Proceedings of 11th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2022 Month: March Volume: 11

Machine Learning based DDoS Detection System in Software-Defined Networking

Diwos Karki [a], Babu Ram Dawadi [b]

[a, b] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, TU, Nepal*
✉ [a] dwos.karki@gmail.com, [b] baburd@ioe.edu.np

Abstract

Software-defined networking (SDN) is a novel networking paradigm that, through an abstraction of network plans, bifurcate the control plane and data plane, and thus providing a flexible, programmable, and dynamic architecture over traditional network. The separation of the control plane and data plane has brought significant network dynamism but still is liable to security vulnerabilities. One of the significant threats is Distributed Denial of Service (DDoS) attack, utilizing the separation of the control plane and data plane, which has been a novel challenge to overcome in SDN. In this paper, a comparative analysis of Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Naïve Bayes (NB) are done on the basis of accuracy, precision, recall, and execution time on CICDDoS2019 and synthetically generated dataset in the SDN environment. The overall performance of RF and KNN is significantly better but their execution time is higher than other algorithms. SVM performance is better if the tradeoff between the accuracy and execution time is taken into consideration. Thus, machine learning algorithms perform significantly well in detection of DDoS at SDN environment if consideration is given to suitable feature selection.

Keywords

SDN, Security, DDoS, Machine learning algorithm

1. Introduction

Evolution in technology, on the one hand provides significant advancement in computer networking, also on the other hand has its own ever evolving issues regarding network security. A significant issue regarding present network security is to detect whether the network related requests are legitimate or attacks. This issue can be mitigated by the use of Intrusion detection system (IDS), which attenuates the risk of network failure and misuse [1]. Infrastructures of traditional network cannot meet the current network requirements like high bandwidth and connection speed, dynamic management, network virtualization and cloud computing. Thus, evolution of new network paradigm i.e. Software- defined networking (SDN) has emerged as a viable alternative of traditional networks [2].

SDN is a new approach of networking, which decouples the network into control logic and network logic to introduce the concept of the control plane and the data plane, thus providing flexibility in network with simplification in management tasks. SDN architecture consists of control, data, and application planes. The central unit, i.e. controller checks the destination address, ensures certain data delivery, and then ultimately selects the process path at the data plane layer devices, i.e. routers, switches etc., thus providing connection to end-user via the network. The application plane, also known as management plane, provides developer a platform to manage networks like fault monitoring and network configuration.

The data plane consists of a flow entry table where devices perform packet transmission as stated by the rules defined by the controller. Communication to the controller is done via secured transport layer protocol, which communicates using southbound interface whenever a new entry occurs that doesn't exist on the flow table. The control plane dictates the new rule for that particular flow entry. The controller communicates with the network applications on application layer via northbound interface. OpenFlow protocol[3] is widely used protocol in this interface. The fundamental architecture of SDN is illustrated in Figure 1.

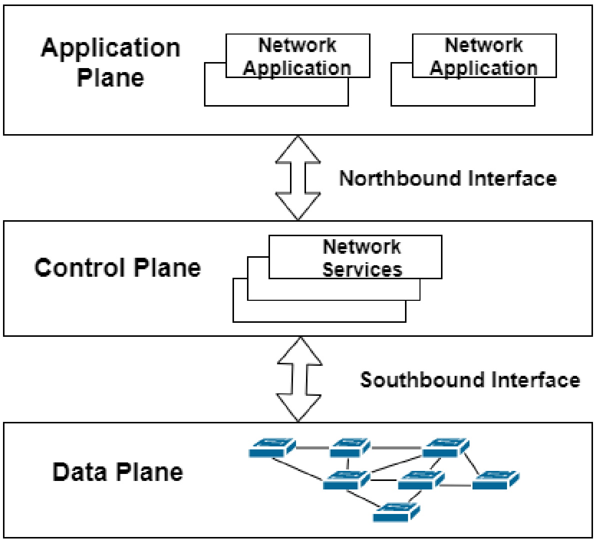

Figure 1: SDN architecture [4]

The agility of software-controlled network flow with topology management provided by SDN, meets the need of present dynamic networks. This new advancement gives rise to different issues in the maintenance of consistent and well-defined network parameters. Seven different threat vectors are identified in SDN networks relating to network management [5]. These are: (1) forged or fake traffic flows, (2) attacks on vulnerabilities in switches, (3) attacks on control plane communications, (4) attacks on and vulnerabilities in controllers, (5) lack of mechanisms to ensure the trust between controller and management applications, (6) attacks on and vulnerabilities in administrative stations, and (7) lack of trusted resource for forensics and remediation.

Triggered by malicious users or faulty device, forged or fake traffic flows can attack both the switches and controllers. The vulnerabilities of the switch can lead to slowdown or drop off the packets, clone or deviation of network traffic or even the injection of forged traffic, which can overload controller or other switches and cause havoc of the entire network. Taking advantage of TLS/SSL communication between the controller and switch, the attacker may gain access to the control plane and even launch DDoS attack thus undermining the control plane communication. Attack on the controller is the most prominent threat of SDN, which may be due to faulty or malicious controller or even due to malicious application, thus compromising an entire network. Lack of ability to establish trusted relationships between the controllers and applications can also create a significant threat to the SDN. The vulnerabilities in administrative stations can have unauthorized access to the network controller and the threat of single compromised machine is compounded in SDN because of its architecture. Also, due to the novel nature of SDN, investigation and remediation requires certain level of authenticity and trustworthiness of data, which can also be a threat to SDNs. Threat vectors 3, 4, and 5 arise due to the bifurcation of the data planes and control planes and the most devastating threat possible on such network is Distributed Denial of Service (DDoS) attacks.

One of the prominent threats defined was DDoS attack, which affects both the controller and OpenFlow switch. Attackers compromise multiple number of computers to send the flood of traffic, which consumes network bandwidth or the resources of the network so that legitimate users are denied the services. DDoS attack appears to be one of the most serious threat to SDN that can have catastrophic effects on the network performance by disrupting network flow as service nodes are directly affected. Thus, legitimate users can't access the service provided by SDN.

In this study, we aimed to contribute to the literature, by developing a system that detects DDoS attacks in SDN by means of ML models. To develop a ML based system, the foremost part is to appropriately process the data and determine the best features using appropriate feature selection technique. This study is focused on selecting the best features so that the ML models can have improved accuracy. Three different feature selection techniques, i.e. Chi- square as filter based, Exhaustive feature selection [6] as wrapper based and Random forest importance as embedded are used and only the combined best features selected by them are fed to the ML models.

The rest of this paper is organized as follows. We present the background and related works on DDoS detection in SDN environment in section II. Methodology regarding DDoS detection in SDN by implementing Machine Learning (ML) is discussed in section III. Section IV presents result and analysis of performance of various ML algorithms. Section V concludes the paper.

2. Background and Related Works

2.1 Need of DDoS Detection

SDN is the new paradigm, where the separation of the control plane and data plane enables the network to be programmable, centralized and flexible. This architecture provides great deal of control to the network administration as the whole network can be controlled via the centralized control plane since the global view of network is available. Thus, construction of an intelligent and automated network is possible via SDN.

However, the centralization of control plane has its own shortcomings as the centralized control plane makes it an ideal target for the attackers. Thus, control plane failure can be the single point of failure of an entire network. One of the attacks is DDoS attack, which can overwhelm the controller and ramification of such attack can make the whole network unavailable. Thus, detection of DDoS attack on SDN is the necessary step for preventing the failure of entire network architecture.

2.2 DDoS Attack

In the SDN, DDoS attack can be performed at the data plane as well as the control plane. When the control plane is attacked, controller is overworked in replying the attackers request thus, wasting valuable computational time of the controller. This can lead to collapse of an entire network as the controller is the heart of SDN network. Hence, legitimate users can't access the network services. DDoS can be classified broadly into two categories, i.e. attack on the data plane and attack on the control plane.

2.2.1 Types of DDoS Attacks

UDP Flood [7] [8]is a form of attack that sends a large number of UDP packets to random ports in order to overwhelm the targeted host.

SYN Flood [8] attacks are performed by exploiting the three-way handshake of TCP connection.

DNS Reflection attack sends DNS requests to the victim's source IP address, resulting in replies that are far larger than the request.

HTTP Flood [8] sends a huge number of requests to a web server and overwhelms it to the point where it cannot respond to legitimate requests.

ICMP Flood [8] is another type of attack that exhausts the resources of the victim by sending a very large number of ICMP pings (echo request), which keeps the server busy in sending responses (echo replies).

2.3 ML Models

K-nearest neighbor (KNN) [9] is a supervised learning algorithm, which can be used for both classification and regression problems. It is a lazy learning algorithm where it stores all the data in the memory during training phase and uses all the data of training while classification. It labels the new instance of data based on the similarity with other instances on the dimensional space based on majority voting. A new instance will be labelled ‘y’, if the majority of the neighboring instances have class ‘y’. For calculation of similarity, a distance metric is used.

Support vector machine (SVM) [9] is a supervised learning algorithm which can be used for both classification and regression problems. The main aim of support vector machine is to find the appropriate hyperplane in an n-dimensional space (n is the number of features) which classifies the data points distinctly. The two step learning process is carried out in following ways. First, the plotting of inputs in an n-dimensional space, where each individual coordinate of attributes are support vectors. Secondly, an optimum hyperplane construction for the separation of instances will be determined. SVM uses Kernel tricks to map complex non-linear functions reducing computational complexity.

Naïve Bayes (NB) [9] is based on Bayes' Theorem with the assumption of event independence among predictors. In general, Naïve Bayes classifier presumes that particular feature in a class is uncorrelated with any other feature.

Random Forest (RF) [9] is a supervised ensemble learning algorithm which creates large number of decision trees and merges them for generating stable and accurate prediction. The “forest” built by RF is an ensemble of decision trees, generally trained using “bagging” method. The concept regarding bagging method is to combine learning methods to increase overall result.

2.4 Related Work

Barki et al. [10] explored implementation of new IDS based on DDoS attack was proposed on the basis of two modules: Signature IDS and Advanced IDS. Various ML algorithms were implemented, such as K-means, NB, KNN, and k-medoids, to classify traffic flow as normal and abnormal, and to find the host set with anomalous behaviors under signature IDS module.

Braga et al. [11]proposed DDoS detection done based on the traffic flow features with low overhead. NOX based controller was used, which uses the OpenFlow Protocol. An unsupervised artificial neural network (ANN) Self- Organizing Maps (SOM) was used to classify the network traffic either as attack or benign.

Polat et al. [12] performed the detection of DDoS by using various ML algorithms like SVM, KNN, NB, and ANN. Here, the feature selection is done by using three different algorithms. These features are fed to ML algorithms to classify the data to be normal or benign. Also the data without feature selection was also fed to ML algorithm and the comparative analysis was performed among those algorithms. Hping3 tool was used to create a DDoS attack dataset and sFlow Docker image was used to record the traffic. POX controller was used as a controller and experiment was simulated in Mininet. Result suggested wrapper-based feature selection with KNN classifier produced the highest accuracy of detection (98.3

Saif Saad Mohammed et al. [13]proposed a DDoS mitigation method based on ML using the NSL-KDD dataset. An attack detection server was set up based on NB algorithm with wrapper based feature selection method to reduce dimensionality of dataset. An authentication mechanism based detection server was set up for the DDoS detection and mitigation.

A DDoS detection framework based on SVM was introduced by Yao Yu et al. for SDN based vehicular network [14]. PACKET_IN was used to extract features, which are then used to train the algorithm. In the controller, the SVM training model was used and suspicious packets were submitted to the SVM model, which classified them into various DDoS attack categories.

Dayal et al. selected the features using a combination of correlation-based feature selection, information gain, and gain ratio in [15]. Then, using a NB algorithm, the reduced dataset was classified with an accuracy of 97.78

According to Alshamrani et al. [16], the current methods for preventing DDoS attacks are ineffective. They collected traffic data from data plane transmission devices on a regular basis and used ML classification algorithms to respond to sudden traffic changes in the SDN architecture at the time of the attack. As a starting point, Packet_In messages flowing between the controller and transmission devices at the time of the attack were examined. For classification, SVM, J48, and NB algorithms were used.

3. Proposed Approach

In this section we present how the evaluation of ML methods for DDoS detection is done. The DDoS attack has been constant threat to SDN because of its inherent centralized controller architecture. Due to rapid development of such attack mechanisms coerces the need of adaption of novel and updated system. Using ML to train the detection system by learning the traffic patterns from new traffic information is more accurate and efficient.

DDoS attacks can be detected mainly by three categories based on detection metric and detection mechanism used, namely: Information-theory based detection, ML-based detection, and ANN-based detection. We have used ML based detection due to its convenience in implementation while achieving relatively high degree of precision than Information-theory based models.

The standard CICDDoS2019 dataset is used to train the ML models and the network data is fed to the trained model to predict whether the data is anomaly or benign. This proposed concept can be depicted by Figure 2.

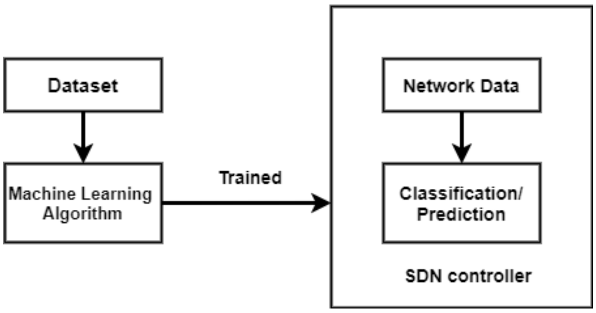

Figure 2: Proposed Methodology

3.1 ML Approach

There are many ML techniques available for the DDoS detection. In this study, we have used classification techniques using RF classifier, KNN Algorithm, SVM, and Naïve Bayes. The selection of the algorithms are based on the following mentioned criteria. First is to include both parametric and non-parametric algorithm, then second is to have mix of algorithms from different categories.

Parametric algorithm summarizes data on the basis of fixed number of parameters and makes assumptions about the dataset thus making it computationally efficient. But it requires its assumption to be correct; otherwise it significantly affects its performance. In this paper, SVM and NB are used as parametric algorithms. In contrast, non- parametric algorithms do not make strong assumptions, thus making it more flexible but can be computationally complex when large dataset is used. Hence, KNN and RF are taken as non-parametric algorithm.

There are four categories of algorithms implemented in this study. **Instance-based:** It is sometimes known as memory-based learning where instead of explicit generalization, every new instance is compared with the training instance stored in its memory. KNN is used as instance based algorithm.

Kernel method: this operates in high-dimensional feature space without the actual computation of the coordinates of data in that space. SVM falls under this category.

Bayesian method: Bayesian reasoning assumption is based on probability distribution and prediction of accurate decision is based on adopting these probabilities on new data. This study uses NB algorithm under this category.

Ensemble Method: It uses multiple learning algorithms in order to enhance the predictive performance compared to a single algorithm, which is done by combining multiple models. RF is used in this study as an ensemble method.

3.2 Experimental setup and data collections

To test the performances of our proposed method, a virtual network is created and simulated using the Mininet [17] [19]with POX controller. Scapy is used as a packet generation tool. The virtual environment consists of a POX controller, nine switches, and 64 hosts as shown in Figure 3.

For the generation of normal traffic and attack traffic, two scripts were created using scapy's feature of python programming, viz. 'launchAttack.py' and 'launchTraffic.py'. The 'launchTraffic.py' script generates random IP addresses resembling the normal traffic among the hosts. Every host was sent UDP packets with destination port 80, simulating web browsing in the host machine. This script generated a packet at the interval of 0.1 second whereas 'launchAttack.py' script generates a packet at the interval of 0.025 second to the destination IP address of controller which was inundated with high volume of UDP Packets. The generated flows were captured using tcpdump and further processed using Wireshark. Total of 2000 datasets were generated where 1000 were normal traffic and 1000 were attack traffic.

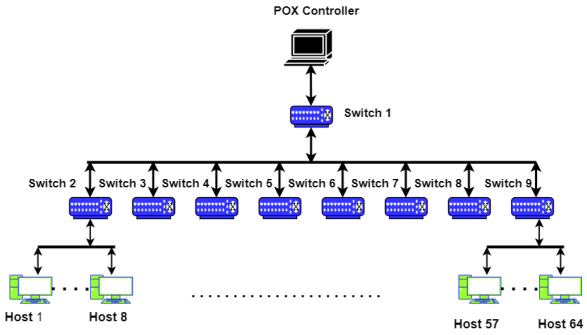

Figure 3: Testbed topology

3.3 CICDDoS2019 dataset for Training Model

CICDDoS2019 [20] contains benign and the most up-to-date common DDoS attacks, which resembles the true real- world data. It also includes the results of the network traffic analysis using CICFlowMeter-V3 with labeled flows based on the time stamp, source and destination IPs, source and destination ports, protocols, and attack. This dataset is used in this research to implement training and evaluation of the proposed model. Four different types of attacks were taken into consideration, i.e. SYN, UDP, UDPLag, and LDAP. There were total 71580 SYN, 6278 UDP, 5940 UDPLag, 10248 LDAP and 2000 Generated data volume for different DDoS attacks and the datasets were split into 60set, 20

4. Result and Analysis

4.1 LDAP dataset

Table 1 shows the evaluation of different ML algorithm in terms of different parameters, while Figure 4 comparatively shows the plots of those parameters. ML models i.e., RF, Naïve Bayes, SVM, and KNN have achieved mean accuracy of 0.99, 0.94, 0.96 and 0.98 respectively. The precision, recall and f-measure of RF is 0.99, 0.99 and 0.99, NB is 0.91, 0.90 and 0.91, SVM is 0.92, 0.93 and 0.91 and KNN is 0.97, 0.92 and 0.94 respectively. Figure 5 shows the comparison of execution time of different ML algorithms. For the execution time, NB has the least execution time of 2.6 ms while KNN has the longest execution time of 16.5 ms. Similarly, execution time for RF is 10.3 ms while SVM is 7.2 ms. RF has provided the best result when considering accuracy and execution time, while SVM has provided the tradeoff between accuracy and execution time.

Table 1: Measures with LDAP Dataset

Measures	RF	NB	SVM	KNN
Mean Accuracy	0.99	0.94	0.96	0.98
Precision	0.99	0.91	0.92	0.97
Recall	0.99	0.90	0.93	0.92
F-Measure	0.99	0.91	0.91	0.94
Execution Time	10.3ms	2.6ms	7.2ms	16.5ms

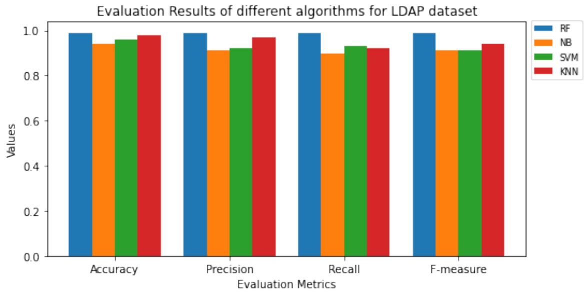

Figure 4: Evaluated result of different algorithms for LDAP dataset

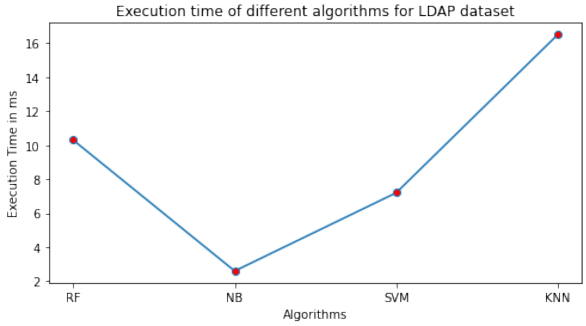

Figure 5: LDAP execution time

4.2 SYN Dataset

Table 2 shows the measures of performance parameters, while Figure 6 shows the comparative chart. For the SYN dataset, RF, NB, SVM and KNN have achieved the mean accuracy of 0.98, 0.45, 0.89 and 0.98. The precision, recall and f-measure of RF is 0.99, 0.98 and 0.99, NB is 0.66, 0.54 and 0.61, SVM is 0.79, 0.98 and 0.88 and KNN is 0.99, 0.98 and 0.98 respectively. The comparison of execution time for different ML algorithms is shown in Figure 7. NB has the least execution time of 7.5 ms, while KNN has the longest execution time 80 ms. Also, the execution time of RF and SVM are 28 ms and 13.2 ms respectively. For SYN dataset, RF provided the best result when accuracy and execution time have been taken into consideration, while SVM has provided the tradeoff between accuracy and execution time. KNN and RF have achieved similar accuracy but the execution time of KNN has been much higher in comparison.

Table 2: Measures with SYN Dataset

Measures	RF	NB	SVM	KNN
Mean Accuracy	0.98	0.45	0.89	0.98
Precision	0.99	0.66	0.79	0.99
Recall	0.98	0.54	0.98	0.98
F-Measure	0.99	0.61	0.88	0.98
Execution Time	28ms	7.5ms	13.2ms	80ms

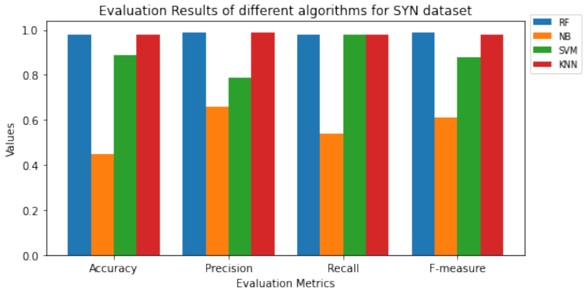

Figure 6: Evaluated result of different algorithms for SYN dataset

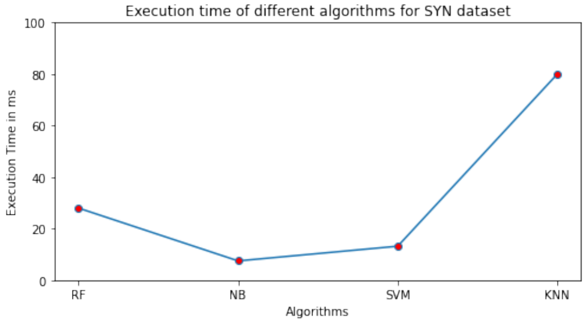

Figure 7: Execution time for SYN datasets

4.3 UDP Dataset

Table 3 shows the measures of performance parameters, while Figure 8 shows the comparative chart. The accuracy achieved by RF, Naïve Bayes, SVM and KNN is 0.98, 0.87, 0.94 and 0.98 respectively. The precision, recall and f- measure of RF is 0.99, 0.98 and 0.99, Naïve Bayes is 0.72, 0.85 and 0.77, SVM is 0.96, 1.0 and 0.98 and KNN is 0.99, 0.98 and 0.98 respectively. Figure 9 shows the comparative execution time for different ML algorithms. NB has achieved the least execution time of 2.3 ms while KNN has the longest execution time of 16 ms. Also, the execution time of RF and SVM are 6ms and 3.5ms respectively. SVM has given the best tradeoff between accuracy and execution time. While considering the accuracy, RF and KNN model have the best result but their execution time have been higher compared to other models.

Table 3: Measures with UDP Dataset

Measures	RF	NB	SVM	KNN
Mean Accuracy	0.98	0.87	0.94	0.98
Precision	0.99	0.72	0.96	0.99
Recall	0.98	0.85	1.0	0.98
F-Measure	0.99	0.77	0.98	0.98
Execution Time	6ms	2.3ms	3.5ms	16ms

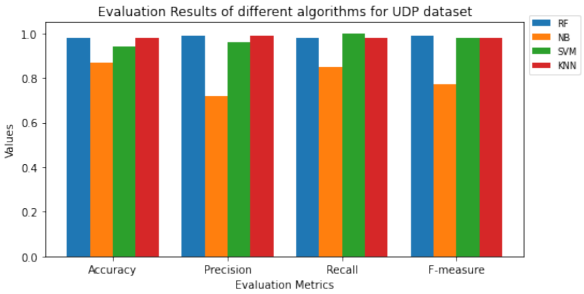

Figure 8: Evaluated result of different algorithms for UDP dataset

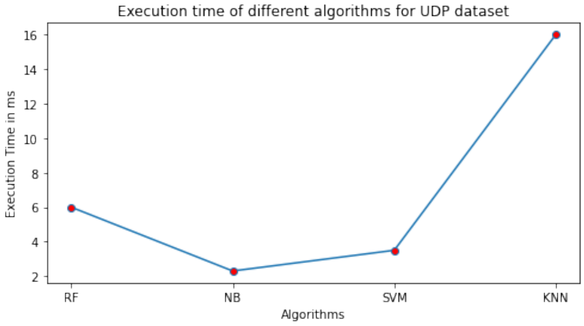

Figure 9: Execution time for UDP datasets

4.4 UDPLag Dataset

Table 4 shows the measures of performance parameters, while Figure 10 shows the comparative chart. The mean accuracy of the RF, NB, SVM and KNN is 0.99, 0.56, 0.70 and 0.99 respectively. The precision, recall and f-measure of RF is 0.99, 1.0 and 0.99, Naïve Bayes is 0.64, 0.67 and 0.65, SVM is 0.74, 0.84 and 0.79 and KNN is 0.99, 0.99 and 0.99 respectively. The execution time of different ML algorithms is shown in Figure 11. When considering the execution time, KNN has taken the longest time to execute of 14.5 ms, while NB has taken the least execution time of 2.2 ms. Also, the execution time of RF and SVM are 5.7 ms and 3.1 ms respectively. KNN and RF have performed relatively better but with the longest execution time.

Table 4: Measures with UDPLag Dataset

Measures	RF	NB	SVM	KNN
Mean Accuracy	0.99	0.56	0.70	0.99
Precision	0.99	0.64	0.74	0.99
Recall	1.0	0.67	0.84	0.99
F-Measure	0.99	0.65	0.79	0.99
Execution Time	5.7ms	2.2ms	3.1ms	14.5ms

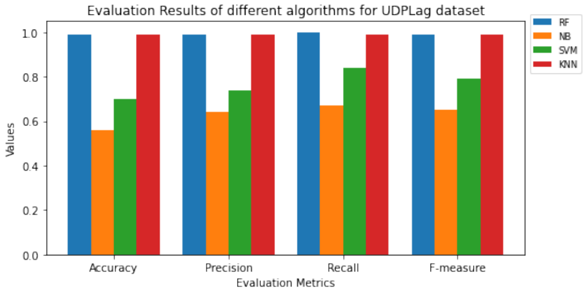

Figure 10: Evaluated result of different algorithms for UDPLag dataset

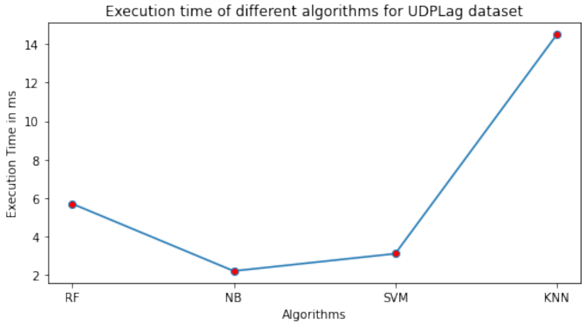

Figure 11: Execution time for UDPLag datasets

4.5 Generated Dataset

Table 5 shows the measure of performance parameters while Figure 12 shows the comparative chart. KNN has achieved the highest accuracy of 0.74 while NB has the least accuracy of 0.52. Also, the mean accuracy of RF and SVM are 0.70 and 0.68 respectively. The precision, recall and f-measure of RF is 0.79, 0.65 and 0.71, NB is 0.37, 0.44 and 0.40, SVM is 0.81, 0.79 and 0.77 and KNN is 0.77, 0.68 and 0.72 respectively. The execution time of the ML algorithms is shown in Figure 13. For the total execution time, KNN has taken the longest execution time of 3.5ms followed by RF with 3 ms, while SVM and NB have taken the least execution time of 1 ms. The reason behind the low accuracy of generated dataset is due to lack of appropriate features to be selected. Also there has been only 2000 dataset which is significantly low, thus hampering the overall performance of the entire machine learning models.

Table 5: Measures with Generated Dataset

Measures	RF	NB	SVM	KNN
Mean Accuracy	0.70	0.52	0.68	0.74
Precision	0.79	0.37	0.81	0.77
Recall	0.65	0.44	0.79	0.68
F-Measure	0.71	0.40	0.77	0.72
Execution Time	3ms	1ms	1ms	3.5ms

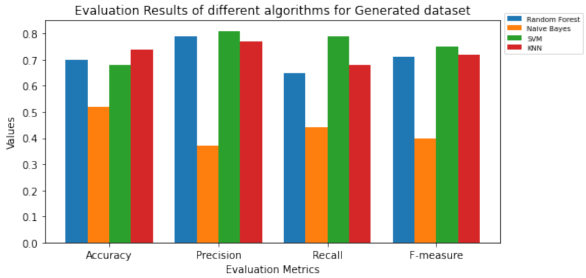

Figure 12: Evaluated result of different algorithms for Generated dataset

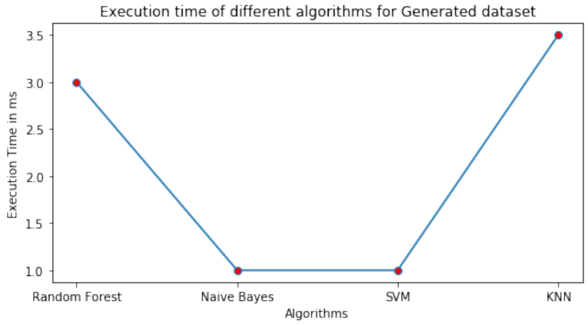

Figure 13: Execution time for Generated datasets

4.6 Limitation and Future Work

F. Limitation and Future Work The scope in this study was limited to the implementation and analysis of supervised machine learning algorithm. The standard dataset used are non-SDN based. Due to lack of standard SDN based dataset, we first trained the model with the standard dataset obtained and additionally implemented the approach over the SDN based traffic dataset by creating environment over Mininet emulator. This generated data only has fewer usable features similar to non-SDN data. Thus, the dataset dimensionality reduction was required for testing the generate data set in this experiment. Further research can be conducted by the implementation and analysis of unsupervised machine learning algorithm. Also, larger generated dataset in Mininet would provide some insightful results for various types of attacks.

5. Conclusion

The DDoS detection was performed on two types of dataset, i.e., standard CICDDoS2019 dataset and generated dataset in virtual environment. In the methodology, various features selection based on attack type were done and further ML procedures were implemented. We compared the different ML algorithms and found the utility of various types of algorithms on the basis of execution time, precision, recall and accuracy. For all the datasets, the execution time is higher for KNN and RF. The reason behind the higher execution time is due to the non-parametric nature of these algorithms. If we consider the tradeoff between the execution time and accuracy, SVM performed better among all the models. Also, in the generated dataset, the overall performance of the model decreased significantly due to lack of complete dataset having high number of features. There were only 3 features selected that were considered while feeding the machine learning algorithm. Hence, ML based DDoS detection in SDN can be performed with high accuracy, if the feature selection process is given enough consideration.

References

[1] Ozgur Depren, Murat Topallar, Emin Anarim, and M Kemal Ciliz. An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert systems with Applications*, 29(4):713–722, 2005.

[2] Merlin James Rukshan Dennis. Machine-learning and statistical methods for ddos attack detection and defense system in software defined networks. 2018.

[3] Junjie Xie, Deke Guo, Zhiyao Hu, Ting Qu, and Pin Lv. Control plane of software defined networks: A survey. *Computer communications*, 67:1–10, 2015.

[4] Babu Ram Dawadi, Danda Bahadur Rawat, and Shashidhar R Joshi. Software defined ipv6 network:

A new paradigm for future networking. *Journal of the Institute of Engineering*, 15(2):1–13, 2019.

[5] Seyed Mohammad Mousavi and Marc St-Hilaire. Early detection of ddos attacks against sdn controllers. In *2015 international conference on computing, networking and communications (ICNC)*, pages 77–81. IEEE, 2015.

[6] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[7] Aarti Singh and Dimple Juneja. Agent based preventive measure for udp flood attack in ddos attacks. *International Journal of Engineering Science and Technology*, 2(8):3405–3411, 2010.

[8] Mohd Azahari Mohd Yusof, Fakariah Hani Mohd Ali, and Mohamad Yusof Darus. Detection and defense algorithms of different types of ddos attacks using machine learning. In *International Conference on Computational Science and Technology*, pages 370–379. Springer, 2017.

[9] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[10] Lohit Barki, Amrit Shidling, Nisharani Meti, DG Narayan, and Mohammed Moin Mulla. Detection of distributed denial of service attacks in software defined networks. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2576–2581. IEEE, 2016.

[11] Rodrigo Braga, Edjard Mota, and Alexandre Passito. Lightweight ddos flooding attack detection using nox/openflow. In *IEEE Local Computer Network Conference*, pages 408–415. IEEE, 2010.

[12] Huseyin Polat, Onur Polat, and Aydin Cetin. Detecting ddos attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability*, 12(3):1035, 2020.

[13] Saif Saad Mohammed, Rasheed Hussain, Oleg Senko, Bagdat Bimaganbetov, JooYoung Lee, Fatima Hussain, Chaker Abdelaziz Kerrache, Ezedin Barka, and Md Zakirul Alam Bhuiyan. A new machine learning-based collaborative ddos mitigation mechanism in software-defined network. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8. IEEE, 2018.

[14] Yao Yu, Lei Guo, Ye Liu, Jian Zheng, and YUE Zong. An efficient sdn-based ddos attack detection and rapid response platform in vehicular networks. *IEEE access*, 6:44570–44579, 2018.

[15] Neelam Dayal and Shashank Srivastava. Analyzing behavior of ddos attacks to identify ddos detection features in sdn. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pages 274–281. IEEE, 2017.

[16] Adel Alshamrani, Ankur Chowdhary, Sandeep Pisharody, Duo Lu, and Dijiang Huang. A defense system for defeating ddos attacks in sdn based networks. In *Proceedings of the 15th ACM international symposium on mobility management and wireless access*, pages 83–92, 2017.

[17] Rogério Leão Santos De Oliveira, Christiane Marie Schweitzer, Ailton Akira Shinoda, and Ligia Rodrigues Prete. Using mininet for emulation and prototyping software-defined networks. In *2014 IEEE Colombian conference on communications and computing (COLCOM)*, pages 1–6. IEEE, 2014.