Classification of Network Traffic using Machine Learning for Software Defined Networks

:

Muhammad Shahzad Haroon
Computer Science Department
SZABIST
Karachi, Pakistan
Shahzad.haroon@szabist.edu.pk

Dr Husnain Mansoor
Computer Science Department
SZABIST
Karachi, Pakistan
Husnain.mansoor@szabist.edu.pk

Abstract

As SDN devices and systems hit the market, security in SDN must be raised on the agenda. SDN has become an interesting area in both academics and industry. SDN promises many benefits which attract many IT managers and Leading IT companies which motivates them to switch to SDN. Over the last three decades, network attacks becoming more sophisticated and complex to detect. The goal is to study how traffic information can be extracted from an SDN controller and open virtual switches (OVS) using SDN mechanisms. The testbed environment is created using the RYU controller and Mininet. The extracted information is further used to detect these attacks efficiently using a machine learning approach. To use the Machine learning approach, a dataset is required. Currently, a public SDN based dataset is not available. In this paper, SDN based dataset is created which include legitimate and non-legitimate traffic. Classification is divided into two categories: binary and multi-class classification. Traffic has been classified with or without dimension reduction techniques like PCA and LDA. Our approach provides 98.58% of accuracy using a random forest algorithm.

Keywords:

SDN, Security, Machine learning, Openflow, dataset

I. INTRODUCTION

Software-Defined Network is a new paradigm that enables flexibility and programmability in the networks. Software-Defined Networking is considered a technology

that can manage the entire network efficiently[1]. SDN can convert the large and complex network architecture into the simple and manageable one. SDN is a separation of control plane and data plane as compared to the conventional networks. In a conventional network, each device act as an intelligent device whereas in SDN each device is controlled centralized. SDN is still in a phase of evolving and research work is currently underway.

SDN architecture is divided into three layers: Application layer, control layer and infrastructure layer. Communication between the application layer and control layer are handled by southbound APIs whereas communication between the control layer and infrastructure is handled by northbound APIs. The application layer is responsible for handling software-related business and security applications. The Control layer is handled by a controller like ODL, floodlight, RYU etc. The infrastructure layer in SDN comprises network devices such as switches(both virtual switches such as Open vSwitch, Indigo and physical switches)

SDN is a centralized control system in which rules are set to be implemented through the forwarding plane. These rules are then followed by devices like routers and switches in the network. SDN presence meets different users requirements according to their needs(SLA, QoS and so on). SDN uses OpenFlow protocol for northbound APIs. OpenFlow protocol can communicate with different heterogeneous devices. Apart from centralized control SDN has another benefit that it can flow traffic on IP-prefix base due to the support of OpenFlow which accommodate 44 different header fields[2]. These header fields are then used to match flow entries, setting traffic flow on IP -prefix base is one of them. Data flow information of the SDN network is made available by OpenFlow which can be used to determine traffic patterns. These traffic patterns can be analyzed using machine learning algorithms for future prediction of network growth, quality of services, traffic engineering, security issues like DDOS, man in the middle etc[3] Advantages of SDN have already been proven as it been implemented in major IT stakeholders but one area which is concerning is SDN security. SDN systems can be explored to enhance security. The monitoring log, the data pattern and various response times can be gathered and analyzed through application reside in a centralized controller. As the controller contain all the information and is a focal point of the whole SDN architecture, security threats can be identified using a different application. Threads like DoS can easily affect the centralized controller and flow table. Another issue between the layers is trust; for example, application layer and controller layer or controller layer and network devices layers. The solution to these

Manuscript received December 5, 2023
Manuscript revised December 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.12.5

challenges is already been proposed in many kinds of literature [1].

Recent studies have shown that old-fashioned networks are not suitable for upcoming challenges[4]. A growing number of devices due to the increased number of users and technology like IoT need more active and flexible networks. Traditional networks only support vendor-specific hardware and software. To upgrade the network, need replacement of hardware which includes reconfiguration of the whole network, usually a costly solution. SDN has provided an alternative solution. While upgrading your network just need software to update.

Rapid growth in the number of users due to technology like IoT and other network-based applications. Intrusion detection becomes a highly important aspect of any network infrastructure. The performance of intrusion detection systems remains a concern to detect, identify and track the attacker footprints. To detect footprints with high accuracy researchers are continuously working and contributing to the network society[5]. IDS are divided into two types; misuse and anomaly. Misuse IDS are based on the databases of attacks signature whereas as Anomaly-based IDS can classify the normal and abnormal activity by monitoring the network traffic. The advantage of anomaly-based IDS is it can detect more types of unknown attacks[6]. Misuse IDS have a very low rate of attack detection whereas anomaly IDS are difficult and time-consuming to analyze a large amount of data[6][7]. Machine learning techniques can be useful to detect and prevent network infrastructure from an intruder.

II. Literature Review

Many researchers have exhaustively used the benchmark dataset KDDCUP99. KDDCUP99 included 22 attack types in the training dataset and testing data contained 15 attack types [11][12]. The major reason for making KDDCUP a benchmark is the public availability of the dataset. The research community also highlighted the disadvantages or disappearance of KDDCUP99 [13][14][15].

NSLKDD, an upgrade version of KDD was introduced as keeping three goals to improve KDD. The first is to remove all the duplication of records. Second, selecting a variety of records from different parts of the original KDD dataset is to achieve reliable results from classifier systems. Third, eliminating the unbalancing problem among the number of records[6]. NSLKDD still lack those scenarios which contain low footprints in modern attacks[7].

The unavailability of a comprehensive network-based dataset that fulfils all the parameters of modern traffic leads to another dataset UNSWNB15. Researchers have used the UNSWNB15 dataset to detect attacks in multiple ways. Multiple works have been reported since the dataset was publicly available[16]. Classification of the incoming network traffic into DoS traffic or normal traffic [11]. Comparison between NSL-KDD and UNSWNB15 dataset by using different machine learning algorithm[12]. Detection of known unknown web attacks using logit boost algorithm[13]. Random Forest performs better while including all the 42 features which in result identify traffic in normal or abnormal[14]. Feature selection has been predicted by using different techniques[15].

The public availability of the dataset is the most traditional based network. The researcher already used this dataset extensively. In this paper, our approach is to generate a dataset based on the SDN network. T the best of our knowledge SDN based public dataset is not available. Although, researchers have created SDN datasets and performed several testing these datasets are not made public[14]–[18].

The paper is divided into two sections; In the first section, SDN based dataset is created which include legitimate and non-legitimate traffic. To generate non-legitimate traffic, attacks have been generated within the SDN network. The flow of the traffic has been generated and captured using various tools. In the second section, the dataset is used to classify legitimate traffic and non-legitimate traffic using various machine learning algorithms. That algorithm that performs well in comparison with others is further discussed in detail to analyze performance parameters. Performance parameter includes Training and testing accuracy, true positive ratio, prediction speed and training time.

III. Methodology

To use the machine learning approach to classify modern security attacks on SDN based networks we have to prepare a dataset.

The study is divided into two major categories, Binary classification and Multi-class classification. Binary classification involves the identification of legitimate and non-legitimate traffic. Multi-classification involves the identification of 6 types of classes which include 5 network attacks and 1 normal traffic.

The testing of machine learning algorithms have been performed in the three-phase; in the first phase, multiple classification algorithms have been applied without any dimension reduction technique; in the second phase, Principle component analysis (PCA) has been applied to before classification; in the third phase, Linear discriminant analysis (LDA) has been applied before classification. The result of all three stages is compared and discussed in the later stage.. The dataset is created by the virtual environment. The virtual environment is developed between two virtual machines. These machines are Linux based operating systems and the distribution we have used is Ubuntu 17.07.

The first virtual machine we used is to create an SDN based controller. Multiple options are available for the SDN controller. A few of them are discussed here.

A. HPE

HPE VAN SDN controller has a modular architecture and it is built upon off-the-shelf Ubuntu Linux, Java 1.7, and OSGI.

B. Cisco

Cisco is one of the leading vendors in network devices. Cisco provides a commercial version of the Open daylight controller. The innovation from Cisco and the open daylight community continuously upgraded the SDN controller concerning modern requirements.

C. Open day Light (ODL)

ODL is a modular open SDN platform that is extensible for networks of any size and scale. And its multi-protocol infrastructure enables network services on top of multivendor environments.

D. Extreme networks one controller

Extreme One controller provides support for the multi-vendor environment in hardware and software. One controller has the capability of backward compatibility. It also has the support of OpenFlow. Through multivendor and backward compatibility, it saves lots of investment of the customer.

E. FloodLight

The Floodlight controller is based on a modular architecture developed in Java and realizes a set of common functionalities to control and inquire an OpenFlow network. It has also a Representational State Transfer Application Program Interfaces (REST APIs), that can be written in any language and exchange information with an external entity at runtime

F. RYU

RYU controller is managed by the RYU community. RYU controller is developed on python and its source code is hosted on Github[22]. All the Ryu code is available free to use under the Apache 2.0 license. RYU support OpenFlow protocol which communicates between southbound API. RYU also supports other network management protocols like NETCONF and PF-Config. For the demonstration of our work, the RYU controller is selected among all the available controllers.

The second virtual machine is used to develop a network topology. Multiple options are available, few of them are discussed here,

G. OPNET

OPNET is owned by Riverbed technology. OPNET is a network simulator used to test the performance and behaviour of any kind of network. The main key feature of OPNET is computational power. A variety of protocols used in the network are available and can be simulated to test the performance evaluation. OPNET is a widely used tool among the research community.

H. GNS3

Graphical Network Simulator-3 is known as GNS3. GNS3 is another network simulator. It is a widely used tool among network professionals who are involved in certification programs. GNS3 also provide support to the real and virtual network. GNS3 provide support for the vendors like cisco juniper and virtual box etc.

I. MININET

Mininet is another widely used network emulator in the research community[23]. Mininet runs on Linux OS. It is used to create a network of switches, hosts, controllers and hosts with their links. To demonstrate our work, Mininet is selected because of its supports OpenFlow with high flexibility in the domain of the software-defined network.

The connection between the RYU controller and mininet refer to Figure 1.

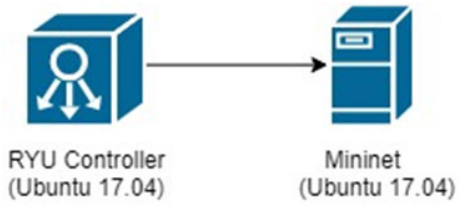

Figure 1

IV. SUPPORTING TOOLS

A. IPERF

IPERF is a tool used to test the network performance parameter like maximum attainable bandwidth. IPERF also allow the user to modify its parameters like tuning the timing, buffer and protocols. It supports protocols like TCP, UDP, SCTP with IPv4 and IPv6. In our work, IPERF is used to generate TCP packets with a maximum reachable bandwidth. The TCP packets generated through IPERF are counted as legitimate network traffic with further captured by tcpdump[24].

B. SCAPY

Scapy is another tool to generate network traffic. Scapy is based on python2 and python3. Scapy allows generating a packet with a wide number of protocols. It is a powerful tool that allows the manipulation of packets. Scapy can generate legitimate and non-legitimate traffic. Non-legitimate traffic includes ARP Cache Poisoning, Attack - Ping of Death, Smurf Attacks, SYN Flooding Attack, Overlapping Fragments and many more[25].

In our work, Scapy is used to generate host-based network attacks. Through Scapy non-legitimate traffic is captured by tcpdump along with the legitimate traffic.

C. TCPDUMP

Tcpdump is a tool to capture network traffic[26]. It allows capturing all the TCP/IP traffic generated in the network. Tcpdump also allows analyzing packets by using tcpdump command line. The traffic that has been captured through tcpdump is further converted in CSV format for further data pre-processing.

D. Microsoft Office Excel

After running the desired tests through the above-mentioned tools. Tcpdump provide a CSV file for further preprocessing of data. Data preprocessing can be done through many programming languages like python, C language etc. In our work, Microsoft Excel has been selected due to building in tools like Data analysis, text to the column, VBA Stripchar function[27]. The network traffic that has been captured by tcpdump is divided into many columns. These columns contain a lot of information that must be used for classifying the network traffic. Few of the columns contain non-numeric data, multiple decimal points like in IPv4 format, and header in a string format. To cater to these problems and to create a dataset, Microsoft Excel is the most suitable choice in terms of a user-friendly GUI platform and with all supporting tools.

We have used the text to column tool to divide the string into multiple columns. VBA based stripchar function is used to eliminate all the non-numeric data.

E. Matlab

Matlab is one of the well-known products for simulation, analysis and design processes [28]. In our work, Machine learning algorithms are implemented by using the Matlab toolbox. Matlab support both supervised learning and non-supervised learning. The dataset we have created is a combination of normal and malicious traffic. Through Matlab, classification algorithms are used to classify legitimate traffic and non-legitimate traffic.

V. TESTBED

Testbed environment has been created on Intel Core™ i7 CPU@3.4GHZ with 16GB of ram. One virtual machine is created for the RYU controller and the other is created for mininet. Network topology has been developed on mininet using one Open vswitch and 20 pcs. RYU controller is connected with Open vswitch. The testbed environment is shown in figure 2.

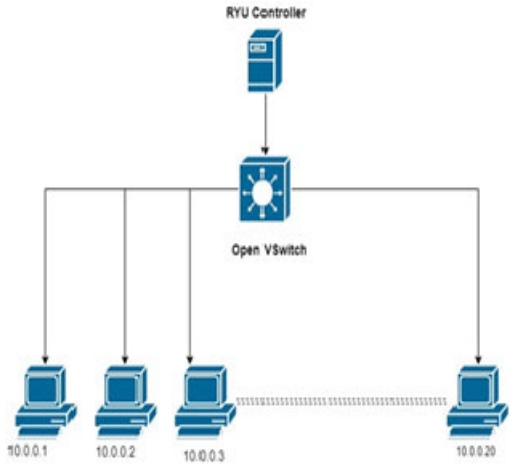

Figure 2

VI. NON-LEGITIMATE NETWORK TRAFFIC

A. ARP Cache Poisoning

Arp poisoning is a method through which an attacker sends spoofed ARP messages. These messages include the attacker Mac address along with the victim IP address. Once the destination ARP table is updated with a spoofed Mac address, an attacker can easily receive messages intended for the victim. This technique can be used to target the host, gateway or any Layer 3 device.

Arp spoofing may lead to a denial of service attack, a man in the middle attack and open many avenues to attack the network. Until the arp table remains in the cache system remain in the vulnerable state.

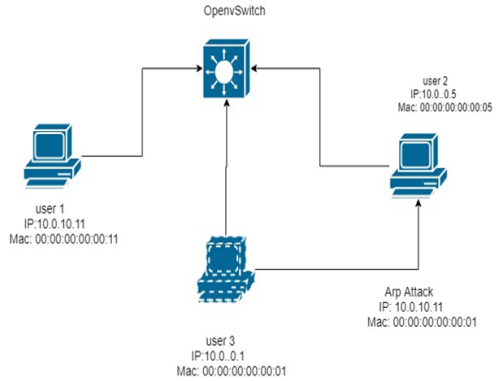

Figure 3

Case 1: User 1, User 2 and User 3 are connected through the same Open vswitch. Use 1 and User are considered legitimate users while User 3 acts as an attacker. From User 3 spoofed ARP messages are generated for User 2 to poison the arp table. All the legitimate traffic and non-legitimate traffic have been captured which involved User 2 response as well against the ARP attack. Refer to Figure 3.

B. Smurf Attacks

A smurf attack is a simple method to create a DOS attack. It is very simple to generate a DOS attack and disturb the network performance. The attacker sends thousands of ICMP packets using a spoof IP address of a victim. These ICMP packets are sent to all the devices in the network as a broadcast message. The victim has to respond to all the ICMP request but is unable to respond all as these are in thousands in number which result in not responding. Ultimately, it also creates a bottleneck in the network and might lead to downtime of whole the network.

Case 2: In this scenarios, 10.0.0.1 is a victim IP while from 10.0.0.2 to 10.0.0.4 are legitimate users. All the legitimate users and the attacker are connected by an Open vswitch. The attacker has used 10.0.0.1 IP to create a smurf attack on the victim pc. Victim pc has to respond to all the legitimate users but a large number of ICMP packets create a bottleneck for the victim. Refer to figure 4.

Tcpdump is configured at open vswitch to capture all the communication. A smurf attack is generated using scapy on the attacker virtual host machine.

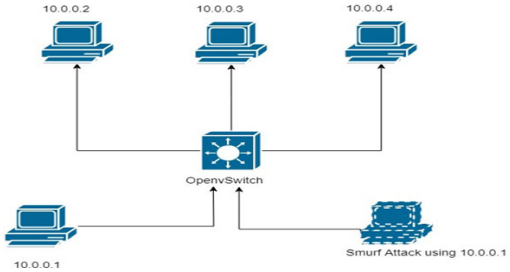

Figure 4

C. SYN Flooding Attack

TCP SYN flood attack is another way to create a broadcast. This type of broadcast usually target web services. Normally, during the session establishment between a web server and a client, a three-way handshake is established. Three-way handshake contains three messages, the first message is initiated by the client to server TCP SYN. The second message is a response to the client request by server known as TCP SYN-ACK. The last message is sent by client SYN-ACK to the acknowledgement of the complete process.

In the TCP syn flooding, a client which act as an attacker will initiate a TCP SYN request and server in response send a TCP SYN-ACK. The last message SYN – ACK which expected to be sent by the client is not generated, hence leaving the server in the waiting mode. These type of messages are sent in large number with the different number of ports. Different ports open different TCP connections and all eventually end up in waiting mode. On the other hand, legitimate users face denial of service attack as the server is busy with the attacker requests.

Case 3: In this scenario, Two pcs are attached through open vswitch. One is acting as a TCP server and other is a client. The client is configured with SYN flood script which initiates each TCP SYN request attack with a different number of the source port. Tcpdump is configured at open vswitch to capture all the communication. Refer to figure 5

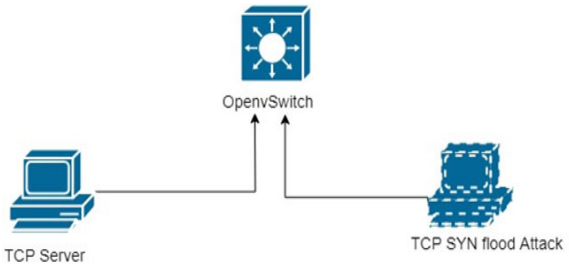

Figure 5

D. Overlapping Fragments

This attack is achieved by sending a packet with a maximum of MTU size. IP layer fragment the packet when it sees that the MTU size of the packet is increasing the limits. IP fragment overlapping occurs when the IP datagram contains the same IP and overlaps the position of the datagram.

Case 4: In this scenario, multiple fragmented packets are sent from one user to another user. Two users are connected through an open v switch. Tcpdump is enabled on the switch to capture all the communication. Overlapping

fragment attack is generated using scapy on the attacker virtual host machine.

E. Attack - Ping of Death

Ping of death is another way to create a broadcast. This type of broadcast results in denial of service to the victim host, multiple users or may lead to the whole network.

A legitimate IPv4 packet has an IP header of 65,535 bytes. If an attacker sends a ping packet more than 65,535 of size, the victim pc start to reassemble the fragmented packets. These ping packets are in large number and fragmentation overflow occurs at the victim pc.

Case 5: In this scenario, two pcs are connected by an open v switch. One pc is legitimate and the other is the attacker. Ping of death is sent with large size. This attack is generated on scapy and 7captured using tcpdump. Tcpdump is enabled on open vswitch.

VII. PRE-PROCESSING OF DATASET

Originally the data has been captured from tcpdump includes seven features. These seven features include Serial number, Time of the packet, source IP, destination IP, Protocol, length of packet and info. The serial number, time of the packet and length were remained unchanged as these three features are pure numeric data. Source IP and Destination IP were in IPv4 format which includes four octets. As the data should be in the numeric format or in a fraction format, four octets were removed from the IPv4 format and converted in a plain numeric form. For some protocols like ARP, the source IP field was received as a source mac address. Mac address format in a semicolon and alphanumeric format due to the hexadecimal form. Mac address format conversion was held in two stages, initially, all the semicolons were removed. Secondly, hexadecimal values are converted in decimal. Protocol field includes ICMP,ARP TCP,HTTP,IPv4 etc. These protocol names were also converted to respective numeric formats. The info field contains lots of information. Each protocol header detail was available in the info feature but in the string format. To extract the information from the string, the data had been sorted by protocol separately in a separate excel file. The protocol bases files are handled separately and merged in a single file after all the pre-processing. Before merging them in a single file, the info field was converted from string to numeric. String to numeric was accomplished by using the text to column function in excel. After all the pre-processing of data, the 7 features were converted into 36 features.

VIII. STANDARD SCALER

Pre-Processing step has provided 37 features. These features are not on the same scale. Few of the features are having big numbers and on the other hand, few of them are dealing with zeros and ones. The vast majority of machine learning algorithms perform well when dataset having to feature on the same scale. Multiple options are available to deal with large number MinMax scale, standard scale and normalize. After a standard scale, the distribution centred around zero with a standard deviation of 1. In this study, a standard scale is implemented based on the below equation.

IX. DATASET

After all the pre-processing, 36 features were developed. The 37th feature was created to label the data. During the test, we have noted the traces for different attacks. With the help of the traces, the dataset has been labelled as 1 for non-legitimate traffic and 2 for legitimate traffic. A total of 127262 records is included in the dataset which includes both legitimate and non-legitimate traffic. The data set was further divided into the train and test CSV files. Train CSV file contains 60% of 127261 which almost become 77228 and test CSV file contain 40% of 127262 which almost become 50033.

A. Binary division of dataset

Table 1 Binary Mapping

Traffic Type	Class label
Legitimate traffic	2
Non-Legitimate traffic	1

In the first category of this study identifies legitimate and non-legitimate traffic. Dataset is classified into two classes. The records contain the traffics which includes multiple network attacks labelled as type 1 and the remaining legitimate traffic labelled as 2.

B. Multi-class division of dataset

The second category of the study identifies all the classes individually. Attacks are labelled in five classes along with normal classes containing legitimate traffic.

Table 2 Multi-Class Mapping

Traffic Type	Class label
ARP Cache Poisoning	1
Legitimate traffic	2
Smurf Attacks	3
SYN Flooding Attack	4
Overlapping Fragments	5
Attack - Ping of Death	6

X. MACHINE LEARNING TECHNIQUES

This section of the paper includes exhaustive testing of the dataset with multiple machine learning algorithms. The testing has been performed in the three-phase; in the first phase, multiple classification algorithms have been applied without any dimension reduction technique; in the second phase, Principle component analysis (PCA) has been applied to before classification; in the third phase, Linear discriminant analysis (LDA) has been applied before classification

A. Principle Component Analysis

PCA is a dimension reduction technique for the unsupervised machine learning algorithm. The purpose behind PCA is to reduce the number of variables, extracting the maximum information with less a variable with a high rate of variance. PCA is normally performed on a large dataset which eventually helps machine learning algorithm to perform better, faster and accurate. variables together to form a smaller number of an artificial set of variables which is called "principal components" that account for the most variance in the data..

B. Linear Discriminant Analysis

LDA is used as a classifier and dimension reduction technique. LDA is also referred to as a supervised algorithm used for machine learning algorithms. LDA basically maximize the separation between the multiple classes.

C. Random Forest

Random forest develops multiple decision trees to classify more accurately. A decision tree is a tree where each root is connected to its branch via the link. The root is represented as the strongest feature. The impact of the features is higher at the root and gradually decreases as the tree grows through its branches. Links are the rules on which root attribute is connected to its branches attributes

D. Naïve Bayes

Naïve Bayes is fundamentally a group of theorems that share a common principle based on Bayes theorems. In naïve Bayes, each feature is classified independently and contribute to the outcome equally. Bayes theorems based on the below expression

E. Kernel SVM

Support vector machine algorithms use the kernel to take data as an input and transform it into the required method[14]. The SVM kernel is a set of mathematical functions. The different variant of SVM uses different types of kernel. The variant of SVM is, for example, Linear SVM, Non-Linear SVM, polynomial SVM, radial basis function and sigmoid. The inner product between two points is returned by the kernel in each feature space. SVM can work with a large dataset with little computational cost

F. K Nearest Neighbor

the k-closest neighbour calculation, regularly condensed K-NN is a way to deal with information order that gauges how likely an information point is to be an individual from one gathering or the other relying upon what amass the information directs closest toward it is in[15]. A case is classified by a majority vote of its neighbours, with the case being assigned to the class most common amongst its K nearest neighbours measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbour.

G. Logistic Regression

Logistic regression is a very commonly used machine learning algorithm. The algorithm is used to predict possibilities from the multiple outcomes.

XI. RESULTS

The result of this study is divided into two categories; binary and multi-class. With binary classification, we have performed three phases which includes classification, classification with PCA and classification with LDA. In the first phase, classification has been performed with multiple classification algorithms. The algorithm includes K-NN, kernel SVM, logistic regression, naïve Bayes and random forest. After examining all the testing results of this algorithm we have concluded that random forest has performed well among these with an accuracy of 98.58%. A comparison of these results is available in figure 6. Similarly, in the second phase, before the classification begins, PCA has been applied to the dataset and the five independent features extracted have more than 50% variance in total. The above-mentioned algorithms have been applied and compared in figure 6. The results show that with the inclusion of PCA, Kernal SVM has performed best among them with an accuracy of 92.99%. Similarly, in the last phase, before the classification begins, LDA has

been applied to the dataset. The above-mentioned algorithms have been applied and compared in figure 6. The results show that with the inclusion of LDA, the random forest has performed best among them with an accuracy of 96.98%. In the comparison between multiple binary classifications that we have performed, we concluded that classification with random forest with an accuracy of 98.58% achieves a high prediction rate.

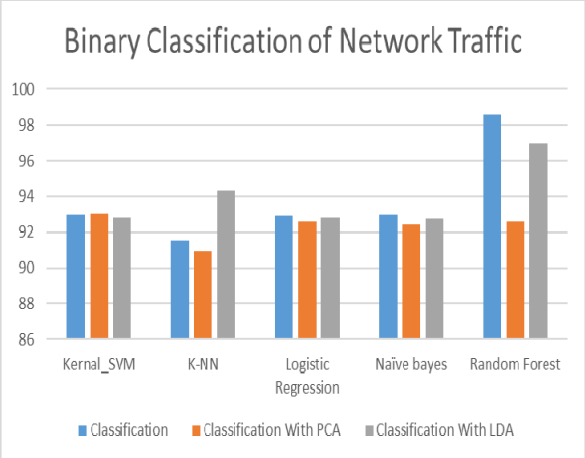

Figure 6 Binary Classification

.In the second category of our study with multi-class classification, we have adopted a similar methodology as in binary classification. We have labelled our dataset into 6 classes. In the first phase, the classification with the random forest has performed well among all with 98.56%, In the second phase, again random forest has predicted better than other with 92.74% accuracy and similarly in the last phase, random forest performed well with an accuracy of 97.24%. These results can be seen in figure 7.

Classification of legitimate traffic and non-legitimate traffic for the Decision tree is shown in figure 7. Figure 7 reflects 1 as non-legitimate traffic and 2 a-s legitimate traffic. Red legend represents train results and blue represents test results.

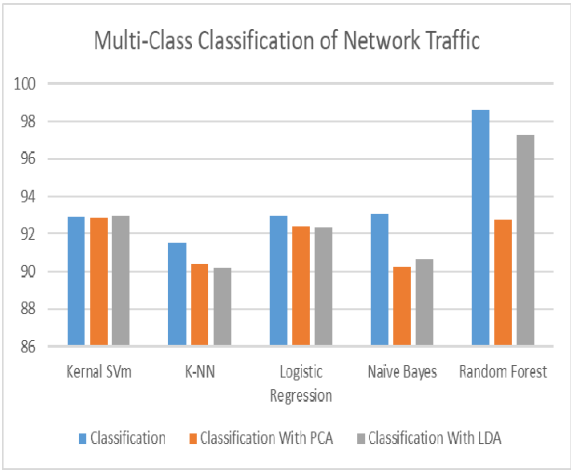

Figure 7 Multi-Class Classification

With the comparison of both binary and multi-class classification, the variant of binary classification with the random forest has been classified better than others. In table 3, the confusion matrix of the binary classification of the random forest shows six classes where class 6 missed classified 269 records out of 2238, Class 2 missed 189 records out of 14400 records.

Table 3 Confusion Matrix-Binary Classification-Random Forest

Confusion Matrix			True Class					
			1	2	3	4	5	6
Random Forest	Predicted Class	6	0	269	0	0	0	1969
		5	0	0	0	0	64	0
		4	0	0	0	74	0	0
		3	0	0	15018	0	0	0
		2	0	14211	0	0	0	189
		1	22	0	0	0	0	0

XII. CONCLUSION

The non-availability of the SDN real dataset motivates this study. To analyze the traffic pattern of the SDN based network to ensure security, the dataset is required which is currently not available as SDN is still in the laboratory testing phase. The goal is to study how traffic information can be extracted by an SDN controller and open virtual switches(OVS) using SDN mechanisms. The extracted traffic is further used to classify legitimate and non-legitimate traffic. The emulated environment for SDN architecture is developed and traffic is captured which

contains all the former traditional network protocols with the inclusion of SDN based protocol OpenFlow. The testbed is designed to generate legitimate and non-legitimate traffic. Non-legitimate traffic is launched within the SDN network. The captured traffic contains seven features which are further converted into 37 features due to the pre-processing of the dataset. The dataset is further used to classify as a binary and multi-class classification. The dataset is further analyzed using machine learning algorithms with dimension reduction techniques like PCA and LDA. With all the different testing results random forest performed well with an accuracy of 98.58%

XIII. Future Work

Multiple options are available to enhance this work to improve attack detections. Attacks that are generated are launched within the SDN network, the attack can also be launched from outside the SDN network. Five types of attacks are launched which are the most common attacks, these attacks can be increased. The results of the methodology adopted are promising. Similar methodology can be used with a large dataset with an increased number of attacks and records..

References

[1] D. B. Rawat and S. R. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey," IEEE Commun. Surv. Tutorials, vol. 19, no. 1, pp. 325–346, 2017.

[2] S. A. Shah, J. Faiz, M. Farooq, A. Shafi, and S. A. Mehdi, "An architectural evaluation of SDN controllers," IEEE Int. Conf. Commun., vol. 1, pp. 3504–3508, 2013.

[3] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, "Machine Learning in Software Defined Networks: Data collection and traffic classification," 2016 IEEE 24th Int. Conf. Netw. Protoc., no. NetworkML, pp. 1–5, 2016.

[4] M. C. Dacier, H. Konig, R. Cwalinski, F. Kargl, and S. Dietrich, "Security Challenges and Opportunities of Software-Defined Networking," IEEE Secur. Priv., vol. 15, no. 2, pp. 96–100, 2017.

[5] C. Applications, " Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling ☆," vol. 87, no. November 2016, pp. 185 – 192, 2017.

[6] N. Moustafa, J. Slay, and I. Technology, "Intrusion Detection systems," 2015.

[7] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," no. Cisda, pp. 1–6, 2009.

[8] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," SDN4FNS 2013 - 2013 Work. Softw. Defin. Networks Futur. Networks Serv., 2013.

[9] S. Jantila and K. Chaipah, "A Security Analysis of a Hybrid Mechanism to Defend DDoS Attacks in SDN," Procedia Comput. Sci., vol. 86, no. March, pp. 437–440, 2016.

[10] A. J. Pinheiro, E. B. Gondim, and D. R. Campelo, "An efficient architecture for dynamic middlebox policy enforcement in SDN networks," Comput. Networks, vol. 122, pp. 153–162, 2017.

[11] K. Afdel, "DoS Detection Method based on Artificial Neural Networks," no. May, 2017.

[12] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," J. Inf. Secur. Appl., vol. 41, pp. 1–11, 2018.

[13] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, "A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks," IEEE Access, vol. 5, pp. 26190–26200, 2017.

[14] M. Belouch, S. El Hadaj, and M. Idlianmiad, "Performance evaluation of intrusion detection based on machine learning using apache spark," Procedia Comput. Sci., vol. 127, pp. 1–6, 2018.

[15] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," IEEE Int. Symp. Ind. Electron., pp. 1881–1886, 2017.

[16] S. K. Fayaz, Y. Tobioka, V. Sekar, M. Bailey, and M. Bailey, "Bohatei : Flexible and Elastic DDoS Defense This paper is included in the Proceedings of the," 2015.

[17] N. Anand, S. Babu, and B. S. Manoj, "On detecting compromised controller in software defined networks," Comput. Networks, vol. 137, pp. 107–118, 2018.

[18] N. Meti, D. G. Narayan, and V. P. Baligar, "Detection of Distributed Denial of Service Attacks using Machine Learning Algorithms in Software Defined Networks," pp. 1366–1371, 2017.

[19] X. You, Y. Feng, and K. Sakurai, "Packet In message based DDoS attack detection in SDN network using OpenFlow," 2017.

[20] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," Inf. Secur. J., vol. 25, no. 1–3, pp. 18–31, 2016.

[21] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc., 2015.

[22] RYU SDN Project Team, "RYU SDN Framework," 2016. [Online]. Available: https://osrg.github.io/ryu-book/en/Ryubook.pdf. [Accessed: 27-Dec-2018].

[23] Mini net team, “Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet,” 2018. [Online]. Available: http://mininet.org/. [Accessed: 27-Dec-2018].
[24] Esnet and Lawrence Berkeley National Laboratory, “iPerf - The TCP, UDP and SCTP network bandwidth measurement tool,” Iperf.Fr, 2016. [Online]. Available: https://iperf.fr/. [Accessed: 27-Dec-2018].
[25] “Scapy.” [Online]. Available: https://scapy.net/. [Accessed: 27-Dec-2018].
[26] M. G. Luis, “TCPDUMP/LIBPCAP public repository,” Online Doc., 2009.
[27] Microsoft Corporation, “Use the Analysis ToolPak to perform complex data analysis,” Microsoft Office Support, 2018. [Online]. Available: https://support.office.com/en-us/article/Use-the-Analysis-ToolPak-to-perform-complex-data-analysis-6C67CCF0-F4A9-487C-8DEC-BDB5A2CEFAB6. [Accessed: 27-Dec-2018].
[28] Mathworka, “Machine Learning with MATLAB - MATLAB & Simulink,” 2016. [Online]. Available: https://www.mathworks.com/campaigns/products/offer/machine-learning-with-matlab.html?s_tid=hp_offer_ml_ebok. [Accessed: 27-Dec-2018].