

Novel Approach for Ddos Attack Mitigation in Software Defined Network

Nirzari Patel¹, Dr. Hiren Patel²

¹Research Scholar, KSV, Gujarat, India

²Principal, VSITR, Gujarat, India²

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Introduction: This research article intends to depict the usage of machine learning (ML) techniques in software defined network (SDN) to address the Distributed Denial of Service (DDoS) attack. Due to expansion in the complex network operations and configurations, SDN has come out as a propitious network model which uses software-based controllers or application programming interfaces (APIs) to manage activity in an organization and connect with the basic equipment framework. Unlike traditional systems which use dedicated hardware (such as switches) to control assemble activities, SDN can create and control a virtual organization or traditional equipment, through computer programmes. With SDN, the online intelligence is concentrated in a software component called SDN Pick, giving organization's admin the ability to effectively manage, protect, and optimize assets as well as programmatically shape the entire organizational activity design. This research comprehensively portrays the usage of ML Algorithms to detect and prevent the DDoS attack. Based on the analysis, to determine the research gaps and opportunities to implement an efficient solution for security in SDN, we summarize the bland system of SDN, identify security problems, find out the optimal solution and provide insights on the long run improvement in this field along with detailed comparison.

Objectives: The objective of this paper is to depict the usage of machine learning (ML) techniques in software defined network (SDN) to address the Distributed Denial of Service (DDoS) attack.

Algorithms: To evaluate the performance and functionality of the proposed SDN, we have carried out independent experiments using Random Forest (RF) Algorithm, Decision Tree (DT) Algorithm, Naïve Bayes (NB) Algorithm, K- Nearest Neighbors (KNN) Algorithm and Linear Regression.

Results: The first scenario performs better at detecting DDoS attack, while other scenarios are more effective at identifying low-frequency attacks. In best scenario using prevention method , over 64.13% of normal data is detected. Additionally, the proposed solution improves the detection and prevention rate of DDoS by 9.67%.

Conclusions The subject of SDN had exclusively gotten colossal consideration from industry and the scholarly community. The anticipated commitments of our work are to reply the investigate questions. We carried out an in-depth examination of security applications conveyed in SDN utilizing m innovation and found out that most ponders included in our paper proposed SDN security and Ddos attack mitigation.

Keywords: software defined networking, Machine Learning, security problem

INTRODUCTION

As the world is getting more and more digitized with the huge consumption of soft-code [5], there is a growing expectation to have on-demand and customizable services for both enterprise and end users. As an end user, one may expect little or no latency and jitter while being online, for instance playing games, watching movies or making video calls. These types of applications require a lot of automation and orchestration. Software Defined Networking (SDN) could be an option to address the challenge. In the initial part of this article, we provide an overview of the role SDN plays and how it transforms the way communication service providers operate their network [5] [6]. Over the past

two decades [24] [6], networks have come under increased traffic demands and increased scrutiny as both organizations and consumers increasingly rely on network connectivity for sales, services, communications, and data sharing. SDN paired with network functions virtualization, is a key technology needed to meet these new demands. SDN is just one piece of the puzzle: Network Functions Virtualization (NFV), SDN, and white box devices; each offers network operators a new way to design, deploy, and manage an SDN architecture and its services. SDN is important because it gives network operators new ways to design, build, and operate their networks [24], [9]. SDN separates the network's control (brains) and forwarding (muscle) planes and provides a centralized view of the distributed network for more efficient orchestration and automation of network services. The SDN controller platforms that organizations use allow communication between the separated network planes [6] [9]. NFV focuses on optimizing the network services themselves. NFV decouples the network functions, such as domain name systems (DNS), caching, firewalls, routing, and load balancing, from proprietary hardware appliances. Decoupled functions can run in software to accelerate service innovation and provisioning, particularly within service provider environments such as the public cloud [6]. NFV ensures the network can integrate with and support the demands of virtualized architectures, particularly those with multi-tenancy requirements [6].

White box devices, such as switches and routers, are based on "generic" merchant silicon networking chipsets available for anyone to buy, as opposed to proprietary silicon chips designed by a single networking vendor. This means that specific software and networking protocols can be applied and adapted through SDN without the restraints of working with one vendor's proprietary limitations within their hardware [5], [6]. Software-defined cloud networking uses white box devices. Cloud providers often use the generic hardware so they can easily bring changes to the cloud data center while saving on capex and opex costs. The promise of SDN is to reduce the administration overhead of managing networks, making them more agile to adapt and adjust based on demand or need, through a centralized controller [9], [6].

Some other expectations of SDN could include [9], [6]:

- Provide visibility over the network state and enable service assurance (close/open-loop) [6].
- Adjust the network on demand or dynamically to deliver services or meet defined SLA [6].
- Configure the network to enable or disable traffic patterns (i.e., traffic steering) [6].
- Configure the network to fulfill the needs of new workloads, and automatically enable cross-workload communication [6].
- Remove the service specific network configuration when it is decommissioned, and adjust impacted network elements accordingly [6].

Software Defined Networking (SDN)

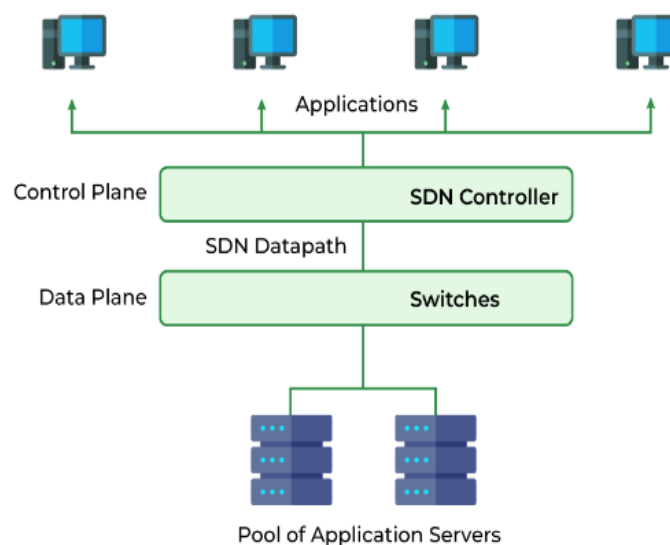


Fig-1 Software Defined Networking [6]

To understand software-defined networks, we need to understand the various planes involved in networking [6].

1. Data Plane
2. Control Plane

Data plane: All the activities involving as well as resulting from data packets sent by the end-user belong to this plane [6]. This includes:

- Forwarding of packets.
- Segmentation and reassembly of data.
- Replication of packets for multicasting.

Control plane: All activities necessary to perform data plane activities but do not involve end-user data packets belong to this plane. In other words, this is the brain of the network [6]. The activities of the control plane include [6]:

- Making routing tables.
- Setting packet handling policies.

Where is SDN Used [6]?

- Enterprises use SDN, the most widely used method for application deployment, to deploy applications faster while lowering overall deployment and operating costs. SDN allows IT administrators to manage and provision network services from a single location [6].
- Cloud networking software-defined uses white-box systems. Cloud providers often use generic hardware so that the Cloud data center can be changed and the cost of CAPEX and OPEX saved [6].

Components of Software Defining Networking (SDN):

- SDN Applications: SDN Applications relay requests or networks through SDN Controller using API [9].
- SDN controller: SDN Controller collects network information from hardware and sends this information to applications [9].
- SDN networking devices: SDN Network devices help in forwarding and data processing tasks [9].

OBJECTIVES

The objective of this paper is to depict the usage of machine learning (ML) techniques in software defined network (SDN) to address the Distributed Denial of Service (DDoS) attack.

METHODS

In this section, we describe the implementation of the different existing machine learning algorithms. The algorithms were integrated using python with software defined networking. In this, we had implemented Software defined network in Mininet and we are using Ryu controller for the same. Along with Software defined network, we had also identified the results of different machine learning algorithms to detect the distributed denial of service attack. After detection of distributed denial of service attack, we had implemented novel approach to low the effectiveness of this attack.

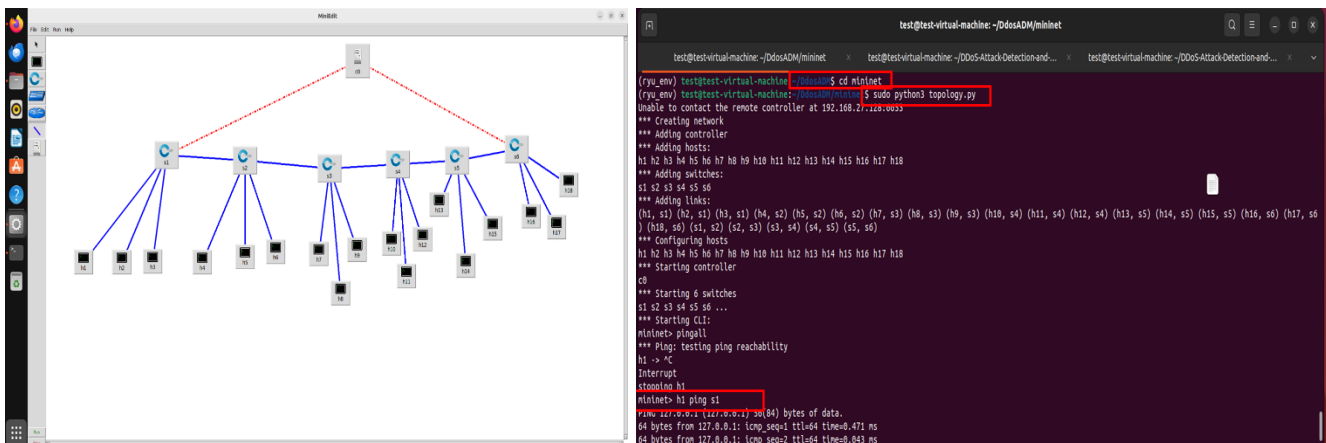


Fig-2 Existing Software Defined Network with 1 controller, 6 switches and 18 hosts

Figure 3 illustrates the proposed Scheme. And Figure 3 illustrates the proposed Algorithm.

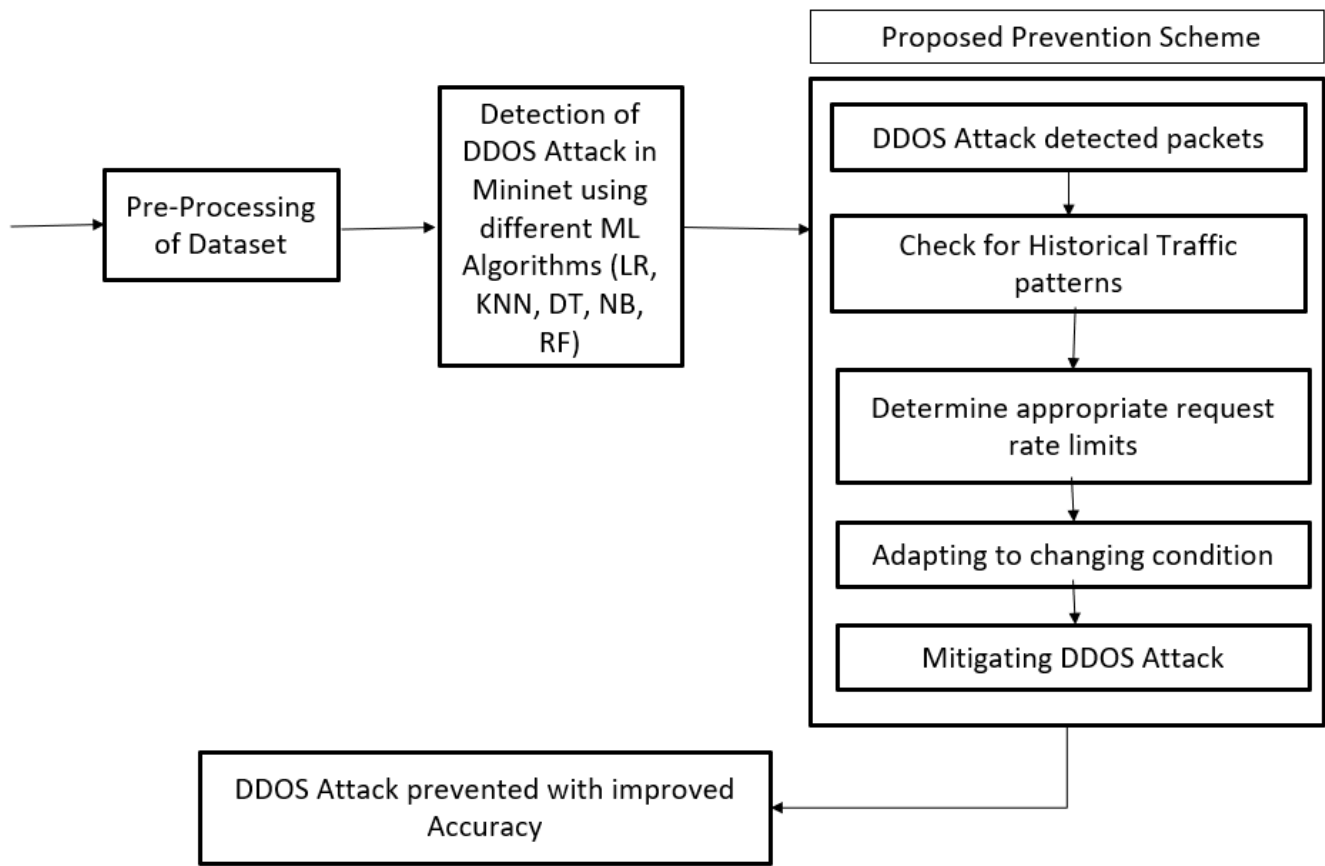


Figure 3: Proposed prevention scheme

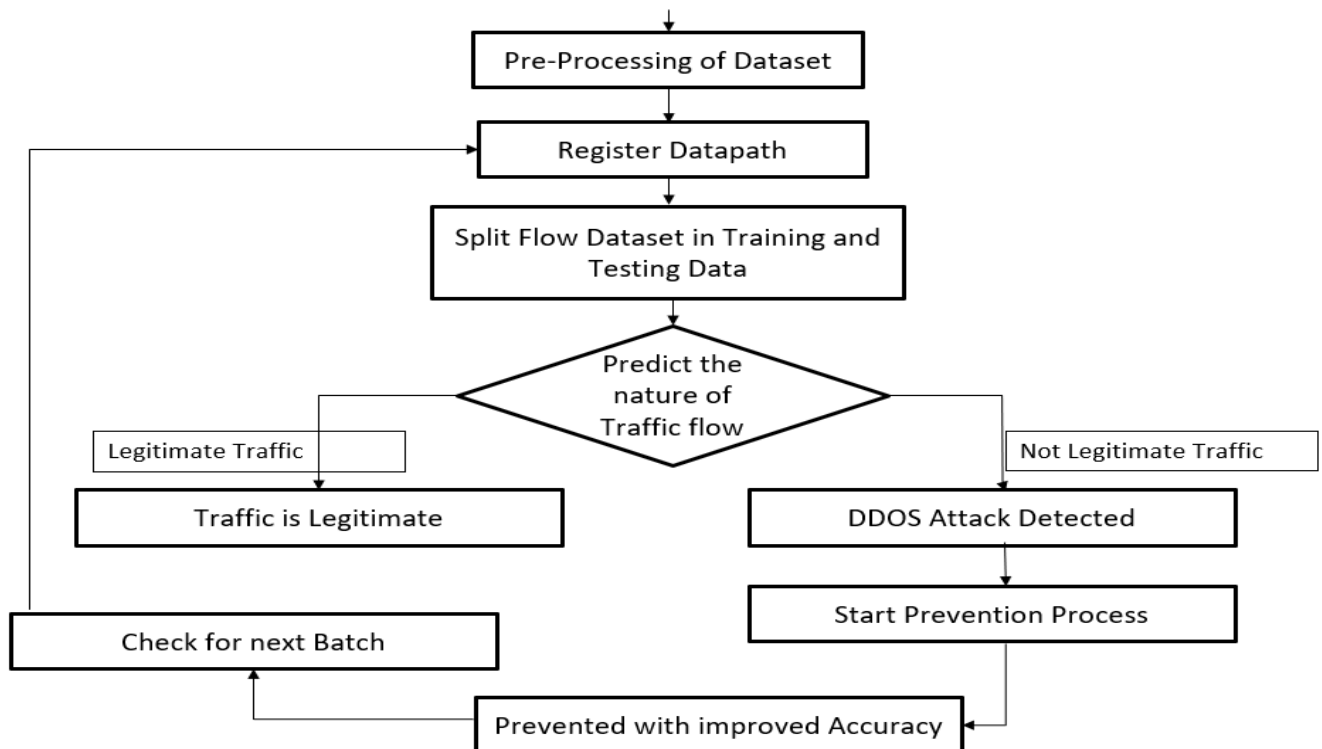
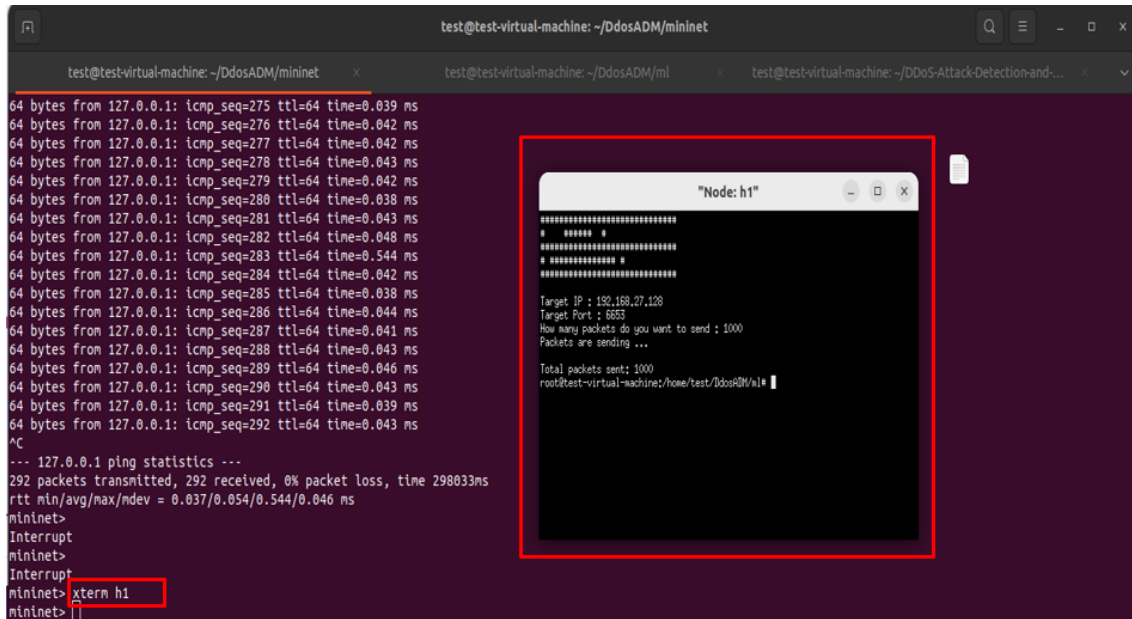


Figure 4: Proposed Algorithm

1. Distributed Denial of Service attack Detection

Now, we will enter targeted ip, port and number of packets for Distributed denial of Service attack in new terminal of node h1. After giving all above information, we had find the Distributed denial of Service attack in our network.



```

test@test-virtual-machine: ~/DdosADM/mininet
64 bytes from 127.0.0.1: icmp_seq=275 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=276 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=277 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=278 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=279 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=280 ttl=64 time=0.038 ms
64 bytes from 127.0.0.1: icmp_seq=281 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=282 ttl=64 time=0.048 ms
64 bytes from 127.0.0.1: icmp_seq=283 ttl=64 time=0.544 ms
64 bytes from 127.0.0.1: icmp_seq=284 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=285 ttl=64 time=0.038 ms
64 bytes from 127.0.0.1: icmp_seq=286 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=287 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=288 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=289 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=290 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=291 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=292 ttl=64 time=0.043 ms
^C
--- 127.0.0.1 ping statistics ---
292 packets transmitted, 292 received, 0% packet loss, time 298033ms
rtt min/avg/max/mdev = 0.037/0.054/0.544/0.046 ms
mininet>
Interrupt
mininet>
Interrupt
mininet>
mininet> xterm h1
mininet>
  
```

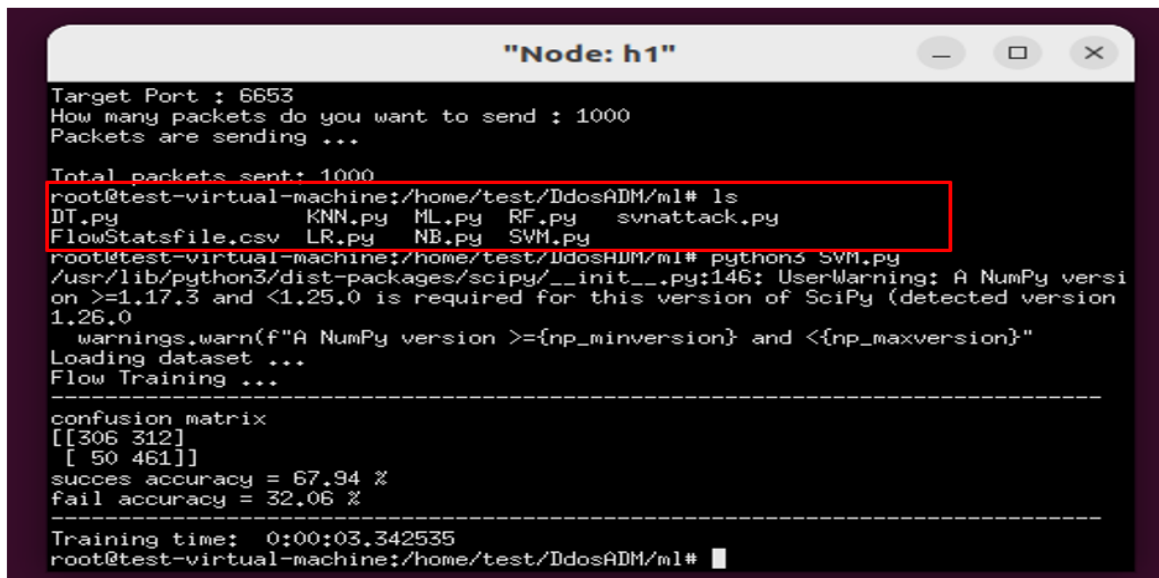
```

"Node: h1"
*****
*****
*****
*****
*****
Target IP : 192.168.0.128
Target Port : 6653
How many packets do you want to send : 1000
Packets are sending ...

Total packets sent: 1000
root@test-virtual-machine/home/test/DdosADM/h1#
  
```

Figure 5: SDN is attacked by Distributed denial of Service attack

After Ddos Attack application on SDN, we had detected attack using different machine learning algorithms. Here, we are using Linear Regression (LR), Random Forest (RF), Decision Tree (DT), Naïve Bayes (NB) and K-Nearest Neighbors (KNN) Algorithms.



```

"Node: h1"
Target Port : 6653
How many packets do you want to send : 1000
Packets are sending ...

Total packets sent: 1000
root@test-virtual-machine:/home/test/DdosADM/ml# ls
DT.py          KNN.py  ML.py  RF.py  svnattack.py
FlowStatsfile.csv LR.py   NB.py  SVM.py
root@test-virtual-machine:/home/test/DdosADM/ml# python3 SVM.py
/usr/lib/python3/dist-packages/scipy/__init__.py:146: UserWarning: A NumPy version
>=1.17.3 and <1.25.0 is required for this version of SciPy (detected version
1.26.0
warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")
Loading dataset ...
Flow Training ...

-----
confusion matrix
[[306 312]
 [ 50 461]]
succes accuracy = 67.94 %
fail accuracy = 32.06 %
-----

Training time: 0:00:03.342535
root@test-virtual-machine:/home/test/DdosADM/ml#
  
```

Figure 6: Using different Machine Learning Algorithms

A. Using Random Forest (RF) Algorithm

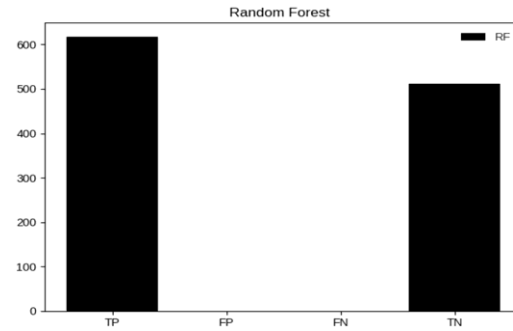


Figure 7: Random Forest (RF) Algorithm results

B. Using Decision Tree (DT) Algorithm

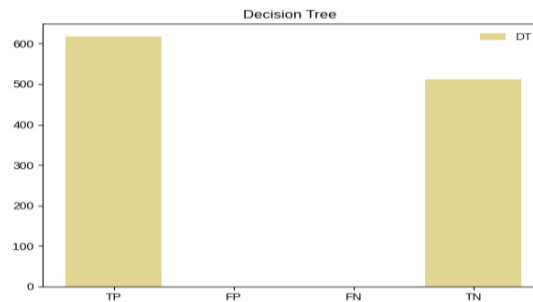


Figure 8: Decision Tree (DT) Algorithm results

C. Using Naïve Bayes (NB) Algorithm

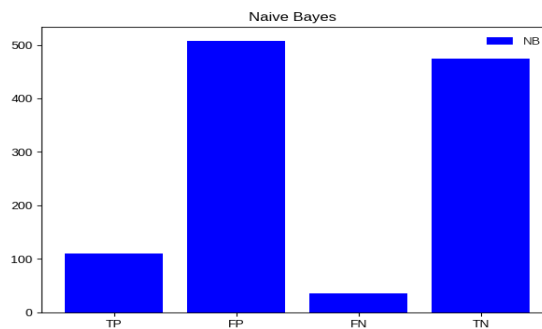


Figure 9: Naïve Bayes (NB) Algorithm results

D. Using K- Nearest Neighbors (KNN) Algorithm

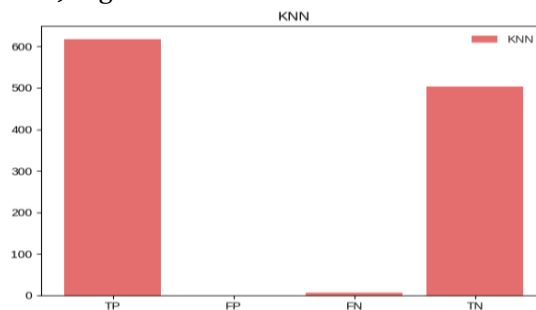


Figure 10: K- Nearest Neighbors (KNN) Algorithm results

2. Distributed Denial of Service attack Prevention (Mitigation)

We had proposed new approach to prevent or low the effect of Distributed Denial of Service attack and after applying that we found that effect of Distributed Denial of Service attack was decreased by 09.67%.

```

Node: h1
958. in mainloop
first_manager.window.mainloop()
File "/usr/lib/python3.10/tkinter/__init__.py", line 1458, in mainloop
self.tk.mainloop(n)
KeyboardInterrupt

root@test-virtual-machine:/home/test/DdosADM/ml# python3 LR.py
/usr/lib/python3/dist-packages/scipy/__init__.py:146: UserWarning: A NumPy version
on >=1.17.3 and <1.25.0 is required for this version of SciPy (detected version
1.25.0
warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")
>Loading dataset ...
Flow Training ...
-----
confusion matrix
[[612  0]
 [347  0]]
succes accuracy = 63.82 %
fail accuracy = 36.18 %
-----
benin = 2458
ddos = 1375
-----

thon3.9 -m venv ryu_env source ryu_env/bin/activate
ontroller/ryu_env/bin/activate'
.9 -m venv ryu_envnew
thon3.9 -m venv ryu_envnew source ryu_env/bin/activate
ontroller/ryu_env/bin/activate'
thon3.9 -m venv ryu_envnew source ryu_envnew/bin/activa
ontroller/ryu_envnew/bin/activate'

```

Figure 11: Existing Software Defined Networking with DDOS attack detection

Figure 12 shows the graphs for Existing Software Defined Networking with DDOS attack detection using three diifferent protocols. These three protocols are ICMP, TCP and UDP. Figure 13 shows the normal and DDOS attacked percentage in above three protocols.

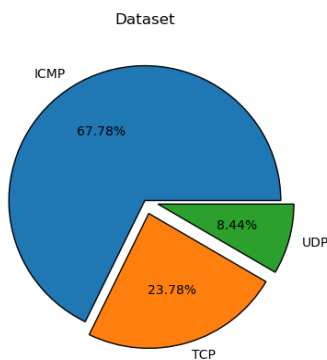


Figure 12: Distributed Denial of Service attack Mitigation

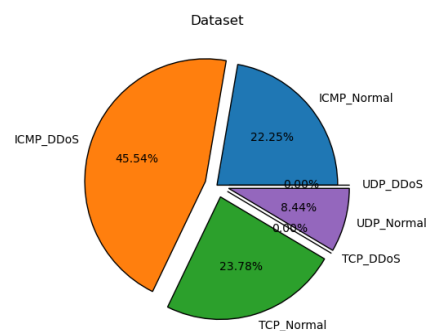


Figure 13: Normal and DDOS attack in ICMP, TCP and UDP

Here are the Hardware requirements and implementation tools details.

TABLE 1:SYSTEM CONFIGURATION AND SOFTWARE ENVIRONMENT

Network Development Tools	
Network Development	Mininet
Controller	Ryu Controller
Controller written in	Python
Machine Learning algorithms performed in	Python
Dataset from	Kaggle
System Configuration	
Operating System	Linux (Ubuntu)
Random Access Memory (RAM)	16 GB

RESULTS

After applying different machine learning algorithms (LR, KNN, NB, DT, RF) to detect the Distributed Denial of Service attack, we had found some comparative analysis as per following.

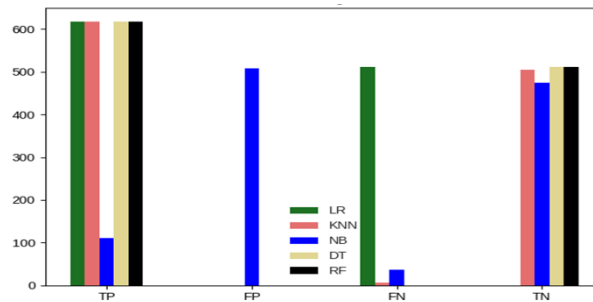


Figure 14: Comparative analysis of all algorithms

Now, we have some more comparison of before prevention and after prevention of the effect of distributed denial of service attack.

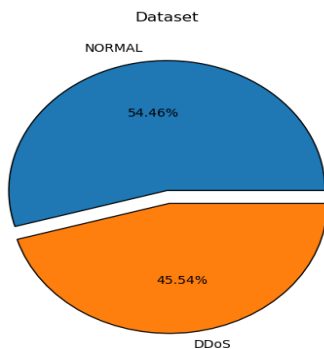


Figure 15: Before distributed denial of service Prevention

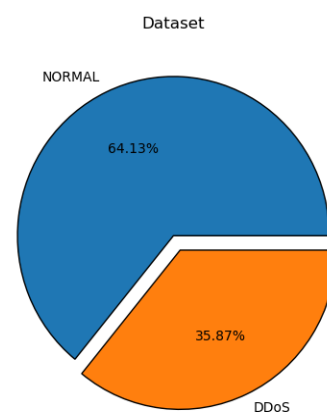


Figure 16: After distributed denial of service Prevention

DISCUSSION

The subject of SDN had exclusively gotten colossal consideration from industry and the scholarly community. The anticipated commitments of our work are to reply the investigate questions. We carried out an in-depth examination of security applications conveyed in SDN utilizing m innovation and found out that most ponders included in our paper proposed SDN security and Ddos attack mitigation. As a future work, more Accuracy in prevention of Ddos Attack can be done.

REFERENCES

- [1] A. Ikram, S. Arif, N. Ayub, and W. Arif, "Load Balancing in Software Defined Networking (SDN)," MAGNT Research Report, vol. 5, no. 1, pp. 298–305, 2018.
- [2] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software-defined networking: State of the art and research challenges," Computer Networks, vol. 72, pp. 74–98, 2014
- [3] <https://www.investopedia.com/terms/b/blockchain.asp>
- [4] T. Alharbi, "Deployment of blockchain technology in software-defined networks: A survey," IEEE Access, vol. 8, pp. 9146–9156, 2020
- [5] <https://www.redhat.com/en/blog/evolution-software-defined-networking>
- [6] <https://www.geeksforgeeks.org/software-defined-networking/>

- [7] S. Khezr, M. Moniruzzaman, A. Yassine, and R. Benlamri, "Blockchain technology in healthcare: A comprehensive review and directions for future research," *Applied Sciences (Switzerland)*, vol. 9, no.9, pp. 1–28, 2019
- [8] L. Ismail, H. Materwala, and S. Zeadally, "Lightweight Blockchain for Healthcare," *IEEE Access*, vol. 7, pp. 149 935–149 951, 2019
- [9] [https://www.euromoney.com/learning /blockchain-explained/what-is-blockchain](https://www.euromoney.com/learning/blockchain-explained/what-is-blockchain)
- [10] W. Li, W. Meng, Z. Liu, and M. H. Au, "Towards blockchain-based software-defined networking: Security challenges and solutions," *IEICE Transactions on Information and Systems*, vol. E103D, no.2, pp. 196–203, 2020
- [11] Shivani Wadhwa, Himanshi Babbar, Shalli Rani "A Survey on Emerging Software-Defined Networking and Blockchain in Smart Health Care" *ICCRDA-2020*, IOP Conf. Series: Materials Science and Engineering 1-10,2021
- [12] T. Ali Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, and T. Alghamdi, "A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations," *IEEE Access*, vol. 7, pp. 176 838–176 869, 2019
- [13] C. Xue, N. Xu, and Y. Bo, "Research on key technologies of software-defined network based on blockchain," *Proceedings - 13th IEEE International Conference on Service-Oriented System Engineering, SOSE 2019, 10th International Workshop on Joint Cloud Computing, JCC 2019 and 2019 IEEE International Workshop on Cloud Computing in Robotic Systems, CCRS 2019*, pp. 239–243,2019.
- [14] M. Khayat, E. Barka, and F. Sallabi, "SDN Based Secure Healthcare Monitoring System (SDNSHMS)," *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, vol. 2019-July, no. 2015, pp. 1–7, 2019.
- [15] F. Sallabi, F. Naeem, M. Awad, and K. Shuaib, "Managing IoT-Based Smart Healthcare Systems Traffic with Software Defined Networks," *2018 International Symposium on Networks, Computers, and Communications, ISNCC 2018*, no. 31, pp. 1–6, 2018
- [16] J. Zhang, N. Xue, and X. Huang, "A Secure System for Pervasive Social," *IEEE Access*, vol. 4, no. December, pp. 9239 – 9250, 2016
- [17] L. Xie, Y. Ding, H. Yang, and X. Wang, "Blockchain-based secure and trustworthy Internet of Things in SDN-enabled 5G-VANETs," *IEEE Access*, vol. 7, pp. 56656–56666, 2019.
- [18] Q. Shafi and A. Basit, "DDoS Botnet prevention using blockchain in software defined Internet of Things," in *Proc. 16th Int. Bhurban Conf. Appl. Sci. Technol. (IBCAST)*, Jan. 2019, pp. 624–628.
- [19] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "DistBlockNet: A distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 78–85, Sep. 2017.
- [20] C. Xue, N. Xu, and Y. Bo, "Research on key technologies of softwaredefined network based on blockchain," in *Proc. IEEE Int. Conf. ServiceOriented Syst. Eng. (SOSE)*, Apr. 2019, pp. 239–2394.
- [21] S. R. Basnet and S. Shakya, "BSS: Blockchain security over software defined network," in *Proc. Int. Conf. Comput., Commun.Automat. (ICCCA)*, May 2017, pp. 720–725
- [22] D. kreutz and F. Ramos "Software-Defined Networking: A Comprehensive Survey" in, *IEEE*, May2014
- [23] Mrs. G. Indumathi, Dr. B.S.E. Zoraida "A Survey on Security And Privacy In Blockchain And Software Defined Network "Webology (ISSN: 1735-188X) Volume 18, Number 6, 2021
- [24] <https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/why-sdn-important/>
- [25] <https://www.vmware.com/in/topics/glossary/content/software-defined-networking.html>