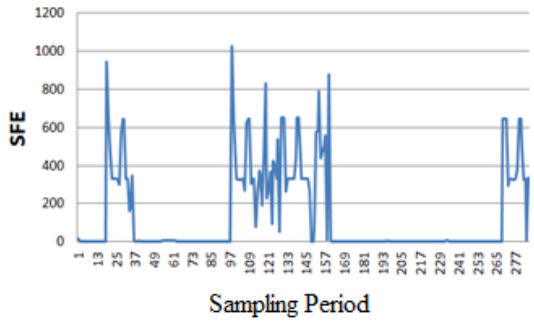


Fig. 8(b): Rate of SFE

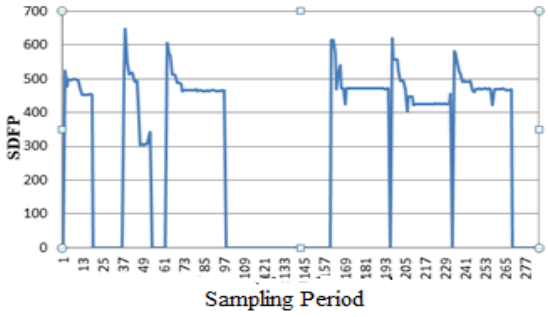


Fig. 8(c): Rate of SDFP.

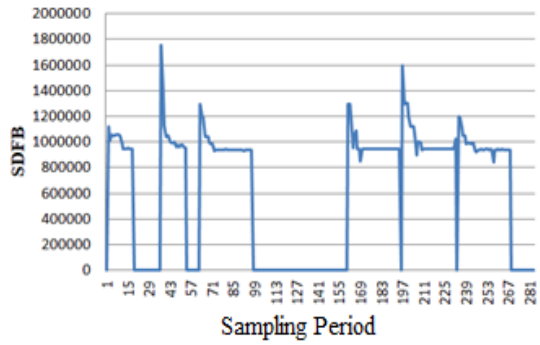


Fig. 8(d): Rate of SDFB.

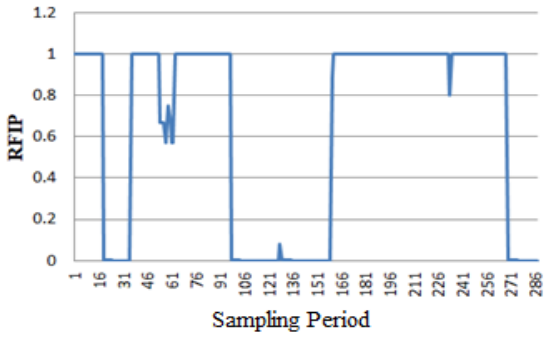


Fig. 8(e): Rate of RFIP.

Figure 9 displays the results of applying the MLP algorithm to the SDN network topology in order to distinguish between malicious and benign network traffic.

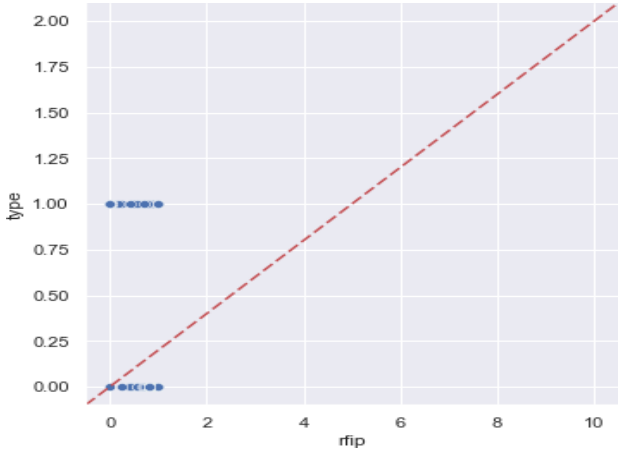


Fig. 9: RFIP classification of standardized and anomalous traffic.

In Figures 10(a) and 10(b), we can see Decision Boundary Graph analysis setting where MLP was used to classify two parameters. A decision boundary is a line that, when applied to two features, displays the distribution of samples along the two classes: one class on one side and the other class on the other. A distinct group is delineated by the line. Using a multi-layer perceptron (MLP) model, decision boundary distinguishes between malicious and benign traffic in this article.

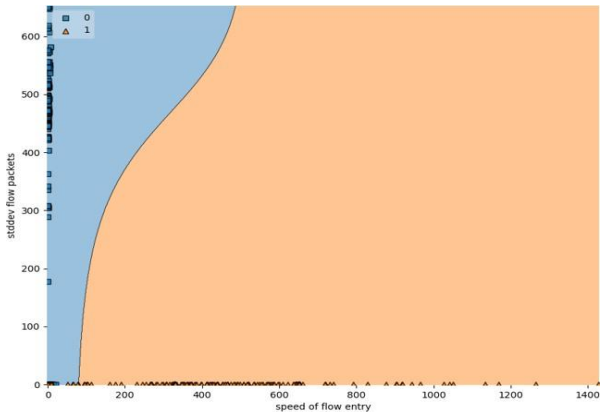


Fig. 10(a) Decision boundary : SDFP v/s SFE

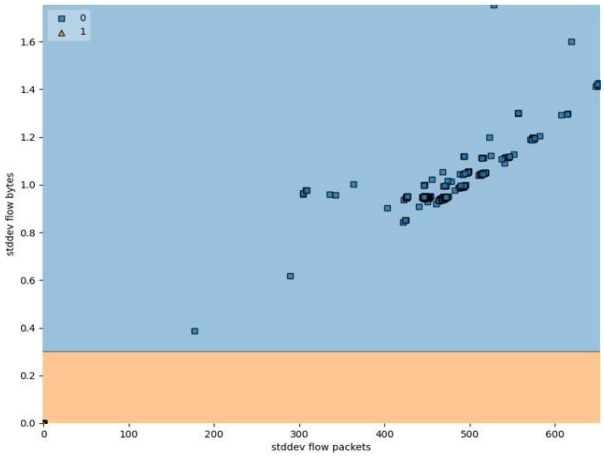


Fig. 10(b) Decision boundary : SDFB v/s SDFP

4 hosts, 5 OpenFlow switches, 3 link pathways, and one congestion controller are used in the experiment. The topology used in this study has 33 distinct path configurations due to the three changeable source-to-destination paths. Each route configuration has its congestion probability given, and The controller builds the paths using the MLP model that was learned in the earlier phase. So, if the route configuration from hl to h3 is throu Sl, S5, or (1, 1, 1), and congestion happens on this 1ink, the control1er will update the route configuration with a new shortest 1ink path if the congestion exceeds a specified threshold limit of 20%. The percentage of packet drops during congestion is used to set the threshold limit. Figure 11 shows the current rate of packet loss for a specific link.

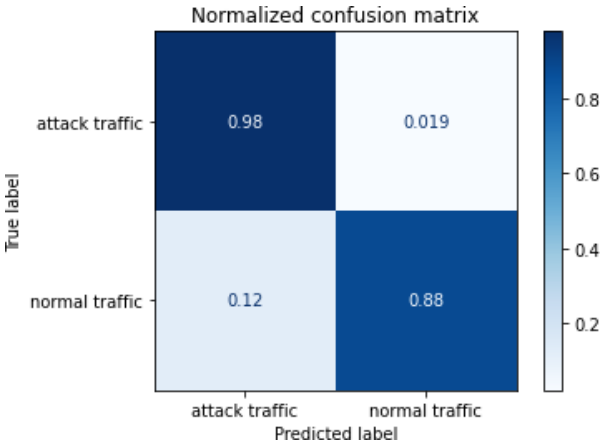


Fig. 11: Rate at which data packets are lost due to connection congestion.

We compare the MLP algorithm's performance to that of other approaches on a variety of graphs. The ROC curves for the MLP and SVM algorithms are shown in Figures 12(a) and 12(b), respectively..

It provides the model's true positive and false positive rates. As illustrated in the graphic, MLP outperforms SVM in terms of accuracy. MLP provides around 99.89% accuracy for this model, while SVM [5] provides 95.24% accuracy.

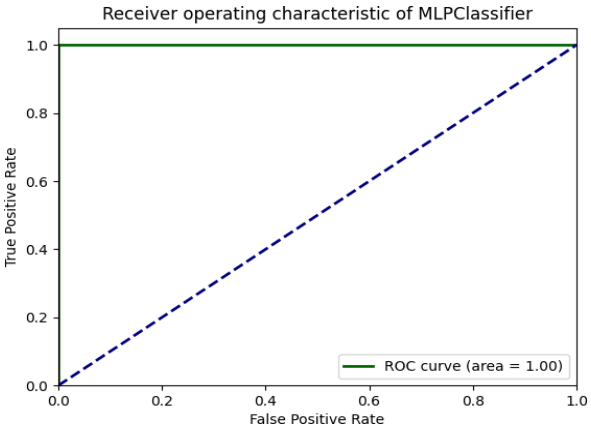


Fig. 12(a): Receiver Operating Characteristic Curve of MLP Classifier

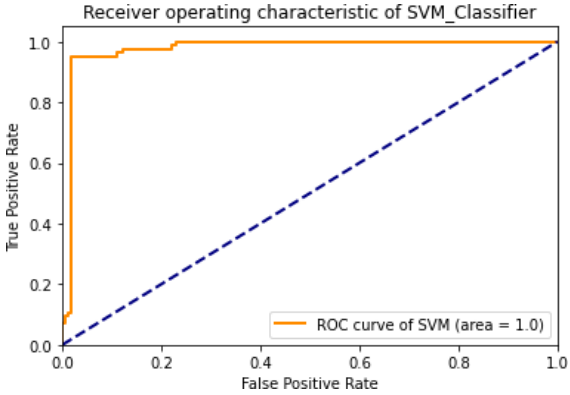


Fig. 12(b): Receiver Operating Characteristic Curve of SVM Classifier.

The MLP method is also tested using a confusion matrix. Figure 13(a) shows the confusion matrix, while Figure 13(b) shows the MLP normalised matrix. Algorithms' true label and rate of false-positive can be found using the confusion matrix, a performance statistic

.

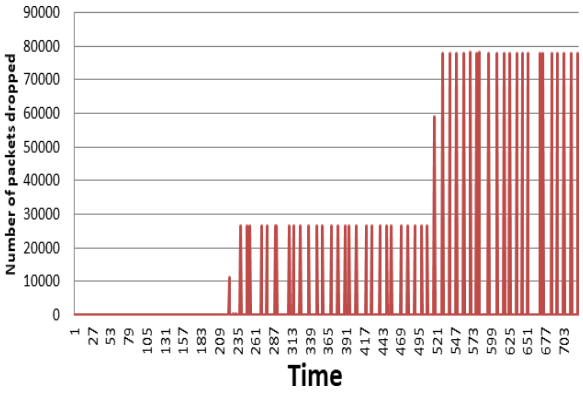


Fig. 13(a): Normalized Matrix of MLP.

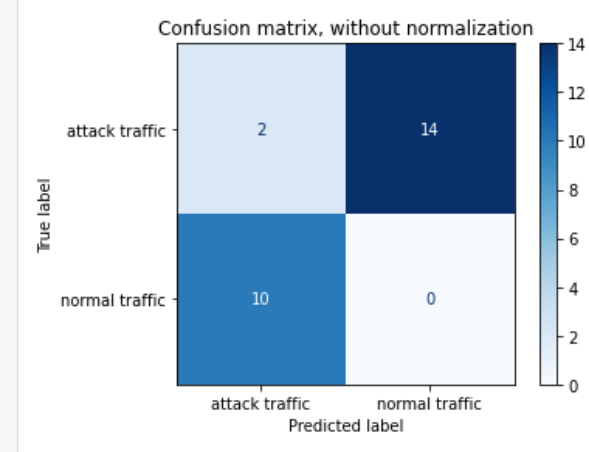


Fig. 13(b): Confusion Matrix of MLP.

Figure 14 shows a comparison of the percentages of three algorithms: the MLP technique, SVM, and Gaussian Navy Bias (GNB). The accuracy scores for GNB(78%), SVM(95.24%), and MLP(99.89%) are all generated using the respective methods. In terms of accuracy, the comparison analysis shows that the Multilayer Perceptron algorithm performs better than the other 2 methods.

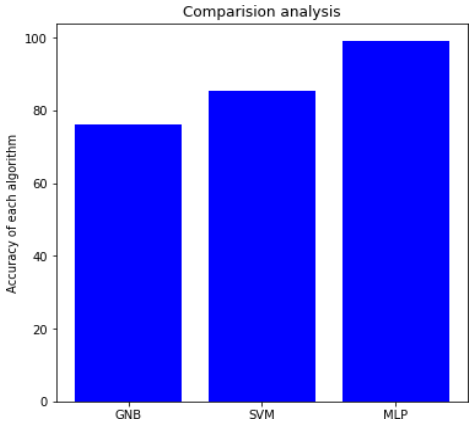


Fig. 14: Comparing three algorithms.

5. Conclusion and Future work

The SDN administration of configurations a11ows programmatic and dynamic network resource control. Centralised control improves performance, but inability to handle heavy traffic volumes causes network security and QoS issues. We use a multilayer perceptron (MLP) to block DDoS attacks on SDN based on five parameters and the Dijkstra algorithm for congestion control. The proposed research algorithm is compared to GNB and SVM. GNB and SVM are less accurate than MLP, according to simulations. Modern methods in a central controller can overcome security issues and preserve QoS.

IT support and content suppliers like Amazon, Twitter, Facebook, and others profit. In the SDN virtualized environment, more accurate and efficient network fault detection systems will be implemented and evaluated. An intelligent routing system can also intervene if a link between two open flow switches breaks, rather than just detecting congestion.

References

[1] W. Xia, Y. Wen, C. H. Foh, D. Niyato and H. Xie, "A Survey on Software-Defined Networking," in IEEE Communications Surveys & Tutorials, vol. 17, no. 1, pp. 27-51, Firstquarter 2015, doi: 10.1109/COMST.2014.2330903.

[2] F. Hu, Q. Hao and K. Bao, "A Survey on Software- Defined Network and OpenFlow: From Concept to Implementation," in IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 2181-2206, Fourthquarter 2014, doi: 10.1109/COMST.2014.2326417.

[3] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," 2017 Seventh International Conference on Emerging Security Technologies (EST), 2017, pp. 138-143, doi: 10.1109/EST.2017.8090413.

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

[5] Mahesh Kumar N and Siddesh G.K, "Comprehensive Survey On Network And Cross Layers Of Cognitive Radio Networks," International Journal of Scientific & Technology Research, vol. 8, no. 9, pp. 230-235, September 2020.

[6] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detection method based on SVM in software defined network," Secur. Commun. Netw., vol. 2018, pp. 1–8, 2018, doi: 10.1155/2018/9804061

[7] N. Ahuja, G. Singal and D. Mukhopadhyay, "DLSDN: Deep Learning for DDOS attack detection in Software Defined Networking," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 683-688, doi: 10.1109/Confluence51648.2021.9376879.

[8] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in Proceedings of the 2015 International Conference on Computing, Networking and Communications, ICNC 2015, pp. 77– 81, Garden Grove, Calif, USA, February 2015.

[9] Iwata K, Ito Y. Proposal of Multi-Pathization Method of UDP with SDN for NFS[C]//2018 International Symposium on Networks, Computers and Communications (ISNCC). IEEE, 2018: 1-5.

[10] J. Wu, Y. Peng, M. Song, M. Cui and L. Zhang, "Link Congestion Prediction using Machine Learning for Software-Defined-Network Data Plane," 2019 International Conference on Computer, Information and Telecommunication Systems (CITS), 2019, pp. 1-5, doi: 10.1109/CITS.2019.8862098.

[11] Khan, S.; Bagiwa, M.A.; Wahab, A.W.A.; Gani, A.; Abdelaziz, A. Understanding link fabrication attack in software defined network using formal methods. In

Proceedings of the IEEE International Conference on Informatics, IoT, and Enabling Technologies, Doha, Qatar, 2–5 February 2020; pp. 555–562.

[12] B. S. Kiruthika Devi, G. Preetha, G. Selvaram and S. Mercy Shalinie, "An impact analysis: Real time DDoS attack detection and mitigation using machine learning," 2014 International Conference on Recent Trends in Information Technology, 2014, pp. 1-7, doi: 10.1109/ICRTIT.2014.6996133.

[13] Yang Y., Wang J., Zhai B., Liu J. (2019) IoT-Based DDoS Attack Detection and Mitigation Using the Edge of SDN. In: Vaidya J., Zhang X., Li J. (eds) Cyberspace Safety and Security. CSS 2019. Lecture Notes in Computer Science, vol 11983. Springer, Cham. https://doi.org/10.1007/978-3-030-37352-8_1.

[14] F. Naeem, G. Srivastava and M. Tariq, "A Software Defined Network Based Fuzzy Normalized Neural Adaptive Multipath Congestion Control for the Internet of Things," in IEEE Transactions on Network Science and Engineering, vol. 7, no. 4, pp. 2155-2164, 1 Oct.- Dec. 2020, doi: 10.1109/TNSE.2020.2991106.

[15] T. Zhang and S. Mao, "Machine Learning for End-to- End Congestion Control," in IEEE Communications Magazine, vol. 58, no. 6, pp. 52-57, June 2020, doi: 10.1109/MCOM.001.1900509.

[16] M. Gholami and B. Akbari, "Congestion control in software defined data center networks through flow rerouting," 2015 23rd Iranian Conference on Electrical Engineering, 2015, pp. 654-657, doi: 10.1109/IranianCEE.2015.7146295.

[17] Yifei Lu and Shuhong Zhu, "SDN-based TCP congestion control in data center networks," 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), 2015, pp. 1-7, doi: 10.1109/PCCC.2015.7410275.

[18] T. Hu, P. Yi, J. Zhang and J. Lan, "Reliable and load balance-aware multi-controller deployment in SDN," in China Communications, vol. 15, no. 11, pp. 184-198, Nov. 2018, doi: 10.1109/CC.2018.8543099.

[19] A. M. Al-Sadi, A. Al-Sherbaz, J. Xue and S. Turner, "Routing algorithm optimization for software defined network WAN," 2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC- MITCSA), 2016, pp. 1-6, doi: 10.1109/AIC- MITCSA.2016.7759945.

[20] P. Dong, X. Du, H. Zhang and T. Xu, "A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows," 2016 IEEE International Conference on Communications (ICC), 2016, pp. 1-6, doi: 10.1109/ICC.2016.7510992.

[21] R. Kandoi and M. Antikainen, "Denial-of-service attacks in OpenFlow SDN networks," 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015, pp. 1322-1326, doi: 10.1109/INM.2015.7140489.

[22] T. G. Gebremeskel, K. A. Gemeda, T. G. Krishna, and P. J. Ramulu, "DDoS Attack Detection and Classification Using Hybrid Model for Multicontroller SDN," *Wireless Communications and Mobile Computing*, vol. 2023, p. e9965945,Jun. 2023, doi: https://doi.org/10.1155/2023/9965945.

[23] Mahesh Kumar N, Ane Ashok Babu, Sathish Shet, Nithya Selvaraj, Jamal Kovelakuntla, "Mitigation of spectrum sensing data falsification attack using multilayer perception in cognitive radio networks", Acta IMEKO, ISSN: 2221-870X, vol.11, no.1, pp. 1-7, 2022. DOI: http://dx.doi.org/10.21014/acta_imeko.v11i1.1199