

CAPITAL UNIVERSITY OF SCIENCE AND  
TECHNOLOGY, ISLAMABAD



# Support Vector Machine Based Distributed Denial of Service Detection and Mitigation in Software Defined Network

by

Sadia Rasheed

A thesis submitted in partial fulfillment for the  
degree of Master of Science

in the

Faculty of Computing

Department of Computer Science

2023

Copyright © 2023 by Sadia Rasheed

All rights reserved. No part of this thesis may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, by any information storage and retrieval system without the prior written permission of the author.

*I want to dedicate this achievement to my parents, teachers and friends who  
always encourage and support me in every crucial time*



## CERTIFICATE OF APPROVAL

### Support Vector Machine Based Distributed Denial of Service Detection and Mitigation in Software Defined Network

by

Sadia Rasheed

Registration No: (MCS193016)

### THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Ammara Gul	Air University
(b)	Internal Examiner	Dr. Amir Qayyum	CUST, Islamabad
(c)	Supervisor	Dr. Muhammad Siraj Rathore	CUST, Islamabad

Dr. Muhammad Siraj Rathore

Thesis Supervisor

December, 2023

Dr. Abdul Basit Siddiqui

Head

Dept. of Computer Science

December, 2023

Dr. M. Abdul Qadir

Dean

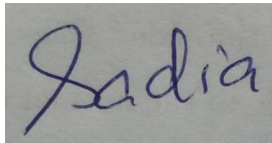
Faculty of Computing

December, 2023

## *Author's Declaration*

I, **Sadia Rasheed**, hereby state that my MS thesis titled “**Support Vector Machine Based Distributed Denial of Service Detection and Mitigation in Software Defined Network**” is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

A rectangular box containing a handwritten signature in blue ink that reads "Sadia".

**(Sadia Rasheed)**

Registration No: (MCS193016)

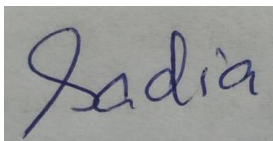
---

## *Plagiarism Undertaking*

I solemnly declare that research work presented in this thesis titled “**Support Vector Machine Based Distributed Denial of Service Detection and Mitigation in Software Defined Network**” is exclusively my research work with no remarkable contribution from any other individual. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the Higher Education Commission and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above-titled thesis declare that no part of my thesis has been plagiarized and any material used as reference is properly cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

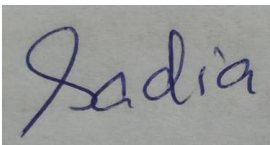


**(Sadia Rasheed)**

Registration No: (MCS193016)

## *Acknowledgement*

First and foremost, I express my gratitude to the most merciful, powerful, and benevolent **ALLAH** Almighty for bestowing upon me the skills, knowledge, and unwavering determination to reach this point and achieve my research goal. I extend my sincere appreciation to my supervisor, **Dr. Muhammad Siraj Rathore**, whose guidance and support were instrumental in completing my research thesis. It was through her inspiration that my interest in research work was ignited.

A handwritten signature in blue ink on a grey background, reading "Sadia".

**Sadia Rasheed**

# *Abstract*

The Software Defined Networking (SDN) is an innovative network architecture that offers flexible and programable networks through a centralized controller. However, If the controller fails the whole system becomes paralyzed. The Distributed Denial of Service (DDoS) attack is one of the main threats to the SDN controller, as it exhausts the resources of the SDN controller which disturbs the whole network and affects the performance of the network. There are several machine learning techniques which can be used to classify DDoS attacks from the normal traffic. The objective is to detect, mitigate and protect controller and switches from DDoS attacks. In this regard we propose a Support Vector Machine (SVM) based machine learning model to detect the DDoS attacks. We also propose a mitigation module that can block all the attack traffic within a short period of time. We evaluate our proposed solutions on two different datasets. i.e., KDD and KDD'99. We select important features from these datasets and train our SVM classifier to accurately detect the DDoS attacks. Our experimental results demonstrate the accuracy of 99.87% for KDD whereas the accuracy of 87.90% is reported for KDD'99. In addition, we also identify how the different selected features can impact the accuracy of the classifier. Finally, the mitigation module is evaluated that blocks attack traffic within a within 11.25 seconds after the detection of the attack.



# Contents

<b>Author’s Declaration</b>	<b>iv</b>
<b>Plagiarism Undertaking</b>	<b>v</b>
<b>Acknowledgement</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Software Defined Network . . . . .	1
1.2 Distributed Denial-of-Service (DDoS) . . . . .	4
1.3 Machine Learning Algorithm . . . . .	6
1.4 Support Vector Machine (SVM) . . . . .	8
1.5 Datasets . . . . .	11
1.6 Motivation . . . . .	12
1.7 Problem Statement . . . . .	12
1.8 Research Question . . . . .	13
1.9 Proposed Research Methodology . . . . .	13
1.10 Organization of Thesis . . . . .	14
<b>2 Literature Review</b>	<b>15</b>
2.1 Introduction . . . . .	15
2.2 DDoS Detection Technique . . . . .	16
2.3 DDoS Detection and Mitigation Technique . . . . .	22
2.4 Research Gap Identification and List of Contribution . . . . .	26
2.4.1 DDoS Detection Trigger Mechanism . . . . .	27
2.4.2 Flow Extraction . . . . .	27
2.4.3 Feature Extraction and DDoS Detection Algorithm . . . . .	27
2.4.4 DDoS Defense Mechanism . . . . .	28

---

<b>3</b>	<b>Research Methodology and Performance Evaluation</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Research Methodology . . . . .	30
3.3	Proposed System Architecture for DDoS Detection and Mitigation	31
3.4	Attributes Extraction . . . . .	34
3.5	Selected Attributed for Classifier of Attack Traffic . . . . .	35
3.6	Summary . . . . .	37
<b>4</b>	<b>Results and Discussions</b>	<b>38</b>
4.1	Results and Discussions . . . . .	38
4.2	Evaluation Metrics and Methods for Solution . . . . .	39
4.3	Experimental Setup . . . . .	41
4.4	Traffic Features Comparison Normal Traffic vs. Attack Traffic . . .	42
4.5	Accuracy and Precision of SVM . . . . .	51
4.6	Mitigation of DDoS Attack . . . . .	54
<b>5</b>	<b>Conclusion and Future Work</b>	<b>56</b>
5.1	Conclusion and Future Work . . . . .	56
	<b>Bibliography</b>	<b>58</b>

# List of Figures

1.1	SDN Architecture Overview	3
1.2	SDN Architecture	3
1.3	Attack Environment	5
1.4	DDoS Mitigation	6
1.5	Machine Learning Algorithm Applied at SDN controller	8
1.6	Ideal Hyper Plane	9
1.7	Data Points in Hyper Plane	10
1.8	SVM Kernel Function	11
2.1	DDoS Attack	16
3.1	Network Topology	31
3.2	Attack Detection and Mitigation	33
4.1	System Architecture for analysis of DDoS attack	38
4.2	Flow Table Entries[1]	39
4.3	System Architecture for DDoS Attack	42
4.4	Flow Arrival Rate (Normal Traffic & Attack Traffic)	43
4.5	Mixed Traffic (Normal+Attack Traffic)	44
4.6	The standard deviation of the Number of Flow Packets (Normal & Attack Traffic)	45
4.7	The standard deviation of the Number of Flow Packets (Mixed Traffic (Normal + Attack Traffic))	46
4.8	The Standard Deviation of the Number of Flow bytes (Normal & Attack)	46
4.9	The Standard Deviation of the Number of Flow bytes(Mixed Traffic (Normal + Attack))	47
4.10	Speed of Flow Entries (Normal and Attack))	48
4.11	Speed of Flow Entries(Mixed Traffic (Normal + Attack))	49
4.12	Ratio of Pair Flow Mixed Traffic (Normal Traffic and Attack Traffic)	50
4.13	Ratio of Pair Flow (Normal + Attack)	51
4.14	Accuracy of Datasets	52
4.15	Precision, Recall, F-Score of KDD dataset	52
4.16	Precision, Recall, F-Score of KDD'99 dataset	53
4.17	Accuracy of KDD with Different Features	53
4.18	Accuracy of KDD'99 with Different Features	54

---

4.19 With Mitigation and without Mitigation . . . . .	55
---	----

# List of Tables

2.1	Comparative Analysis of Surveyed Machine Learning Techniques to Detect DDoS Attack . . . . .	20
2.2	Comparative Analysis of Surveyed Machine Learning Techniques to Detect and Mitigate DDoS Attack . . . . .	25
3.1	Selected Attributes of Data Sets . . . . .	31
3.2	Features of KDD Dataset . . . . .	34
3.3	Features of KDD'99 Dataset[2] . . . . .	35
4.1	Flow Table Information . . . . .	40
4.2	Comparative Analysis of Datasets on Classifier . . . . .	51

# Chapter 1

## Introduction

### 1.1 Software Defined Network

With the increase in number of network devices leads to exponential increase in number of users. With this increased network the communication channel also increases, which form many new networks. Computer networks is a complex architectures as it comprises of many switches, router etc.

This rapid increase in modern technologies also needed new infrastructure that can manage problems like Attacks detection and prevention.

SDN controller separates software and hardware resources. SDN architecture uses control plane which helps the network to identify where the data traffic coming from the software is forwarded towards the data plane whose task is to forward it to the hardware.

In the traditional network that are used earlier the switches in the network have separate data and control , which are not interconnected and they works separately. The major role of the control plane which consists of number of switches is to manage a forwarding table (each switch has it own forwarding table) whose major task is to forward the incoming data packet towards its path by using data plane. Software-defined networking (SDN) manage all tasks of data and control plane by itself. So, the major role of control plane is integrated in SDN controller, which

is the centralized controller. Hence, without managing each individual switch the data traffic is centralized via SDN controller without involvement of the humans.

The data plane is the one which is used as earlier. when any switch of the network receives a data packet, it forward the data packets to the path according to the defined flow table entry and these entries are pre assigned and done by controller. Flow table has many entries some of important of them are port number and packet header and instruction. When a packet is received by the switch it is matched against the flow table. Packet is then forwarded according to the path defined in the flow table. Each data packet is forwarded by using one or more ports, the action taken by flow table are drop the packet or add new header to the packet.

If new packets are received by the switch and did not have any flow table entry, the switch then request the controller to make a new entry in the flow table. Then switch then take action according to the new defined entry made by the controller .

Software Defined Network (SDN) is a new centralized platform which takes attention of many users. Because of its enormous advantages Google has redesigned its data centers into SDN networks globally [3].

The SDN architecture is a centralized network. The SDN architecture is combination of data plane and control plane same like traditional architecture.

The data plane is comprises of switches and routers. It interacts with the controller via southbound interface. The control plane is the brain of the Software Defined architecture, which controls the overall transmission of the packet. SDN allows hardware abstraction layer which helps to interact hardware layer to the operating system of the network. It interacts with the application interface via northbound interface.

The application interface provides services to the end user which it receives from the controller. The application interface tasks are intrusion detection system, load balancing , and firewall. The Infrastructure layer helps the operating system to receives and response the new requests, it is comprises of hardware/switches which helps to received the data packets and response the network by identifying the path of the flow.

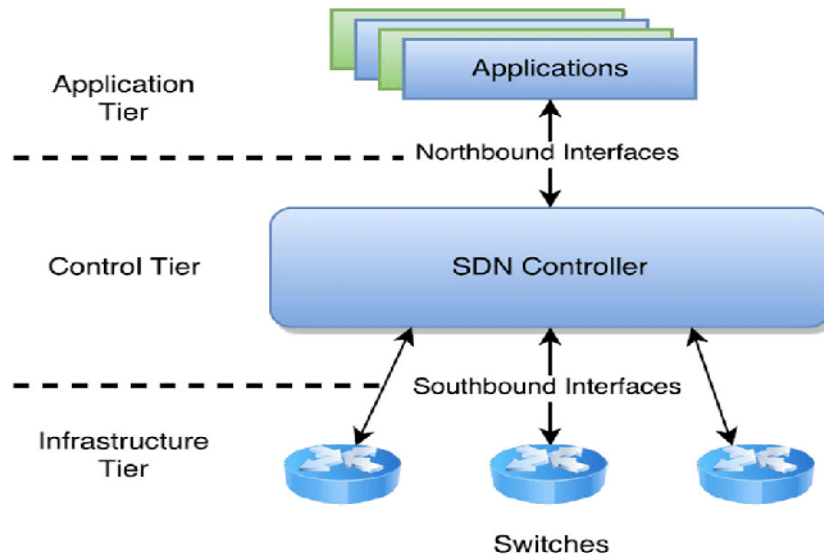


FIGURE 1.1: SDN Architecture Overview

The two layers of the SDN controller infrastructure layer and the control layer communicates with each other by using API's. The southbound API's is used to make communication among infrastructure layer and control layer. On the other hand, the northbound API helps to make communication among control layer and application layer.

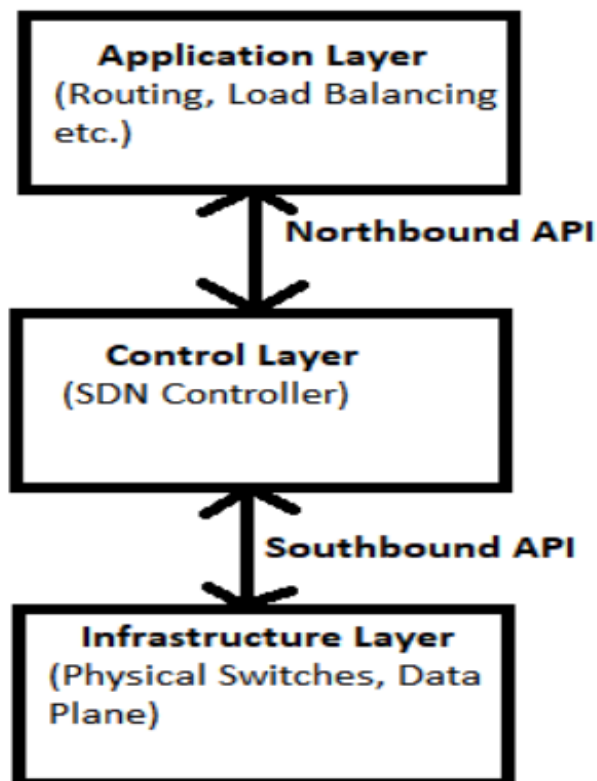


FIGURE 1.2: SDN Architecture



The southbound interface is used in infrastructure layer which helps to hardware and the software layer. The northbound interface is used in control layer which helps to communicate among the control layer and infrastructure layer the among the control layer and application layer the software and the software layer.

## 1.2 Distributed Denial-of-Service (DDoS)

Traditional networks are not flexible enough to deal with modern requirements e.g. Denial of Services Attack (DoS) attack on a single machine can crash whole network and make resources unavailable to the user. Traditional networks comprises of two networking planes, data plane: forwards the packet and control plane; compute the routes. Distributed Denial of Service (DDoS) attack are the most difficult types of attack to identify, it uses multiple systems which flood the target machine. DDoS leads to significant loss [3].

DDoS attack is a type of attack that can be started online whose major goal is to make the resources of the network unavailable for the users whenever it happened. A large number of data packets or maximum flow arrival rate is initiated in order to imitate a DDoS attack.

DDoS Attack is different from the DOS attack. A DOS attack is initiated by a single user from a single machine, whereas a DDoS attack is imitated by many bot in the network that can attack from different locations by using many kind of resources.

However the major goal of both types of attacks remains the same which is to exhaust the resources of the server and interrupt its services. Because DDoS attacks originate from many different types of bots which sends a huge data traffic towards the controller at once, which makes it difficult for controller to quickly detect and mitigate the incoming threat. As such, DDoS cause more damage to the network than the older type of DOS attack. Since one cannot identify who and which are the sender of it and from which it coming, and it completely damage the central network.

The DDoS is the most popular attack in recent era [4]. The idea is to launch the DDoS attack consumes resources of the victim. Attacker sends huge amount of data traffic towards victim, which exhausts the resources of the victim results in networks failure. With this the services of the network would not be offered to the specific time and are not accessible to the normal legal users.

Also this is most frustrating challenged faced by the normal user. These DDoS attacks are occurring on the controller daily and their attack duration increases also they are also hard to identify because packets of long duration and huge volume are coming from various sources. Many well-known users like Amazon, Azure, AWS, Mirai Krebs and OVH and GitHub have suffered DDoS Attack. The DDoS attack are increasing day by day, as the botnets are increasing is getting bigger and are hard to track as it is difficult to differentiate between normal traffic and attack traffic.

To generate and attack traffic many systems are involved. In order to generate the attack traffic many Botnets are involved. If attacker wants to initiate an attack to the target controller, it just sends order to the handlers, which by using zombie's ahead huge flow arrival rate towards the victim side. the consequences of this attack is the network resources are inaccessible to the normal user shown in figure 1.3 below.

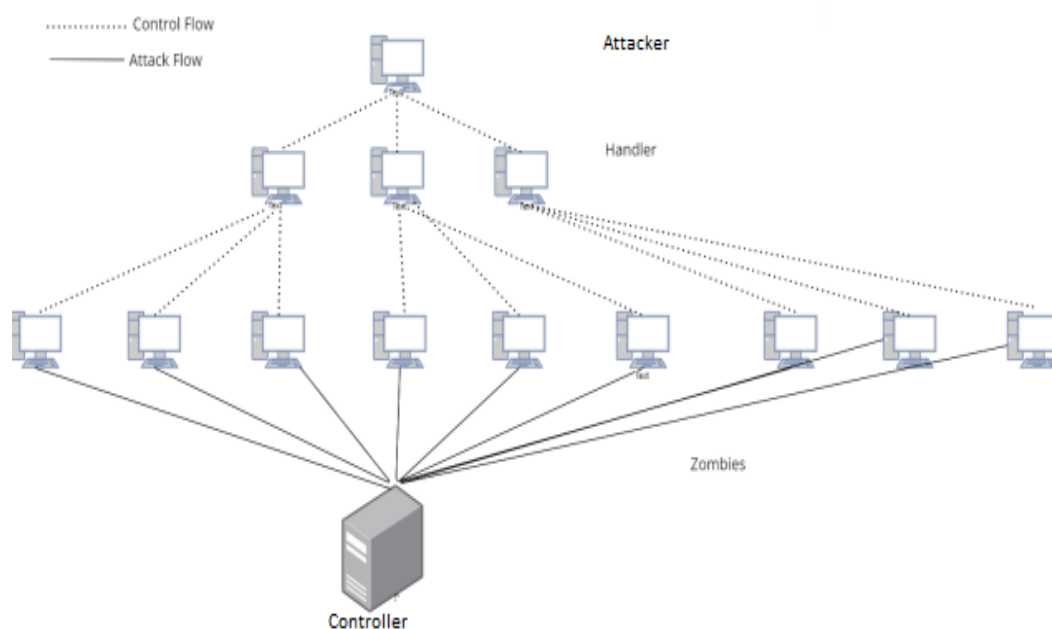


FIGURE 1.3: Attack Environment

DDoS attack normally held on network layer, causing sudden traffic jam like traffic block on highways, stopping normal traffic from arriving to its destination. Examples include:

- TCP SYN flooding
- User Datagram Protocol (UDP) flood
- Internet Control Message Protocol (ICMP) flood

A DDoS mitigation is a process that helps to reduce/ minimize the impact of denial-of-service attack on the SDN controller.

A mitigation process is done by monitoring/Analyzing the network traffic. Since DDoS attacks are initiated by using high data traffic.

The key point in mitigating a DDoS attack is to differentiate between attack and normal traffic. Since a normal user can also generate high flow arrival rate during some peak timings. on the other hand the attackers only generate the high flow arrival rate, whenever it generates an attack traffic.

Whenever the flow arrival rate exceeds a from a defined threshold, then the network must get some alert message or notification that the network can be under attack. This process helps to identify the DDoS attack as early as possible and mitigate damaged caused by this flow.



FIGURE 1.4: DDoS Mitigation

### 1.3 Machine Learning Algorithm

Machine learning derived from the Artificial intelligence; it is a set of machine learning algorithms that improve automatically at a given task by accumulating data and experience. Machine learning algorithms can also be used to identify the

solution of many complex type of problems [5]. These machine learning algorithms can also be used to identify the DDoS attacks. These algorithms are likely to be trained in order to identify the abnormal behavior of the network traffic with better accuracy.

The most important part is to select the most correlated and appropriate attributes of the dataset. As there are large amount of data in the network during a DDoS attack, so this data has to be analyzing. The goal of this analysis is to achieving improved accuracy, improved precision, improved recall and improved F-Score with reducing computational complexity.

The of machine learning algorithms depend on “training” data sets to develop the ability to solve problems for a particular task.

Machine learning algorithms are distinguished into four categories: supervised, unsupervised, semi-supervised and reinforcement learning [6].

Supervised machine learning is used to train datasets and provide the relationship between input and output; in other words they help to build a system model. When an input is fed to the model it generates the desired output [7]. It is a type of task where an algorithm is provided a training set of labeled input-output examples that it uses to infer a function. It is then tested and validated on further datasets before being deployed. Supervised Learning can be further divided into Classification, Regression and Forecasting.

Unsupervised machine learning algorithm aim to find the pattern by grouping the data into different sub groups and then identify the resemblance among them. So there is no output and input is fed without labels. It widely used in clustering techniques [6]. In Unsupervised Learning, large, unlabeled datasets are provided to the algorithm; due to the absence of labeling, the algorithm creates its own hidden structures abstractly to solve the task.

Semi-Supervised machine learning algorithm is combination of supervised and unsupervised machine learning algorithm. The goal is to learn the function that can accurately predict the output based on input variables [8].

Reinforcement learning is a machine learning technique that is based on rewards or punish. It is a trial-and-error method where a conditioning “reward-system” is used where, for every output generated by the algorithm, an interpreter classifies

it as favorable or unfavorable, causing the algorithm to adjust its parameters according to maximize the favorability of its outputs. Agents are used to find the best possible decision taken in a specific situation. Each decision is made by using last decision feedback and after each action agent receives a feedback that helps to determine whether choice was correct or not. It based on trial and error method [9].

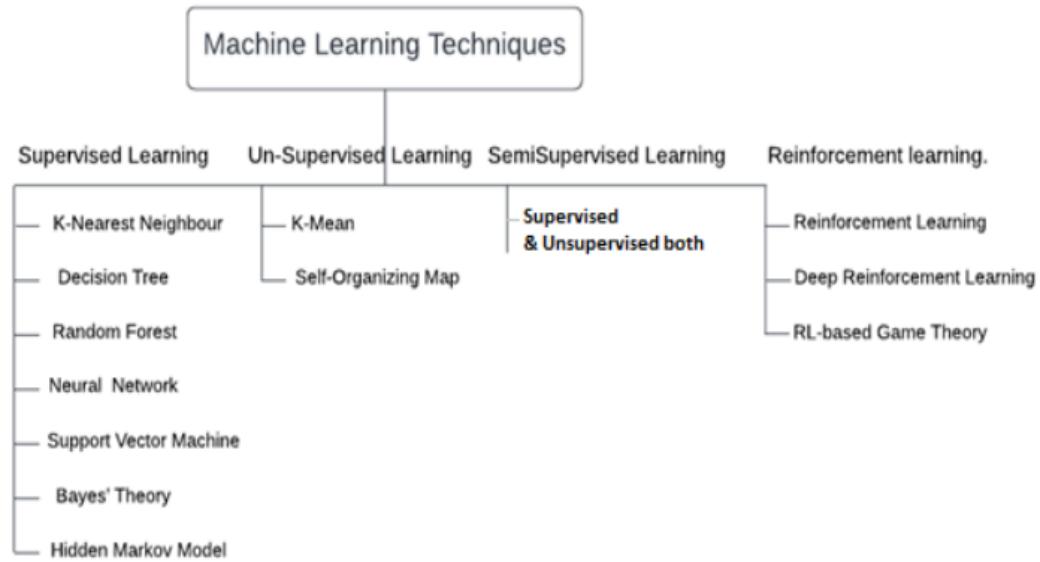


FIGURE 1.5: Machine Learning Algorithm Applied at SDN controller

Many techniques to detect DDoS attack using machine learning algorithm are proposed in [10]. The goal is to enhance accuracy and reduce computational complexity [10]. In [11] centralized SDN network improves DDoS detection and mitigation capabilities. Different machine learning algorithms are discussed in [12], [13], [14], where Random Forest (RF), Decision Tree (DT), K-nearest Neighbor, Naive Bayes (NB) are discussed.

## 1.4 Support Vector Machine (SVM)

SVM (Support Vector Machines) is a type of machine learning and is from supervised type, that are used in Regression Analysis or to classify data values by creating boundaries known as hyper planes. With a strong focus on highly complex, small datasets, SVM allows memory efficient classification of data in higher dimensional spaces however it struggles with very large, feature-rich (where the

sum of features exceeds number of data points) and probabilistic data sets. SVM machine learning algorithm is preferred used in research work because researchers reported the highest DDoS attack detection accuracy.

The main benefit of SVM is the ability to utilize “kernel tricks” which map difficult-to-classify data into a higher dimensional form that can be more easily separated by a boundary. Hence, SVM is a candidate algorithm for identifying and separating DDoS indicative traffic vs. safe traffic.

Within the context of broader Machine learning algorithms such as: Decision Trees, Naïve Bayes, K-Nearest-Neighbor, Random Forest, etc., SVM occupies the niche of being most useful for data classification where the dataset is relatively small but complex, where there exists a clear margin of separation and the data occupies higher dimensions. Here, SVM is a memory efficient means of classifying between two types of data, in this case safe vs. malicious traffic. In the context of data classification, a subset of SVM known as SVC (Support Vector Classifier) is used. The basic “rule” followed is to find the hyperplane such that the classes are separated with the largest margin possible (leaving maximum room for error).

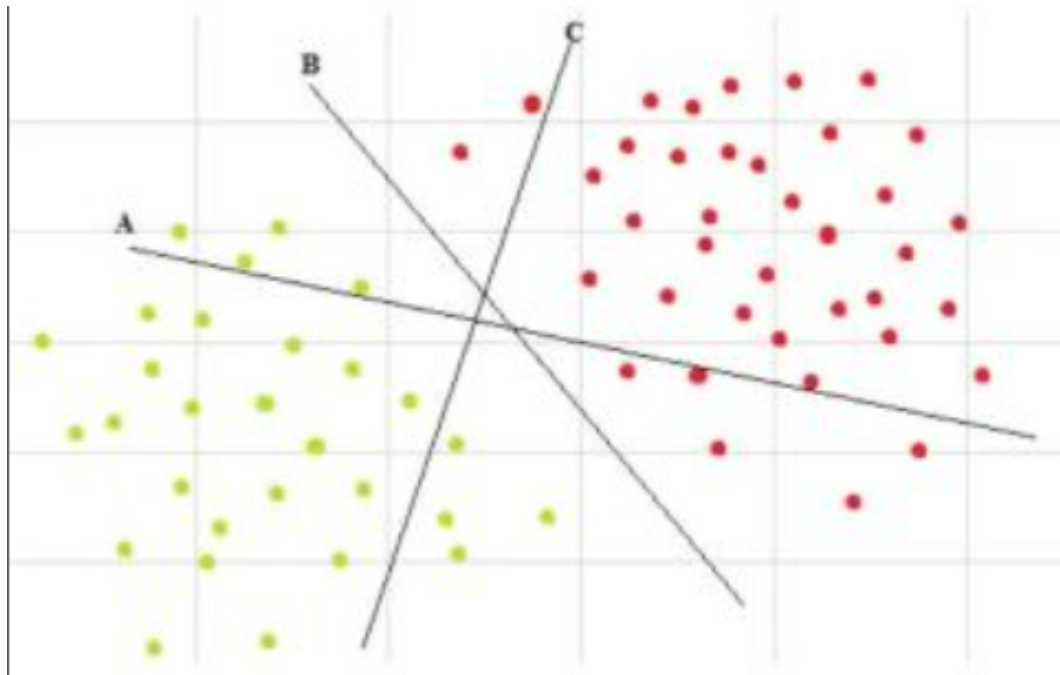


FIGURE 1.6: Ideal Hyper Plane[5]

As shown above in fig 1.5, B is the ideal hyper plane because it leaves the largest margin between datasets A and C. To do this, the algorithm will pick certain data

points from each class (known as Support Vectors) such that this largest margin hyper plane can be created.

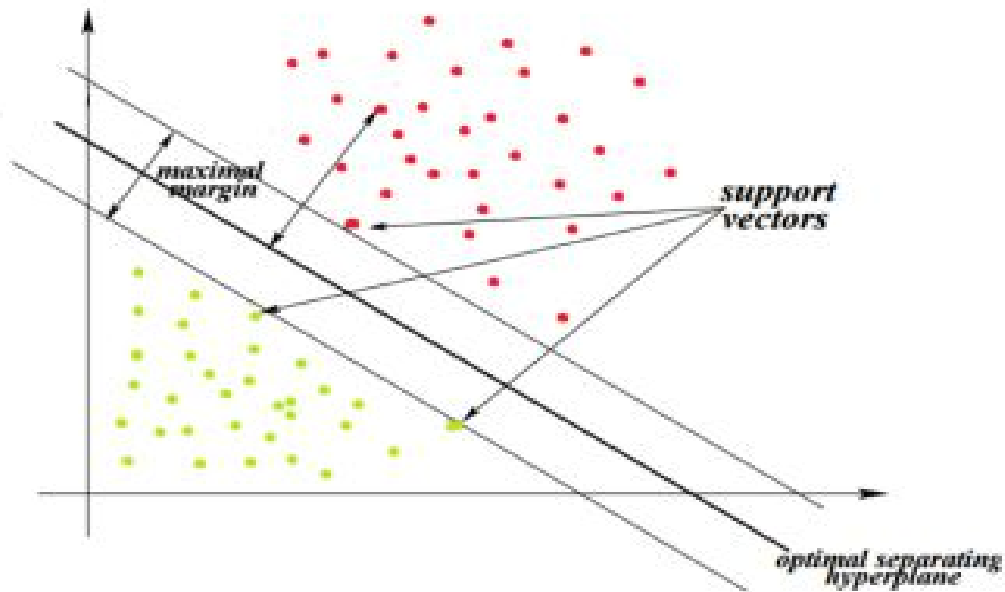


FIGURE 1.7: Data Points in Hyper Plane[5]

The hyper plane will divide the data into classes having either vectors of class +1 or -1 (i.e positive or negative data points. Mathematically, this can be represented as:

$$\text{Argmax}(w^*, b^*) \frac{2}{\|w\|} \text{ such that } y_i(\vec{w}, \vec{x}) \geq 1$$

Where  $w$  is the difference of the Support Vectors,  $x$  is the test data point projected onto  $W$  via dot product, and the term  $2/\|w\|$  represents the largest possible margin. However, in real life, datasets are not so easily classifiable hence to leave margin for error. To account for this we add the regularization parameter  $C$  to control the trade-off between mis-classifications and margin width. The parameters  $\|w\|$  and  $\text{argmax}(w^*, b^*)$  are inverted, and the parameter  $C$  is added to the resulting equation to create the following relation:

$$\text{Argmax}(w^*, b^*) \frac{\|w\|}{2} + c \sum_{i=1}^n C_i$$

As  $C$ , increases, the room for error decreases which leads to a smaller margin. The goal is to find the optimal value of  $C$  that maximizes the margin while decreasing error rate. Maximizing the margin leaves the greatest room for error which

increases the overall accuracy of the model (although this is offset if there are a large number of misclassifications inside the margin itself). One of the main challenges of employing SVM is finding the optimal value of  $C$ . Techniques such as cross-validation and Grid Search CV are used for this.

A kernel is a type of function that takes vectors in a given space as an input and returns their dot products in a different (usually higher dimensional feature space). This allows separation of data with a single hyper plane in higher dimensions where it would not be otherwise possible.

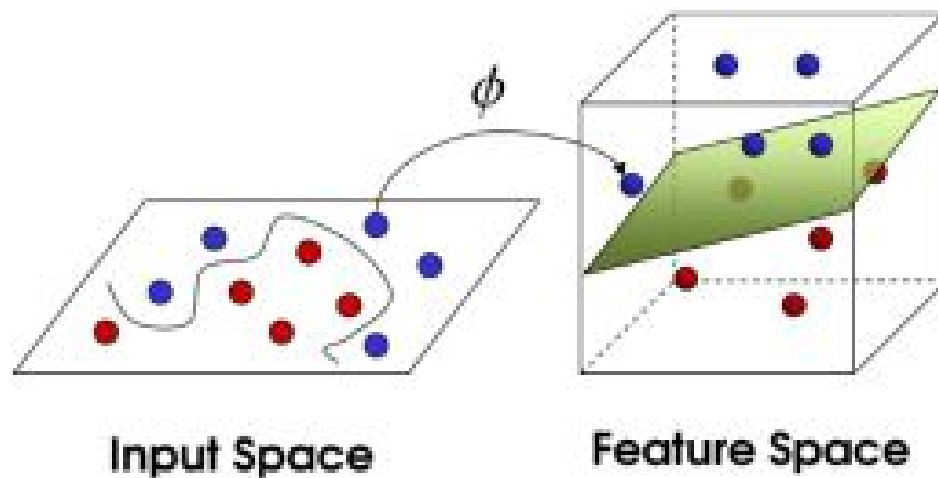


FIGURE 1.8: SVM Kernel Function[5]

As seen above, the SVM Kernel function  $\varphi$  modified input space feature space where a single hyper plane could now separate the data. SVM provides several Kernels such as Sigmoid, Bessel Function, RBF, Polynomial etc.

## 1.5 Datasets

### KDD'99:

KDD'99 is most widely used data set was created by DARPA and is the original intrusion detection dataset [15]. The dataset contains 41 features. This dataset is used to identify the attacks in the network. The attack types are categorized as: U2R, DoS, R2L and probing attack. The feature of this dataset can be continuous, discrete and symbolic. This complete dataset is used to train and test the



outcomes.

Some problem related to KDD'99 dataset are [16]:

- Include redundant data.
- Duplicate record.
- No statistics of drop packets.

### **KDD:**

KDD dataset solve some inherit problems of KDD'99 dataset[17]. This dataset also contains 41 features. This dataset removed redundant and duplicate records. This dataset is more calculate more accurate results with machine learning algorithms. Also the results are more consistent.

## **1.6 Motivation**

Several researches have been done on DDoS attack in SDN controller, which not only detect and also prevents the DDoS attack. Machine learning based techniques helps to detect DDoS attack by achieving maximum accuracy. As controller is the central part and brain of network. It is important to protect it from the attacker. Study shows that many research papers also achieve maximum accuracy but they did it only with one dataset. The goal of this research is to study which data set among two helps to achieve maximum accuracy. The other goal of this is to achieve low false positive rate with. Following point is considered in our research.

1. Identify those attributes from the two data sets KDD and KDD'99 that achieve maximum accuracy, maximum precision, maximum recall and maximum f-score with low false positive rate.

## **1.7 Problem Statement**

A DDoS attack offer high data rate to the switch and controller to exhaust their resources. It is difficult to detect a DDoS attack and takes time to mitigate that

attack. When an attacker sends flooding of packets (a packet flow) to the network it first reaches to the switch, the switch task is to check its flow table entry according to the coming data packet, since it is a new flow entry, it does not match with any existing entry in flow table. The switch after that forwards the incoming data packet to the controller.

The job of controller is to add flow table entry for this new flow. During flooding, the arrival rate of new flows is very high which ultimately exhausts resources at the controller and makes it unusable. The controller works as the backbone of the SDN network and the failure of a controller results in the failure of the whole network.

In different researches that are conducted many machine learning algorithms are proposed to accurately detect and mitigate the DDOS attack and attributes like flow arrival rate and length of a flow can be used to classify normal traffic from DOS attack. However, it is not examined in detail how different attributes can impact the accuracy of a solution.

## 1.8 Research Question

The problem statement raised questions are as below:

1. Which attributes should be used to accurately detect a DDOS attack?
2. Can we improve accuracy of DDOS detection algorithm by using different features and different datasets?
3. Which feature and dataset have maximum accuracy to detect DDOS attack using SVM?

## 1.9 Proposed Research Methodology

Research methodology proposed in our research is discussed as follows:

1. Explore new domain and new research topics.

2. Perform literature review in order to identify the limitations to the topic.
3. Analyze the behaviour of data traffic in order to know the system response.
4. Used SVM for DDoS detection.
5. Setting up simulations for the research proposal.

## 1.10 Organization of Thesis

Chapter 1 is the introduction portion, Chapter 2 is about literature review of the different techniques used to detect and mitigate DDoS attacks on SDN controller. This chapter is divided into 2 sections; DDoS detection techniques and their limitations, DDoS detection and mitigation techniques and their limitation, comparative analysis of the techniques on the basis of machine learning techniques and software defined network. In chapter 3, research methodology and performance evaluation is defined, In chapter 4, Results are discussed. In chapter 5, conclusion and future work are discussed.

# Chapter 2

## Literature Review

### 2.1 Introduction

In the context of DDoS attack, it is important to differentiate between an attack flow and a normal flow. In this chapter, different techniques which are proposed earlier in order to detect the DDoS attacks on a SDN platform. There are many studies and we have divided them into two categories.

In Section 2.1, we discuss those studies whose focus is only on DDoS attack detection on an SDN controller. However, these studies did not discuss the follow up action plan which has to be taken after the detection of DDoS attack. This chapter, different techniques which are proposed earlier in order to detect the DDoS attacks on a SDN platform. Naturally, an attacker's flow (after detection) must be blocked to protect against any possible service unavailability and this action of defense is known as mitigation.

In Section 2.2, we present those studies which focus both on DDoS attack detection and mitigation. In this regard, we have considered the research article from past 7 years, because we have identify the solution of the problem from them.

There are many research papers in which DDoS detection system is proposed and there are many surveyed papers in which the DDoS detection and mitigation techniques are proposed in detailed. Finally in Section 2.3, we identify the research gap in the existing literature and highlight the contributions of our work.

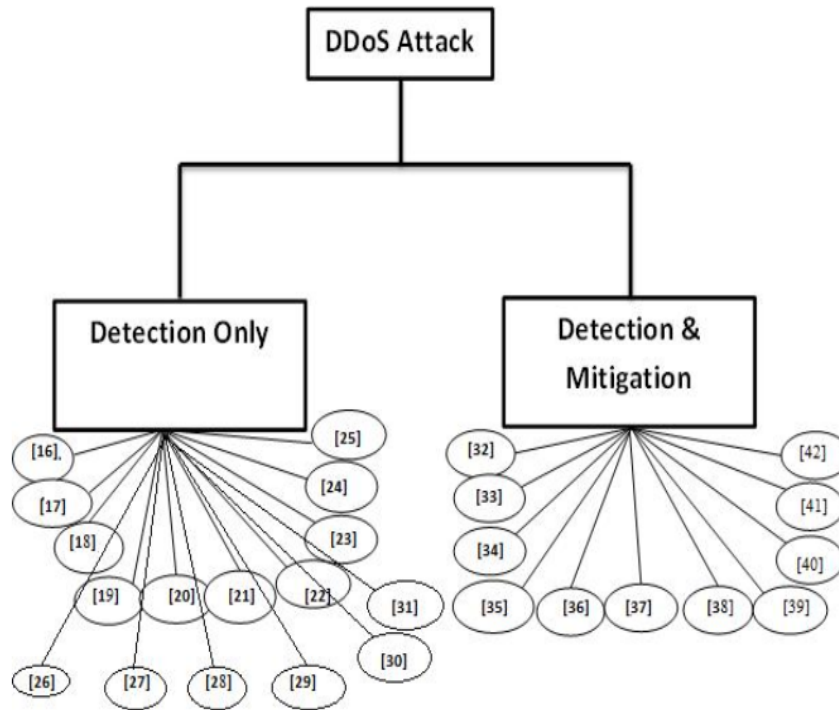


FIGURE 2.1: DDoS Attack

## 2.2 DDoS Detection Technique

In DDoS detection, the major symptom is the service of the controller suddenly becomes slow or unavailable. This can also be possible in normal traffic, so further investigation is required to classify a DDoS attack from normal traffic.

In this regard, some features/attributes of network traffic (e.g., flow arrival rate, ratio of pair flow) are selected which may help to differentiate between attack and normal traffic. It is due to the fact that characteristics of an attacker's traffic might be different than the normal traffic and this information can be used to differentiate between two flows.

Therefore, the selection of appropriate features is an important task. The traffic features are normally extracted from publicly available datasets (Such as KDD and KDD 99) and then a machine learning model is trained on these features to classify between attack traffic and normal traffic.

There are several studies which are presented in this section which use different datasets, set of features and machine learning models in order to detect DDoS attack. Once a model is trained to detect DDoS attack, it is then evaluated and the accuracy is calculated. The details of these studies are given below.

In [18], authors proposed a technique based on K-means and KNN machine learning algorithm, in order to detect the DDoS attack on controller using 15 features out of 41. The research achieved 98.85% accuracy. where as the false positive rate calculated is very low because of asymmetric characteristics to detect the DDoS attacks, also the proposed model effectively differentiated among bursty normal flow and attack flow.

In [19], authors proposed Internet2's network topology in order to detect DDoS attack on the controller. In this study KDD99 dataset is used and 11 features are selected for model training (Total out of 41 available features in the dataset). The reported DDoS detection accuracy is 85%. However, there are reported false positive as well which means that sometimes the victim is detected as an attacker. Furthermore, authors suggest two methods to detect IP address of victim and attacker. Sequential approach is preferably used if the main idea is to identify the victim controller as it can detect the IP address of the victim controller easily.

In [20], authors use Entropy based machine learning algorithm on controller, which only detects the DDoS attack without mitigation. Although authors claim the attack detection accuracy of 96% (within first 250 packets of the traffic). However, we could not find details regarding the numbers of features and long-term accuracy of the proposed solution. The study claims to detect DDoS attack on both controller and switch. For switch protection the authors proposed an idea of the statics checked mad measured on flow table. After certain period of time the controller monitors the existing flows and if there is an inactive flow it will remove the flow, since the number of the hosts occur at a time in the network can be caluclated.

In [21], authors proposed a machine learning technique that is based on k-means and K nearest machine learning algorithms in order to detect the DDoS attack. In order to train the model, 15 features are used (out of 41 KDD '99 dataset). It is demonstrated that K-nearest have higher precision and lower false rate of attack traffic; whereas the recall remains stable. In [22], author proposed Signature based SNORT machine learning technique using KDD '99 dataset, which both detect the SDN controller, achieving accuracy of 74%. Attributes of the dataset are not identified; it did not show any information regarding switch protection. In [23],

authors proposed Entropy based and SVM machine learning algorithm in order to detect the DDoS attack on SDN controller. Author calculates the threshold value of the network traffic, when it exceeds the certain threshold the alert is generated and sends to the controller. Here machine learning algorithm is used to identify the attack traffic and then attack flow is blocked. Author uses 6 features out of 41, achieving 95.24% accuracy.

In [24], author proposed Joint entropy method to detect and prevent the SDN controller using KDD '99 dataset achieving 80% of accuracy. Author uses 11 features out of 41 to detect the DDoS attack on the SDN controller. Experiments shows that the 80% success rate of attack detection for unfamiliar packet and 70% success rate for mixed traffic. Also, it can work efficiently with minimum storage and low processing requirements.

In [25], author proposed Random Forest and Decision Tree techniques in order to detect the DDoS attack on controller. Author calculates the threshold value of the network traffic, when it exceeds the certain threshold the alert is generated and sends to the controller. Here machine learning algorithm is used to identify the attack and normal traffic. When the incoming flow is identified as the attack flow it is blocked. The accuracy measured is 92.18% and selected features are not identified in the paper.

In [26], author proposed DDoS detection method on using SVM machine learning algorithm, where author proposed the idea of selection of best features from DDoS dataset. Author proposed that when DDoS attack happened on controller, the controller then updates the rule of forwarding or denying of any packets are updated, also flow table entries are updated. Author uses 20 features out of 41 by achieving 95.71% accuracy.

In [27], author proposed DDoS detection method using SVM and KNN machine learning algorithms at POX SDN controller. Author proposed an idea to detect abnormal behavior of data traffic in SDN controller. It is demonstrated that that SVM achieved higher accuracy of 98.1423%, with low false positive rate.

In [28] author proposed DDoS detection method on using SVM, KNN, DT, MLP, and CNN machine learning algorithm, where author proposed the idea characterizing DDoS assaults in an elastic technique owing to a decoupled SDN architecture.

Author uses CICIDS2017 and CICDDoS2019 datasets which includes 79 features on different machine learning model. Author shows that SVM achieved maximum accuracy during training and prediction in both datasets. SVM correctly detect the DDoS attack among all proposed machines learning algorithm. Results show that CICDDoS2019 performs better to achieve maximum accuracy as compared to CICIDS2017.

In [29] author proposed DDoS detection method on IPV6 enabled SDN using SVM machine learning algorithm. Author proposed an idea where normal and attack traffic were generated 500,000 packets for 20 min in order to test the proposed model. The proposed model generates a pattern map in order to distinguish between normal and attack traffic. Results show that SVM achieved an accuracy of 99.69% and the DDoS attacks detection rate on SDN controller is of 100%.

In [30] author proposed intrusion detection system to find out the DDoS detection method on SDN controller using SVM machine learning algorithm on vehicular network. SDN network receives the huge amount of DDoS attack traffic, which then pass through the SVM in order to differentiate the normal and attack traffic. If the attack traffic is detected then it shows the notification of DDoS attack traffic and results are passed through to control unit. Results show that the proposed model achieved 99.33% accuracy and 99.22% attack detection rate.

In [31] author proposed RBF-SVM based DDoS attack detection method on SDN controller of vehicular space. Author proposed an attack detection using 'poly' kernel that have best accuracy and low detection rate. Results show that the poly kernel SVM model is best as compared to linear kernel SVM model. The accuracy achieved with this model is 99.4% and attack detection rate is 99.22%.

In [32] author proposed Extreme Gradient Boosting (XGBoost), SVM, Random Forest (RF) and Decision Tree (DT) machine learning based DDoS attack detection method on SDN controller. Author proposed a model, when the controller detects the DDoS attack it informs the switch and the port number to block that port where heavy traffic is generated. The accuracy achieved with this model is 99.94% using SVM.



TABLE 2.1: Comparative Analysis of Surveyed Machine Learning Techniques to Detect DDoS Attack

S.No.	Ref.	Year	Dataset	NOA	MLM	Accuracy	FP	FN	CP	SP
1	[18]	2020	KDD	15	K-Mean,KNN	98.85%	✓	-	✓	✓
2	[19]	2016	KDD99	11	Internet2's Network Topology	85%	✓	✓	✓	-
3	[20]	2015	KDD	-	Entropy	96%	-	-	✓	✓
4	[21]	2019	KDD99	-	K-Means++,K-nearest	99.01%	✓	✓	✓	✓
5	[22]	2017	KDD99	6	Entropy,Signature SNORT	-	✓	✓	✓	-
6	[23]	2017	KDD	6	Entropy based, SVM	95.24%	✓	✓	✓	-
7	[24]	2018	KDD99	11	Joint Entropy	80%	✓	✓	✓	-
8	[25]	2021	KDD	-	RF,DT	92.18%	-	-	✓	-

S.No.	Ref.	Year	Dataset	NOA	MLM	Accuracy	FP	FN	CP	SP
9	[26]	2017	KDD	11	SVM	95.71	✓	✓	✓	-
10	[27]	2019	KDD	11	SVM,KNN	98.1423% 81.4123%	✓	✓	✓	-
11	[28]	2023	CICIDS2017	50	SVM, KNN, DT, MLP, CNN	97.808%	✓	✓	✓	-
			CICDDoS2019						-	
12	[29]	2022	DARPA 2000	-	SVM	99.69%	✓	✓	✓	-
13	[30]	2022	CICDDoS2019 SDN-DDoS	21	SVM	99.33%	✓	✓	✓	-
14	[31]	2022	SDN DDoS attack	-	RBF-SVM	99.4%	✓	✓	✓	-
15	[32]	2023	CICDDoS2017 dataset	16	XGBoost,RF, SVM, DT	99.94%	✓	✓	✓	-
16	[33]	2023	CICDDoS2017	19	RF, J48, NB, SVM	98.86%	✓	✓	✓	-

In [33] author proposed signature based and machine learning algorithm to detect the DDoS attack. The proposed machine learning algorithm are Random Forest (RF), J48, Naive Bayes (NB), and Support Vector Machine (SVM) machine learning algorithm on SDN controller. Author proposed that developing and implementing intrusion detection systems (IDS) in SDN is necessary as SDN is single point of failure and IDS effectively detect the DDoS attack using signature based and machine learning algorithm. The accuracy achieved with this model is 98.86% prediction accuracy and a train time of 1.46s.

The summary of DDoS detections techniques is provided in Table 2.1:

Here in table:

NOA refers No. of Attributes

MLM refers Machine Learning Model

CP refers Controller Protection

SP refers Switch Protection.

## 2.3 DDoS Detection and Mitigation Technique

As discussed earlier a DDoS attack is a targeted attack where attacker sends packets flooding from more than one users called bots. If appropriate preventative methods are not taken, then it makes the whole network to crash and unresponsive. A DDoS mitigation is a process that helps to reduce/ minimize the impact of DOS attack on the SDN controller.

DDoS prevention techniques are used not only detects but also mitigates the DDoS attacks. The first phases of each technique are detection of an attack; the second phase is to differentiate the normal and attack traffic and block the flow if it is a high flow arrival rate. There are many different DDoS prevention techniques which are discussed in this section.

In [34], authors proposed an idea of DDoS attack detection on controller based on SVM machine learning algorithm. Attack traffic type is TCP/UDP. Authors show if the false accuracy achieved in DDoS detection is 99.27% and accuracy in mitigation process is 99.3% having false positive rate 0.67.

In [35], author proposed a technique which detects and also mitigation the SDN controller by using DosDefender in floodlight controller using IP and MAC spoofing technique. However, the limitation is that the proposed solution cannot mitigate the attack that comes from the multiple sources. The performance metrics are accuracy, CPU and memory utilization. Author uses 11 features out of 41. Results shows that the normal traffic has low CPU utilization and attack traffic cannot detect easily which come from multiple resources and CPU utilization increases.

In [36] author proposed a Machine learning algorithm the context of SDN controller is used to take decision automatically whenever attack happened and also mitigate the controller. Proposed algorithm effectively detects and mitigates the DDoS attack and also improves the DDoS attack detection rate and reduces the false positive rate. Author uses 15 features out of 41. The accuracy achieved in this study is 87%.

In [37], author uses KDD datasets by using SVM and decision tree machines learning algorithm, which detects and mitigates the DDoS attacks on SDN controller. Experimental results shows that whenever switch detect any abnormal traffic, it send an alert message to the controller. When flow is detected as an attack traffic controller then drop the articular flow and update flow table entries. The performances metrics of proposed model are accuracy, F-Score, precision and recall. Author uses KDD99 data set. Experimental results shows that SVM have better accuracy rate then decision tree which is 85%. The precision, recall and f-score value of SVM is better.

In [38], author proposed advanced SVM technique with TCP, SYN and UDP flooding technique to detect and mitigate the DDoS attack on SDN controller. The features include the fast training and testing time, the performance evaluation includes accuracy. The accuracy achieved is 98.18%. The problem authors face is it cannot detect the low volume DDoS attack.

In [39], author uses dual entropy and SVM machines learning algorithm using UDP flooding. The proposed algorithm detects and mitigates the attack. Author uses 12 out of 41 features. The performances metrics are false positive rate. The goal is to restores the normal communication after the attack is detected and malicious host is identified. The weakness of this research is it only detects the high

data rate attack traffic. Whereas the low data rate DDoS attack traffic cannot be detected. Accuracy is not calculated in this study. In [40], author proposed idea of DDoS attack detection and mitigation on SDN controller by using KNN machine learning algorithm. The performance metrics that author used are accuracy, precision, recall and F-Score. The weakness includes that the computational cost for communication increases on controller side. The experiment shows 97% of flows are dropped. For DDoS mitigation the new dataset is used to improve accuracy of the system. In [11], author proposed a system using SVM that provides a powerful method for tracking internet traffic and can also easily prevents the DDoS attacks from the malicious users. Authors give the idea of queuing theory which is build a flow table. This strategy uses the unused flow table of other OpenFlow switches in the network to shield the switch table from overload. Features are not identified whereas the false positive rate is also not measured. In [41], author uses SVM, K- nearest and Naive based machine learning algorithm. They use two different datasets to calculate the accuracy of the algorithm whereas features are not clearly identified, when DDoS attack is detected on the SDN controller by using all the above three methods. The results show that the KNN achieved more accuracy than other proposed algorithms i.e. 99.18%.

In [42] author proposed a detection and mitigation on SDN controller. Initially the signature based SNORT detects the DDoS attack on the controller. Then classifiers uses SVM and deep neural networks machine learning algorithms are used. Result shows that the accuracy achieved from SVM is 74.3%, however, the deep neural network achieved more accuracy which is 92.3%. In [32], author proposed an idea of DDoS attack Detection and mitigation on SDN controller. The proposed machined learning algorithms are SVM and Shannon entropy machine learning method. The key parameters include: Source IP, Destination IP and Destination Port Protocol. The approaches detect and mitigate the DDoS attack using Classifiers. The entropy method only task is to identify the changes that happened on the SDN controller when a DDoS attack occur. whenever it happened. The DDoS Detection Module has to perform three tasks; information collection, feature extraction, and attack detection. The maximum accuracy achieved is from SVM which is 98.75% and author uses 15 out of 41 features.

TABLE 2.2: Comparative Analysis of Surveyed Machine Learning Techniques to Detect and Mitigate DDoS Attack

S.No.	Ref.	Year	Dataset	NOA	MLM	Accuracy	FP	FN	CP	SP
1	[34]	2019	KDD	-	SVM	99.27%	✓	✓	✓	-
2	[35]	2019	KDD	11	DosDefender in floodlight controller	-	✓	✓	✓	-
3	[36]	2019	KDD	16	SVM	7%	✓	✓	✓	-
4	[37]	2021	KDD'99	-	SVM and Decision Tree	85%	✓	✓	✓	-
5	[38]	2019	KDD	11	SVM	98.18	-	-	✓	-
6	[39]	2016	KDD	12	Dual Entropy SVM	-	✓	✓	✓	-
7	[40]	2019	KDD99	-	KNN	97%	-	-	✓	-
8	[11]	2019	KDD'99	-	SVM	-	-	-	✓	-
9	[41]	2020	KDD	15	SVM, K-Nearest, Naive Based	99.18	✓	✓	✓	-
10	[42]	2018	KDD	12	SVM, K-Nearest, Naive Based	92.3%	✓	✓	✓	-
11	[32]	2023	KDD	15	SVM, Shannon Entropy	98.75%	✓	✓	✓	-
12	[43]	2022	Intrusion Detection DARPAR	19	SVM	98.76%	✓	✓	✓	-

In [43] author proposed a hybrid approach to detect DDoS attack detection and mitigation method on SDN controller using SSAE-SVM machine learning algorithm for suspected network traffic. Author shows that the normal traffic entry is made in flow table. Is abnormal traffic is detected. The proposed system consists of two model, detection and defense. The DDoS attack detection is works as: when attack is detected then controller-to-switch message are initiated by controller regarding data entry in flow table and asynchronous message are initiated from switch regarding network traffic and make entry in flow table. When an attack is detected then defense filtering module identifies the port which receives the attack traffic and blocks it, therefore protect the SDN controller. The accuracy achieved from proposed algorithm is 98.73% with detection time 67.57 seconds. The summary of DDoS detections and mitigation techniques is provided in Table 2.2:

Here in table:

NOA refers No. of Attributes

MLM refers Machine Learning Model

CP refers Controller Protection

SP refers Switch Protection.

## 2.4 Research Gap Identification and List of Contribution

Different approaches discussed in literature to detect and mitigate DDoS attack in SDN controller. In these techniques various datasets used to perform experiments. Similarly, various detection and prevention technique have been suggested which produce effective results against DDoS attack.

Some limitations also have been discussed in these techniques. Tables 1 and 2 are created on the basis of the techniques they used. In table 1 ten parameters are discussed: those are s.no, ref, No. of attributes, Machine Learning Model, Accuracy, False Positive, False Negative, Controller Protection and Switch Protection. In table 2 ten parameters are discussed: those are s.no, ref, Location, Method for DDoS Detection, DDoS Detection, DDoS Mitigation, Controller Protection,

Switch Protection, Accuracy % and Dataset Attributes. SVM is the most popular machine learning algorithm. It helps to solve the real time problem.

As Support vector machine is used for both classifications as well as regressions [1]. It helps to gather accurate results with high performance. Nearly all existing reach search papers uses SVM with only one data set and with defined attributes where they measure the accuracy but no research paper have used more than one dataset in their research with SVM in order to check whether there is any change in accuracy with varying set of attributes. Nearly all authors proposed solution of DDoS attack on controller or switch and protect both, with some performance metrics using machine learning algorithm. Different authors use different attributes for their machine learning algorithm in order achieve maximum accuracy. But none of them explains how these attributes can help to accurately detect the DDoS attack? Also if same attributes are used on different datasets which algorithm achieves maximum accuracy?

#### **2.4.1 DDoS Detection Trigger Mechanism**

This mechanism is implemented on data plane which helps to count the in coming packets from the switch using packet\_in messages. During DDoS attack the flow arrival increases drastically. This shows that their is some suspicious flow and controller have to start mitigation process in order to identify the DDoS attack.

#### **2.4.2 Flow Extraction**

This checks the normal or abnormal flow. When any switches detect the abnormal flows passing through it, switch sends a message to the controller. The controller task is to check the flow information of the incoming flow and detect whether it is normal traffic or abnormal traffic.

#### **2.4.3 Feature Extraction and DDoS Detection Algorithm**

For the proposed solution SVM based traffic detection module is used, which helps to improve the DDoS attack detection and checks that whether it an attack traffic or a normal heavy flow.



#### **2.4.4 DDoS Defense Mechanism**

When controller detects the DDoS attack, it needs to mitigate it as soon as possible in order to reduce the impact of DDoS attack and ensuring normal operation. Controller sends a message to the switch to remove the address from it flow table entry.

# Chapter 3

## Research Methodology and Performance Evaluation

### 3.1 Introduction

SDN architecture is a single point of failure architecture as the entire network is centralized. Controller handles it all. A DDoS is an attack that sends high data rate packets to the switch and controller to exhaust their resources.

It is difficult to detect a DDoS attack as many bots send huge number of packets to the network from different sources and spread more quickly. DDoS detection and mitigation of DDoS attack is much needed in SDN architecture. When DDoS attack happened a huge data flow are forwarded to the controller, as attacker main target is to target the controller. If switch does not found the entry of the incoming flow, it forwards the flow to the SDN controller.

SDN main task is to identify the path for this flow. During flooding, the arrival rate of new flows is very high which ultimately exhausts resources at the controller and make it unusable. The controller is the backbone SDN network and the crash of a controller results in the failure of the whole network.

In literature, different machine learning algorithms are proposed to detect and mitigate the DDOS attack and many different attributes can be used to classify normal traffic from DOS attack. However, it is not examined in detail how different attributes can impact the accuracy of a solution.

In this chapter the process of traffic attributes selection and machine learning based threat detection technique is discussed. This technique helps to prevent the SDN controller from the DDoS attack which helps to save resources.

Proposed technique differentiates between normal users and the attackers and prevents the controller from the DDoS attack. Our proposed machine learning technique works as follows: Firstly, the features of the base paper for datasets KDD and KDD'99 are used. This technique helps to choose number of features. Lastly Support Vector Machine (SVM) classifier is used in our research work which helps to classify the normal traffic and attack traffic.

## 3.2 Research Methodology

The steps to evaluate the proposed solution are: Firstly, we deploy a network topology in a virtual environment using Mininet. Next step is to deploy the DDoS algorithm which generates heavy traffic.

Attributes are derived from the incoming packet. It is useful that the features set must be reduced as the number of features to minimize the complexity of algorithm while maintaining maximum accuracy. Attribute selection method is used to reduce the number of features of the dataset.

It is a two-step operation firstly subsets are generated and then ranking is performed. Subset helps to find the good subset till better outcome achieved and process continues till find the best subset and termination condition stops.

After capturing heavy traffic, next step is to apply SVM algorithm on the intrusion detection evaluation dataset and chosen attribute.

Last step is to measure the accuracy from the proposed approach.

The proposed system receives incoming traffic from different users it first has to differentiate whether it is a normal flow or the attack flow.

DDoS monitoring system monitors all this traffic and calculates threshold of incoming packets. If the flow arrival rate exceeds the threshold, then switch and sends a signal to the controller.

Switch receives alert message from the controller and discard each suspected entry from flow table. Controller starts mitigating the network.

Initially, the dataset is exported to Weka tool. After that the data 80% of the data is features selection and 20% is used for testing purpose.

The number of features are reduced 41 to 6 for KDD and from 42 to 6 only for KDD'99 dataset. These features are for KDD and KDD'99 data sets are given in Table 3.1:

TABLE 3.1: Selected Attributes of Data Sets

S.No.	Dataset	Attributes
1	KDD	srcIP, desIP, switch, pktcount, bytecount , pairflow,pktrate, protocol, duration
2	KDD'99	sourcebyte, destbyte, land, wrongfragment , diffsrvrate, dsthostdsrportrate ,desthostsrverrorrate, Duration, protocoltype.

### 3.3 Proposed System Architecture for DDoS Detection and Mitigation

The network topology of SDN environment shown in figure 3.1 consists of controller, switches and PC's are deployed in Mininet in operating system Ubuntu. PC1 is attacker while PC2 is the normal user. PC1 and PC2 generate traffics.

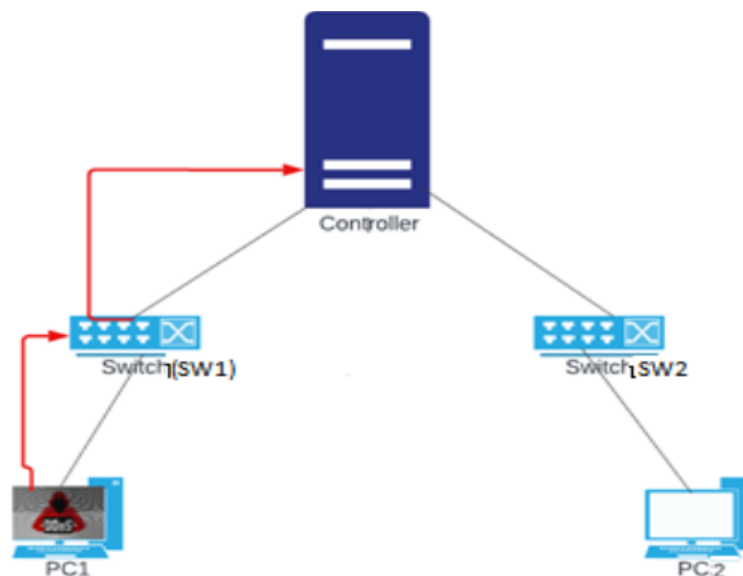


FIGURE 3.1: Network Topology

Figure 3.1 shows the network topology of the simulation. PC2 is the normal user, which generate traffic of fluctuation flow arrival rate. PC1 is the attacker whose task is to exhaust the resource of the controller. It generates the traffic with consistently high flow rate. All traffic of normal user and attacker traffic firstly received by OpenFlow switch from the input port, as switch are responsible for forwarding the packet [44].

Switch on the other hand checks the flow table entries for the incoming packet. Theses entries include: switch input port, VLAN ID, VLAN priority, Ethernet source address, Ethernet destination address, Ethernet frame type, IP source address, IP destination address, IP protocol, IP Type of Service (ToS) bits, TCP/UDP source port, TCP/UDP destination port. If a matching flow entry is found, then packet is forwarded predefined path or necessary action is taken against it. If the flow table did not match against any entry also called a table-miss, then packet is forwarded to the SDN controller.

If the flow arrival rate increases on the switch side, the switch then checks the threshold of the incoming flow.

In proposed solution the threshold value is 100K Bytes and above. This is because in proposed solution it is observed that if flow arrival rate increases the controller performance decreases. However threshold might be changed if this is applied on the scenarios other than the this setup. In our setup, we have manually observed that controller response slows down after 100 K Bytes and therefore selected this value as a threshold.

Threshold helps to figure out whether the network is coping with the heavy traffic and classifier have to take necessary measures in order to find out if the network is under attack or not. If controller does not check the threshold values, then its only tasks is to find out the forwarding path, which it continuously started to assign and as a result it is exhausted. So, if threshold is within the range then it takes action against the request.

If flow arrival rate on the controller cross the threshold then it send it to the classifier to check whether the incoming flow is normal or attack flow. SVM algorithm is working on classifier. Classifier task is to check whether the incoming traffic is from the normal user with maximum flow arrival rate or the attacker.

if it is not suspected as a high flow arrival rate from the normal user then the classifier sends allows this traffic and make entry in the flow table. if it suspects as an attack traffic then classifier block that user that is generating this high data traffic and informs the switch to block the traffic that is coming from this route.

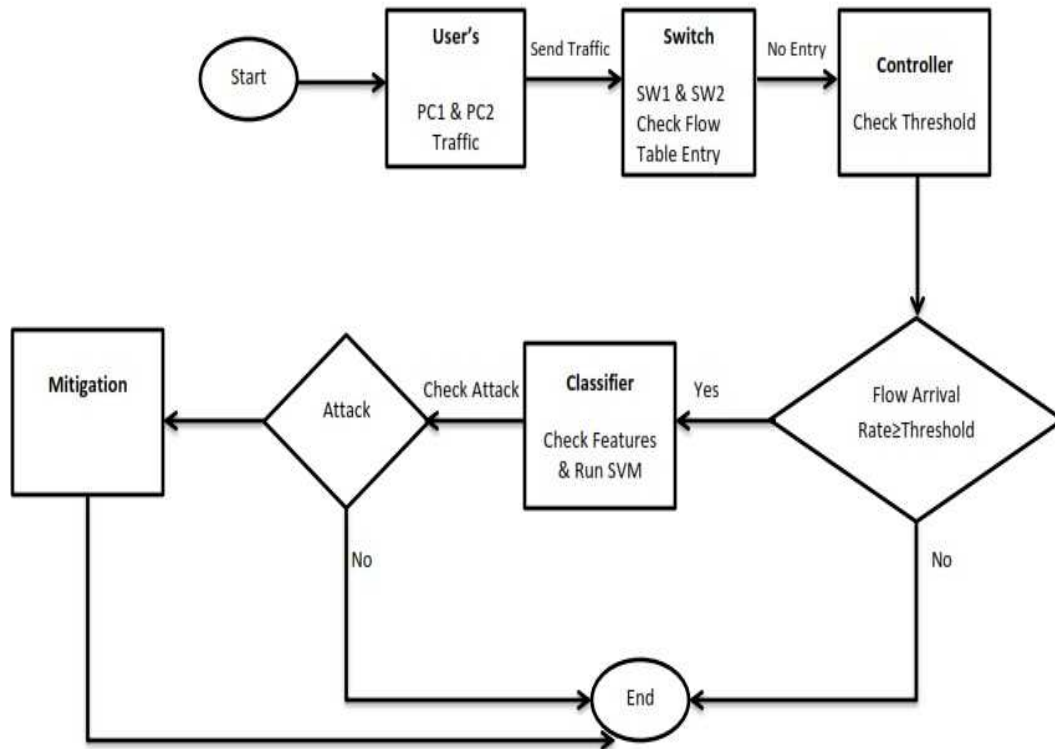


FIGURE 3.2: Attack Detection and Mitigation

Figure 3.2 shows the broader view of the Attack Detection process. When firstly the packets are generated by the user's both normal or Attacker received by the switches. Switches check the packets entries in there corresponding Flow table entries, if matches then take necessary action. If it did not match the entry then these flows are sending towards the controller.

The task perform by the controller is to checks the threshold value of the flow, if it is with in the threshold range then make the entry in the Flow Table, if it exceeds the threshold the controller send the flow entries to the classifier.

Classifier checks the features and runs the classifier. If it is normal traffic then make the entry in the flow table, if it is detected attack traffic, then mitigation process started which block the port which received the attack traffic. After that the mitigation process starts.

### 3.4 Attributes Extraction

The attributes for the proposed system are derived from the incoming traffic. Information regarding these attributes is managed in a log file. These features help to differentiate between normal and attack flow.

The KDD and KDD'99 datasets are used to determine the efficiency of the proposed machine learning technique. Collection of downloadable files is available for research purpose. KDDTest+.ARFF is used.

It consists of 42 features for KDD dataset. Features of this dataset are listed in table 3.2 [45].

TABLE 3.2: Features of KDD Dataset

S.No.	Feature Name	S.No.	Feature Name
1	Duration	22	Is_guest_login
2	Protocol type	23	Count
3	Service	24	Serror_rate
4	Src_bytes	25	Rerror_rate
5	Dst_bytes	26	Same_srv_rate
6	Flag	27	Same_srv_rate
7	Land	28	Diff_srv_rate
8	Wrong_fragment	29	Srv_count
9	Urgent	30	Srv_serror_rate
10	Hot	31	Srv_rerror_rate
11	Num_failed_login	32	Srv_diff_host_rate
12	Logged_in	33	Dst_host_count
13	Num_compromised	34	Dst_host_srv_count
14	Root_shell	35	Dst_host_same_srv_rate
15	Su_attempted	36	Dst_host_diff_srv_rate
16	Num_root	37	Dsthostsamesportrate
17	Num_file_creations	38	Dsthostsrvidiffhostrate
18	Num_shells	39	Dst_host_serror_rate
19	Num_access_files	40	Dst_host_rerror_rate
20	Num_outbound_cmds	41	Dsthostdiffsrvrerrorrate
21	Is_hot_login	42	Class

For KDD '99 it consists of 41 features. Features of this dataset are listed in table 3.3 [2].

TABLE 3.3: Features of KDD'99 Dataset[2]

S.No.	Feature Name	S.No.	Feature Name
1	Duration	22	Is_guest_login
2	Protocol type	23	Count
3	Service	24	Serror_rate
4	Src_bytes	25	Rerror_rate
5	Dst_bytes	26	Same_srv_rate
6	Flag	27	Diff_srv_rate
7	Land	28	srv_count
8	Wrong_fragment	29	Srv_error_rate
9	Urgent	30	Srv_rerror_rate
10	Hot	31	Srv_diff_host_rate
11	Num_failed_login	32	Dst_host_count
12	Logged_in	33	Dst_host_srv_count
13	Num_compromised	34	Dsthostsamesrvrate
14	Root_shell	35	Dst_host_diff_srv_rate
15	Su_attempted	36	sthostsamesrportrate
16	Num_root	37	Dsthostsrvidffhostrate
17	Num_file_creations	38	Dsthostserrorate
18	Num_shells	39	Dst_host_rerror_rate
19	Num_access_files	40	Dsthostdiffsrvrerrorate
20	Num_outbound_cmds	41	Dst_host_Diff_srv_rate
21	Is_hot_login		

Since the features present in the dataset have the different values. There next step is to transform the data into some standard scale so that machine learning techniques can be applied. The data is then divided into training and testing datasets.

### 3.5 Selected Attributed for Classifier of Attack Traffic

Feature Selection is the most useful method to identify those features that are useful and takes less time for calculation. It is the most traditional process in order to get improved accuracy.



When DDoS attack occurs on the controller by the bot, the bot will generate random and huge amount of IP addresses and large flow rate with fixed packet size to the target, also large number of new source port addresses was randomly generated when DDoS attack occurs. Controller checks the threshold of the incoming traffic, if it exceeded, then these flows are send to the classifier, which runs the SVM algorithm by checking the features. In this thesis, the following six features related to DDoS attacks are used for DDoS attack detection. The Selected attributes from the base paper [18] are:

1. The Speed of Source IP (SSIP)[18] number of source IP addresses per unit of time:

$$SSIP = \frac{Sum\_IPsrc}{T} \quad (3.1)$$

Where Sum\_IPsrc is source IP and T is the time interval to take sample. As large number of data packets are generated, so the number of IP will also be increases.

2. The Speed of Source Port (SSP) [18] is the number of source ports per unit of time:

$$SSP = \frac{Sum\_Portsrc}{T} \quad (3.2)$$

Sum\_Portsrc are the total number of attack from any source ports. As during DDoS attack large numbers of ports are randomly generated.

3. Standard Deviation of Flow Rate (SDFR)[18],is the total number of packets in the in T period define as follows:

$$SDFR = \sqrt{\frac{1}{N} \sum_{i=1}^N (packets\_i - Mean\_packets)^2} \quad (3.3)$$

Where mean packets represents the average number of bits in T period of time. This will be used in order to differentiate between normal and attack traffic.

4. The Deviation of Flow Bytes (SDFB)[18], is the number of bits in T period of time defined as:

$$SDFR = \sqrt{\frac{1}{N}} \sum_{i=1}^N (bytes_i - Mean\_bytes)^2 \quad (3.4)$$

Where mean byte are the average number of bytes in T period of time. As the packet size is smaller, so the standard deviation is smaller than normal flow.

5. The speed of flow entries (SFE)[18] is the total number of flow entries on switch per unit time.

$$SFE = \frac{N}{T} \quad (3.5)$$

Where N is number of flow entries per unit time. As during DDoS attack the flow entries are much higher than normal traffic.

6. The Ratio of Pair-Flow (RPF)[18] is the ratio of interactive flow entries to flow entries defined as:

$$RPF = \frac{2 * Pair\_sum}{N} \quad (3.6)$$

Where Pair Sum is total number of interactive flow entries per unit time Classifier classifies the data packets and detects whether the data packets are normal or DDoS attack.

All of the above defined features are used to detect the DDoS attack on the controller. These

### 3.6 Summary

In this chapter, proposed machine learning based classification of DDoS was discussed. KDD and KDD'99 datasets are taken as input. This data is imported in Weka tool. There are 42 attributes for KDD and 41 attributes for KDD'99 present in the dataset. 80% data is used for training and 20% data is used for testing purpose. Traffic is generated including normal and attack contains TCP, UDP and ICMP data packets.

# Chapter 4

## Results and Discussions

### 4.1 Results and Discussions

In this chapter, experiment results of our proposed technique are discussed. Different performance calculating metrics are used for evaluation. The performance metrics used are true positive, true negative, false positive false negative, precision, recall and F-score.

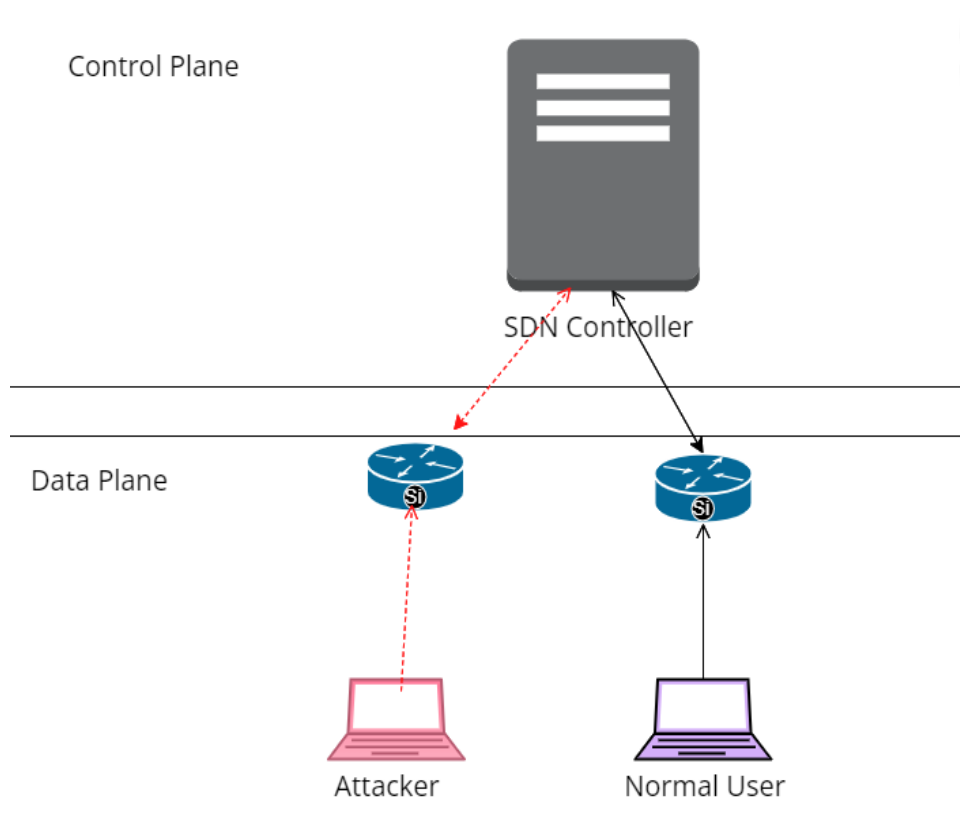


FIGURE 4.1: System Architecture for analysis of DDoS attack

## 4.2 Evaluation Metrics and Methods for Solution

In this experiment, the Ryu SDN controller and Openflow switch are used in Ubuntu to generate the network topology diagram in Figure 4.1. The experimental setup is generated in Mininet. There are an attackers and a normal user. Attacker generates attack traffic and normal user sends normal packets to generate normal flows. Flows include TCP, UDP, and ICMP traffic.

When switch receives the data packets from the hosts (attacker + normal), it started to perform the matching in the flow table. If flow entries matches with the flow table it applies the operation defines according to flow table entry and forwards it to the corresponding path. If flow receives are from the normal user, then controller defines the flow rules according to the network management policy and sends reply to the switch to add this new flow table entry.

If corresponding flow table entries does not matched, then the packets are forwarded to the SDN controller.

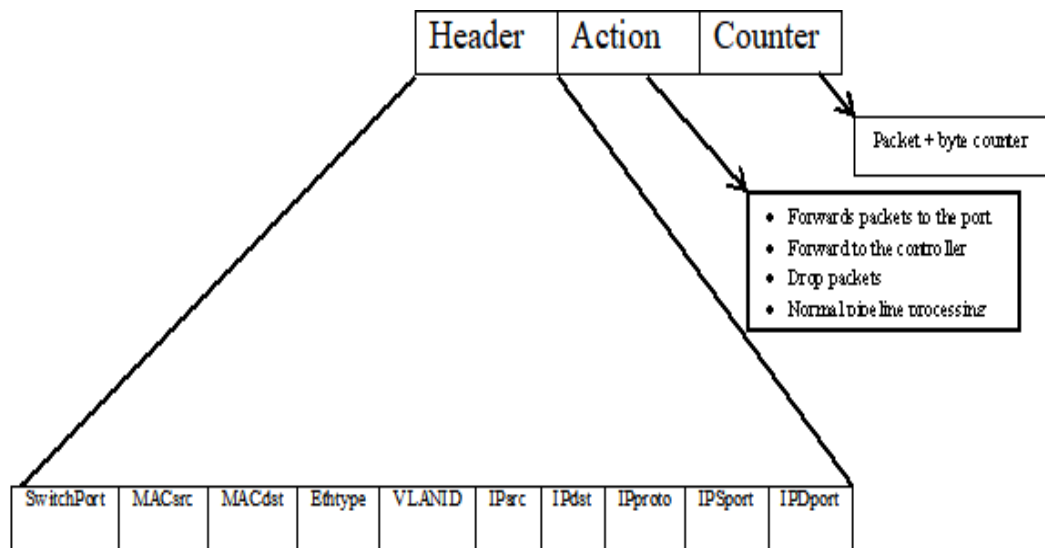


FIGURE 4.2: Flow Table Entries[1]

As SDN controller major task is to forward the incoming data packets and manage the switch information. Packet forwarding is done with the help of flow table. Flow table entry helps to forward the packet received to one or more paths. Each entry of the switch has the header field, counter and actions.

In the Flow State Collection phase the flow table status is accessed using Openflow protocol. Switch send message to controller to make an entry against the packet. The flow table structure is defined in figure 4.2.

With the above defined structure of flow table entries the flow table entries filed in flow table during simulations are as follows. These flow table entries are collected with `sudo ovs-ofctl dump-flows s1`". The flow table information of switch is given as follows:

TABLE 4.1: Flow Table Information

NXST_FLOW reply cookie=0x0	duration=3.750s	table=0
n_packets=0	n_bytes=0	idle_age=3
priority=1	tcp in_port=2	dl_src=0e:ac: ba:f7:a1:38
dl_dst=a6:5d:ec:fd:fd:03	nw_src=76.136.213.90	nw_dst=10.0.0.1
tp_src=10699	tp_dst=0	actions=output:

In order to measure the efficiency of the research, following parameters are used. Those are accuracy, true positive, true negative, false positive, false negative, precision, recall and f-score.

- **Accuracy:** It is percentage of the total variables that were correctly detected in the given dataset. It is calculated using formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

- **True Positive:** It is the percentage of correctly detection in a dataset which are detected in reality. It is calculated using formula:

$$TP = \frac{TP}{ActualPositive} \quad (4.2)$$

- **True Negative:** It is the percentage of correctly not detection in a dataset which are not detected in reality. It is calculated using formula:

$$TN = \frac{TN}{ActualNegative} \quad (4.3)$$

- **False Positive:** It is the percentage of correctly detection in a dataset which are incorrect in reality. It is calculated using formula:

$$FP = \frac{FP}{ActualNegative} \quad (4.4)$$

- **False Negative:** It is the percentage of incorrectly detection in a dataset which are true in reality. It is calculated using formula:

$$FN = \frac{FN}{ActualPositive} \quad (4.5)$$

- **Precision:**

It is the percentage of correct prediction that is actually correct. It is calculated using formula:

$$Precision = \frac{TP}{TP + FP} \quad (4.6)$$

- **Recall:** It is the percentage of incorrect prediction that is actually correct. It is calculated using formula:

$$Recall = \frac{TP}{TP + FN} \quad (4.7)$$

- **F-Score:** It is the percentage of actual positive instance that are present in a data set. It is calculated using formula:

$$Fscore = 2 * \frac{precision + recall}{precision * recall} \quad (4.8)$$

### 4.3 Experimental Setup

At start the User's box of Figure 4.3 Hping3 is used to generate both attack and normal network traffic which is a classical traffic generator and it can generate TCP SYN flood, UDP flood, and ICMP flood attack traffic.

80% data is used for training and 20% data for testing. When this traffic reached at switch it checks the flow table entries against the traffic, if it does not find any entry against a flow.

It sends this flow to the controller. Controller then checks the threshold value, if it exceeds to threshold, it send it to the classifier. On classifier Pandas is used to work with Numpy to provide support for dimension array, which helps to read into dataset. scikit-learn is used, which helps to train the dataset.

Matplotlib is used accurate predictor of possible intrusions on a network at classifier. If the attack is detected then mitigation process starts and blocks the source port.

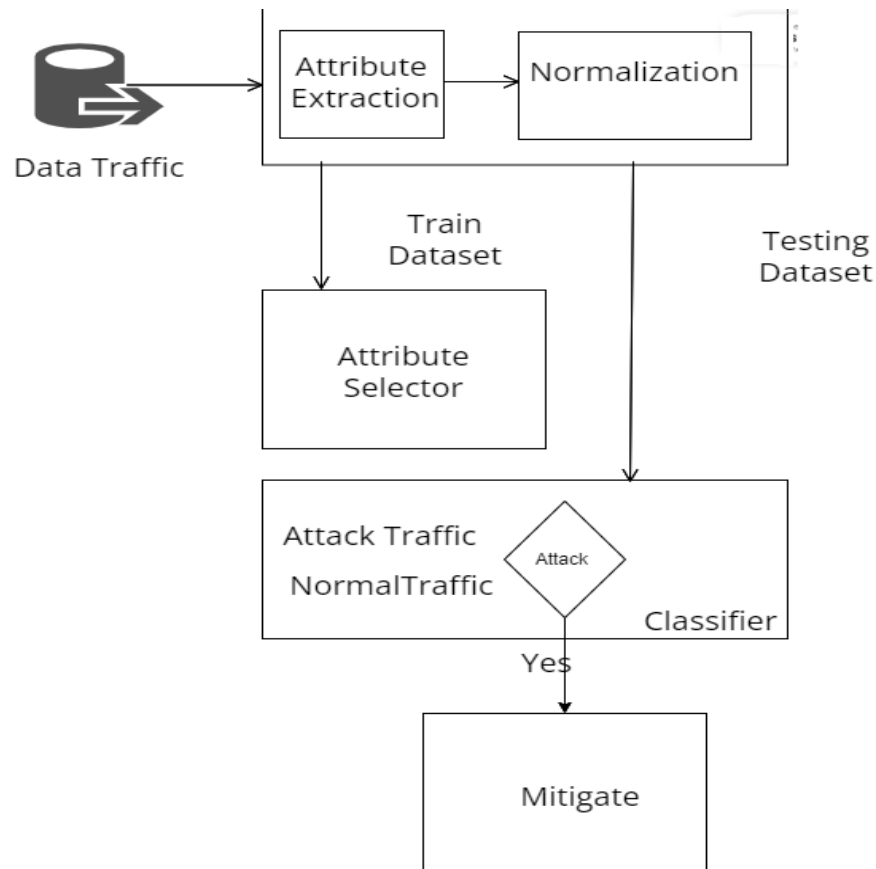


FIGURE 4.3: System Architecture for DDoS Attack

#### 4.4 Traffic Features Comparison Normal Traffic vs. Attack Traffic

When an attacker initiates a DDoS in the network it started sending large number of flows of data packet from different IP addresses. The six defined features of flow status as define in chapter 3 stated to extract the information of the flows; we compare characteristics of these features both for normal traffic and attack traffic.

This analysis is useful to learn how attack traffic can be distinguished from normal traffic.

Figure 4.4 shows the traffic which is a normal traffic and attack traffic generated by an attacker and a normal user towards the controller at different amount of time. First, we transmit normal traffic only and measure the flow arrival rate and then we transmit attack traffic separately and measure the flow arrival rate. The red line shows the attack traffic while the blue line shows the normal traffic. The maximum flow arrival rate of normal user reaches to 125 k and the attacker flow rate touched 250K. Figure shows that flow arrival rate of normal user fluctuates whereas the attacker continuously sending the high flow rates.

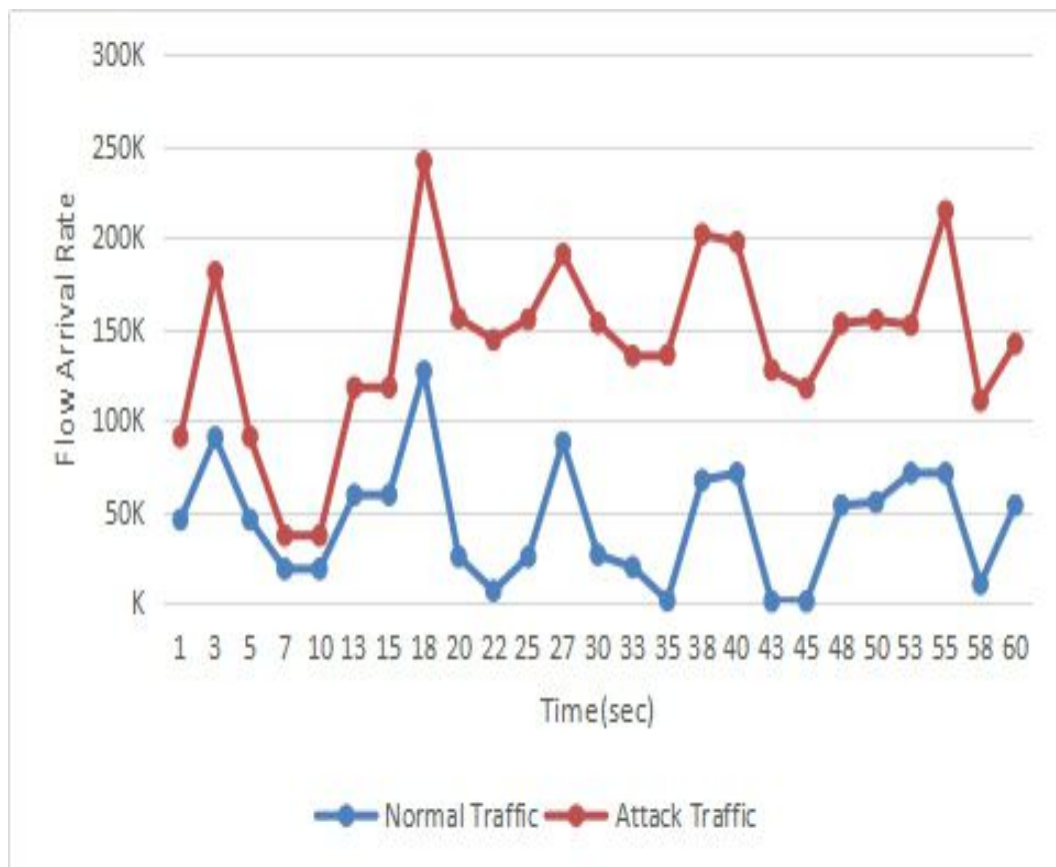


FIGURE 4.4: Flow Arrival Rate (Normal Traffic & Attack Traffic)

Figure 4.5 shows the mixed traffic (normal + attack (mixed traffic)) traffic generated by both normal user and attacker at the same time in parallel towards the controller, where the average flow arrival rate lies between 100k to 140k, As it exceeded the defined threshold value. At time 16 sec to 35 sec and from 38 to 55 sec the flow arrival rate is continuously high. There is a drop in between two



peaks but again reaches the 140K. Figure shows that the flow arrival rate remains high maximum time which shows that the network is under attack continuously. The controller continuously receives high flow arrival rate, which can affect the performance of the controller.

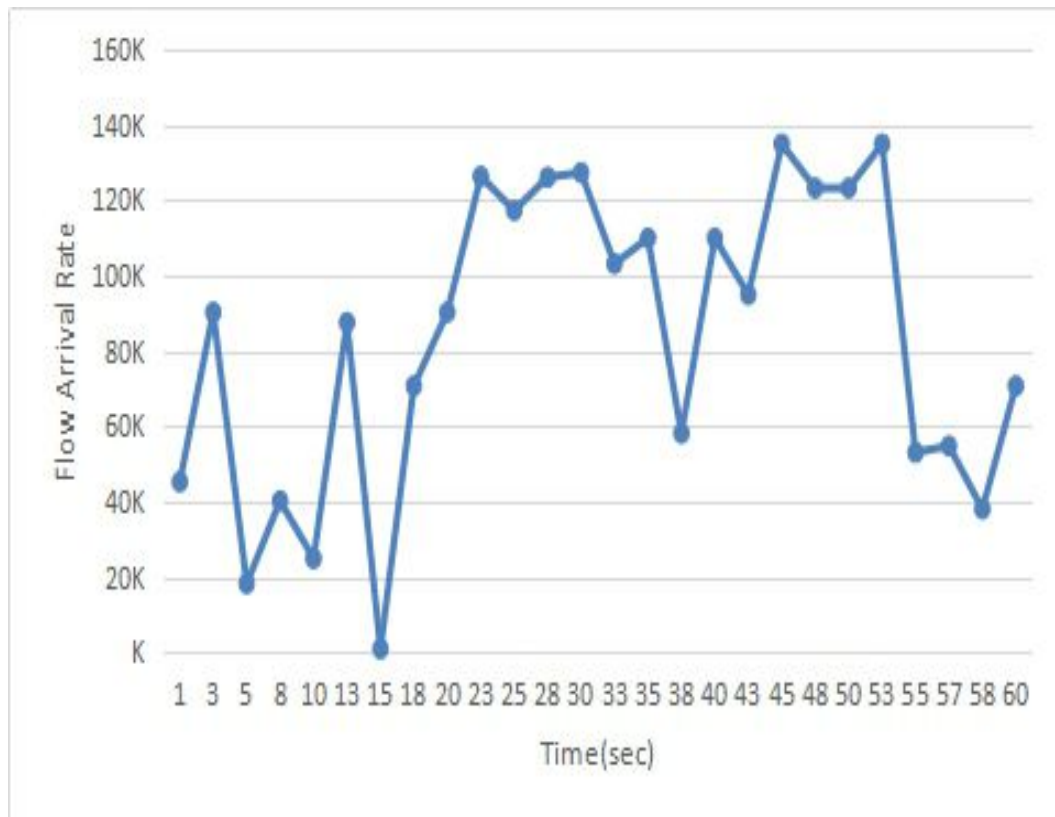


FIGURE 4.5: Mixed Traffic (Normal+Attack Traffic)

The above two defined figure 4.4 and 4.5 shows the first defined feature results as when the attack occurs, a large number of IPs and source port are randomly generated. So the flow arrival rate at switch increases, which forward it to the controller for flow table entry. With this increased number of flow the resources of controller can exhausts.

Figure 4.6 shows the standard deviation of the Number of Flow Packets (Normal and Attack Traffic), generated by both normal user and attacker separately/turn-wise towards the controller. It can be noticed that in case of attack traffic there is little deviation since attacker prefer to transmit continuously at high rates. While the incoming traffic from the normal user fluctuate every time. it can remains high, if normal user send high traffic rate continuously. But this can be done not often.

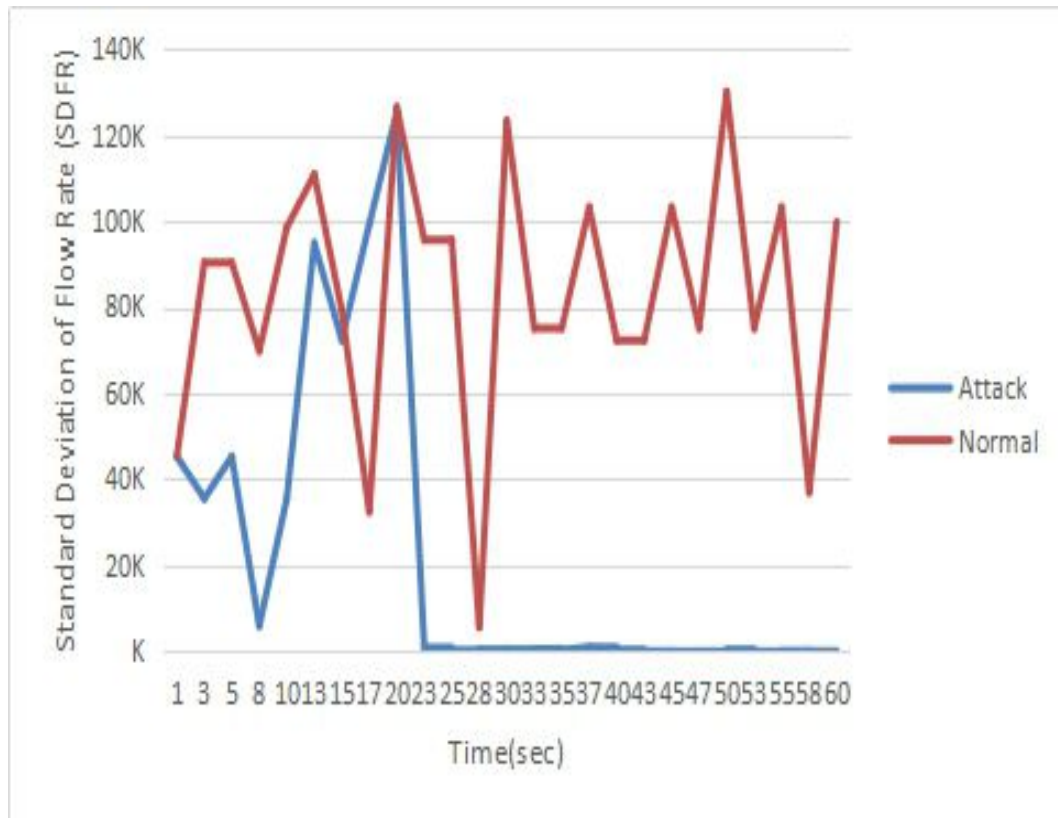


FIGURE 4.6: The standard deviation of the Number of Flow Packets (Normal & Attack Traffic)

Figure 4.7 shows the standard deviation of the Number of Flow Packets (Normal + Attack Traffic at the same time) generated by both normal user and attacker at the same time (parallel) towards the controller. Figure shows that the standard deviation drops at time 20 sec to 45 sec, this is because the attacker major task is generated more traffic so for this packets size remained unchanged, so, the standard deviation of flow packets will be smaller than the normal flow. On the other hand normal packet size fluctuates so, after time 45 sec it again started to fluctuate and touches the highest peak upto 225k.

Figure shows that when there is a mixed traffic, then the packet size remains smaller in most of the time.

Figure 4.8 shows the Standard Deviation of the Number of Flow bytes (Normal & Attack) in bytes generate by normal user and attacker at different time. Under normal circumstances the more fluctuation can be observed as compared to attack traffic. When an attack occurs the standard deviation of flow bytes decreases since bytes are constantly transmitted at high rate to exhaust the controller resources.

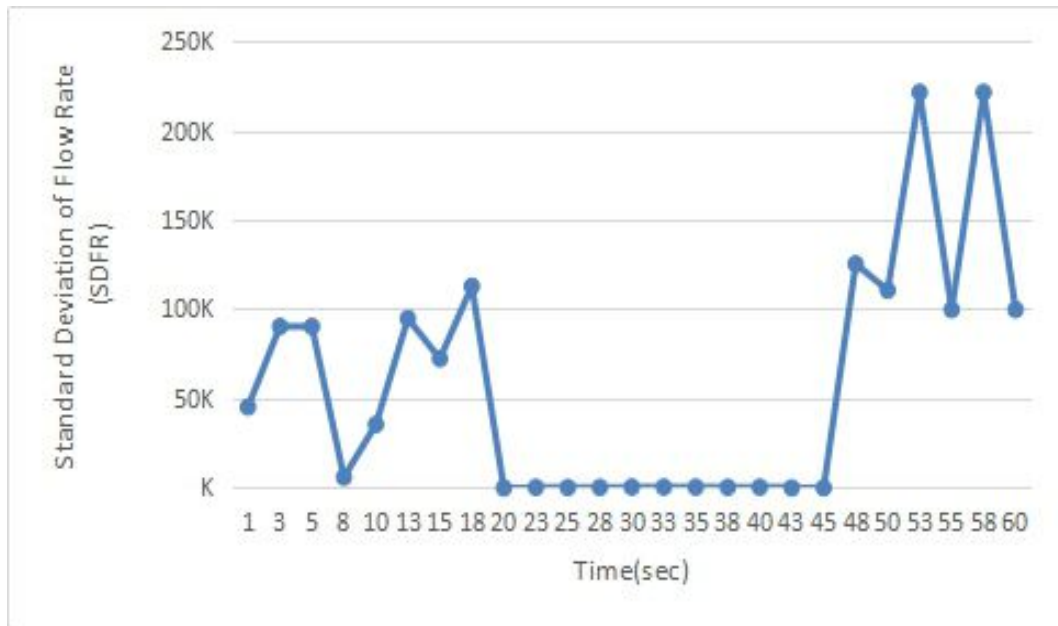


FIGURE 4.7: The standard deviation of the Number of Flow Packets (Mixed Traffic (Normal + Attack Traffic))

The small packet size is the main approach of the attacker. Since it does not have to send the important data packets, but it only send the packets with no data. The attacker continuously sends the flow with high packet arrival rate. So, instead of make a huge data packet, it continuously send the smaller data packets in order to exhaust the resources of the controller.

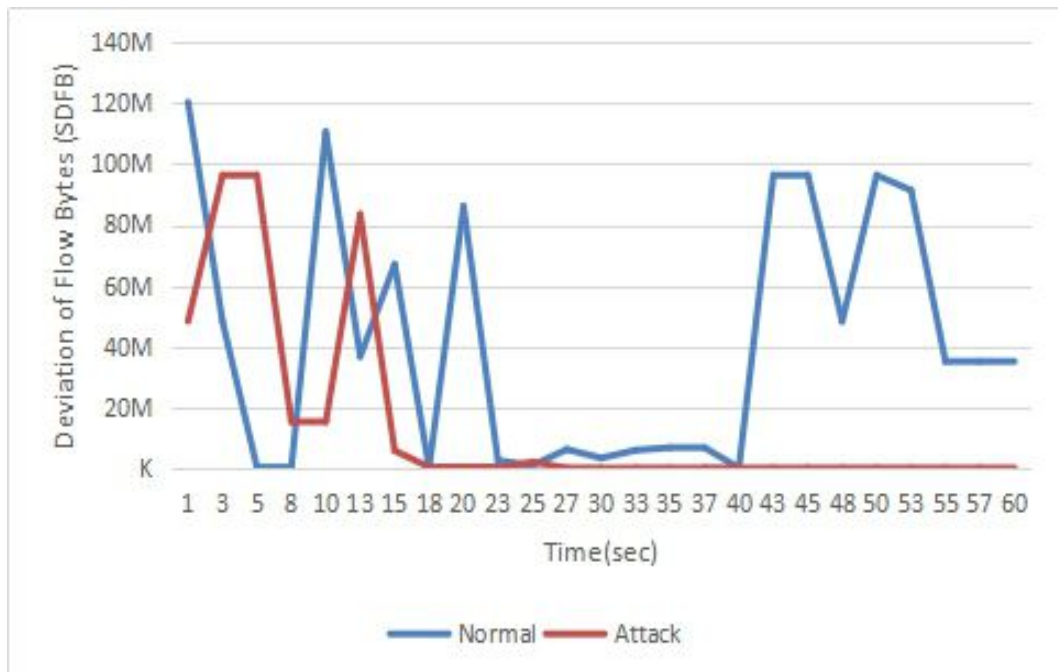


FIGURE 4.8: The Standard Deviation of the Number of Flow bytes (Normal & Attack)

Figure 4.9 shows the Standard Deviation of the Number of Flow bytes for mixed traffic (Normal + Attack) generated by both normal user and attacker in bytes at the same time towards the controller. At time 8 sec the SDFB drops, but again reaches to 100M and remained in this state upto time 24 sec.

At time 25 sec it drops to nearly 50K and remained in this state upto 45 sec. Figure shows that when attack occur the flow arrival rate increases but on the other hand the packet size is smaller than normal flow. As major goal of the attacker is to exhaust the resources of the controller so instead of sending varying size packet that attacker sends smaller size packets.

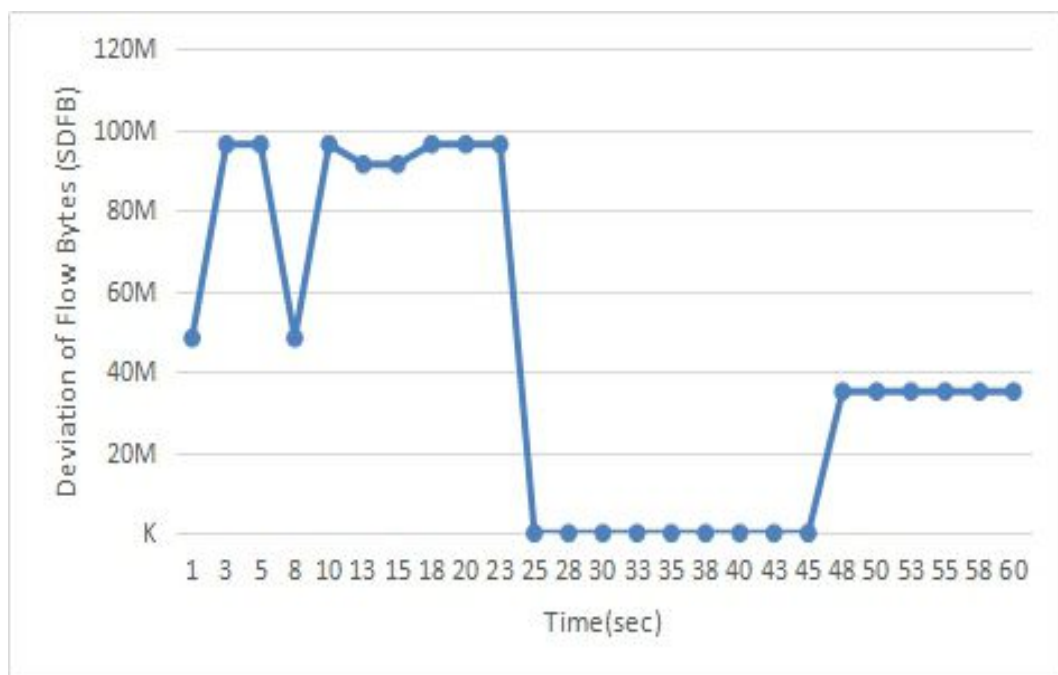


FIGURE 4.9: The Standard Deviation of the Number of Flow bytes(Mixed Traffic (Normal + Attack))

Figure 4.10 show the speed of flow entries of Normal Traffic and attack traffic generated by normal user and attacker at different amount of time towards the controller. Figure shows the when attack occurs the number of flows per unit time increases at controller since during attack the flow entries are much higher than normal traffic.

Figure shows that flows remain consistently high during attack traffic, whereas the normal traffic flows shows some fluctuation. The major of attacker is to exhaust the resources of the controller so the flow entry increases per unit time than normal flow.

Normal flow generated from the normal user can also remain high, as some times many normal users can generate the requests in large amount of time. But this situation can occur not so often, it can be rare. But from the attacker, it remains high always and the attack duration is also prolonged.

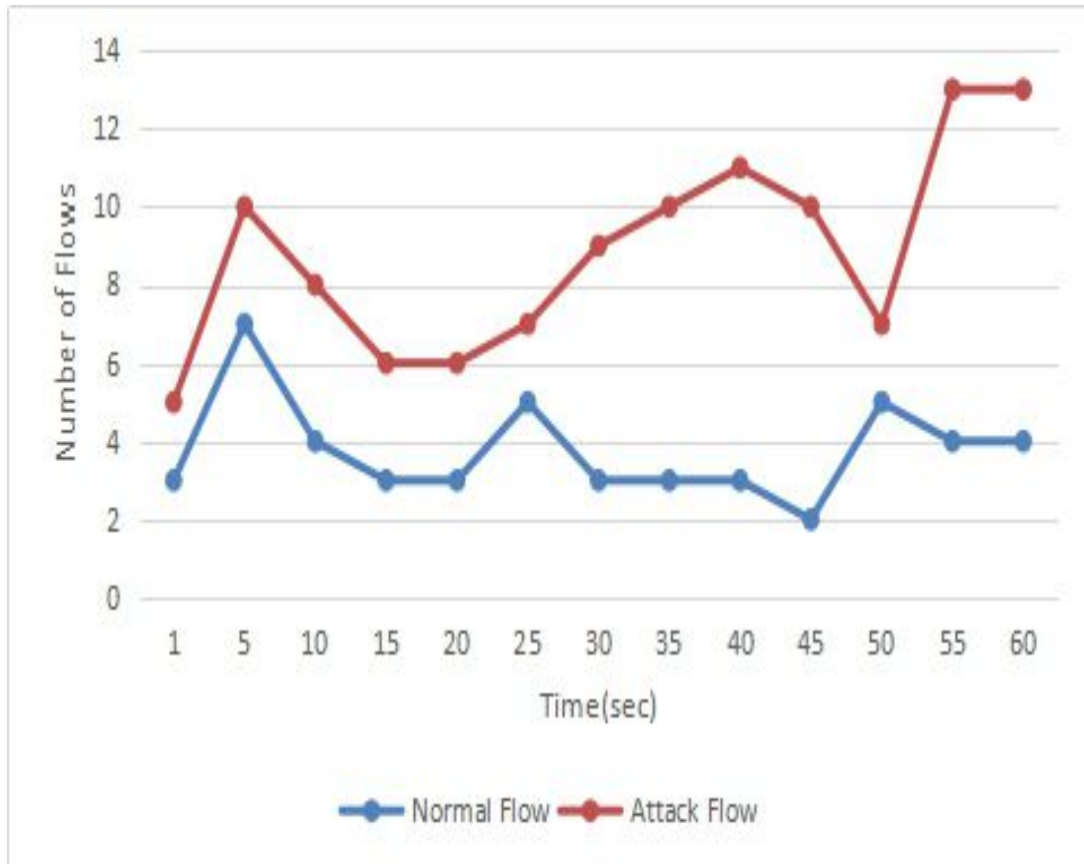


FIGURE 4.10: Speed of Flow Entries (Normal and Attack))

Figure 4.11 show the speed of flow entries of mixed normal traffic and attack traffic generated by normal user and attacker. Figure shows the when attack occurs the number of flows per unit time increases suddenly at switch, since during attack the flow entries are much higher than normal traffic.

when a mixed traffic received BY the switch it can be of various speeds. When normal user sends the flow, it can be of speed, some time is too low and some time it can be high. but from the attacker it is always high. so when this mixed traffic is received by the switch it remains mostly high due to attackers flow arrival rate, which is always high.

The duration of flow entries on switch increases, and remains in that in maximum amount of time. With this the performance of switch also reduces.

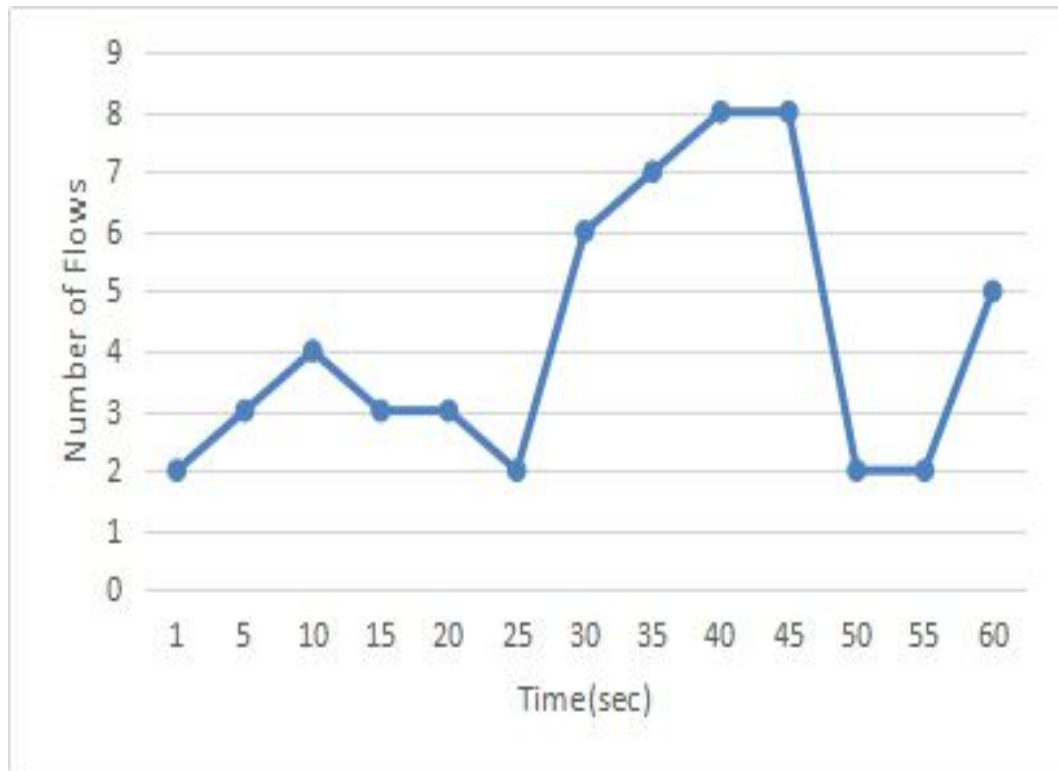


FIGURE 4.11: Speed of Flow Entries(Mixed Traffic (Normal + Attack))

Figure 4.12 show the Ratio of Pair Flow of Normal Traffic and attack traffic generated by normal user and attacker at same time towards the controller. Figure shows that under normal circumstances the controller responds back to the switch after making some flow table entries.

Figure shows there are fluctuation, but when attack occur on the controller then the controller does not respond back to the switch because the flow arrival rate increases drastically towards the controller.

So under normal circumstances the controller sends back the response to the switch. But during attack there in also an effect on the performance of the controller, since it continuously receiving the request and remains in the execution state, and unable to respond back.

Since, the controller is unable to respond back, figure shows that that it became unresponsive and pair flow request drops.

Figure 4.13 show the Ratio of Pair Flow (Normal + Attack Traffic) generated at the same time towards the controller. Figure shows that there is certain drop at time 8 sec but controller started to responds back. But when flow arrival rate increases, the ratio of pair flow decreases from time 30 sec and above.

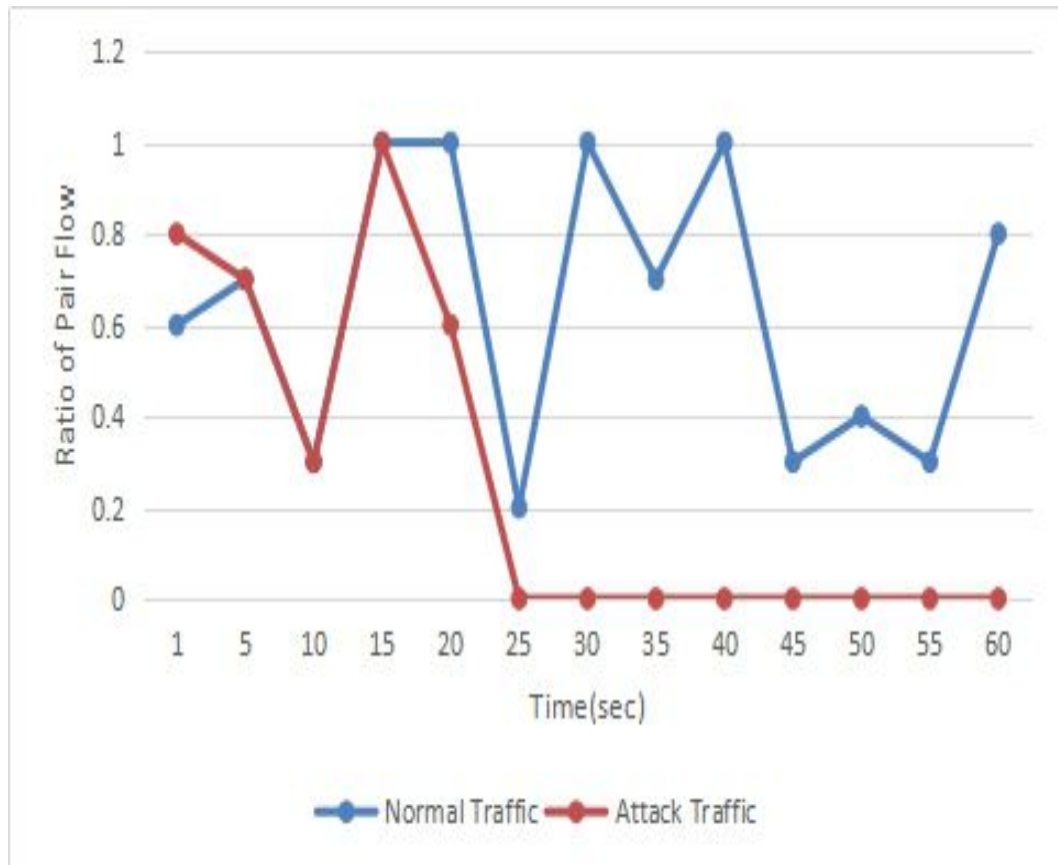


FIGURE 4.12: Ratio of Pair Flow Mixed Traffic (Normal Traffic and Attack Traffic)

Figure shows that when attack happens on the controller and flow rate increases so the interactive ratio drops and controller resources exhaust and it stops responding back to the switch.

When an attack occurs the controller continuously receiving the requests from the switch. Before fulfilling the earlier request it receives bundle of new requests, so it started to remains in this state in order to fulfill all the request. with passage of time the controller stops sending response to the request it receives, so the pair flow response drops drastically.

Above results shows that when attack is detected then classifier task is to check whether the coming flow is attack traffic or a normal flow.

The classifier major task is to extract the defined six tuples and identify whether the flow is an attack flow or a normal flow. When flow is identified as normal flow then necessary flow table entries are made accordingly and controller sends back the response to the switch. But when the flow entry is identified as an attack flow then SVM machine learning algorithm is used for classification.



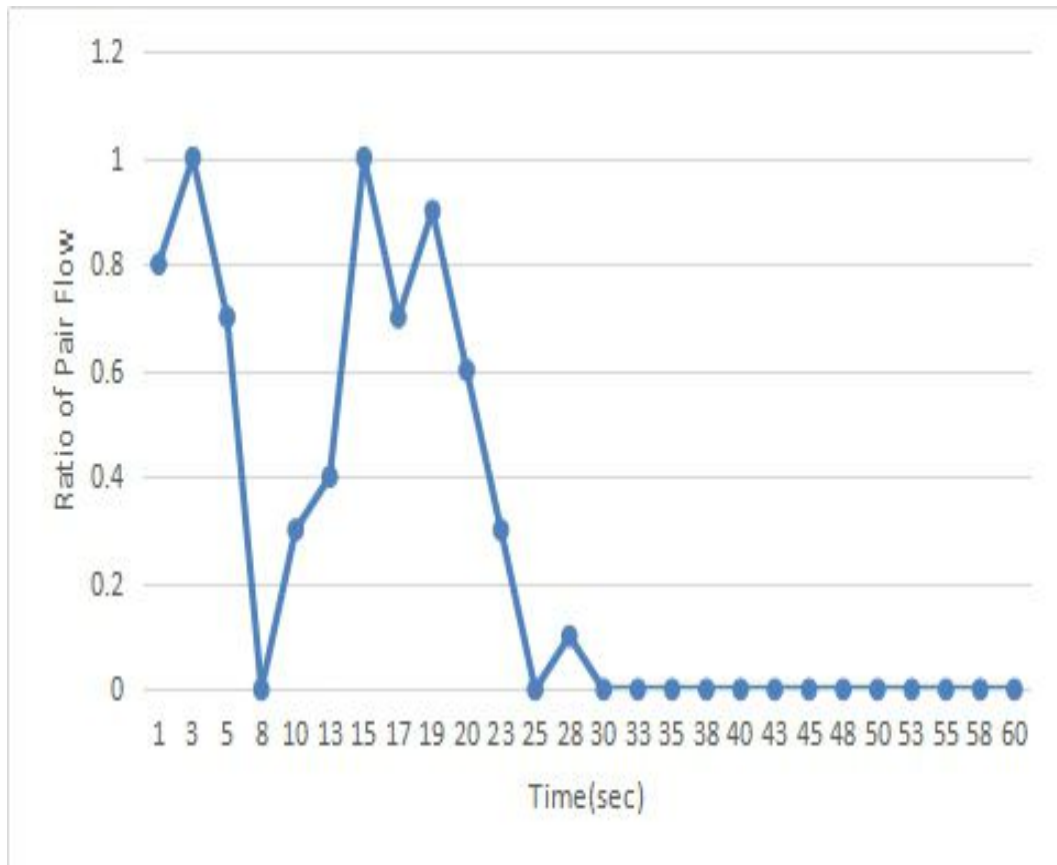


FIGURE 4.13: Ratio of Pair Flow (Normal + Attack)

## 4.5 Accuracy and Precision of SVM

The SVM algorithm is trained on the aforementioned six traffic features to classify the attack traffic from normal traffic. The accuracy of SVM is reported in Table 4.2 and Figure 4.14 for two different datasets with six features.

TABLE 4.2: Comparative Analysis of Datasets on Classifier

Classifier	Dataset	Accuracy%	Time(sec)
SVM	KDD	99.87	11.25
	KKDD'99	87.9	121.26

The traffic is generated from normal user and attacker of Figure 4.1 30 times, and their accuracy is measured each time. Results show that the average accuracy for KDD is 99.87% and for KDD'99 is 87.89% on six features.



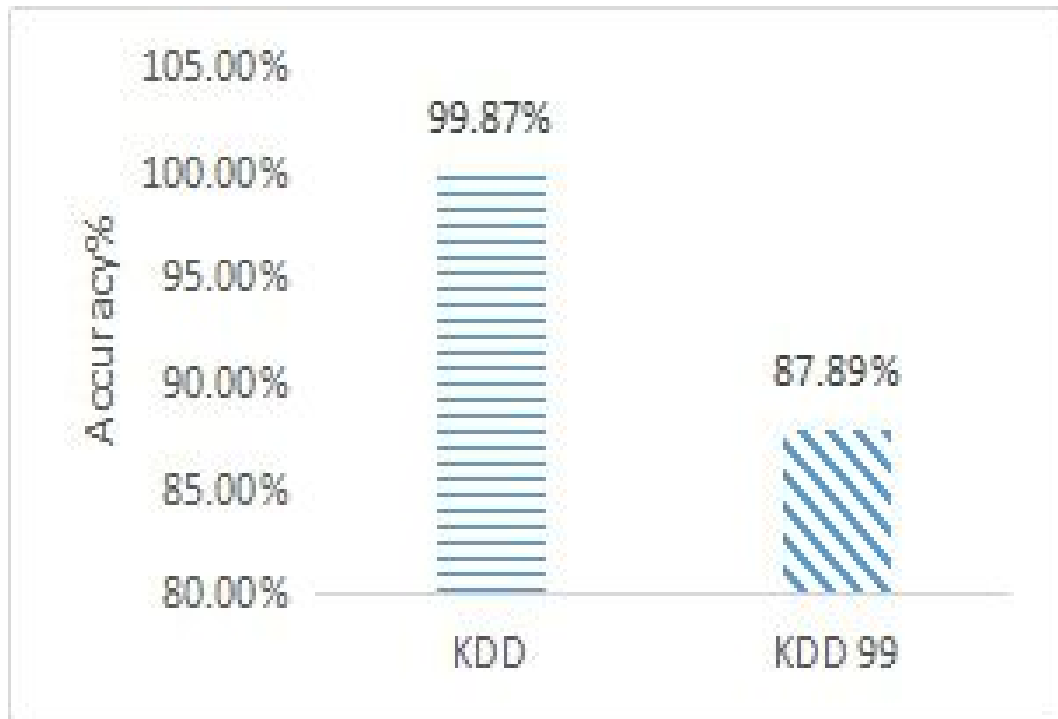


FIGURE 4.14: Accuracy of Datasets

Figure 4.15 and Figure 4.16 show the evaluation matrices of two datasets with above defined six features. The KDD datasets have precision of 0.97% while on the other hand the precision of KDD'99 dataset is 0.87%. The Recall of KDD dataset is 0.88% and KDD'99 is 0.81%. The F-score of KDD is about 0.92% and of KDD is 0.84%.

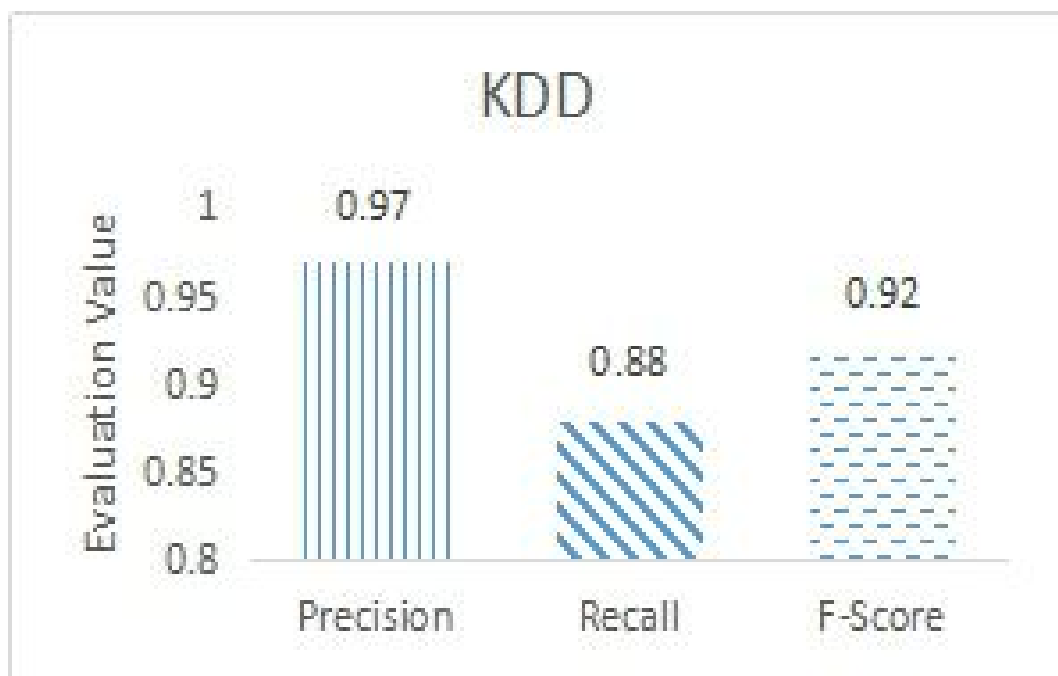


FIGURE 4.15: Precision, Recall, F-Score of KDD dataset

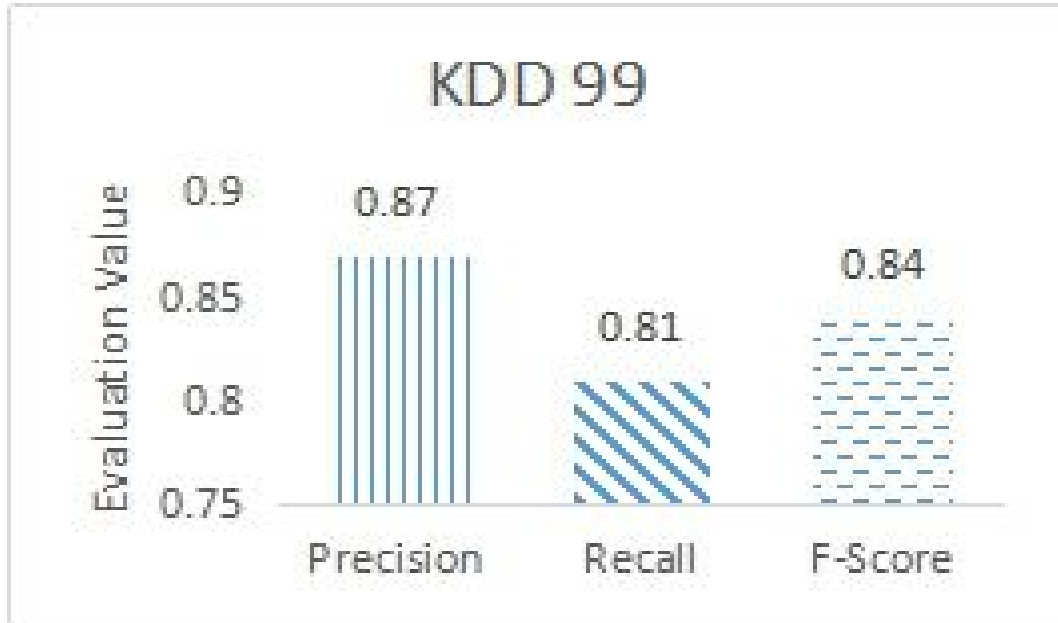


FIGURE 4.16: Precision, Recall, F-Score of KDD'99 dataset

Figure 4.17 shows the graph of accuracy of KDD with different features set. In this figure the maximum accuracy achieved are from the six tuple defined in section 3.4.

Graph 4.17 shows many combinations of features. Figure shows that the feature 1, 3 and 6 are most important feature as it has huge impact on the accuracy. If we have these three features in any combination the accuracy improves, on the other hand the accuracy drops when any one is not included.

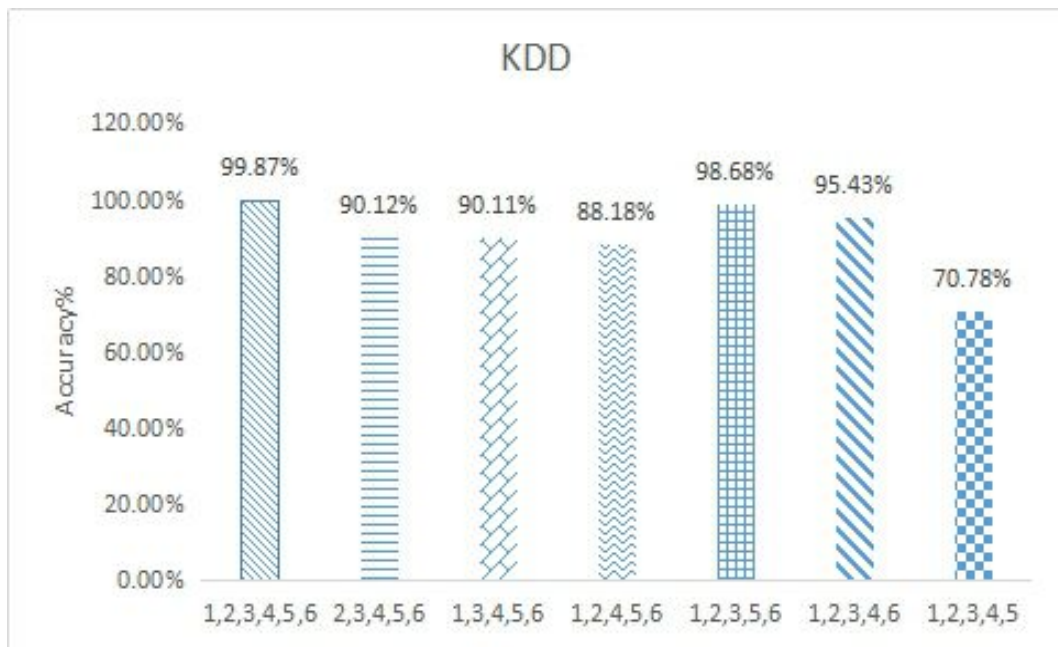


FIGURE 4.17: Accuracy of KDD with Different Features

Figure 4.18 shows the graph of accuracy of KDD 99 with different features set. In this figure the maximum accuracy achieved are from the six defined tuple.

Many different combinations of features are defined and their impacts on accuracy are shown in figure below. Figure shows that the feature 1, 3 and 6 (The Speed of Source IP (SSIP), The Deviation of Flow Bytes (SDFB), The Ratio of Pair-Flow (RPF)) are most important feature as it has huge impact on the accuracy.

If we have these three features in any combination the accuracy improves, on the other hand the accuracy drops when any one is not included.

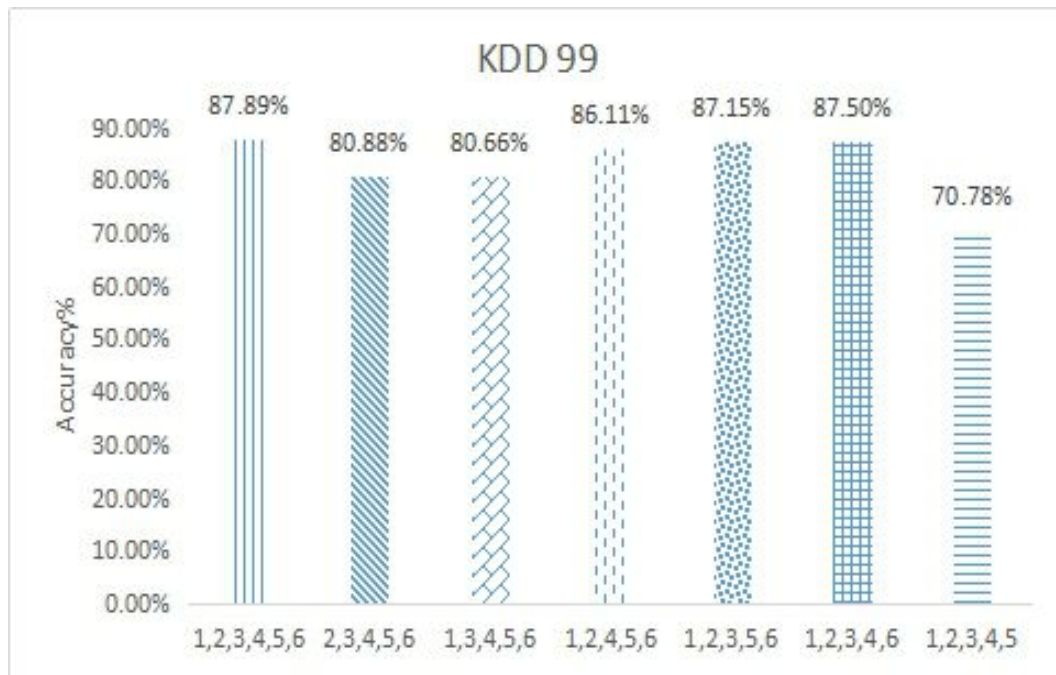


FIGURE 4.18: Accuracy of KDD'99 with Different Features

## 4.6 Mitigation of DDoS Attack

When attack occurs and the classifier identifies that it is attack traffic not a normal flow, then the next major task is to start the mitigation process in order to secure the controller. So the mitigation process starts.

Mitigation process from the controller helps to blacklist the abnormal flows and inform switch about it.

Figure 4.19 shows that the controller before mitigation and controller after mitigation. When heavy flow arrival rate is detected and Classifier detects the incoming traffic is attack traffic, then it started to mitigate and block all incoming flows

entries by blocking the port where the attack occurs.

When mitigation starts the controller takes 11.25 seconds to responds back to the request.

Figure 4.19 shows that from time 65-76 seconds the controller respond suddenly drops and after mitigation the controller responds back.

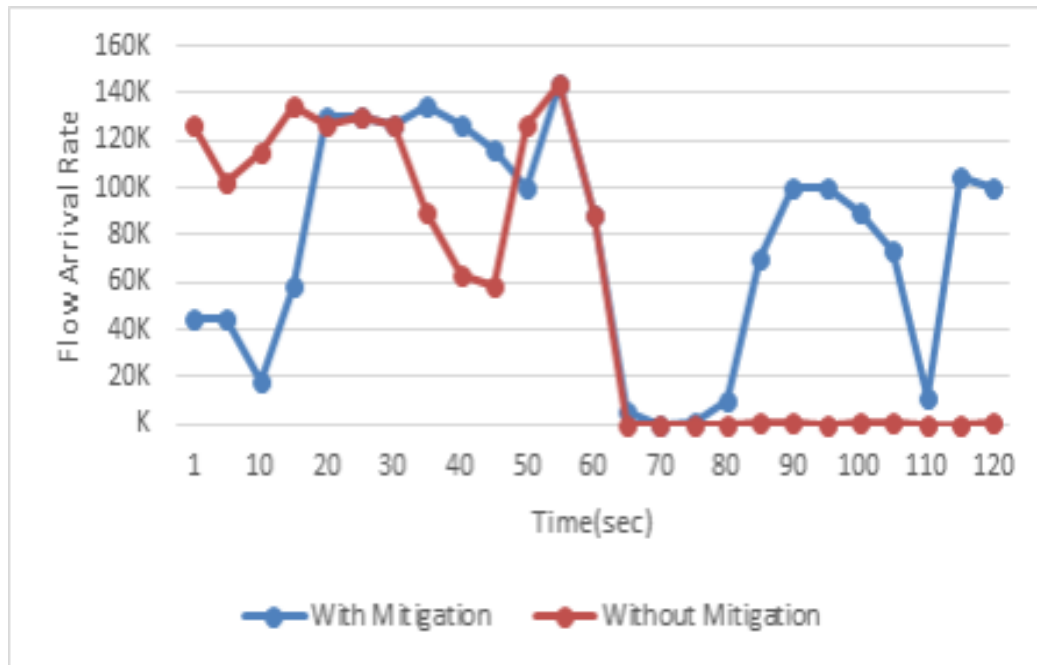


FIGURE 4.19: With Mitigation and without Mitigation

# Chapter 5

## Conclusion and Future Work

### 5.1 Conclusion and Future Work

Our proposed DDoS detection and Mitigation solution is based on SVM. It consists of three modules; pre-processing, attributes selection and attack detection, and finally the mitigation system. For attack detection, the incoming traffic is first evaluated against a packet arrival threshold rate, if traffic exceeds the threshold it is forwarded to SVM classifier for further inspection. If the classifier detects it as an attack flow, it is reported to mitigation module to initiate the prevention mechanism by blocking all attack traffic flows. Our SVM classifier is trained on two different datasets i.e. KDD and KDD99. The total number of features in KDD dataset are 42 whereas there are 41 features in KDD 99. The numbers of features are reduced to 6 by selecting the most relevant features for our study. The SVM classifier is trained using 80% of data whereas 20% of data is used for testing. The accuracy achieved for KDD is 99.87% whereas for KDD99 the accuracy of 87.9% is achieved. The precision, recall and f-score of KDD are 0.97, 0.88 and 0.92 which is better than KDD'99 which is 0.87, 0.81 and 0.84. Overall, the results are impressive on both datasets, however, SVM classifier performed better with KDD in terms of accuracy, precision, recall and F-score. Once the attack is detected correctly, our mitigation module responded quickly and the controller restores normal behavior within 11.25 seconds. In future, we are planning to evaluate the accuracy of our solution in a larger setup under different scenarios and different sources of traffic using different network topology's.

Our experimental network scenario is small LAN. But practical networks can have different scenarios such as bigger LAN, ISP, and data center networks etc. The proposed network topology is a small network with only two SDN switches.

It would be more interesting as a future work to evaluate the performance of our proposed solution for these real network scenarios. In such networks, the network traffic might be more distributed on multiple SDN switches. In this case individual flows may not exceed the threshold limit, however in aggregate all traffic would be directed to controller and generate a DDoS attack as a combine effects of different flows. In such a setup, in additional to traffic profile it would also important to see the relevance between different flows in order to accurately detect a DDoS attack using our machine learning model. We would like to further explore this idea in our future work.

# Bibliography

- [1] Imran, Zeba Ghaffar, Abdullah Alshahrani, Muhammad Fayaz, Ahmed Mohammed Alghamdi, and Jeonghwan Gwak. A topical review on machine learning, software defined networking, internet of things applications: Research limitations and challenges. *Electronics*, 10(8):880, 2021.
- [2] B. Senthilnayagi, K. Venkatalakshmi, and Kannan Arputharaj. Intrusion detection system using feature selection and classification technique. *International Journal of Computer Science and Application*, 3:145, 01 2014. doi: 10.14355/ijcsa.2014.0304.02.
- [3] Shubham Kumar, Sumit Kumar, and Valluri Sarimela. Software-defined networks and methods to mitigate attacks on the network. In *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2017, Volume 1*, pages 317–327. Springer, 2019.
- [4] Amjad Alsirhani, Srinivas Sampalli, and Peter Bodorik. Ddos attack detection system: utilizing classification algorithms with apache spark. In *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)*, pages 1–7. IEEE, 2018.
- [5] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.
- [6] Enterprise ai, August 16, 2023. URL <https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning>. August 16, 2023.
- [7] Ihsan Abdulqadder, Deqing Zou, Israa Aziz, and Bin Yuan. Validating user flows to protect software defined network environments. *Security and Communication Networks*, 2018:1–14, 02 2018. doi: 10.1155/2018/1308678.

- 
- [8] Geeksforgeeks, April 18, 2023. URL <https://www.geeksforgeeks.org/what-is-reinforcement-learning/>. June 20, 2023.
- [9] Ayon Dey. Machine learning algorithms : A review. 2016. URL <https://api.semanticscholar.org/CorpusID:40455026>.
- [10] KE Elliott and CM Greene. A local adaptive protocol. argonne national laboratory, argonne. Technical report, France, Tech. Rep.: 916-1010-BB, 1997.
- [11] Kriti Bhushan and Brij B Gupta. Distributed denial of service (ddos) attack mitigation in software defined network (sdn)-based cloud computing environment. *Journal of Ambient Intelligence and Humanized Computing*, 10: 1985–1997, 2019.
- [12] Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.
- [13] Tushar Ubale and Ankit Kumar Jain. Taxonomy of ddos attacks in software-defined networking environment. In *Futuristic Trends in Network and Communication Technologies: First International Conference, FTNCT 2018, Solan, India, February 9–10, 2018, Revised Selected Papers 1*, pages 278–291. Springer, 2019.
- [14] Felipe S Dantas Silva, Esau Silva, Emidio P Neto, Marcilio Lemos, Augusto J Venancio Neto, and Flavio Esposito. A taxonomy of ddos attack mitigation approaches featured by sdn technologies in iot scenarios. *Sensors*, 20(11): 3078, 2020.
- [15] Richard P Lippmann, David J Fried, Isaac Graf, Joshua W Haines, Kristopher R Kendall, David McClung, Dan Weber, Seth E Webster, Dan Wyschogrod, Robert K Cunningham, et al. Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, volume 2, pages 12–26. IEEE, 2000.



- [16] Mahbod Tavallaei, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. Ieee, 2009.
- [17] Sathyanarayanan Revathi and A Malathi. A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)*, 2(12): 1848–1853, 2013.
- [18] Liang Tan, Yue Pan, Jing Wu, Jianguo Zhou, Hao Jiang, and Yuchuan Deng. A new framework for ddos attack detection and defense in sdn environment. *IEEE Access*, 8:161908–161919, 2020.
- [19] Yang Xu and Yong Liu. Ddos attack detection under sdn context. In *IEEE INFOCOM 2016-the 35th annual IEEE international conference on computer communications*, pages 1–9. IEEE, 2016.
- [20] 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.
- [21] Yuhua Xu, Houtao Sun, Feng Xiang, and Zhixin Sun. Efficient ddos detection based on k-fknn in software defined networks. *IEEE access*, 7:160536–160545, 2019.
- [22] Dingwen Hu, Peilin Hong, and Yixin Chen. Fadm: Ddos flooding attack detection and mitigation system in software-defined networking. In *GLOBECOM 2017-2017 IEEE global communications conference*, pages 1–7. IEEE, 2017.
- [23] Narmeen Zakaria Bawany, Jawwad A Shamsi, and Khaled Salah. Ddos attack detection and mitigation using sdn: methods, practices, and solutions. *Arabian Journal for Science and Engineering*, 42:425–441, 2017.
- [24] Kübra Kalkan, Levent Altay, Gürkan Gür, and Fatih Alagöz. Jess: Joint entropy-based ddos defense scheme in sdn. *IEEE Journal on Selected Areas in Communications*, 36(10):2358–2372, 2018.

- [25] Yunhe Cui, Qing Qian, Chun Guo, Guowei Shen, Youliang Tian, Huanlai Xing, and Lianshan Yan. Towards ddos detection mechanisms in software-defined networking. *Journal of Network and Computer Applications*, 190: 103156, 2021.
- [26] 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.
- [27] V Deepa, K Muthamil Sudar, and P Deepalakshmi. Design of ensemble learning methods for ddos detection in sdn environment. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, pages 1–6. IEEE, 2019.
- [28] Tariq Emad Ali, Yung-Wey Chong, and Selvakumar Manickam. Comparison of ml/dl approaches for detecting ddos attacks in sdn. *Applied Sciences*, 13(5):3033, 2023.
- [29] Oluwashola David Adeniji, Deji Babatunde Adekeye, Sunday Adeola Ajagbe, Ademola Olusola Adesina, Yetunde Josephine Oguns, and Matthew Abiola Oladipupo. Development of ddos attack detection approach in software defined network using support vector machine classifier. In *Pervasive Computing and Social Networking: Proceedings of ICPCSN 2022*, pages 319–331. Springer, 2022.
- [30] Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Jae-Min Lee, and Dong-Seong Kim. Optimization of rbf-svm kernel using grid search algorithm for ddos attack detection in sdn-based vanet. *IEEE Internet of Things Journal*, 2022.
- [31] Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Jae-Min Lee, and Dong-Seong Kim. Appropriate svm kernel selection for ddos attack detection in sdn-based vanet. , pages 1251–1252, 2022.
- [32] Basem A Almohagri, Mogebeeb A Saeed, Haroon M Alazaby, and Ayman I Mohammed. Machine learning approach for distributed daniel of service attack

- detection in sdns. In *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, pages 01–07. IEEE, 2023.
- [33] Oluwapelumi Fakolujo and Amna Qureshi. Analysis of detection systems in a software-defined network. In *Science and Information Conference*, pages 1342–1363. Springer, 2023.
- [34] Trung V Phan and Minh Park. Efficient distributed denial-of-service attack defense in sdn-based cloud. *IEEE Access*, 7:18701–18714, 2019.
- [35] Shuhua Deng, Xing Gao, Zebin Lu, Zhengfa Li, and Xieping Gao. Dos vulnerabilities and mitigation strategies in software-defined networks. *Journal of Network and Computer Applications*, 125:209–219, 2019.
- [36] Nasrin Sultana, Naveen Chilamkurti, Wei Peng, and Rabei Alhadad. Survey on sdn based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, 12:493–501, 2019.
- [37] K Muthamil Sudar, M Beulah, P Deepalakshmi, P Nagaraj, and P Chin nasamy. Detection of distributed denial of service attacks in sdn using machine learning techniques. In *2021 international conference on Computer Communication and Informatics (ICCCI)*, pages 1–5. IEEE, 2021.
- [38] Myo Myint Oo, Sinchai Kamolphiwong, Thossaporn Kamolphiwong, Sangsaree Vasupongayya, et al. Advanced support vector machine-(asvm-) based detection for distributed denial of service (ddos) attack on software defined networking (sdn). *Journal of Computer Networks and Communications*, 2019, 2019.
- [39] Yunhe Cui, Lianshan Yan, Saifei Li, Huanlai Xing, Wei Pan, Jian Zhu, and Xiaoyang Zheng. Sd-anti-ddos: Fast and efficient ddos defense in software-defined networks. *Journal of Network and Computer Applications*, 68:65–79, 2016.
- [40] Nguyen Ngoc Tuan, Pham Huy Hung, Nguyen Danh Nghia, Nguyen Van Tho, Trung V Phan, and Nguyen Huu Thanh. A robust tcp-syn flood mitigation

- scheme using machine learning based on sdn. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 363–368. IEEE, 2019.
- [41] 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.
- [42] BV Karan, DG Narayan, and PS Hiremath. Detection of ddos attacks in software defined networks. In *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, pages 265–270. IEEE, 2018.
- [43] Zhang Long and Wang Jinsong. A hybrid method of entropy and ssae-svm based ddos detection and mitigation mechanism in sdn. *Computers & Security*, 115:102604, 2022.
- [44] Mamoun Alazab. Profiling and classifying the behavior of malicious codes. *Journal of Systems and Software*, 100:91–102, 2015.
- [45] Senthilnayaki Balakrishnan, Kannn Venkatalakshmi, and A Kannan. Intrusion detection system using feature selection and classification technique. *International journal of computer science and application*, 3(4):145–151, 2014.

## Turnitin Originality Report

Support Vector Machine Based Distributed Denial of Service Detection and Mitigation in Software Defined Network by Sadia Rasheed

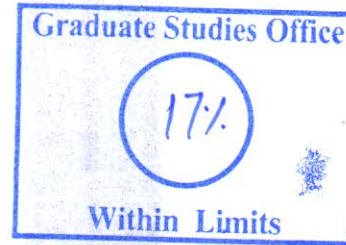


From Quick Submit (Quick Submit)

- Processed on 06-Dec-2023 11:48 PKT
- ID: 2249781397
- Word Count: 12899

Similarity Index  
17%  
Similarity by Source

Internet Sources:  
12%  
Publications:  
12%  
Student Papers:  
7%

**sources:**

- 1 2% match (Internet from 04-Feb-2023)  
<https://thesis.cust.edu.pk/UploadedFiles/Abdul%20Moqeet-MCS183056.pdf>
- 2 1% match (Internet from 02-Feb-2023)  
[https://www.researchgate.net/profile/Anand-Kannan-2/publication/236353735\\_Genetic\\_Algorithm\\_Based\\_Feature\\_Selection\\_Algorithm\\_for\\_Effective\\_Intrusion\\_Detection\\_in\\_Cloud\\_Networks/in-Algorithm-Based-Feature-Selection-Algorithm-for-Effective-Intrusion-Detection-in-Cloud-Networks.pdf](https://www.researchgate.net/profile/Anand-Kannan-2/publication/236353735_Genetic_Algorithm_Based_Feature_Selection_Algorithm_for_Effective_Intrusion_Detection_in_Cloud_Networks/in-Algorithm-Based-Feature-Selection-Algorithm-for-Effective-Intrusion-Detection-in-Cloud-Networks.pdf)
- 3 1% match (Internet from 25-Nov-2022)  
<https://downloads.hindawi.com/journals/specialissues/414209.pdf>
- 4 1% match (Liang Tan, Yue Pan, Jing Wu, Jianguo Zhou, Hao Jiang, Deng Yuchuan. "A New Framework for DDoS Attack Detection and Defense in SDN Environment", IEEE Access, 2020)  
[Liang Tan, Yue Pan, Jing Wu, Jianguo Zhou, Hao Jiang, Deng Yuchuan. "A New Framework for DDoS Attack Detection and Defense in SDN Environment", IEEE Access, 2020](https://doi.org/10.1109/ACCESS.2020.3000000)
- 5 1% match ("Intelligent Computing", Springer Science and Business Media LLC, 2023)  
["Intelligent Computing", Springer Science and Business Media LLC, 2023](https://doi.org/10.1007/978-1-4939-9888-8_1)
- 6 1% match (student papers from 07-Sep-2020)  
[Submitted to University of Oxford on 2020-09-07](https://www.oxfordjournals.org/doi/10.1093/oxfordjournals/itp.a011111)
- 7 < 1% match (Internet from 11-Jun-2022)  
[https://www.researchgate.net/profile/Zouhair-Chiba/post/Machine\\_Learning\\_in\\_SDN/attachment/5ccca2a6cfe4a7968d9c502e/AS%3A754569743200256%401556914854775/download/A-+Research+Issues+and+Challenges-2018.pdf](https://www.researchgate.net/profile/Zouhair-Chiba/post/Machine_Learning_in_SDN/attachment/5ccca2a6cfe4a7968d9c502e/AS%3A754569743200256%401556914854775/download/A-+Research+Issues+and+Challenges-2018.pdf)
- 8 < 1% match (Internet from 28-Jan-2023)  
[https://www.researchgate.net/figure/TCP-Stream-name-Value-comparisons-filtering-packets-based-on-lengths-secure-BSSIDs\\_fig1\\_46280039](https://www.researchgate.net/figure/TCP-Stream-name-Value-comparisons-filtering-packets-based-on-lengths-secure-BSSIDs_fig1_46280039)
- 9 < 1% match (Internet from 21-Mar-2023)  
[https://www.researchgate.net/publication/335144411\\_Defense\\_Mechanisms\\_Against\\_DDoS\\_Attacks\\_in\\_a\\_Cloud\\_Computing\\_Environment\\_of-the-Art\\_and\\_Research\\_Challenges](https://www.researchgate.net/publication/335144411_Defense_Mechanisms_Against_DDoS_Attacks_in_a_Cloud_Computing_Environment_of-the-Art_and_Research_Challenges)
- 10 < 1% match (Internet from 11-Nov-2022)  
[https://www.researchgate.net/publication/350955612\\_Predicting\\_Absenteeism\\_at\\_Work\\_Using\\_Machine\\_Learning\\_Algorithms](https://www.researchgate.net/publication/350955612_Predicting_Absenteeism_at_Work_Using_Machine_Learning_Algorithms)
- 11 < 1% match (Internet from 13-Dec-2020)  
<https://www.hindawi.com/journals/scn/2018/9804061/>
- 12 < 1% match (Internet from 06-Jan-2023)  
<https://www.hindawi.com/journals/jcnc/2019/8012568/>