

Real Time Performance Analysis on DDoS Attack Detection using Machine Learning

by

Debashis Kar Suvra

16301009

Tanusree Sen

15201046

Maysha Maliha Mou

19241028

Asifur Rahman

20141017

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
April 2020

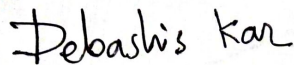
© 2020. Brac University
All rights reserved.

Declaration

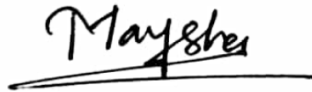
It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

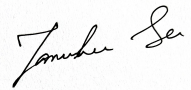
Student's Full Name & Signature:



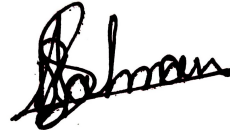
Debashis Kar Suvra
16301009



Maysha Maliha Mou
19241028



Tanusree Sen
15201046



Asifur Rahman
20141017

Approval

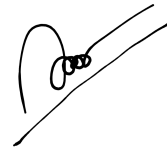
The thesis titled “Real Time Performance Analysis on DDoS Attack Detection using Machine Learning” submitted by

1. Debashis Kar Suvra (16301009)
2. Tanusree Sen (15201046)
3. Maysha Maliha Mou (19241028)
4. Asifur Rahman (20141017)

Of Spring, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on April 7, 2020.

Examining Committee:

Supervisor:
(Member)



DR. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)



DR. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac University

DR. Mahbubul Alam Majumdar
Professor
Department of Computer Science and Engineering
Brac University

Abstract

In recent years, Distributed Denial of service (DDoS) attacks have led to a tremendous financial loss in some industries and governments. Such as banks, universities, news and media publications, financial services, political or governmental servers. DDoS attack is one of the biggest threats for cyber security nowadays. It is a malicious act that slows down the server, makes loss of confidential data and makes reputation damage to a brand. With the advancement of developing technologies for example cloud computing, Internet of things (IoT), Artificial intelligence attackers can launch attacks very easily with lower cost. However, it is challenging to detect DDoS traffic as it is similar to normal traffic. In this era, we rely on the internet services. Attackers send a huge volume of traffic at the same time to a specific network and make the network null and void. So that the server cannot respond to the actual users. As a result, clients cannot get the services from that server. It is very essential to detect DDoS attacks and secure servers from losing important information and data. However, many detection techniques are available for preventing the attack. But it is very challenging to choose one method among those as some are time efficient and some are result oriented. In our paper, we mainly focused on the top machine learning classification algorithms and evaluated the best model according to the dataset. The experimental result shows that the Decision Tree algorithm achieved the excellent accuracy of 98.50 percent with very less time consumption. Therefore, we are using a better approach to detect DDoS attacks in real time.

Keywords: DDoS attacks, detection, Machine Learning, Artificial Intelligence.

Dedication

We would like to dedicate our thesis report to our parents who always supported us. Without their support we may not be able to complete this. Our supervisor helped us throughout the year. We want to also dedicate this to him also. Special gratitude towards our close friends who helped us to do better.

Acknowledgement

Firstly, we would like to thank Muhammad Iqbal Hossain for his constant guidelines and supervision. He provided necessary information for completing the thesis whom we will always be indebted to. We also want to thank all the faculty members for providing us friendly and such a wonderful environment to conduct the research.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Dedication	iv
Acknowledgment	v
Table of Contents	vi
List of Figures	viii
List of Tables	1
1 Introduction	2
1.1 Motivation	4
1.2 Problem Statement	4
1.3 Objectives and Contributions	5
1.4 Thesis Structure	5
2 Algorithms and Related Work	6
2.1 Background	6
2.1.1 Literature Review	7
2.2 Algorithms	8
2.2.1 K-Nearest Neighbor (KNN)	8
2.2.2 Random Forest (RF)	9
2.2.3 Naive Bayesian	10
2.2.4 Support Vector Machine (SVM)	11
2.2.5 AdaBoost	12
2.2.6 XGBoost	13
2.2.7 Decision Tree	14
2.2.8 Logistic Regression	15
3 Our Proposed Model	16
3.1 Dataset description	16
3.1.1 Dataset	16
3.1.2 Dataset Feature Description	16
3.2 Proposed Prediction Model	18

3.2.1	Data preprocessing	18
3.2.2	Feature Selection	19
3.2.3	Selected Features	21
3.2.4	Training or Testing machine learning classifier	22
3.3	Data visualization	23
3.3.1	Heat map	23
3.3.2	Bar Graph	24
3.3.3	Correlation Matrix	25
3.3.4	Count Plot	26
3.3.5	Pie Plot	27
4	Result Analysis and Real Time Implementation	28
4.1	Result Analysis	31
4.1.1	Logistic Regression	31
4.1.2	K-Nearest Neighbour (K-NN)	32
4.1.3	Random Forest	34
4.1.4	AdaBoost	35
4.1.5	Support Vector Machine	37
4.1.6	Decision Tree	38
4.1.7	XGboost	40
4.1.8	Naive Bayesian	41
4.2	Analyzing the Results	45
4.3	Real Time Implementation	46
5	Conclusion and Future Work	49
	Bibliography	52

List of Figures

1.1	Types of DDoS Attack, 2018	3
1.2	Percentage of attack	3
2.1	KNN Algorithm	8
2.2	RF Algorithm	10
2.3	Naive bayes Algorithm	10
2.4	SVM Algorithm	11
2.5	Linear SVM Algorithm	12
2.6	Non-Linear SVM Algorithm	12
2.7	AdaBoost Algorithm	13
2.8	XGBoost Algorithm	14
2.9	Decision Tree Algorithm	14
2.10	Logistic Regression Algorithm	15
3.1	Workflow Diagram of Prediction model	18
3.2	PCA for selecting features using WEKA	20
3.3	Heat map	23
3.4	Bar graph	24
3.5	Correlation Matrix	25
3.6	Count Plot	26
3.7	Pie Plot	27
4.1	Confusion Matrix of Logistic Regression	31
4.2	Classification report of Logistic Regression	31
4.3	ROC Curve for logistic regression	32
4.4	Confusion Matrix of K-Nearest Neighbour	32
4.5	The classification report of KNN	33
4.6	ROC Curve for KNN	33
4.7	Confusion Matrix of Random Forest	34
4.8	The classification report of Random Forest	34
4.9	ROC Curve for Random Forest	35
4.10	Confusion Matrix of AdaBoost	35
4.11	The classification report of AdaBoost	36
4.12	ROC Curve for AdaBoost	36
4.13	Confusion Matrix of Support Vector Machine	37
4.14	The classification report of Support Vector Machine	37
4.15	ROC Curve for SVM	38
4.16	Confusion Matrix of Decision Tree	38
4.17	The classification report of Decision Tree	39

4.18	ROC Curve for Decision Tree	39
4.19	Confusion Matrix of XGboost	40
4.20	The classification report of XGboost	40
4.21	ROC curve for XGBoost	41
4.22	Confusion Matrix of Naive Bayesian	41
4.23	The classification report of Naive Bayesian	42
4.24	ROC Curve for Naive Bayesian	42
4.25	Accuracy rate of Various Algorithms	44
4.26	Execution Time of various Algorithms	44
4.27	Client-Server workflow model of Real-time implementation	46
4.28	Real Time implementation Dashboard	48

List of Tables

3.1	Selected Features Description	21
3.2	Selected Features Description	22
4.1	Performances of various Machine Learning classifying Algorithms Part 1	43
4.2	Performances of various Machine Learning classifying Algorithms Part 2	43
4.3	Average Execution Time of various Algorithms in Google Colab . . .	45

Chapter 1

Introduction

A distributed denial-of-service (DDoS)[34] attack is a malignant attempt that inflicts the usual traffic of a targeted server by sending fake internet traffic. For the developing technologies the attack can be done easily with very low cost and time. It can be done using botnets and zombie computers. So, it is very easy to launch an attack. Moreover, this fake incoming traffic can flood the victim originating from many different sources which are distributed. It sends a huge volume of traffic at the same time. As a result, the bandwidth of the server gets slower or sometimes it cannot respond to the real users. As the attackers use botnet and zombie computers, it is impossible to stop attacks by blocking a single IP address. So, the real user cannot get the services from the server. In consequence, the reputation or the brand value of the business slowly gets down. Sometimes, users move on to another company for the denial of services. It creates a huge financial loss to a company. For the attack, a server can lose important data and information which is a great loss for a business. It is a criminal act that is punishable according to the cyber law. In a nutshell, it is one of the most seen problems nowadays as well as a huge threat for cyber security now.

The attackers usually target news and media publications, universities, online services, online financial industries and government or political server etc[24]. But sometimes attackers target small businesses too. Symantec's report shows that about one in forty small businesses are at risk of being victimized by DDoS attack. For DDoS attacks business servers can be disrupted and consumers may flock. [31] Many cyber criminals use DDoS simply as a smokescreen so that they can control the server and sometimes they blackmail the companies or business.

DDoS attacks can be divided into many types based on the characteristics. One of the types which focus on particular network layers. The Network layer, the Transport layer and the Application layer attack are very common among them. Network layer includes ICMP floods, UDP floods and others. The aim of the attack is saturation of the bandwidth of sites. Transport layer attacks aim to exhaust server resources instead of bandwidth for example firewalls and load balancers. This includes SYN floods, fragmented packet attacks, Ping of Death (PoD) etc. Application layer attack mainly, HTTP-encrypted attacks.

Some of the common and mostly used types of DDoS attack are. UDP Flood

attacks a target with UDP packets. This approach drains host resources that can lead to inaccessibility to the server. In NTP Amplification, the attacker attacks NTP servers to flood the targeted with User Datagram Protocol (UDP) traffic. In HTTP Flood[27], the assailant controls HTTP and POST undesirable demands so as to destroy a web server or application. SYN flood is mostly used. In a SYN flood situation, the requester sends many SYN requests and does not respond from the host's SYN-ACK request. In Fig 1.1, the distribution of DDoS attack by type is shown using a pie chart.

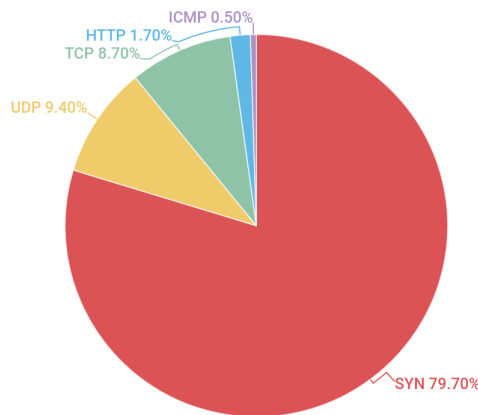


Figure 1.1: Types of DDoS Attack, 2018

GitHub DDoS attack is the world's greatest DDoS attack till now. On 28th February, 2018 this well-known developer platform faced the terrible attack [30]. It is recorded that 1.35 terabits per second traffic hit the platform all at once. Attackers took advantage of the caching system and used a cached approach to attack GitHub. Due to this attack GitHub was unavailable from 17:21 to 17:26 UTC again from 17:26 to 17:30. However, the attack did not last more than 10 minutes. GitHub automatically called a DDoS mitigator service (Akamai Prolexic) for help. It scattered all the traffic coming up to GitHub. Then find out the malicious packet and dropped it off. In Fig 1.2, the percentage of attacks is shown using bar graphs.

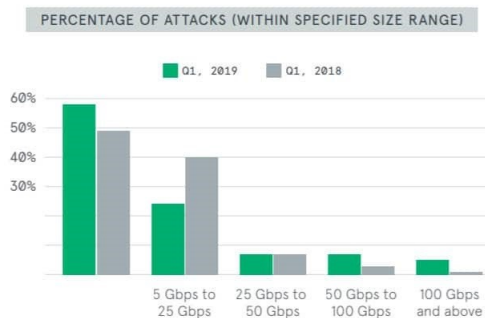


Figure 1.2: Percentage of attack

In short, DDoS attacks distress the normal usage of services. As a result, it slows

down the speed of the server or sometimes it makes the server down. This is hectic for the real users, clients as well as the business.

1.1 Motivation

The Internet has become one of the most basic things in our today's life. Now we are dependent on the internet in every aspect of our life. From online transactions to hotel booking we rely on the internet. Though the IT industry has evolved enormously in the last few decades, it is still not resilient enough. Among many web threats DDoS is the tremendous one. A DDoS attack can make an online platform unreachable within a blink of an eye. A website can lose their confidential data and decrease the productivity of the site. Many companies also lose their reputation on this issue. Moreover, its toxic effect can completely disable a website. Despite much research has been done on this alarming issue yet there is a lot of scope to improve the effectiveness. Rate limiting, active filtering, IP traceback are some renowned techniques to detect DDoS attacks[4]. But the constant evolution of attacks are making it challenging to defend DDoS attacks with these existing techniques. Furthermore, these methods are moderately outdated. We are using a large dataset made up of huge amounts of data to get the best result. We believe, in this approach we will bring the best result.

1.2 Problem Statement

The internet traffic can be connection requests, fake packets or incoming messages. These packets originate from thousands of different computers without sender's concern which makes it difficult to block the incoming source. Culprit sends more packets than the server can accommodate to render the site. Criminals mainly target the high valued web servers like banks, credit card payment gateways and ecommerce sites. For example, a recent study says that more than 50% of the DDoS attack that took place in 2013 to 2014 have targeted the ecommerce site. The main goal of attackers is to make the website unreachable or crushing down.

HTTP Flood and PoD are some of the best known approaches to attack a victim to disrupt their website. In the UDP flood victim's server is attacked by UDP packets and with these overwhelming incoming packets the targeted server stops responding. Ping flood also works as similar as the UDP flood. In DoS attacks it is easier to detect the IP address and block the source. Sadly, in DDoS the request comes from thousands of unknown sources. So it is quite impossible to block all the sources. Thus, in our model we are considering the time period of request and number of requests. If a request takes more time than usual, that request will be evaluated as a fake request. Moreover, if one IP source makes several requests then that will also be added on the list of fake IP. Therefore, our aim is to design an efficient model that has less time complexity with higher accuracy.

1.3 Objectives and Contributions

In this research paper, we intended to design a model to DDoS attack by combining machine learning algorithm and neural network algorithm. In machine learning we are determined to use Logistic Regression, KNN, Random Forest, SVM, Naïve Bayes algorithm etc. Along with these machine learning algorithms we use some classifiers to differentiate DDoS and non-DDoS attacks. Firstly, we collect our dataset from UNB CICDdos2019. This dataset contains lots of data which contains lots of DDoS and non-DDoS attacks together. Then we processed our data to perform in our system. Thereafter, we applied this dataset on our selected algorithm. We applied this dataset in different machine learning algorithms to understand which algorithm provides the finest result.

DDoS attack detection becomes highly obligatory to make clients feel safe about their website. Therefore, our objective is to develop a system which will detect DDoS files in real time.

1.4 Thesis Structure

In Chapter 1, there is Introduction where motivation, problem statement, objectives and contribution are discussed. In chapter2, we reviewed the previous work. Background, Literature Review and Algorithm has been summarized in this part of the paper. In chapter 3, we talked about our proposed model. Here, we discussed our methodology to detect the DDoS Attacks, flask server setup, and real-time implementation. Chapter 4 comes with the Result and Analysis part. In this portion, we showed the outcome of our project. And finally in chapter 5, we concluded the paper with our future plan about this project.

Chapter 2

Algorithms and Related Work

2.1 Background

DoS attack is the origin of DDoS attack. In DoS attacks, perpetrators use a single computer to flood a targeted system whereas, in DDoS traffic floods that come to the victim's systems are from multiple computers and many different sources. For the first time, in 1974 David Dennis, a 13 years old boy performed DDoS attack. This incident took place at CERL at the University of Illinois Urbana-Champaign. Though this child attacker did not have any destructive intention behind this attack yet the world encountered its first DoS attack through this event. David learned about a new command "external" that could be run on CERL's PLATO terminals and the command was for allowing the interaction with external devices that are connected to the other terminals. But if there is no external device to connect then the terminal will block up.

Many years after this incident the DDoS took place, on July 22 1999. One day suddenly 114 infected computers attacked a computer at the University of Minnesota with malicious scripts and made the university computer disable for two days. After that, very soon this unpleasant tactic spread out and many innocent became the victim of it. From Amazon, Yahoo, Github to Dyn many other reputed companies became the victim of it. Among many DDoS attacks some of the major attacks from the initial period are Melissa, Code Red, the Morris Worm, SQL Slammer, Estonia and so on. Melissa affected MS Office documents with a simple mail macro-virus. It also brought down the White House's site. Compared to other DDoS attacks, SQL slammer can be considered very small but it is very much malicious. This malicious attack is not only used in business but also in political issues. In 2011-2012, there was high political pressure in Russia. In that time, both opposition and government sites were DDoSed. In Russia that was the first time cyber-criminal methods were applied in political issues so widely. These malicious packets cause financial loss as well as data loss. Moreover, reputed companies lose their reputation.

Over the years many methods have been developed to prevent DDoS attacks[23], [8]. Some of the well-known techniques are shortly discussed here. Filtering is the very popular one. Filtering is a technique that filters out the IP addresses. Second one is monitoring, it is developed by Cisco monitoring traffic patterns. This popular tool for DDoS attacks is used by ISPs to monitor traffic in both directions on all router

interfaces. Scrubbing is another helpful technique for DDoS prevention. Previously these techniques were used to prevent DDoS in the application layer. But nowadays many advanced methods have been developed for DDoS prevention and in those methods Machine learning and Neural Network algorithms helps to a great extent.

2.1.1 Literature Review

Over the past years, many research work has been done to detect DDoS attacks. In this area of the paper, we will briefly go over some of the previous research work that has been conducted for DDoS detection and prevention.

In paper [13], the researcher applied a different machine learning algorithm in the cloud computing environment and made a comparison between the obtained results. Here, they developed a DDoS detection model using C.4.5 algorithm along with that author used signature detection techniques to identify DDoS attacks. This method will generate a decision tree which will perform automatically to detect signature attack when a server will be flooded by DDoS attack.

In paper[17] the author proposed a plan for the traffic generated in HTTP and TCP. They applied the SVM algorithm, K-NN and predicted the accuracy. As SVM is better than K-NN so he applied the SVM algorithm using python to detect it. The author used a limited amount of data for training purposes, then applied the SVM algorithms in the entire dataset. This algorithm will perform as normal and abnormal binary classification to differentiate DDoS and non-DdoS packets.

To rescue the government and different corporations from DDoS attack, paper [16] determined to develop a new method using DBSCAN clustering algorithms. The three phases of this method are analyzing phase, clustering phase and prevention phase. Considering the DDOS attack type the dataset can be created. The attack can be blocked in 15-20 sec by creating some firewall rules according to the features.

In [25], the author proposed an efficient solution against the distressing cyber threat Distributed Denial-of-Service (DDoS) attacks in the SDN environment. Two classification algorithms, SVM and SOM have been used here. Support Vector Machines provide high accuracy but the flaw it has is taking more time. On the other hand, the Self Organizing Map (SOM) makes a reliable prediction based on their neurons to minimize the resource consumption. Results show that this approach achieves 99.27% success rate for attack detection and 99.30% for accuracy. Despite the fact that this model gets very high accuracy but the false alarm rate and legal packet drop is still a concerning factor. This paper [18], aims to analyze the effectiveness between two algorithms SVM and deep feed forward (DFF). To conduct this estimation the author used DARPA 2009 and DARPA scalable Network Monitoring DDos attack dataset.

In paper [14], the author gave an overview of different DDoS attacks that occurred in cloud environments and in application layers. They write down all potential DDOS attacks and the methods used in the cloud to detect and prevent DDoS attacks. And researcher also talked about anomaly-based detection, signature-based

detection, network-based approach as well as host-based approach for cloud environment. In addition to this, the researcher talk about Dynamic detection method, Packet percentage and queue caching method, Shrewd sending rates and Randomising retransmission timeout for attacks in application layer.

For cloud environments a hybrid model has also been introduced in paper [5], based on the entropy and covariance matrices. Both of the systems are alike in heuristically as the two classify DDoS attacks by estimating the heightened dependency in the data. In paper [15], the author analyzes different artificial intelligence techniques to detect and prevent DDoS attacks. Researchers review the DeepDefense approach which is a deep learning based DDoS detection approach. It is a multi-layer approach which has a number of defensive mechanisms to protect the server. This approach is mainly built with CNN and RNN algorithms. Along with this, the author discussed ANN[21], Random forest tree, Naive bayes. And finally suggested that Random forest tree and Naive Bayes is superior in performance measurement.

2.2 Algorithms

In our proposed model, we are using a supervised model which represents all the features have been labeled, in this case. We have to classify the information that has been provided in the dataset. For the classification and regression, we are using different machine learning algorithms[22], [19]. Because machine learning is the most satisfying solution for any kind of model for that machine learning approaches are being used in many industrial sectors nowadays[26]. Keeping these aspects in mind here, we have used different algorithms to cluster the supervised learning. The description of the algorithms explained below:

2.2.1 K-Nearest Neighbor (KNN)

It is very easy to implement a supervised learning model for applying KNN machine learning algorithms.

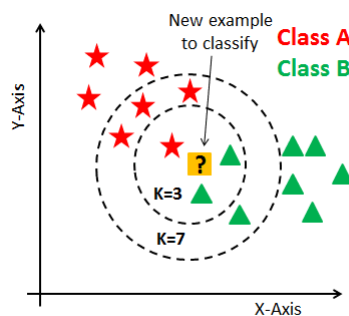


Figure 2.1: KNN Algorithm

KNN performs very well in these terms remembering these parameters we used KNN algorithm to get the satisfying output and less calculation time. Now we have to know about the implementation processes to execute the KNN algorithm.

First of all, we have to import all the information as data. Then we need to choose the parameter. 'K' is the nearest neighbour in KNN algorithm. Here initializing the 'K' is very challenging. Because the value of 'K' in the algorithm influences the result. So we have to initialize the value of 'K' properly. To get the estimated class, we need to iterate from 1 to sum up the total number of training data points. After that we need to calculate the distance between test data and each row of training details. Here, we use Euclidean distance as our interval metric because it gives the most satisfied outcome. We can also use other matrix techniques such as Chebyshev, Cosine etc.

Now we need to sort the calculated data using ascending order with respect to the distance values. After that we can get the top K rows in this ascending sorted array. Obtain the most periodic class of these rows. As shown in Fig 2.1, we need to return the estimated class again.

Therefore, we can say that the KNN algorithm is the most uncomplicated approach for any supervised learning model.

2.2.2 Random Forest (RF)

Random Forest algorithm is a very flexible algorithm for classification and regression which is widely used in many machine learning models. It is conducted by building up multiple decision trees and accommodating those decision trees and predicting more accurate outcomes.

Working of Random forest algorithm:

First of all, the algorithm chooses some random illustrative from the given dataset. Secondly, the algorithm will build a decision tree for every illustrative by using an aggregating classifier(hyperparameter)

After that, the algorithm will obtain the prediction from every decision tree.

Then, in this step, the decision trees will be accumulated and predict more accurate results.

First of all, we have to know how random forest algorithm works. Random Forest algorithm is the procedure of searching the root node and deassociating the feature nodes will run randomly. Random forest algorithm is processed in two phases. In the beginning it constructed the forest by obtaining the data. The other phase is assemble the prediction from the constructed forest. In the period of building up the decision trees, random forests choose closely unchanged hyperparameters as decision trees which is also called aggregating classifier.

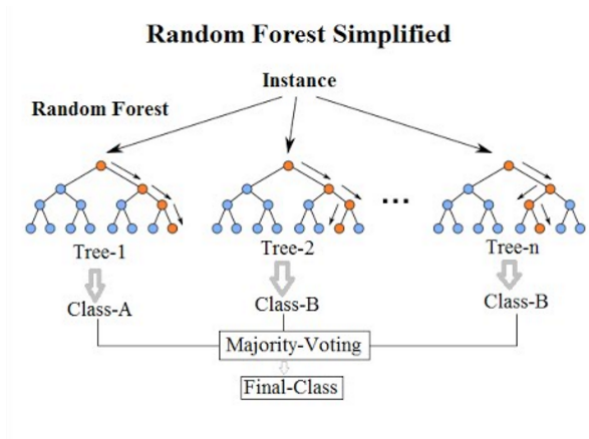


Figure 2.2: RF Algorithm

Here, we apply a random forest algorithm on the dataset of DDoS attack prevention model where, the dataset is supervised learning model. Generally, when we apply any algorithm on dataset of any machine learning problem the most serious issue is overfitting the result. Nevertheless, random forest can be used with some benefits. The advantages of applying random forest algorithm are [11]:

2.2.3 Naive Bayesian

Naive bayes algorithm can be used in only classification problems. when we use any dataset of numerous number of datas with a couple numbers of tuples, it will be the wisest decision to apply naive bayes algorithm. Naive bayes model is simple to construct and principally convenient for extensive records of dataset. naive bayes model is popular for defeating vastly complicated tasks. [10]

As Naive Bayes algorithm constructs based on bayes theorem, we need to know about bayes theorem. Bayes theorem serves toward calculating subsequent probability $P(A|B)$ from $P(A)$, $P(B)$ and $P(B|A)$. Formula of the Bayes theorem:

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

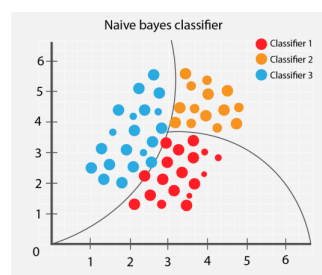


Figure 2.3: Naive bayes Algorithm

Naive Bayes algorithm is actualized into different steps. The steps are:

Load the information of the dataset into a frequency table. After getting multiple classes, the equation of subsequent probability of the class given predictor is used to compute the posterior probability of each class. As shown in Fig 2.3, the uppermost

posterior probability is the result of the prediction. Moreover, Naive Bayes algorithm can be applied on multi class prediction. This is one of the most important criteria of the Naive Bayes algorithm. By applying this algorithm, we can get the probability of multiple classes of targeted volatile. In contrast, most of the time, Naive Bayes algorithm works on small dataset. It hardly fit in large record of dataset.

Therefore, we have used the Naive Bayes algorithm in our supervised learning model for expecting the convenience outcome. Naive Bayes algorithm is very simple and gives a very spontaneous outcome as prediction of the test class of the dataset as well as this algorithm accomplishes well in multi class prediction. The performance of the Naive Bayes algorithm is very fast and well in the function of unambiguous input variables.

2.2.4 Support Vector Machine (SVM)

SVM algorithm performs well in analyzing information as data. Mainly, SVM algorithm is widely used in machine learning for classification problems.[12] The purpose of SVM algorithm is to generate the best determination edge which is supposed to discriminate the n-dimensional spaces into classes. This algorithm is good for text categorization [1] and face recognition[2]. The goal of this discrimination is that the new data point can be put into the proper category area in further. Here, choosing the best determination edge is known as hyperplane. If n is the dimensional vector space ,n-1 will be the subspace of the hyperplane. For constructing the hyperplane, the SVM determine the acute vectors and these determining the vector points are called support vectors[28]

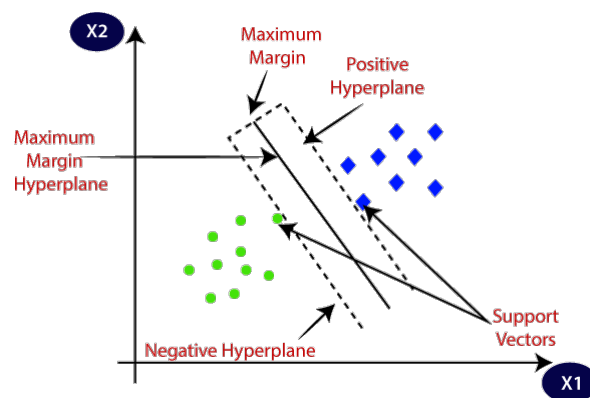


Figure 2.4: SVM Algorithm

There are two categories of SVM algorithm:
 First one is Linear SVM algorithm:

In linearly arranged data where dataset has two type of feature only(x_1 , x_2) then these features can be separated by one straight line.The SVM helps to choose hyper-plane and then finds the nearest points of hyperplane from both of the features.These points are called support vectors. Shown in Fig 2.4.

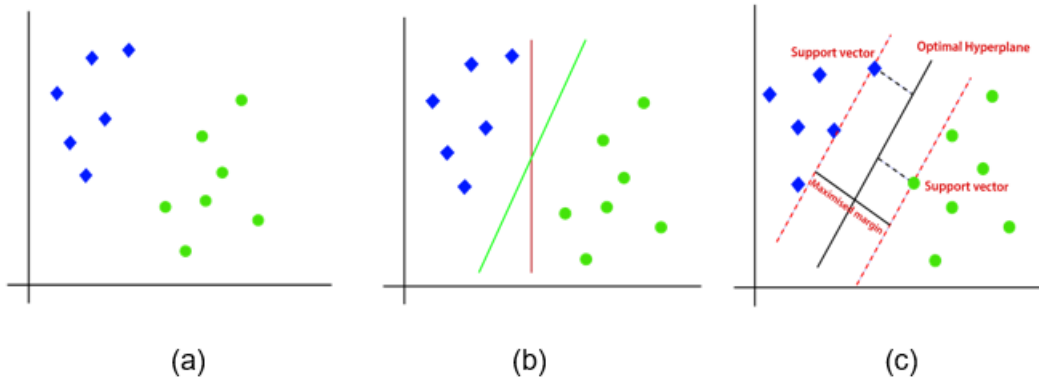


Figure 2.5: Linear SVM Algorithm

Another one is Non Linear SVM algorithm:

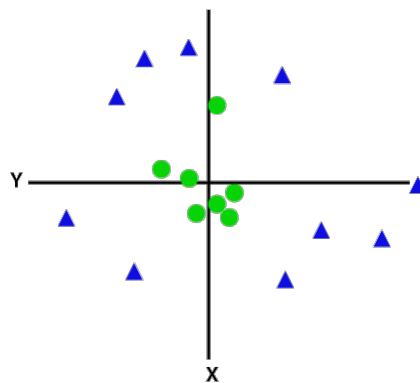


Figure 2.6: Non-Linear SVM Algorithm

$$z = x^2 + y^2 \quad (2.1)$$

In Fig 2.6, non-linearly arranged data,there are multiple features in the dataset.So,one straight line can not separate the features.For separating the features ,the data will be represented as three dimension.

2.2.5 AdaBoost

It also known as Adaptive Boosting. Boosting is a group of machine learning algorithms where multiple weak classifiers together turn into a strong classifier. In Adaptive Boosting multiple classifiers also called weak learners, combine together and form a unique classifier. This algorithm is to construct strong classifier by the supposition of many weak classifiers. AdaBoost can improve the performance of any machine learning algorithm and achieve the best accuracy over any other classifier.

The performance of the Adaboost can be improved based on three hyperparameters and they are the number of estimators, maximum number of splits and learning rate. One of the important features of the Adaboost algorithm is that it is very responsive to the noise label. Moreover, Adaboost can easily regulate adaptively the errors of the weak hypotheses by the weak classifiers. [32] This algorithm puts more weight in those training data that is hard to classify whereas easily manageable instances get less weight. Thus, AdaBoost makes predictions about weak classifiers. In the training process AdaBoost decreases the number of features by cutting off insignificant features and selects only the important features which can enhance the predictive power of a model. The working process of AdaBoost is briefly described in the following steps –

First, it generates its training data and puts initial weight. After that, every decision projection produces one decision input variable and that results in a +1.0 or -1.0 value for the value of first or second class. Then, this algorithm calculates error for the trained model. The equation to generate the error is:

$$error = (correct - N)/N \tag{2.2}$$

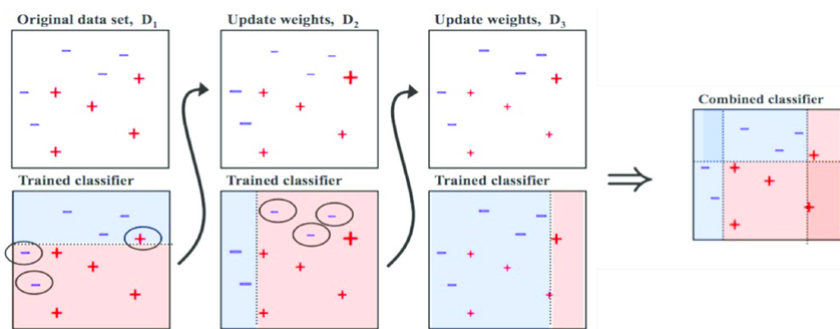


Figure 2.7: AdaBoost Algorithm

Then, shown in Fig 2.7, the weight update again. The weight goes up or down for correctly or incorrectly classification. And this process continued for several times to bring out the best classifier.

2.2.6 XGBoost

XGBoost is another highly efficient and flexible Boosting Algorithm. This algorithm [6] is designed with machine learning algorithms in the Gradient Boosting framework. It is a utilization of gradient boosted decision tree that is good at solving various data science problems with high accuracy. It is basically an open source library that presents a gradient boosting framework for different programming languages.

Fig 2.8, The work process of XGBoost is discussed in the following few steps:-

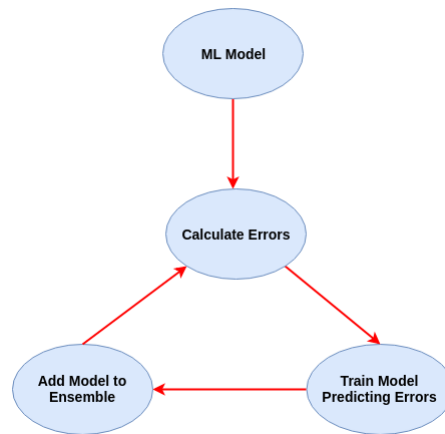


Figure 2.8: XGBoost Algorithm

We can use XGBoost for mainly two purposes[20]:

- i. Calculate the implementation Speed.
- ii. Measure the performance of the model.

2.2.7 Decision Tree

Decision tree is similar to a tree structured where the node of internal produces a “test” on the attribute. Decision tree contains two types of nodes, one is internal nodes which produces to an attribute and every sections presents the result of the test and every leaf node produces a class label. In the tree the path which is coming from the root to the leaf node represent classification processes. Decision tree is more likely to the supervised learning model. Moreover, it produces excellent outcome for solving the classification and regression problems.

Fig 2.9, The Decision Tree can be generated by the following steps:

Initially, we have to put the best tuples of the dataset at the base of the tree. At that point, we will part the preparation set into subsets. Here, the subsets are made in the process where each subset comprises of the information with the comparable incentive for a trait. Reiterate process 1 and procedure 2 on each subset until the leaf nodes are found in each part of the tree. Lastly stage 1 and 2 will be on every subset until you discover leaf hubs in all the parts of the tree.

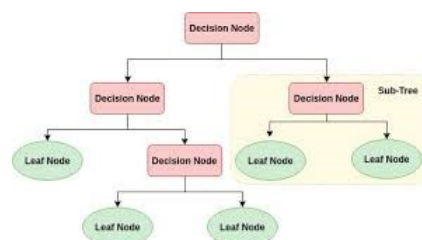


Figure 2.9: Decision Tree Algorithm

2.2.8 Logistic Regression

It is a model which consists of decision rules that anticipated the probabilities of the result[7]. According to this way, Logistic regression can be used as a regression model as it predicts the of the class membership in the way of multi linear function of the feature. In the regression cases,it performs well in finding out the continuous dependent variables.

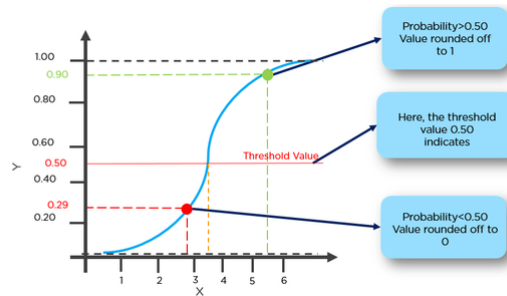


Figure 2.10: Logistic Regression Algorithm

$$y = \frac{e^{(b_0 + b_1 \cdot x)}}{1 + e^{(b_0 + b_1 \cdot x)}}$$

Here,

y= output values after prediction

x=input values

Chapter 3

Our Proposed Model

3.1 Dataset description

3.1.1 Dataset

Dataset is the assembly of data with related attributes which can be discrete that is retrieved independently or as in consolidation. Dataset is very efficient part for any machine learning approach. It is very challenging to make the dataset appropriate for machine learning method. There are two types of dataset. One is Training dataset which is used to train machine learning algorithm for different activations and another one is Testing dataset which is used to legitimize the accuracy of the proposed model. The dataset can be collected from websites or can be constructed by accommodating data. The sources are kaggle CIC (Canadian Institute for Cyber security) and so on. Our dataset is collected from CIC (Canadian Institute for Cyber security). To train the data we used a dataset. We collected the dataset from CIC. There were 87 Traffic Features in the dataset. Each feature has its own unique value and importance. Here is a brief description of the dataset below.

3.1.2 Dataset Feature Description

To train the data we used a dataset. We collected the dataset from the CIC 2019 DDoS Evaluation Dataset. There were 87 Traffic Features in the dataset. Each feature has its own unique value and importance. Here is a brief description of the dataset below.

To talk about the selected features we have selected the source port and the destination port. They are the port number of the source device and the destination feature respectively. The duration of the flow. The number of flow packet per second and the number of flow bytes per second. Total forward packets and backward packets which indicates the number of packets which is forward and backward. After that the length of the forward and backward packets. Forward Packet length max and backward packet length max which indicates the maximum size of packets in forward direction and the minimum size of the packets in backward direction. The mean value of the forward packet length. Forward packet length std or the standard deviation of the forward packet length. Just like the forward length of the packet we have also selected the maximum length of the backward packet, minimum length

of the backward packet, mean length of the backward length.

Flow IAT means the mean value between two flows. Flow IST Std is the standard deviation between two flows. Flow IAT Max shows the maximum time between two flows. Flow IAT min is the minimum time between two flows. Fwd IAT total time between two packets when the packets are sent forward. Fwd IAT mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min is the mean value, standard deviation, Maximum number of packets, Minimum time between two flows. Similarly, Backward IAT Total is total time between two flows sent in the backward position. Bwd IAT std is standard deviation time between two flows sent in backward position. Bwd IAT Max and Min is the maximum and minimum amount of time between two values sent in backward direction.

Forward and backward push flag is the number of times the push flag is set in forward and backward position. Forward URG Flag and Backward URG Flag is the number of times this flag is set while moving forward and backward. Fwd Header Length and Bwd header length indicates the bytes of the header moving forward and backward. Forward packets is number of forward packet per second and backward packet is number of backward packets per second. Minimum packet length and maximum packet length indicates the minimum and maximum length of a flow Download and upload ratio is indicated in the down up ratio column. Forward and Backward byte bulk rate average calculates the average number of byte bulk rate that is directed forward and backward. Forward and Backward packet bulk rate average calculates the average number of packet bulk rate that is directed forward and backward. Forward and Backward bulk rate average calculates the average number of packet rates that are directed forward and backward. Sub flow forward and backward packet calculates the average number of sub flow that is directed forward and backward. Forward and backward window bytes is the total number of bytes in the initial window that is sent forward and backward. Active average indicates before becoming idle the mean time the flow was active. Active Standard deviation indicates before becoming idle the standard deviation time a flow was active.

3.2 Proposed Prediction Model

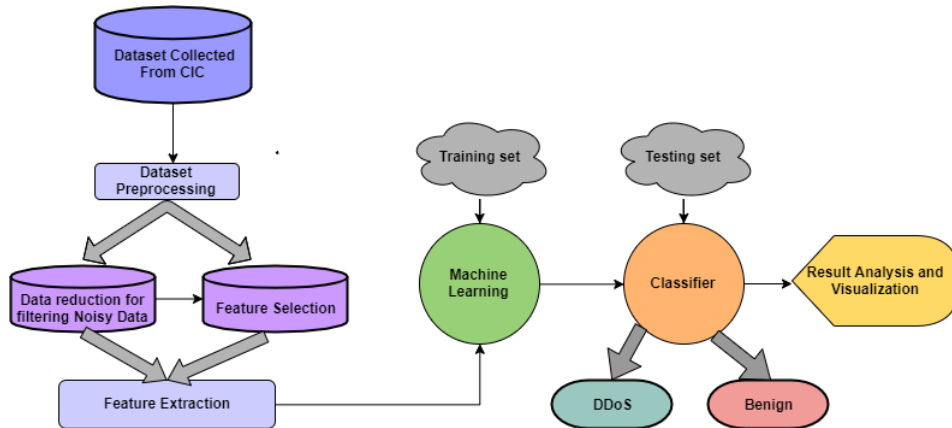


Figure 3.1: Workflow Diagram of Prediction model

The above Fig:3.1 diagram briefly represents the approach of our work to identify DDoS and non-DDoS. First, we took a large dataset for our model to distinguish between DDoS and non-DDoS which was taken from CIC 2019 DDoS. Then we preprocess our data and remove infinity and noisy data from our dataset. We used standard scaler for the standardization of our dataset. After that we did feature selection using WEKA. We used various techniques which are discussed broadly in the feature selection topic. Finally, we put our preprocessed data into the various machine learning classifiers. We got the results from there which have been discussed in the result analysis part.

3.2.1 Data preprocessing

Without preprocessing the raw data, we can not build a good model. Normally, we can divide our data preprocessing into some steps. They are:

Removing Noisy and Missing values: Firstly, removing noisy and missing data. A raw dataset may have some noisy data and missing values. Noisy data are value-less data. Firstly we remove those noisy and missing data. There were also some infinity/nan values in our raw dataset. We also resolved the problem.

Removing Duplicate values: In the raw dataset, there were a few duplicate values. We removed those duplicated values from the dataset.

Data Standardization: Data standardization means modifying the data. There are many methods available. Here, we used standardscaler for standardization. That means we set a range for the data. We implemented it using sklearn.

Table 3.1: Before data preprocessing with StandardScaler

	Source Port	Destination Port	Protocol	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets
0	443	50458	6	2	0	0
1	443	50465	6	2	0	0
2	0	0	0	52	0	0
3	56085	80	6	9	8	8
4	56131	443	6	29	30	5844

Here, we used sklearn to generate a standard scaler for standardization before data preprocessing which is shown in Table 3.1

After that we used sklearn preprocessing library to preprocess the data which is shown in the below table 3.2

Table 3.2 : After data preprocessing with StandardScaler

	Source Port	Destination Port	Protocol	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets
0	[0.96796225 -0.26229	-0.09915471	-0.08643441	-0.50730514	-0.02403028]	
1	[0.96829084 -0.26229	-0.09915471	-0.08643441	-0.50730514	-0.02403028]	
2	[-1.4005851 -0.26285116	0.04173247	-0.08643441	-0.50730514	-0.02403028]	
3	[1.12160004 -0.04970369	-0.05688856	0.19954211	-0.50730514	0.00451493]	
4	[0.73236553 3.01629034	-0.05688856	0.19954211	-0.50730514	0.00451493]	

3.2.2 Feature Selection

There are a number of tools and ways for data mining and preprocessing. In our case we have used WEKA in order to process our dataset. WEKA [33] is considered as the best software for data mining. Generally, WEKA is used for data mining. It has some built in machine learning algorithms. Some of the main features of WEKA are data mining, preprocessing, machine learning, clustering, classification, attribute selection, regression. We have used it for preprocessing our data.

The dataset we have used is a very large file. Initially it has 84 features. There was 10 lac data in the dataset. Among that 10 lac 40% were Benign data and the rest of them were DDoS data. There are many types of DDoS attack. We have worked with several types of DDoS attack such as UDP flood, NTP amplification, SYN flood, HTTP flood, ICMP (ping) flood.

Our target was to find the most important features for that attack. We were also concerned about the amount of features. We tried to keep a moderate number of

tuples. At first we used a filter for preprocessing. Using filters gives only 6 features which is too less. Later we found out that there were some problems with the dataset. After solving that I was giving 11 features. Which was not fulfilling our expectation. After that we used the very famous Principle Components Analysis (PCA) method[9]. This time we got 24 features which were not too low nor so high.

For using PCA we select the select attribute option. Then we select the Attribute Evaluator as Principal Component Analysis (PCA). After that we select the Ranker search method. We used the full training set for this. After executing the task it gave us 24 important features. After selecting the features we used the classifier option and selected the logistic algorithm for a test run.

We have also used some other techniques like wrapper method, Attribute selection with Naïve Bayes and Logistic algorithm. We have used ten fold cross validation as well. Comparing the result we found out that PCA is giving the best result. Lastly, we decided to take the features we got from PCA.

There were some difficulties as well. We have faced different problems like NaN values, Infinity Values. We have removed all the NaN values and the infinity values. Besides, the dataset was too big so there were a huge number of duplicate values. The duplicate values were affecting our result. Along with the Infinity values we removed the Duplicate values as well. It took a lot of time for us to process the dataset. The excel software was crashing frequently because of that large file and Weka was also taking more time to calculate the values.

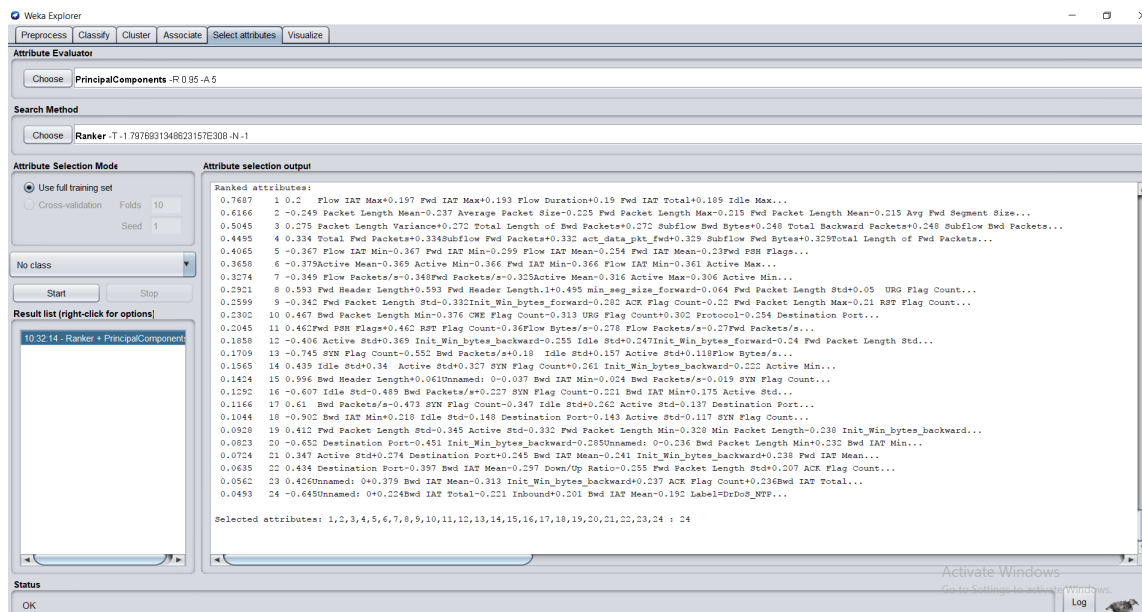


Figure 3.2: PCA for selecting features using WEKA

3.2.3 Selected Features

In Table 3.3, we have discussed the description of selected features.

Table 3.1: Selected Features Description

Serial No.	Name of tuple	Description
1	Source Port	Port number of the source
2	Destination Port	Port number of the destination
3	Protocol	Protocol
4	Total Fwd Packet	Number of packets directed to forward
5	Total backward Packet	Number of packets directed to backwards
6	Total length of Fwd Packet	All packets that are directed to forward
7	Total length of backward Packet	All packets that are directed to backward
8	Fwd packet length Max	Maximum size of the packets that are directed to forward
9	Fwd packet length Min	Maximum size of the packets that are directed to backward
10	Fwd packet length Mean	Mean size of the packets that are directed to forward
11	Fwd packet length Std	In forward direction the Standard deviation
12	Bwd packet length Max	Maximum size of the packets that are directed to Backward
13	Bwd packet length Min	Minimum size of the packets that are directed to Backward
14	Bwd packet length Mean	Mean size of the packets that are directed to Backward
15	Bwd packet length STD	In Backward direction the Standard deviation
16	Flow Bytes/s	The number of that bytes are transferred in a second is Flow Bytes/s
17	Flow packets/s	The number of packets that are transferred in a second is Flow Packets/s

Table 3.2: Selected Features Description

Serial No.	Name of tuple	Description
18	Flow IAT Mean	Total mean time between two flows
19	Flow IAT STD	It calculates between two flow what is the Standard deviation time
20	Flow IAT Max	It calculates between two flow what is the Maximum time
21	Flow IAT Min	It calculates between two flow what is the Minimum time
22	Fwd IAT Total	In forward direction what is the total time between two packets
23	Fwd IAT STD	In forward direction what is Standard deviation time between two packets

Our dataset is mainly a customized dataset. From the actual dataset, we have merged Benign data, DDoS NTP, DDoS DNS and DDoS UDP data into the customized dataset. It is an imbalanced dataset.

3.2.4 Training or Testing machine learning classifier

We started to train and test the machine learning classifiers after completing the feature selection process. For that, we split our dataset into two sets which are training dataset and testing dataset. To train and test the dataset we used some machine learning classification algorithms. To train the model training data is used. It makes the model fit for the operation. Testing data is used to test the machine and test the models capability of detecting new attacks and give new results. We have used forty percent data for testing and sixty percent data for training. To train our model we have used eight classification algorithms they are:

- Logistic Regression
- K Nearest Neighbor (KNN)
- Random Forest
- AdaBoost
- Support Vector Machine (SVM)
- Decision Tree
- XGboost
- Naïve Bayesian

3.3 Data visualization

Depending on the outcome they were visualized. Data visualization refers to graphical representation of data. It shows us the relationships among the given variables or tuples. Here, we used python libraries for data visualization. Some of the very common and most used libraries of python for data visualization are Matplotlib, Seaborn and Scikit Plot. We have generated Heat map, Bar Graph, Correlation Matrix, Count Plot, Pie Plot and Scatter plot.

3.3.1 Heat map

Heatmap is a very powerful tool to visualize data. It is a colorful matrix which contains many cells. Each of the cells in the matrix represent data intensity. It has a color scale to measure the intensity. Darker color represents the low intensity and darker color higher intensity. In order to generate the heatmap of the dataset we used the seaborn library of python to visualize it. For generating the Heat map of all characteristic variables where these characteristic variables are used as column headers and row headers. Heat map map can be produced by using software or using python code. Here, we used python code to generate heat map for our supervised learning dataset. After generating the heat map, shown in Figure 3.3, we get a correlation of these characteristic variable in an illustrated visualization as high dimensional 2D space.

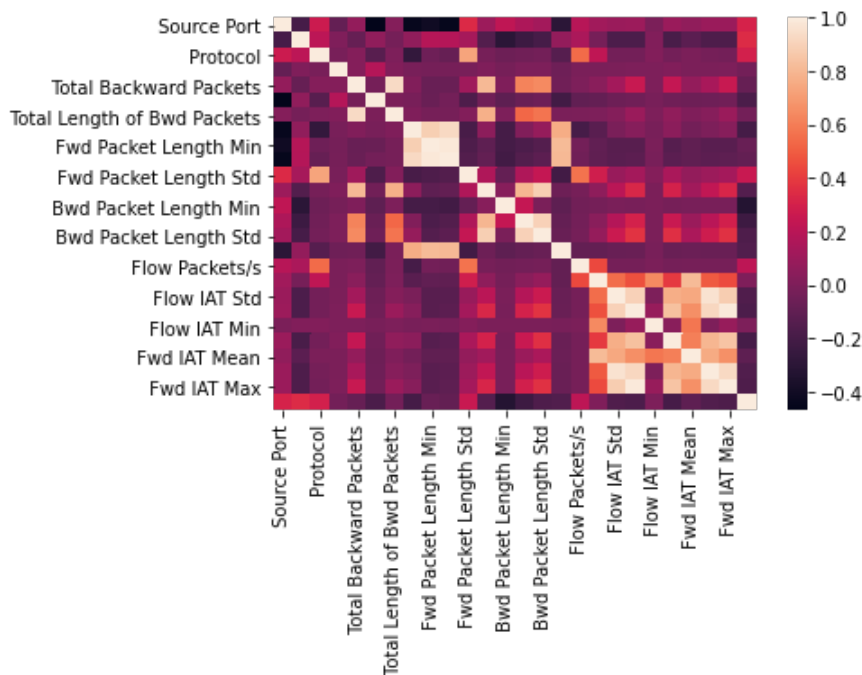


Figure 3.3: Heat map

3.3.2 Bar Graph

Bar graph is a data visualization technique that is shown in Figure 3.4 by rectangular bars. Bar graphs are mainly used for making comparisons between attributes. Bar graph is constructed by using two axes(X-axis,Y-axis).

Bar graph represents four aspects:[29]

- i.A bar graph simply presents the comparison of data series between different attributes.
- ii.The Bar graph visualizes the classification on one axis and a discontinuous value.
- iii.Bar graph illustrates the interrelationship between two axes.
- iv.Based on the time Bar graph can visualize the changes in data.

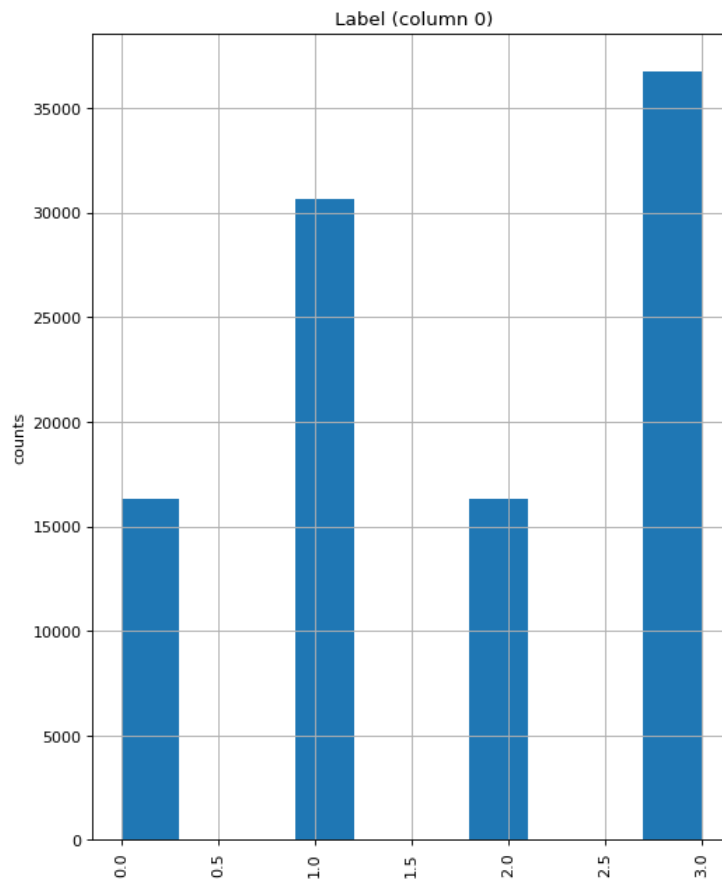


Figure 3.4: Bar graph

3.3.3 Correlation Matrix

Correlation matrix is another visualization technique of representing the data series. Here, correlation represents the relation of the changes between two attributes. If the two attributes change in the opposite direction, then the attributes will be correlated negatively. On the other hand, if the attributes change in the same direction, then the correlation between two attributes will be positive. By generating, correlation matrix, the correlation between the attributes can be visualized. Here, we have generated correlation matrix plot for our model using python program.

In this correlation matrix plot which is shown in Figure 3.5, the plot is represented symmetrically. Here, the left bottom value of the matrix is the same as the top right point. The right bottom point and the top left point represent the relationship of the attributes diagonally.

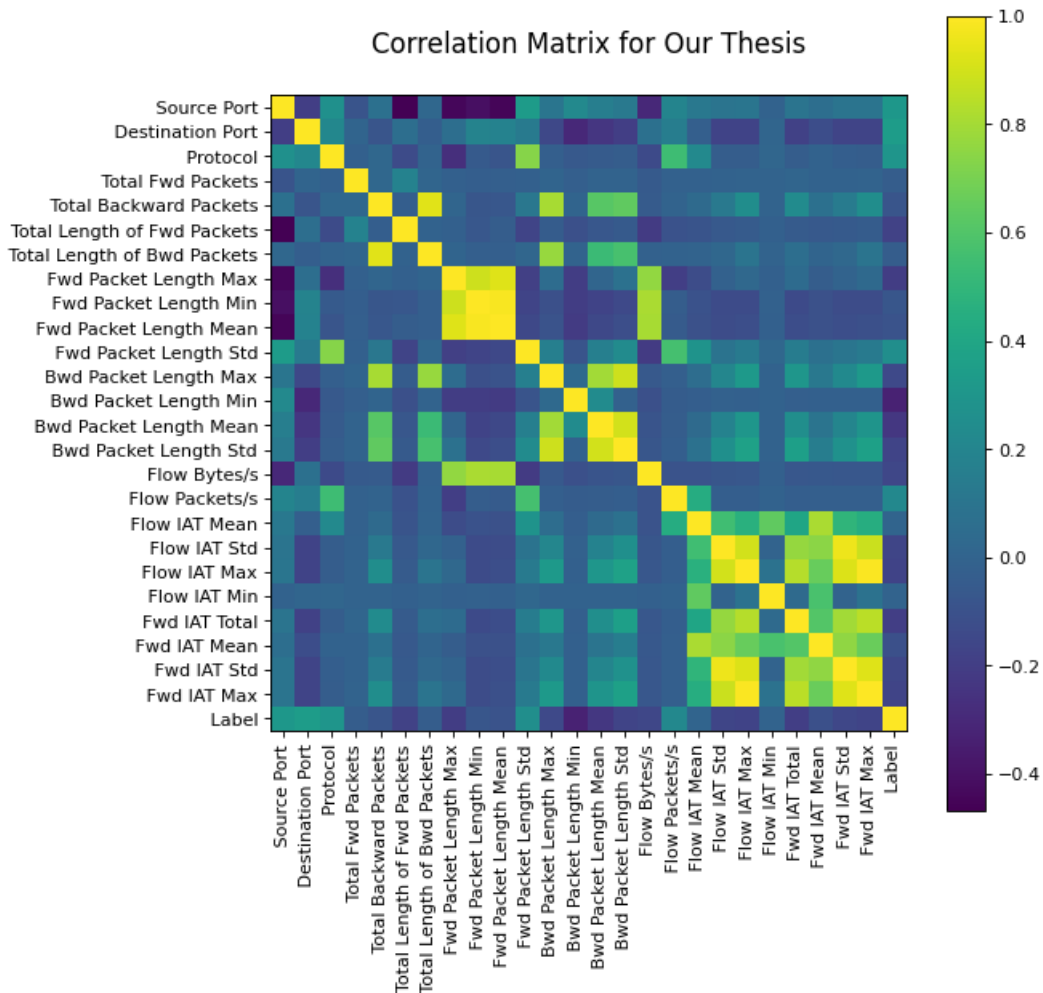


Figure 3.5: Correlation Matrix

3.3.4 Count Plot

Count plot is another representation of data. This plot is quite similar as the histogram which works in particular area and demonstrates the number of occurrences of same component on a particular feature. Here, we have built a count plot on the certain category named 'Label' using python. In the dataset there are four type of items in the "label" feature named 'BENIGN', 'DDoS-DNS', 'DDoS-UDP' and 'DDoS-NTP'. The count plot count the number of occurrence of 'BENIGN', 'DDoS-DNS', 'DDoS-UDP' and 'DDoS-NTP' and then illustrate the plot. Using this plot which is shown in Figure 3.6 ,at a glance we can see which item occurs how many times.

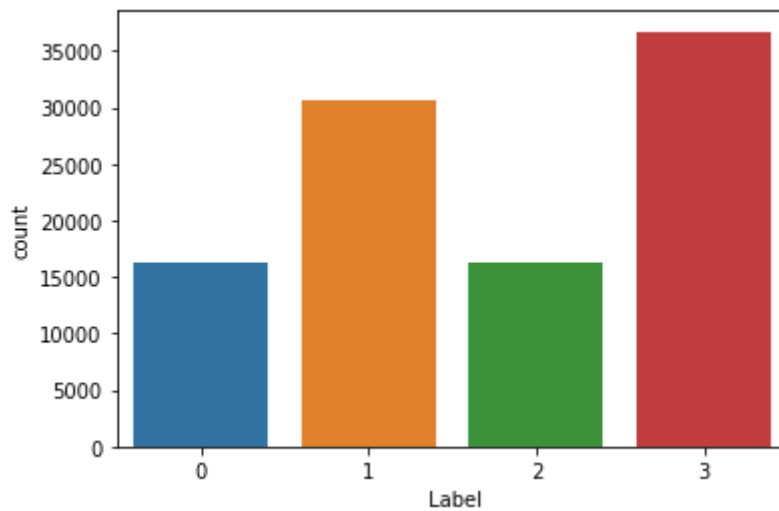


Figure 3.6: Count Plot

3.3.5 Pie Plot

Pie plot is one kind of histogram presentation of data. In pie chart, there are two or more slices constructed the pie plot where, each slice represents each type of data. The slice of the each is represented by percentage. Here, in our model, we have constructed the pie plot based on the 'Label' attribute where it gives four type of data, 'BENIGN', 'DDoS-DNS', 'DDoS-UDP' and 'DDoS-NTP'. With the help of the pie plot, they are illustrated statically in the below Figure 3.7:

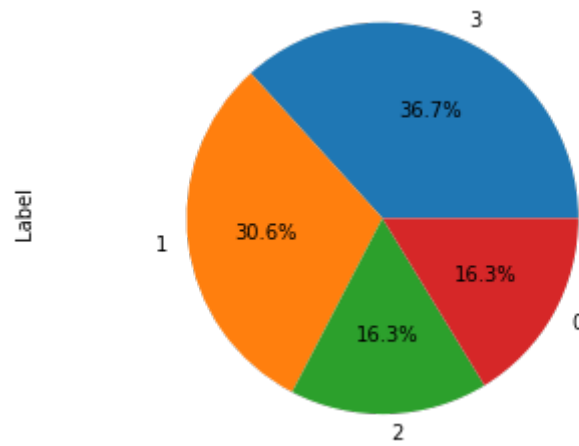


Figure 3.7: Pie Plot

Chapter 4

Result Analysis and Real Time Implementation

To work with any dataset, we need to analyze it properly to get a very good outcome. We have generated different calculation matrices such as ROC curve, accuracy result, confusion matrix, Classification report. Here, we have applied nine machine learning classification algorithms to detect DDoS and Benign. After preprocessing our dataset we get some selected features. Then, we split the data into training set and testing set. The analyzed matrices are discussed below:

Confusion Matrix:

Confusion matrix is a kind of matrix that is often used in Machine learning to evaluate the performance of algorithms. It summarizes the total correct and incorrect value that is predicted by the machine learning algorithms.

Confusion Matrix for Binary class

	<i>Class 1 Predicted</i>	<i>Class 2 Predicted</i>
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Confusion Matrix for Four classes

	PREDICTED				
	1	2	3	4	
ACTUAL	1	TP ₁	E ₁₂	E ₁₃	E ₁₄
	2	E ₂₁	TP ₂	E ₂₃	E ₂₄
	3	E ₃₁	E ₃₂	TP ₃	E ₃₄
	4	E ₄₁	E ₄₂	E ₄₃	TP ₄

Details of the confusion matrix,

- **Positive (P)** : Actually positive
- **Negative (N)** : Actually is not positive
- **True Positive (TP)** : Actually positive and predicted as positive.
- **False Negative (FN)** : Actually positive but predicted as negative.
- **True Negative (TN)** : Actually negative, and is predicted as negative.
- **False Positive (FP)** : Actually negative but is predicted as positive.
- E_{12} : It indicates error, It is actually 1 but classified as 2. Same process for the rest.

- **True Positive Rate (T.P.R):**

We know to calculate TPR, we have to divide TP(True Positive) with the Positives(TP+FN). So,

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **False Positive Rate(F.P.R):**

To calculate FPR, we will divide FP(False Positive) with Positives(TP+FN). So it becomes,

$$\text{FPR} = \frac{\text{FP}}{\text{TP} + \text{FN}}$$

- **Accuracy Score:**

Accuracy score is a very common way to evaluate a model. Here, in our research, we used a python sklearn library to find the accuracy of various models. The general formula to find the accuracy of a model is given below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Recall:**

It is the fraction of correctly predicted positive classification to the total predicted actually positive sample.

$$\mathbf{Recall} = \frac{TP}{TP + FN}$$

- **Precision:**

It is the proportion of actual positive classification from the cases that is predicted as positives.

$$\mathbf{Precision} = \frac{TP}{TP + FP}$$

- **F-measure:**

It is a single metric that combines recall and precision using the harmonic mean. The formula of F-measure is given below:

$$\mathbf{F - measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

- **Support:**The support is the quantity of tests of the genuine example that is in that class.
- **Micro average:** We can get micro average by averaging the total true positives, false negatives and false positives samples.
- **Macro average:** We can calculate macro average by averaging the metrics for each label, and finding their unweighted mean.
- **Weighted average:** We calculate weighted average by the metrics for each label and find their average weighted by support.
- **Receiver Operating Characteristic (ROC Curve):** To evaluate any machine learning model, ROC curve is one of the best options. The full form of ROC is Receiver Operating Characteristic. ROC is mainly a probability curve. It is the plot of the true positive rate(T.P.R) against the false positive rate(F.P.R).
- **Area under the curve (AUC):** It is a measurement to ascertain the general execution of a characterization model dependent on a region under the ROC curve.

4.1 Result Analysis

4.1.1 Logistic Regression

For logistic regression we got the accuracy score 0.91745.

The confusion matrix for logistic regression is as follows:

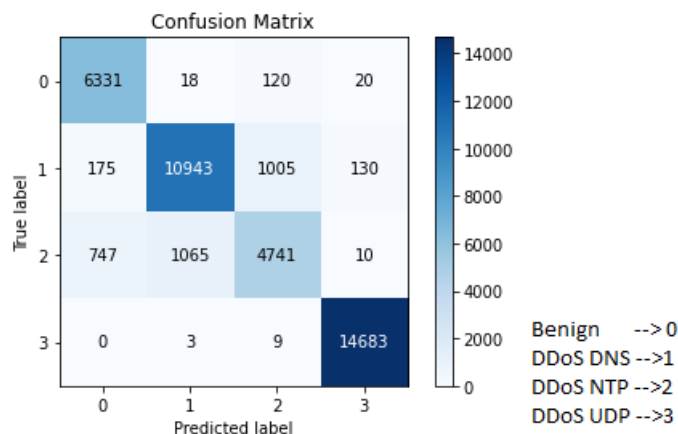


Figure 4.1: Confusion Matrix of Logistic Regression

Here, It is the classification matrix of logistic regression. It is the confusion matrix for four classes. Here, 0 denotes benign data, 1 means DDoS DNS, 2 denotes DDoS NTP and 3 denotes DDoS UDP. We can see True Positives(TP) are diagonally related. However, $TP_0 = 6331$, $TP_1 = 10943$, $TP_2 = 4741$ and $TP_3 = 14683$.

The classification report is given below:

For Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.87	0.97	0.92	6639
1	0.85	0.92	0.88	12173
2	0.84	0.57	0.68	6514
3	0.98	1.00	0.99	14674
accuracy			0.90	40000
macro avg	0.89	0.87	0.87	40000
weighted avg	0.90	0.90	0.90	40000

Figure 4.2: Classification report of Logistic Regression

As we can see from the classification report from Fig 4.2, for benign precision, recall and F1 score are .87, .97 and .92 respectively. For DDoS DNS precision, recall and F1 score are .85, .92 and .88 respectively. Now DDoS NTP precision, recall and F1 score are .84, .57 and .68 respectively. For DDoS UDP precision, recall and F1 score are .98, 1 and .99 respectively.

The ROC curve for logistic regression is as follows:

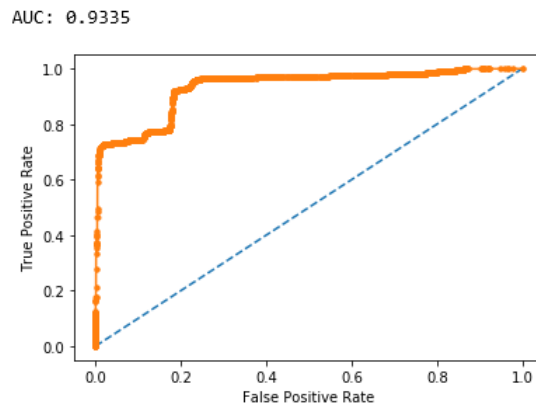


Figure 4.3: ROC Curve for logistic regression

By analyzing the ROC curve from for Fig 4.3 we can see, the orange line which denotes the logistic regression model is near tpr and AUC is 0.9335. So the overall prediction is good.

4.1.2 K-Nearest Neighbour (K-NN)

For K-NN we got the accuracy score 0.9688

The confusion matrix of K-Nearest Neighbour algorithm is as follows:

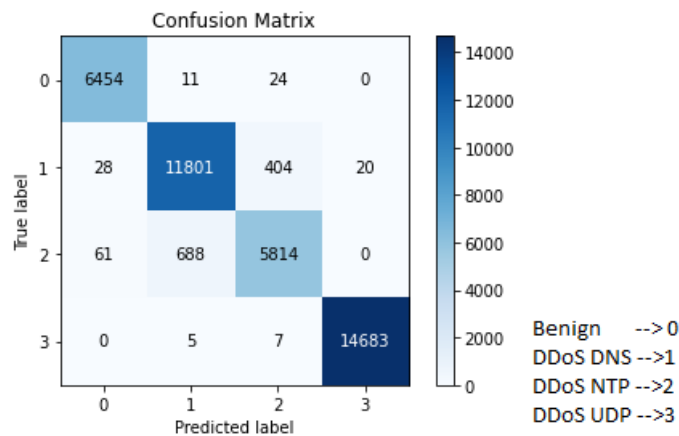


Figure 4.4: Confusion Matrix of K-Nearest Neighbour

On Fig 4.4, is the classification matrix of K-NN. We can see True Positives (TP) are diagonally related. However, $TP_0 = 6454$, $TP_1 = 11801$, $TP_2 = 5814$ and $TP_3 = 14683$.

The classification report is given below:

For KNN Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	3272
1	0.95	0.97	0.96	6199
2	0.94	0.89	0.91	3245
3	1.00	1.00	1.00	7284
accuracy			0.97	20000
macro avg	0.97	0.96	0.97	20000
weighted avg	0.97	0.97	0.97	20000

Figure 4.5: The classification report of KNN

As we can see from the classification report from Fig 4.5, for benign precision, recall and F1 score are .99, .99 and .99 respectively. For DDoS DNS precision, recall and F1 score are .95, .97 and .96 respectively. Now DDoS NTP precision, recall and F1 score are .94, .89 and .91 respectively. For DDoS UDP precision, recall and F1 score are 1, 1 and 1 respectively.

The ROC Curve of KNN is as follows:

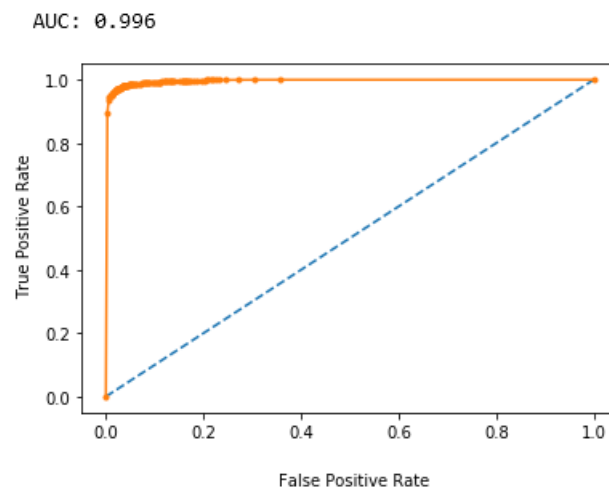


Figure 4.6: ROC Curve for KNN

By analyzing the ROC curve from Fig 4.6, we can see, the orange line which denotes the KNN model is near tpr and AUC is 0.996. So the overall prediction is very good.

4.1.3 Random Forest

For Random Forest accuracy score is 0.9880

The confusion matrix of Random Forest is as follows:

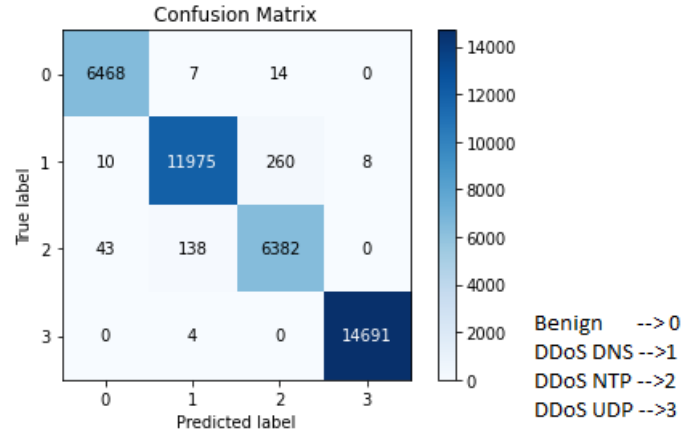


Figure 4.7: Confusion Matrix of Random Forest

From Fig 4.7, We can see, True Positives(TP) are diagonally connected. However, $TP_0 = 6468$, $TP_1 = 11975$, $TP_2 = 6382$ and $TP_3 = 14691$.

The classification report is given below:

For Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	0.99	3272
1	0.99	0.98	0.98	6199
2	0.96	0.97	0.96	3245
3	1.00	1.00	1.00	7284
accuracy			0.99	20000
macro avg	0.98	0.99	0.98	20000
weighted avg	0.99	0.99	0.99	20000

Figure 4.8: The classification report of Random Forest

From Fig 4.8, for benin precision, recall and F1 score are .99, 1 and .99 respectively. For DDoS DNS precision, recall and F1 score are .99, .98 and .98 respectively. Now DDoS NTP precision, recall and F1 score are .96, .97 and .96 respectively. For DDoS UDP precision, recall and F1 score are 1, 1 and 1 respectively.

The ROC curve for Random Forest:

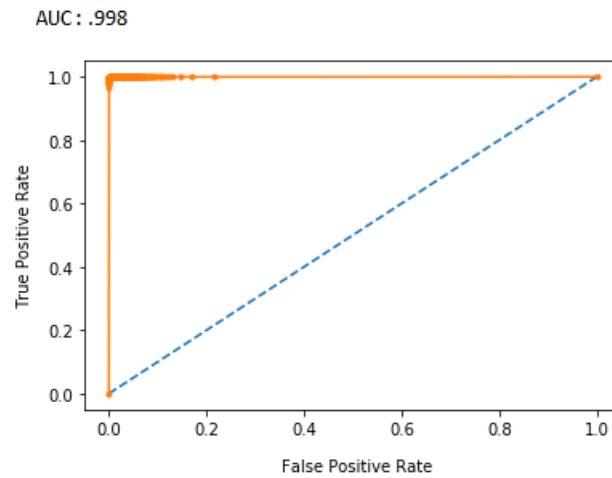


Figure 4.9: ROC Curve for Random Forest

By analyzing the ROC curve from Fig 4.9, we can see, the orange line which denotes the Random Forest model is near tpr and AUC is 0.998. So the overall prediction is excellent.

4.1.4 AdaBoost

For AdaBoost we got the accuracy score of 0.98765.

The confusion matrix for AdaBoost is as follows:

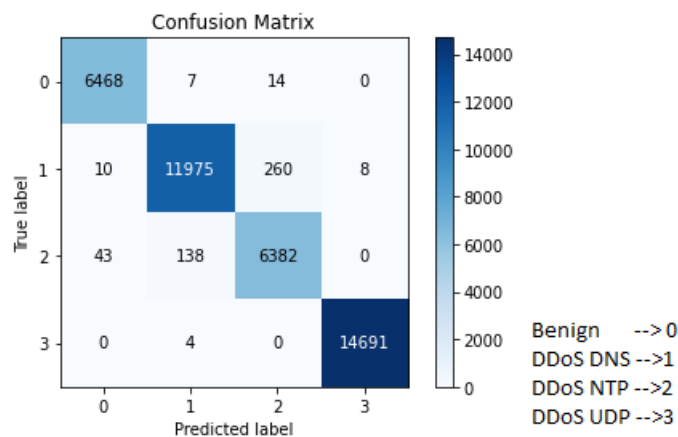


Figure 4.10: Confusion Matrix of AdaBoost

Here is the classification matrix of AdaBoost. From Fig 4.10, We can see, True Positives (TP) are diagonally connected. However, $TP_0 = 6468$, $TP_1 = 11975$, $TP_2 = 6382$ and $TP_3 = 14691$.

The classification report is given below:

For Adaboost Classification Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	0.99	3272
1	0.99	0.98	0.98	6199
2	0.96	0.97	0.96	3245
3	1.00	1.00	1.00	7284
accuracy			0.99	20000
macro avg	0.98	0.99	0.99	20000
weighted avg	0.99	0.99	0.99	20000

Figure 4.11: The classification report of AdaBoost

By analyzing Fig 4.11, benign precision, recall and F1 score are .99, 1 and .99 respectively. For DDoS DNS precision, recall and F1 score are .99, .98 and .98 respectively. Now DDoS NTP precision, recall and F1 score are .96, .97 and .96 respectively. For DDoS UDP precision, recall and F1 score are 1, 1 and 1 respectively.

The ROC Curve of AdaBoost is as follows:

The classification report is given below:

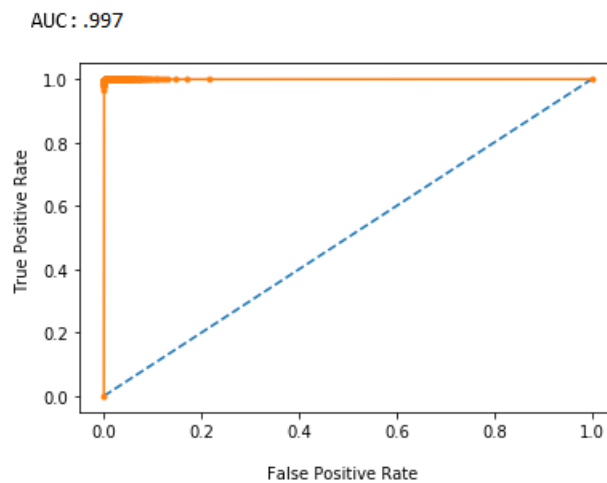


Figure 4.12: ROC Curve for AdaBoost

By analyzing the ROC curve from Fig 4.12, we can see, the orange line which denotes the AdaBoost model is near tpr and AUC is 0.997. So the overall prediction is very good.

4.1.5 Support Vector Machine

For SVM we got the accuracy score 0.9212

The confusion matrix is as follows:

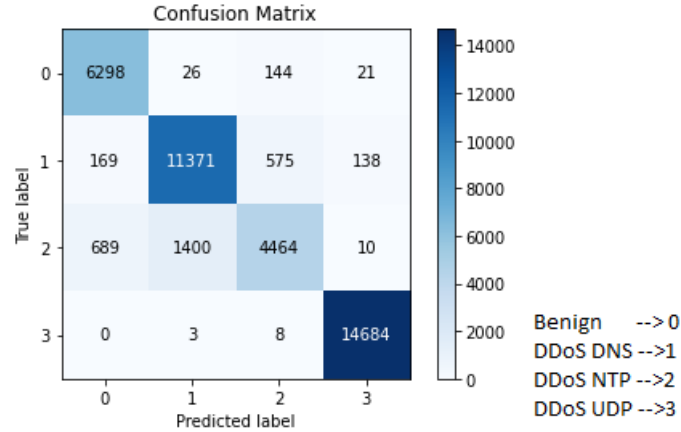


Figure 4.13: Confusion Matrix of Support Vector Machine

Here from Fig 4.13, the classification matrix of SVM. It is the confusion matrix for four classes. We can see, True Positives (TP) are diagonally connected. However, $TP_0 = 6298$, $TP_1 = 11371$, $TP_2 = 4464$ and $TP_3 = 14684$.

The classification report is given below:

For SVM Classification Report is

	precision	recall	f1-score	support
0	0.89	0.99	0.94	4859
1	0.89	0.93	0.91	9273
2	0.87	0.67	0.76	4815
3	0.99	1.00	0.99	8525
accuracy			0.92	27472
macro avg	0.91	0.90	0.90	27472
weighted avg	0.92	0.92	0.91	27472

Figure 4.14: The classification report of Support Vector Machine

As we can see from Fig 4.14 the classification report, for benin precision, recall and F1 score are .89, .99 and .94 respectively. For DDoS DNS precision, recall and F1 score are .89, .93 and .91 respectively. Now DDoS NTP precision, recall and F1 score are .87, .67 and .76 respectively. For DDoS UDP precision, recall and F1 score are .99, 1 and .99 respectively.

The ROC Curve of SVM is as follows:

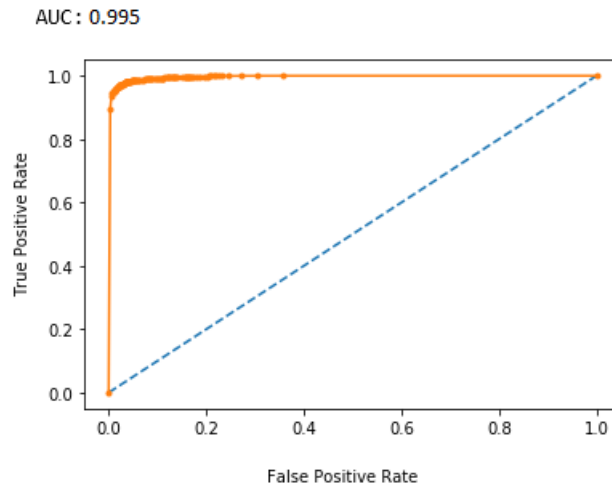


Figure 4.15: ROC Curve for SVM

By analyzing the ROC curve from Fig 4.15, we can see, the orange line which denotes the SVM model is near tpr and AUC is 0.995. So the overall prediction is good.

4.1.6 Decision Tree

For Decision Tree we got the accuracy score 0.985075

The confusion matrix for Decision Tree is as follows:

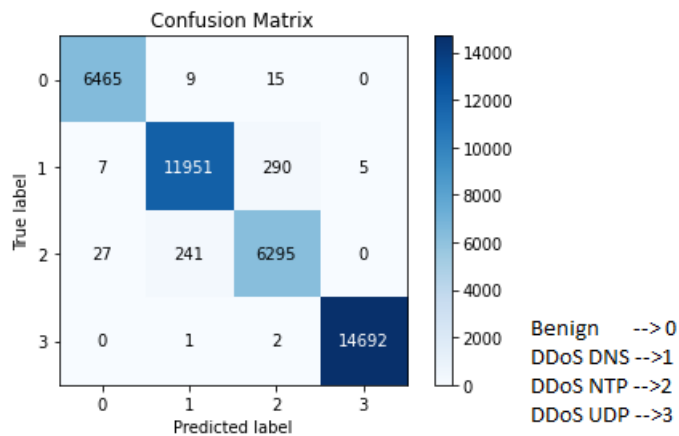


Figure 4.16: Confusion Matrix of Decision Tree

Here from Fig 4.16, the classification matrix of the Decision Tree. As, it is the confusion matrix for four classes. We can see, True Positives (TP) are diagonally connected. However, $TP_0 = 6265$, $TP_1 = 11951$, $TP_2 = 6295$ and $TP_3 = 14692$.

The classification report is given below:

For Decision Tree Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	4859
1	0.98	0.98	0.98	9273
2	0.96	0.95	0.96	4815
3	1.00	1.00	1.00	8525
accuracy			0.98	27472
macro avg	0.98	0.98	0.98	27472
weighted avg	0.98	0.98	0.98	27472

Figure 4.17: The classification report of Decision Tree

As we can see from Fig 4.17 the classification report, for benign precision, recall and F1 score are .99, .99 and .99 respectively. For DDoS DNS precision, recall and F1 score are .98, .98 and .98 respectively. Now DDoS NTP precision, recall and F1 score are .96, .95 and .96 respectively. For DDoS UDP precision, recall and F1 score are 1, 1 and 1 respectively.

The ROC Curve of Decision Tree is as follows:

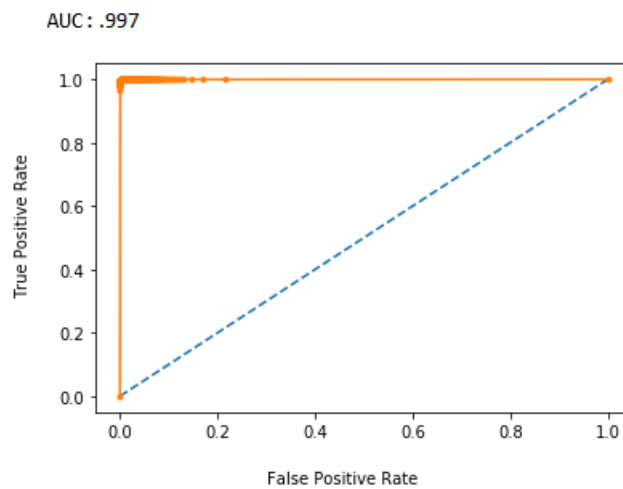


Figure 4.18: ROC Curve for Decision Tree

By analyzing the ROC curve from Fig 4.18, we can see, the orange line which denotes the Decision Tree model is near tpr and AUC is 0.997. So the overall prediction is very good.

4.1.7 XGboost

For XGboost we got the accuracy score 0.9824

The confusion matrix for XGboost is as follows:

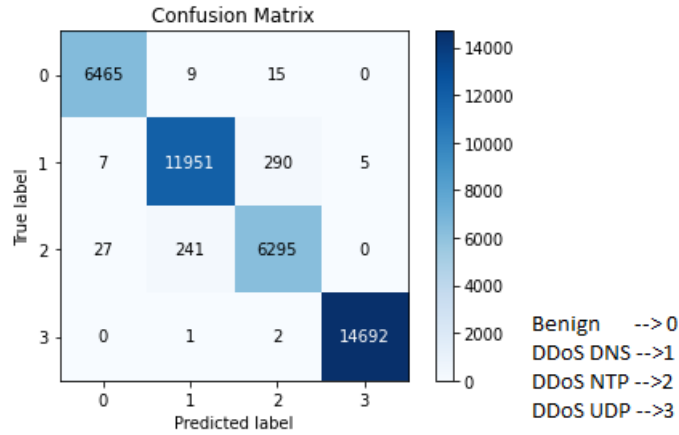


Figure 4.19: Confusion Matrix of XGboost

From Fig 4.19, the classification matrix of XGboost. We can see, True Positives (TP) are diagonally connected. However, $TP_0 = 6265$, $TP_1 = 11951$, $TP_2 = 6295$ and $TP_3 = 14692$.

The classification report is given below:

For XGboost Classification Report is

	precision	recall	f1-score	support
0	0.98	1.00	0.99	4859
1	0.99	0.97	0.98	9273
2	0.95	0.96	0.95	4815
3	1.00	1.00	1.00	8525
accuracy			0.98	27472
macro avg	0.98	0.98	0.98	27472
weighted avg	0.98	0.98	0.98	27472

Figure 4.20: The classification report of XGboost

As we can see from Fig 4.20, the classification report, for benin precision, recall and F1 score are .98, 1 and .99 respectively. For DDoS DNS precision, recall and F1 score are .99, .97 and .98 respectively. Now DDoS NTP precision, recall and F1 score are .95, .96 and .95 respectively. For DDoS UDP precision, recall and F1 score are 1, 1 and 1 respectively.

The ROC Curve for XGboost:

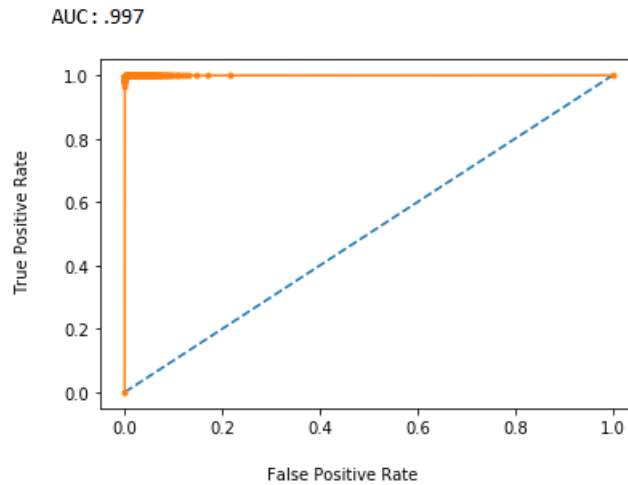


Figure 4.21: ROC curve for XGBoost

By analyzing the ROC curve from Fig 4.21, we can see, the orange line which denotes the XGBoost model is near tpr and AUC is 0.997. So the overall prediction is very good.

4.1.8 Naive Bayesian

For Naive Bayesian we got the accuracy score 0.773375

The confusion matrix for Naive Bayesian is as follows:

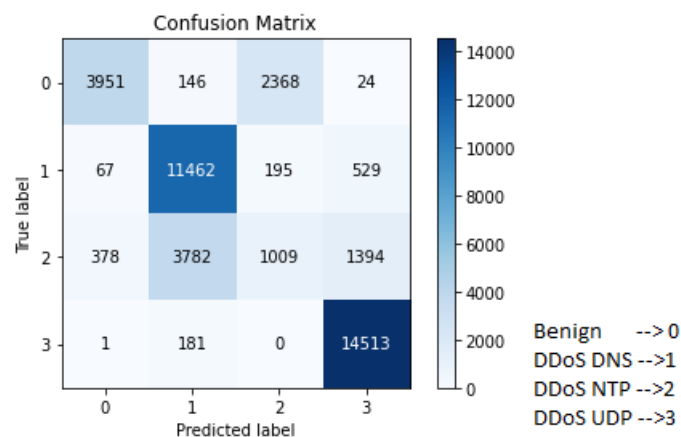


Figure 4.22: Confusion Matrix of Naive Bayesian

Here is the classification matrix of Naive Bayesian on Fig 4.22. It is the confusion matrix for four classes. However, $TP_0 = 3951$, $TP_1 = 11462$, $TP_2 = 1009$ and $TP_3 = 14513$.

The classification report is given below:

For Naive Bayes Classification Classification Report:

	precision	recall	f1-score	support
0	0.90	0.64	0.75	6639
1	0.74	0.94	0.83	12173
2	0.31	0.17	0.22	6514
3	0.89	0.99	0.94	14674
accuracy			0.78	40000
macro avg	0.71	0.68	0.68	40000
weighted avg	0.75	0.78	0.76	40000

Figure 4.23: The classification report of Naive Bayesian

As we can see from Fig 4.23 the classification report, for benin precision, recall and F1 score are .90, .64 and .75 respectively. For DDoS DNS precision, recall and F1 score are .74, .94 and .83 respectively. Now DDoS NTP precision, recall and F1 score are .31, .17 and .22 respectively. For DDoS UDP precision, recall and F1 score are .89, .99 and .94 respectively.

The ROC Curve for Naive Bayesian is as follows:

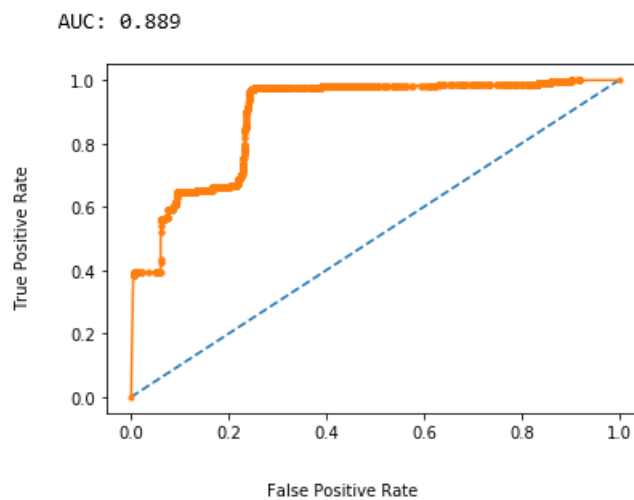


Figure 4.24: ROC Curve for Naive Bayesian

By analyzing the ROC curve from Fig 4.24, we can see, the orange line which denotes the Naive Bayesian model is near tpr and AUC is 0.889. So the overall prediction is average good.

Table 4.1: Performances of various Machine Learning classifying Algorithms Part 1

Algorithm	Accuracy	0			1		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Logistic Regression	91.74	.87	.97	.92	.85	.92	.88
K-NN	96.88	.99	.99	.99	.95	.97	.96
Random Forest	98.80	.99	1	.99	.99	.98	.98
AdaBoost	98.76	.99	1	.99	.99	.98	.98
SVM	92.12	.89	.99	.94	.89	.93	.91
Decision Tree	98.50	.99	.99	.99	.98	.98	.98
XGboost	98.24	.98	1	.99	.99	.97	.98
Naive Bayesian	77.33	.90	.64	.75	.74	.94	.83

Table 4.2: Performances of various Machine Learning classifying Algorithms Part 2

Algorithm	2			3			AUC Scores
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
Logistic Regression	.84	.57	.68	.98	1	.99	0.9335
K-NN	.94	.89	.91	1	1	1	0.996
Random Forest	.96	.97	.96	1	1	1	0.998
AdaBoost	.96	.97	.96	1	1	1	0.997
SVM	.87	.67	.76	.99	1	.99	0.995
Decision Tree	.96	.95	.96	1	1	1	0.997
XGboost	.95	.96	.95	1	1	1	0.997
Naive Bayesian	.31	.17	.22	.89	.99	.94	0.889

The table 4.1 and 4.2 show the Precision, recall, F1-score and AUC scores of the various machine learning algorithm which are applied into our dataset.

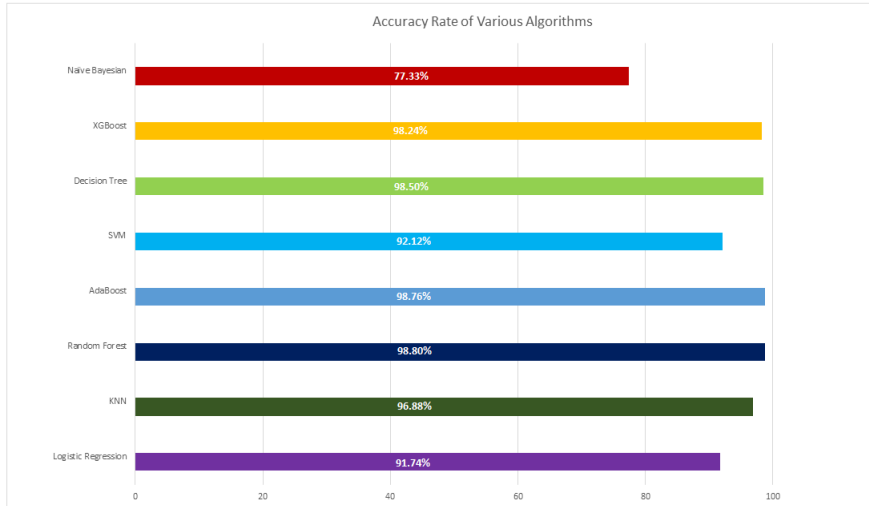


Figure 4.25: Accuracy rate of Various Algorithms

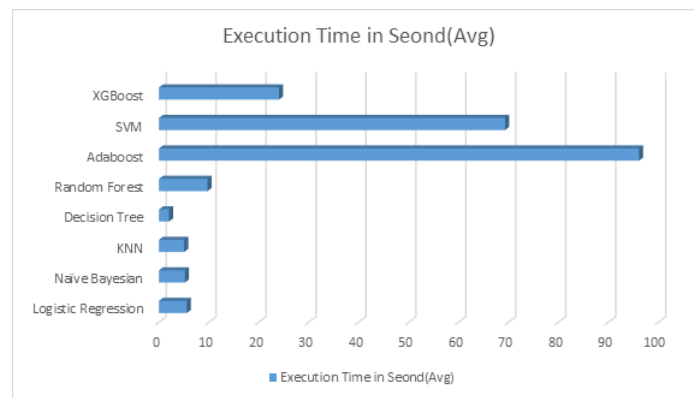


Figure 4.26: Execution Time of various Algorithms

Here, in the above Figure 4.26, the execution time of different algorithms is shown where we can evaluate easily which algorithm takes the lowest computation time to execute.

Table 4.3: Average Execution Time of various Algorithms in Google Colab

Algorithm	Execution Time in second (Avg)	Accuracy
Logistic Regression	5.556 s	91.74%
Naive Bayes	5.189 s	77.33%
KNN	5.045 s	96.88%
Decision Tree	2.015 s	98.50%
Random Forest	9.71 s	98.80%
Adaboost	96.122 s	98.76%
SVM	69.301 s	92.12%
XGboost	24.037 s	98.24%

4.2 Analyzing the Results

Now, if we analyze the results from Fig 4.25, we will notice that the accuracy of Naive Bayesian, SVM and Logistic regression is below 98%. And the rest of the algorithms performed better in the dataset and all the accuracy is above 98%. This happened because ensemble based algorithms like Random forest, XGBoost, AdaBoost perform better in tabular formed short dataset [36]. Among those algorithms Random Forest acquired the highest accuracy 98.80%. But time consumption is high in comparison with Decision Tree. On the other hand, the Naive Bayesian algorithm is based on the Gaussian approach and Logistic regression is based on regression algorithms. However, ensemble based algorithms are powerful and stable so it gives good performances. But ensemble based algorithms take more time to execute so it might not be good for real time implementation.[3]. But here, Decision Tree acquired the excellent accuracy of 98.50% with less time consumption because it can maintain accuracy if there is a large number of missing data, it has methods to reduce errors on imbalanced dataset. For these reasons Decision Tree acquired excellent accuracy with less time consumption among the ensemble based algorithms.

4.3 Real Time Implementation

After analyzing our data and obtaining the result we have constructed a model for applying the machine learning procedures in real time implementation in practical life. In DDoS attack detection model if we want to execute this model in our real life we need to detect the DDoS attack practically as well as the calculation time should be determined. We have constructed an architectural model of Client-Server[3]. Here we used Flask model. The workflow model of the real time implementation is below:

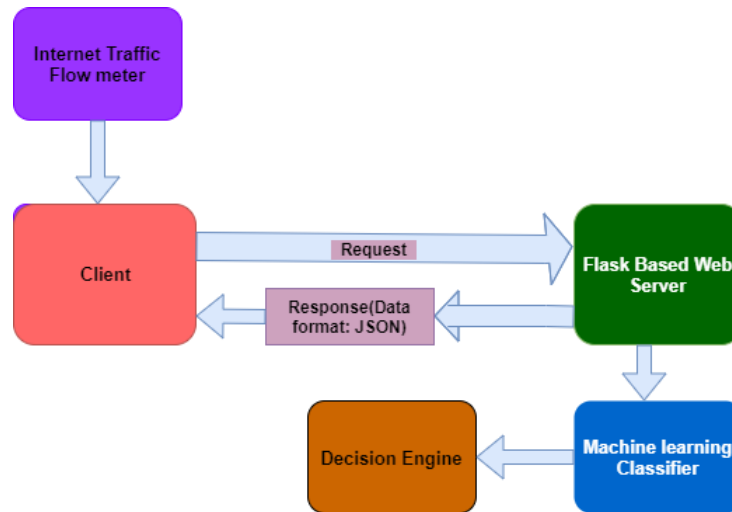


Figure 4.27: Client-Server workflow model of Real-time implementation

For building up the model we have gone through process by process:

(i) Building up web server:

To build a web server we have primarily set a local server. Server is basically a platform where the client can send request and the server serves data accordingly to the sent request. The server will respond and serves data to that particular request remembering the IP address from the client. Here, we have used flask and python for setting up the web server because flask brings up all python tools which can be control easily.

Introducing Flask:

Flask is mainly a web based application framework which is written on python. So at the beginning we have to install python and then import the flask. Flask package can be installed from the Python Package Index (PPI) .There are many advance benefits of using flask. Flask is lightweight and simply classified as a micro-framework. Flask is used for Back-end purposes. This framework need not depend on external libraries or tools. Flask supports the client with all python tools, libraries and some built in functions. The main advantage of using flask is that there is no restriction of using this framework. Any user can build anything in it. In work with Rest API, flask provides an excellent and fast outcome. Moreover, Flask works very intelligently with high weighted database for this reason, we have used flask to build our server.

(ii) Data Processing:

Here, we have decided to use an Internet Traffic Flow Meter device which is used for gathering the data in real life about the traffic flow. The device will work at a certain point with the range of the network, this certain point is called 'meter'. After collecting the data, this will be processed in a dataset.

(iii) Client sends Requests to the Server:

Client will receive the data from the dataset, then it will send the file to the Server by sending a request with API call. Client side will use GET request for API call. GET request can be saved in browsing history as well as bookmarked when the Client-side sends a GET request to the server. The server will remember its information such as IP address. After sending the requests, the client side will wait for the response and ping to the server continuously.

(iv) Response from the Server:

After obtaining the requests from the Client side, the Server will accept the requests and save the data into the database and train data to the machine learning classifier for the calculation.

Machine learning classifier:

So far, we have trained machine learning algorithms with the training dataset. In real-time implementation the classifier will classify and compute all computational values from real time data which is need to be shown and the data will be provided by the server. After the calculation, the result will be saved in a file. Then, the file will be formatted into JSON data type.

JSON (JavaScript Object Notation):

JSON is a very uncomplicated regulated format for exchanging the data. JSON only can be medium between client and server which mainly format the response before responding to the particular request. JSON works as a parsing free excellent formatter. JSON is basically text-based format. It is mainly used for serialization the data. JSON can be valid for six types of data:

1. String
2. Object
3. Number
4. Boolean
5. Null
6. Array

Here, we have imported jsonify library for activating the JSON.

Other libraries: We have imported some other libraries such as OS, render-template. OS is imported for reading the file and render-template is imported for showing the html.

(v) **DDoS Detection Prototype:** We have designed a dashboard as a prototype for showing the result. Here, we have kept three types of results they are DDoS Status, Accuracy and Calculation Time.

DDoS Dashboard:

Ddos status:true
Accuracy: 92%
Calculation Time: 27 seconds

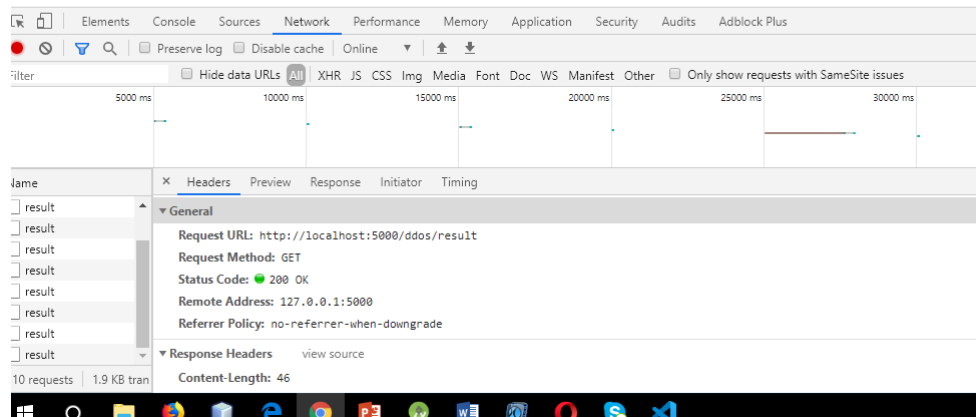


Figure 4.28: Real Time implementation Dashboard

Chapter 5

Conclusion and Future Work

In our paper, we used a supervised learning dataset for the purpose of detecting DDoS attacks. In our proposed model, our dataset is finely labeled. We have propagated different evaluation reports and visualization models after preprocessing the dataset. So far, we have applied nine machine learning algorithms to detect DDoS attacks. The methods are KNN, Naive Bayesian, Support Vector Machine (SVM), Logistic Regression, Random Forest, Artificial Neural Network (ANN), AdaBoost, XGBoost and Decision Tree. After generating all these nine methods we have come up with a result that the Decision Tree algorithm gives a satisfied result with the highest accuracy which is 98.50 percent with the lowest execution time which is 2.015 seconds. We have got convinced results from the Decision Tree algorithm because this method can handle large datasets and missing values as well as it can reduce error. We have also evaluated those results based on the features of each classifier. Moreover, for advanced visualization, we have designed a model of real-time implementation. Moreover, we have constructed a Client-Server model using the Flask framework because Flask is an advanced microweb framework. In the system, we have built a dashboard which will be able to show the accuracy, computational time and DDoS Status. In our paper, we have built the system in a file-based manner. In the future, we have planned to set the web server in real-time [3] and database-based. Moreover, we have planned to construct a hybrid classifier which will produce more accurate results with less computational time compared with other machine learning classifiers. Therefore, we keep the hope to handle the detection model more faster and accurate with our proposed system. Thus, we can say that, our proposed model will be helpful for detecting with the aim to security demands.

Bibliography

- [1] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features”, in *European conference on machine learning*, Springer, 1998, pp. 137–142.
- [2] P. J. Phillips, “Support vector machines applied to face recognition”, in *Advances in Neural Information Processing Systems*, 1999, pp. 803–809.
- [3] S. N. Shiaeles, V. Katos, A. S. Karakos, and B. K. Papadopoulos, “Real time ddos detection using fuzzy estimators”, *computers & security*, vol. 31, no. 6, pp. 782–790, 2012.
- [4] S. Gupta, D. Grover, and A. Bhandari, “Detection techniques against ddos attacks: A comprehensive review”, *International Journal of Computer Applications*, vol. 96, no. 5, 2014.
- [5] A. Girma, M. Garuba, J. Li, and C. Liu, “Analysis of ddos attacks and an introduction of a hybrid statistical model to detect ddos attacks on cloud computing environment”, in *2015 12th International Conference on Information Technology-New Generations*, IEEE, 2015, pp. 212–217.
- [6] J. Brownlee, *A gentle introduction to xgboost for applied machine learning. [online] machine learning mastery*, <https://machinelearningmastery.com/>, 2016.
- [7] *Logistic regression for machine learning*, <https://machinelearningmastery.com/>, 2016.
- [8] Q. Niyaz, W. Sun, and A. Y. Javaid, “A deep learning based ddos detection system in software-defined networking (sdn)”, *arXiv preprint arXiv:1611.07400*, 2016.
- [9] M. Galarnyk, *Pca using python (scikit-learn)*, <https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60>, 2017.
- [10] S. Ray, *6 easy steps to learn naive bayes algorithm with codes in python and r*, <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>, 2017.
- [11] —, *How random forest algorithm works in machine learning*, <https://medium.com/>, 2017.
- [12] *Understanding support vector machine(svm) algorithm from examples (along with code)*, <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>, 2017.

- [13] M. Zekri, S. El Kafhali, N. Aboutabit, and Y. Saadi, “Ddos attack detection using machine learning techniques in cloud computing environments”, in *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, IEEE, 2017, pp. 1–7.
- [14] B. Zhang, T. Zhang, and Z. Yu, “Ddos detection and prevention based on artificial intelligence techniques”, in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, IEEE, 2017, pp. 1276–1280.
- [15] —, “Ddos detection and prevention based on artificial intelligence techniques”, in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, IEEE, 2017, pp. 1276–1280.
- [16] U. Dincalp, M. S. Güzel, O. Sevine, E. Bostanci, and I. Askerzade, “Anomaly based distributed denial of service attack detection and prevention with machine learning”, in *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, 2018, pp. 1–4.
- [17] K. Gurulakshmi and A. Nesarani, “Analysis of iot bots against ddos attack using machine learning algorithm”, in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, 2018, pp. 1052–1057.
- [18] P. Khuphiran, P. Leelaprute, P. Uthayopas, K. Ichikawa, and W. Watana-keesuntorn, “Performance comparison of machine learning models for ddos attacks detection”, in *2018 22nd International Computer Science and Engineering Conference (ICSEC)*, IEEE, 2018, pp. 1–4.
- [19] P. Sarkar, *An intro to ensemble learning in machine learning*, <https://towardsdatascience.com/>, 2018.
- [20] *A beginner’s guide to xgboost*, <https://towardsdatascience.com>, 2019.
- [21] *Artificial neural networks for machine learning – every aspect you need to know about*, <https://data-flair.training>, 2019.
- [22] J. Brownlee, *A tour of machine learning algorithms*, <https://machinelearningmastery.com/>, 2019.
- [23] F. S. d. Lima Filho, F. A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, and L. F. Silveira, “Smart detection: An online approach for dos/ddos attack detection using machine learning”, *Security and Communication Networks*, vol. 2019, 2019.
- [24] A. G. Oleg Kupreev Ekaterina Badovskaya, *Ddos attacks in q3 2019*, <https://securelist.com/ddos-report-q3-2019/94958/>, 2019.
- [25] T. V. Phan and M. Park, “Efficient distributed denial-of-service attack defense in sdn-based cloud”, *IEEE Access*, vol. 7, pp. 18 701–18 714, 2019.
- [26] K. Saeedi, *Machine learning for ddos detection in packet core network for iot*, 2019.
- [27] I. Sreeram and V. P. K. Vuppala, “Http flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm”, *Applied computing and informatics*, vol. 15, no. 1, pp. 59–66, 2019.
- [28] *Support vector machine algorithm*, <https://www.javatpoint.com>, 2020.

- [29] *Bar graph - learn about bar charts and bar diagrams*, <https://www.smartdraw.com>.
- [30] *Ddos attacks: Their top 5 favorite industry targets*. <https://www.cloudbric.com>.
- [31] *Ddos statistics, facts and trends for 2018-2019*, <https://www.comparitech.com>.
- [32] A. Rajwade, *Face detection using adaboost*, https://www.cse.iitb.ac.in/~ajitvr/CS763.Spring2017/Adaboost_FaceDetection.pdf.
- [33] *Weka tutorial: Machine learning data mining*, <https://wekatutorial.com>.
- [34] *What is a distributed denial of service attack (ddos) and what can you do about them?*, <https://us.norton.com/internetsecurity-emerging-threats-what-is-a-ddos-attack-30sectech-by-norton.html>.