DDOS ATTACK DETECTION USING NETWORK TRAFFIC CLASSIFICATION TECHNIQUES

REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

By

Hriman Krishna Mahanta
Roll Number: 180102014
&
Vikash Bhuyan
Roll Number: 180103014

UNDER THE GUIDANCE

OF

Dr. Satyajit Sarmah

Assistant Professor, Dept. of IT, GU

DEPARTMENT OF INFORMATION TECHNOLOGY

GAUHATI UNIVERSITY

GUWAHATI, INDIA

JUNE –2022

GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

DECLARATION

We, Hriman Krishna Mahanta, Roll No 180102014, and Vikash Bhuyan, Roll No 180103014, B.Tech. students of the department of Information Technology, Gauhati University hereby declare that we have compiled this report reflecting all our works during the semester long full time project as part of our BTech curriculum.

We declare that we have included the descriptions etc. of our project work, and nothing has been copied/replicated from other's work. The facts, figures, analysis, results, claims etc. depicted in our thesis are all related to our full time project work.

We also declare that the same report or any substantial portion of this report has not been submitted anywhere else as part of any requirements for any degree/diploma etc.

Hriman Krishna Mahanta
Branch: CSE
Date: 27-06-2022

Vikash Bhuyan
Branch: CSE
Date: 27-06-2022

GAUHATI UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

Date: 27-06-2022

CERTIFICATE

This is to certify that Hriman Krishna Mahanta bearing Roll No: 180102014 and Vikash Bhuyan bearing Roll No: 180103014 has carried out the project work “DDoS attack detection using network traffic classification techniques” under my supervision and has compiled this report reflecting the candidate’s work in the semester long project. The candidates did this project full time during the whole semester under my supervision, and the analysis, results, claims etc. are all related to their studies and works during the semester.

I recommend submission of this project report for the 8th semester examination of Bachelor of Technology in Computer Science & Engineering of Gauhati University.

Dr. Satyajit Sarmah
Assistant Professor, Dept. of IT, GU

GAUHATI UNIVERSITY

DEPARTMENT OF INFORMATION TECHNOLOGY

Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

Date:27-06-2022

TO WHOM IT MAY CONCERN

This is to certify that Hriman Krishna Mahanta bearing Roll No 180102014 and Vikash Bhuyan bearing Roll No 180103014, B.Tech. students of the department of Information Technology, Gauhati University, has submitted the softcopy of their project for undergoing screening through anti-plagiarism software and the similar report is found to be ____

Dr. Satyajit Sarmah
Assistant Professor, Dept. of IT, GU

GAUHATI UNIVERSITY

DEPARTMENT OF INFORMATION TECHNOLOGY

Gopinath Bordoloi Nagar, Jalukbari Guwahati-781014

External Examiners Certificate

This is to certify that Hriman Krishna Mahanta, bearing Roll No 180102014 and Vikash Bhuyan, bearing Roll No 180103014 has delivered their project presentation on 27/06/2022 and I examined their report entitled "DDoS attack detection using network traffic classification techniques" and recommend this project report as a part for partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering of Gauhati University.

(External Examiner)

ACKNOWLEDGEMENT

We would like to express our special thanks and gratitude to our project supervisor Dr. Satyajit Sarmah under whose supervision we were able to complete this project.

Secondly, we would also like to thank our parents and friends for the support and encouragement they have given us in completing this project.

Hriman Krishna Mahanta
Roll no: 180102014
Branch: CSE

Vikash Bhuyan
Roll no: 180103014
Branch: CSE

TABLE OF CONTENTS

ABSTRACT

The Internet has become an almost indispensable part of the modern world. We are accessing the Internet when using our computers, smartphones and other smart devices. When we are accessing the Internet, lots of protocol data units called packets are actually getting transferred between various computers across the world. These packets contain lots of information about the type of network traffic. By using these information, we can analyse the network better. We can also use this information to detect if there is any security issues in the network. Hence, Network Traffic analysis is important both for traffic engineering and network security. With the increase in the usage of Internet throughout the world, new kinds of network security threats are also becoming more and more prominent. One of the biggest and most costliest threat to the modern Internet is the Distributed Denial of Service (DDoS) attack. In this project, we have tried to find a solution for countering a DDoS attack by trying to analyze different packets flowing in a computer network and finding the relevant features in order to detect a DDoS attack. To classify the packets, we have used various machine learning algorithms and created a model. We have tried to make the model as accurate as possible so that it can be used in some real world servers to detect an attack. We have also derived some interesting statistics about the features of a network traffic which plays the most important part in the classification.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol	Explanation
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
BOOTP	Bootstrap Protocol
FTP	File Transfer Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
LOIC	Low Orbit Ion Cannon
DDOS	Distributed Denial of Service

1. INTRODUCTION

In the past 50 years, computer networks have continued to expand in size, complexity, and overall user count while also undergoing constant innovation. As a result, the volume of network traffic passing through their nodes has dramatically increased. In order to keep the network operating smoothly and increase economic efficiency, network administration, maintenance, and monitoring are crucial as network technology develops and becomes more widely used.

However with the increase in usage of Internet, network security has become a very important issue in the modern world. Attackers are always trying to come up with some new ways to find the vulnerabilities in networking devices. One of the most prominent and costliest network security threat in the modern world is the DDoS attack. A DDoS attack occurs when an attacker compromises a huge number of computers and then uses those computer to flood a server with a overwhelming amount of packets. The affects of DDoS attack are immense. DDoS attacks have been known to cause servers to go offline ranging from 24 hours to even weeks.

In this project, we have first captured packets from a network traffic using Wireshark and then performed some statistical analysis on those packets. We have found some statistics like distribution of transport layer protocols, application layer protocols. We have also found the various regions of the world from which the network packets in our computer are coming from. For this, we have used a database called “GeoLiteCity” which contains the corresponding location of each IP address. We have also plotted a map using “Google My Maps” feature which shows the network traffic from our computers to various servers all around the world. For this, we have again used the “GeoLiteCity” database from which we have extracted the latitude and longitude in order to plot the map.

After this, we have tried to simulate a DDoS attack in our lab by using a software known as Low Orbit Ion Cannon (LOIC). This software simulates a DDoS attack by sending a huge number of packets to a particular IP address. In our lab, we have send the packets from one computer to another. To make sure that the traffic simulates a DDoS attack as closely as possible, we have used multiple threads while simulating the attack. Then, we have captured the packets in the network. We have also downloaded some packet capture files online to make our dataset bigger and more variable.

After capturing and accumulating packets for both normal and attack traffic, we have extracted various features from the packets by creating a time window. It means the features that we have extracted are actually the consolidated data of all the packets in each time window. Creating a time window gives us a better idea about the kind of network traffic as it reflects the nature of the traffic over a period of time. We have extracted a total of 16 features which are later used in the classification using various machine learning algorithms.

Finally, we have classified the network traffic using the features we have extracted by using six binary classification algorithms. We have used the following algorithms:

(i) Logistic Regression

(ii) Decision Tree

(iii) Random Forest

(iv) K Nearest Neighbours

(v) Naive Bayes

(vi) Standard Vector Machine

We have found the accuracy of each algorithm by creating a confusion matrix and also calculating various accuracy parameters. We have also plotted a feature importance graph by calculating the importance of each features in the classification process.

2. RELATED WORKS AND LITERATURE REVIEW

This section consists of the previous research and studies done on this topic. Lots of research has been done to provide a suitable and effective way to mitigate DDoS attacks. We have looked at some of the previous works that have done related to this project. We have mentioned some of them in this section.

In [1], a probabilistic method is used to classify the packets. The author has tried to make a very flexible and easily configurable framework so that it can be used for various networks. They have created a model using machine learning which uses the statistics of sequences of packets to differentiate between known traffic from unknown traffic. Their method of classification is mainly based on likelihood estimation.

In [2], the authors have tried to investigate the performances of various network traffic capture tools for feature extraction and also find the efficiency of various machine learning tools. They used six different Internet applications and two different network anomalies. Their result showed that the Colasoft Capsa network capturing tool gave the best result in terms of classification. On the other hand, according to their results all the machine learning algorithms that they used gave almost similar accuracy.

In [3], the authors have used Naive Bayes algorithm and Standard Vector Machine to classify the packets in a network traffic into normal or DDoS traffic. They have designed a system which consists of various operations like capturing packets after DDoS attack, processing the data and then classifying them.

In [4], the authors have tried to extract the most relevant features that can be most useful in classification of the traffic. They have then compared the performance of various features in the classification process. Then, they have used tree based machine learning algorithms to classify the data.

In [5], the authors have carried out an analysis of various traffic classification techniques. They have also tried to investigate whether it was feasible to use machine learning techniques to classify network traffic. Their results showed that machine learning is the most promising and efficient technique presently for traffic classification. They have also done a comparison of supervised and unsupervised learning for this topic.

In [6], the authors have proposed an artificial neural network to create a botnet detection model. The author have used the BoT-IoT dataset. They have implemented a data resampling technique known as SMOTE to resample a real time data into balanced data. Their results showed that the system was quite effective with a basic configuration of ANN.

In [7], the authors considered two machine learning algorithms for classifying the network traffic, namely the Support Vector Machine and K-Means. The authors studied the impact of feature selection and model tuning on the performance of the classifier. They also used five fold cross validation for the classification process. Their result showed that the accuracy obtained by supervised learning was better than that of unsupervised learning.

In [8], the authors proposed a machine learning based IP traffic classification in operational networks. The authors used Internet application traffic as the input and then classified them using various machine learning algorithms. They also tried to outline the critical operational requirements of a real time classifier.

In [9], the authors proposed a network traffic classification model to identify unknown network traffic classes using supervised learning techniques. They have used four machine learning algorithms namely, C4.5, Support Vector Machine, BayesNet and NaiveBayes. They have also used ten fold cross validation to build the classification model. Their result showed that C4.5 algorithm gave the best accuracy compared to the other algorithms.

In [10], the authors performed an analysis to differentiate between normal and abnormal data received on the Internet. They used the KDD Cup 99 dataset for the classification. They classified the dataset by using Naive Bayes, bayes Net, Random Forest, Multilayer perception and sequential minimal optimization algorithms. Their result showed that random forest and multilayer perception gave the best accuracy.

3. BACKGROUND

3.1 MOTIVATION

With an increase in demand for accessible services and applications over the Internet, a good network without any security threats has become more important than ever before. Various issues such as loss of packets, latency of packet flow, frequent downtime of networks, errors in transmission and jitters can hamper the overall user experience. By analyzing the packets flowing in a network, we can have a good idea about the kind of traffic flowing in a network and also detect any anomalies in the network.

Now a days, there are a lot of attacks happening in a network. One of the most prominent and costliest attacks in the modern world is the DDoS attack. In DDoS attack, a server is congested with lots of requests and the server stops working. DDoS attacks have been known to cause resources to be offline for 24 hours, multiple days or even a week depending on the severity of the attack. We have tried to classify the packets to either normal or attack traffic using machine learning to detect the packets which may be part of a DDoS attack. The model created by the classifier can be used to detect DDoS traffic in a network.

The aim of this project is to analyse the traffic flowing in a network by using the various data present in a packet. By performing various analysis of the network traffic, we can collect a real time or historical record of what is happening in our network, detect malware activity or use of vulnerable protocols and ciphers and use this information to troubleshoot a slow network. It can be used to troubleshoot performance issues such as bandwidth consumption/utilization and network downtime. In this project, we have tried to derive some useful statistics about the network traffic that might be helpful to solve some of these issues.

3.2 NETWORK PACKETS

In networking, a packet is a single part of a larger message. Data sent over the Internet, is first divided into packets. Then, these packets are recombined by the computer that receives them. For example, if we consider the case of a loading an image in their computer or mobile device. The image file is not sent from the web server to the user’s device in a single piece. Instead, it is broken down into small units called packets that are sent over the wires, cables, and radio waves of the Internet, and then reassembled by the user's device into the original photo.

The Internet is a "packet switching" network. Packet switching means that the devices in the network process the packets independently from each other. It also means that packets can take different paths through different routers to the same destination. Another property of packet switching is that the packets can travel in any order. This enables multiple connections to take place over the same networking devices at the same time. That is why, billions of devices can connect to the Internet to exchange data at the same time, instead of just a handful.

There are two portions in every packets: a header and a payload. The header contains information about the packet like its origin and destination IP addresses, its source and destination port, length of the packet. On the other hand, the payload is the actual data in the packet. Each packet consists of multiple headers. There are headers for each of data link layer, network layer, transport layer and application layer. The format or structure of a each header is determined by the protocol that is used in that particular header. There are multiple protocols that are used in most of the OSI layers.

For network analysis or attack detection, the information present in each packets especially the information present in the header is very useful for collecting data about the network traffic. In our project, we have relied on the data from the headers in each captured packets to create our features.

3.3 THE OSI MODEL

The OSI Model (Open Systems Interconnection Model) is a theoretical model used to describe a network packet. It was published in 1984 by the International Organization for Standardization (ISO). The 7 layers of the OSI Model are:

(i) Physical Layer: It is the lowest layer in the OSI model. Its function is to transmit raw unstructured data across a network. The physical layer consists of devices hubs, cable, repeaters, network adapters and modems.

(ii) Data Link Layer: The data link layer performs hop to hop transfer of data where data is sent in the form of frames. The data link layer also has mechanism for correcting errors that may have occurred at the physical layer.

(iii) Network Layer: The network layer uses IP address to find the destination of each packets. In this layer, routers are used to find the path through which a packet must traverse to reach its destination..

(iv) Transport Layer: Delivery and error checking of data packets is the main objective of the transport layer. It regulates the amount of data transferred at a particular time to avoid congestion in the network. Two popular protocols are TCP and UDP.

(v) Session Layer: The purpose of session layer is to set up a session between two computers and manage some tasks like authentication between the computers.

(vi) Presentation Layer: The presentation layer translates data into a form which the application layer accepts. It also handles encryption and decryption of data..

(vii) Application Layer: The application layer is the layer through which the user interacts with the network. This layer consists of all the network services provided to users of a computer like web browser or any other application which needs access to the Internet.

3.4 NETWORK PROTOCOLS

Network protocols can be defined as a set of rules that determine how data is transmitted between devices in the Internet. It's main function is to allow all the network devices to communicate with each other, even if they are different in terms of their internal processes, structure or design. We can easily and efficiently communicate with people all over the world because of network protocols. Hence, network protocols play a critical role in modern digital communications.We can think of protocols as the language with which the networking devices communicate with each other. Similar to the way that speaking the same language makes it possible for people to communicate with each other, network protocols make it possible for networking devices to communicate with each other over the Internet.

There are various network protocols used in different layers of the OSI model. Some of them are:

(i) Transport Layer - TCP, UDP

(ii) Data Link Layer - Ethernet

(iii) Network Layer - IP

(iv) Application Layer - HTTP, HTTPS, SMTP, FTP, BOOTP, DNS, DHCP

3.5 DDOS ATTACK

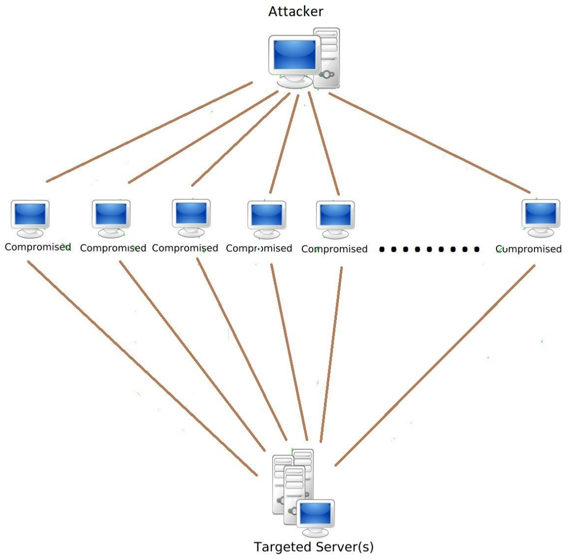

Fig 3.5: Structure of a DDoS attack

DDoS is short for Distributed Denial of Service. A DDoS attack is one of the most prominent and costliest network security threats in the modern world. Various DDoS attacks have caused servers around the world to go offline ranging from a single day to even weeks. A DDoS attack can be defined as an attempt to stop the working of a server by overwhelming the server with a huge amount of Internet traffic. The attacker of a DDoS attack first compromises multiple computers or network devices around the world and then uses those devices to initiate an attack on a particular server by sending lots of packet to that server at the same time. When the victim server tries to serve all the requests that it received from all the computers, it's resources gets overwhelmed making it unable to serve normal user. This phenomenon is called as "denial of service" because the users who are trying to connect to the server are denied service.

3.6 DIFFERENT TYPES OF DDOS ATTACKS

There are various types of DDoS attacks which uses multiple vulnerabilities of a server to exploit its resources and hence make it unusable for a normal user. Some of the popular types of DDoS attacks are:

(i) SYN Flood: The aim of a SYN Flood attack is to exploit weakness in the three way handshake method used to initiate a TCP connection. In a SYN Flood attack, a huge number of connection requests are sent and the connections do not close, hence eating up the resources of the server.

(ii) UDP Flood: The aim of a UDP Flood attack is to target random ports of a server with UDP packets. The host server checks for the application at those ports, but it does not find any application.

(iii) HTTP Flood: A HTTP Flood attack sends a huge number of GET or POST requests to a server, causing the server to run out of resources. Even though the HTTP Flood uses relatively less bandwidth than other types of attacks, it still is able to force a server to use a very huge amount of resources.

(iv) Ping of Death: A ping of death is a type of DDoS attack in which an attacker manipulates IP protocols by sending malicious pings to a system. Ping of death was a popular attack around 20 years ago, but now a days it is very less effective.

(v) Smurf Attack: A smurf attack uses a malware program called smurf to spoof an IP address and then it pings IP address on a network by using ICMP packets.

3.7 BINARY CLASSIFICATION ALGORITHMS

Binary classification is the process of classifying the data of a set into one of two classes on the basis of a classification rule. In this project we have used 6 of the most popular binary classification algorithms in order to classify the packets. These algorithms are briefly discussed below:

(i) Logistic Regression: Logistic regression is a type of classification algorithm which is used to predict a binary outcome, such as yes or no, 1 or 0, etc. The goal of Logistic regression is to find a correlation between the features and the likelihood of a specific outcome. Logistic Regression uses a cost function known as the “Sigmoid function” or “Logistic function” to classify the data.

(ii) Decision Tree: Decision Tree is a tree structured classifier where data is continuously split according to certain conditions. In a decision tree, the internal nodes represent the features, branches represent the decision rules or conditions and leaf node represents the outcome of the classification. On every node, a decision tree asks a question and then it splits the subtrees based on the answer.

(iii) Random Forest: Random Forest is a classifier that consists of multiple decision trees which classify on various subsets of the given dataset and takes the average to improve the accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, it predicts the final output.

(iv) K Nearest Neighbours: K Nearest Neighbours algorithm assumes the similarity between the new data and available data and puts the new data into the category that is most similar to the available categories. It uses the proximity of different points to a data in order to classify the data. Basically, it looks at k neighbours closest to a data and then it chooses the most prominent class of those neighbours and assigns it to the data.

(v) Naive Bayes: Naive Bayes classifier uses the principle of Bayes algorithm in order to classify the data based on their probabilities. It is called “naive” because it assumes that all the features are independent of each other. Naive Bayes classifier gives a good performance when the dataset is very large.

(vi) Standard Vector Machine: Standard Vector Machine plots each data item into an n-dimensional space where n is the number of features in the dataset. It then creates a decision boundary such that it has the maximum distance between data points of both the classes. Then it classifies new data into a particular class depending on which side of the decision boundary that data is plotted.

4. SOFTWARE TOOLS USED

4.1 WIRESHARK

Wireshark is a free and open source network traffic capture and analyzer tool that enables users to view the data traffic on a computer network. It was initially called Ethereal, but it was renamed to Wireshark in 2006. Wireshark has been contributed by many network developers from all around the world who have developed its network analysis, troubleshooting, software development and communication protocols tools.

Wireshark is basically a network protocol analyzer or it can also be called an application that captures packets from a network connection, for example from our computer to our home office or the internet. Packet is the name given to a single unit of data in a typical Ethernet network. Wireshark is the one of the most popular packet capturing tools in the world. Wireshark mainly has three functions:

(i) Packet Capture: Wireshark listens to a network connection in real time and then captures all the network packets flowing in the traffic.

(ii) Filtering: Wireshark can filter the packets that it captured by using various filtering conditions. This can be used by a user to capture only the packets that he/she is interested in.

(iii) Visualization: Wireshark also has a very good graphical user interface which makes it easier for a user to visualize the captured packets and their data.

Wireshark has many applications, including troubleshooting networks with performance issues. Wireshark is very often used by cybersecurity professionals for tracing connections, viewing the contents of suspected network transactions and identifying bursts of network traffic. Wireshark is a open source software which is used by educational institutions, networking corporations, etc to troubleshoot network issues. Additionally, Wireshark can also be used as a learning tool for students.

4.2 VISUAL STUDIO CODE

Visual Studio Code is an Integrated Development Environment (IDE) created by Microsoft for Windows, Linux and Mac operating system. It is a very advanced IDE with a rich set of features such as debugger, syntax highlighter, intelligent code completion, refactoring of code and embedded Git. Now a days, Visual Studio Code is the most popular IDE among developers.

The first version of Visual Studio Code was released by Microsoft on 14 April 2016. Since then, it has gained popularity and is now considered the most used code editor. The popularity of Visual Studio Code can be attributed to the fact that it supports almost all the languages with various application in a single platform. Another important feature of Visual Studio Code is that it is fully embedded with support for Git. As Git has become very popular among developers, so this feature makes it easier for a developer to integrate the code that they write in visual studio code into Git.

One of the outstanding feature that Visual Studio Code introduced was the use of extensions. The extensions are stored in a central repository and they can be installed whenever a user needs to use them. The use of extensions makes Visual Studio Code a very dynamic IDE because a user can add even more features and support more languages just by installing the required extension.

5. IMPLEMENTATION

5.1 EXPERIMENTAL SETUP FOR CAPTURING DDOS TRAFFIC

To simulate a DDoS attack, we have installed a software called Low Orbit Ion Cannon (LOIC) in the IT department lab. LOIC simulates a DDoS attack by sending a huge amount of TCP SYN request to a particular IP address. We have send the packets from one computer in the lab to another computer. To make sure that the packets captured actually simulate packets coming from multiple computers, we have used multiple threads. This ensures that at any point of time, all the packets in each thread are parallely transferred from one computer to the other. This is done so that this simulation comes as close as possible to that of a real world DDoS attack. In the victim computer, we have installed the software “Wireshark” to capture the packets. We have filtered out the packets coming from the outside Internet so that the captured packets only contain the packets that we have generated using the LOIC software. We have done this experiment for almost two weeks in the lab and captured a huge amount of network traffic data.

However, the amount of data captured was still not big enough to give a good accuracy for our machine learning algorithms. Hence, in addition to this we have also downloaded some packet capture files online which captured some DDoS traffic. Adding this data to our dataset ensured that we get a big dataset as well as a more variable dataset which contained different types of DDoS attacks.

5.2 CAPTURING THE NORMAL PACKETS

To capture normal packets i.e packets which are not part of a DDoS traffic, we have installed wireshark on our computers and captured the packets flowing in and out of our computer when we are usually surfing the Internet for our day to day use. We have used the packet sniffing application “Wireshark” to capture the packets. Wireahark is a very popular software which is used to capture the details of each packets flowing in an interface of a computer. It captures the data contained in all of the OSI layers and their respective headers. It is these values contained in the headers that we have extracted information from in order to create the features of our classification algorithms. After capturing the packets, we have saved them as a .pcap file. pcap files are data files which contain the packet data of a network. We have captured packets at various different intervals and then combined them as a single .pcap file using the merge feature in Wireshark. We have filtered out IPv6 packets and only taken the IPv4 packets to make the data more consistent and easy to manipulate in order to extract information from the packets. Since IPv6 contains completely different structure of header and fields, it would not have been possible to extract the same features that we have extracted from the IPv4 packets. For now, filtering out the IPv6 packets do not make much of a difference because the amount of IPv6 packets are almost negligible but in the future if the number of IPv6 packets increases, then we might need to write an alternative code to extract features from the IPv6 packets as well. And similar to capturing the DDoS traffic, in order to bring more variability to the dataset and make the dataset even bigger, we have also downloaded some pcap files online which captures normal traffic.

5.3 EXTRACTING THE FEATURES

To extract features from the pcap files, we have created a time window of 500 milliseconds. For each time window (i.e for every 500 ms), we have extracted and accumulated the features from all the packets captured in that time window. Then, we have added a new row in the dataset with the data values of each features. Thus, the total number of rows in the dataset is directly dependent on the amount of time we have captured the dataset. Since we have set the time window as 500 ms, the total number of rows in our dataset will be double the number of second we have captured the dataset. The advantage of creating a time window for feature extraction is that it gives a more clearer idea about the kind of network traffic. And since our main aim is to identify DDoS attack, creating a time window will give much more relevant features. As we know, a DDoS attack occurs when a stream of packets are send in a short period of time in order to flood a server with huge number of requests and make it unable to process all the requests. Hence, it makes more sense to create a time window and extract the features from the packets rather than extracting features from one packet at a time.

To extract the features, we have written code using Python language. We have used the Object Oriented Programming paradigm to structure our code. There are four stages through which features are extracted from the pcap file and then stored in a csv file. In the first stage, we have converted the pcap files into JSON objects representing key value pairs. In the second stage, we have extracted the various information related to the transport layer. In the third stage, we have extracted the information related to the application layer. And finally in the fourth stage, we have calculated the numerical values of the features. The total number of features that we have created are 16. We have extracted the following features from each time window:

(i) tcp_frame_length: It is the total length of a TCP packet in the data link layer.
(ii) tcp_ip_length: It is the total length of a TCP packet in the network layer.

(iii) tcp_length: It is the total length of a TCP packet in the transport layer.

(iv) udp_frame_length: It is the total length of a UDP packet in the data link layer.

(v) udp_ip_length: It is the total length of a UDP packet in the network layer.

(vi) udp_length: It is the total length of a UDP packet in the transport layer.

(vii) num_tls: It is the total number of TLS connections in a particular time window.

(viii) num_http: It is the total number of HTTP packets in a particular time window.

(ix) num_dhcp: It is the total number of DHCP packets in a particular time window.

(x) num_dns: It is the total number of DNS packets in a particular time window.

(xi) num_tcp: It is the total number of TCP packets in a particular time window.

(xii) num_udp: It is the total number of UDP packets in a particular time window.

(xiii) num_igmp: It is the total number of IGMP packets in a particular time window.

(xiv) num_connection_pairs: It is the total number of connected pairs in a particular time window.

(xv) num_ports: It is the total number of ports in a particular time window.

(xvi) num_packets: It is the total number of packets in a particular time window.

5.4 FINDING NUMBER OF TRANSPORT LAYER PROTOCOLS

In each packet, the protocol used in the transport layer is present in the “Protocol” field of the IP header. There are only two transport layer protocols: Transport Layer Protocol (TCP) and User Datagram Protocol (UDP). Each packet contains any one of these two protocols in the “Protocol” field in the IP header. For each packet captured, we have stored the value present in the protocol field of the IP header and stored them in a list in python. Then, we have created a dictionary which contains the key as the protocol name (TCP and UDP) and the value as the number of time it appears. So, now the dictionary contains the frequency of each protocol. We have used this to plot a bar graph which gives a better visual representation of the statistics. We have used the matplotlib.pyplot library in python to plot the bar graph.

5.5 FINDING NUMBER OF APPLICATION LAYER PROTOCOLS

The information about the application layer protocol used is not directly present in each packet. However, we can use the source port and destination port to get an idea about the application layer protocol used. Each application layer protocol is assigned a corresponding port number. Hence we can use this to find the application layer protocol that a packet is using. The source port and destination port addresses are present in the TCP or UDP header. We have extracted the source port and destination port of each packets and stored them in a list. Now to find the application layer protocol we have written if else statements in python for the following port numbers:

(i) 80 - HTTP

(ii) 443 - HTTPS

(iii) 53 - DNS

(iv) 67 - DHCP

(v) 68 - BOOTP

(vi) 21 - FTP

5.6 FINDING NUMBER OF PACKETS FROM VARIOUS COUNTRIES

To find the location of each IP address, we have used the “GeoLiteCity” database. The “GeoLiteCity” database contains various information corresponding to each IP address such as the latitude, longitude, country, etc. We have stored the source IP address and destination IP address present in the IP header of each packet in a list in python. Then, using the “GeoLiteCity” database we have found the country corresponding to each IP address. Then, we have created a dictionary in order to find the frequency of packets coming from various countries. Finally, we have plotted a bar graph using the matplotlib.pyplot library in order to give better visual representation.

5.7 PLOTTING THE MAP SHOWING THE NETWORK TRAFFIC

We have used the “Google My Maps” feature to plot the map showing the distribution of network traffic coming from different parts of the world. The “Google My Maps” feature can be used to upload a file containing the latitude and longitude of various locations and plot a map corresponding to those locations. We have stored the source IP address and destination IP address from each packets and found their corresponding latitude and longitude using the “GeoLiteCity” database. Then, using python we have created a .kml file which we have uploaded in Google My Maps in order to plot the map. The kml file contains various information such as the colour and width of the lines in the map as well as the latitude and longitude. After uploading the kml file, the map showing the distribution of network traffic from various regions of the world is plotted.

5.8 CLASSIFYING THE PACKETS USING VARIOUS ALGORITHMS

We have used various machine learning algorithms to classify the network traffic into either normal or attack traffic. We have used the 16 features that we have extracted as the input parameter in the classification.

We have used six of the most popular binary classification algorithms in order to find their accuracy for this problem. The classification algorithms we have used are:

(i) Logistic Regression
(ii) Decision Tree Classifier
(iii) Random Forest Classifier
(iv) K Neighbors Classifier
(v) Naive Bayes Classifier
(vi) Standard Vector Machine

We have imported and used the sklearn library in Python to implement the various machine learning algorithms. For each classifier, we have calculated their performance by finding their corresponding accuracy, precision, recall, f1 score and confusion matrix.

5.9 FINDING THE IMPORTANCE OF THE FEATURES

We have also tried to find the relative importance of each of the 16 features in the classification process. This gives an interesting statistics about the usefulness or relevance of all the features that we have extracted in the classification algorithm. To calculate the feature importance, we have used the “feature_importances_” property in Python. We have chosen the random forest classifier model to calculate feature importances because it gave the best accuracy compared to other classification algorithms. Each feature is given a value between 0 and 1 depending on their relative importance. Higher the value of a particular feature means that the feature is more important. Similarly, lower value means that the feature is less important. Using the results of feature importance, we have also removed some features which had almost no effect on the classification.

6. RESULTS

6.1 DISTRIBUTION OF TRANSPORT LAYER PROTOCOLS

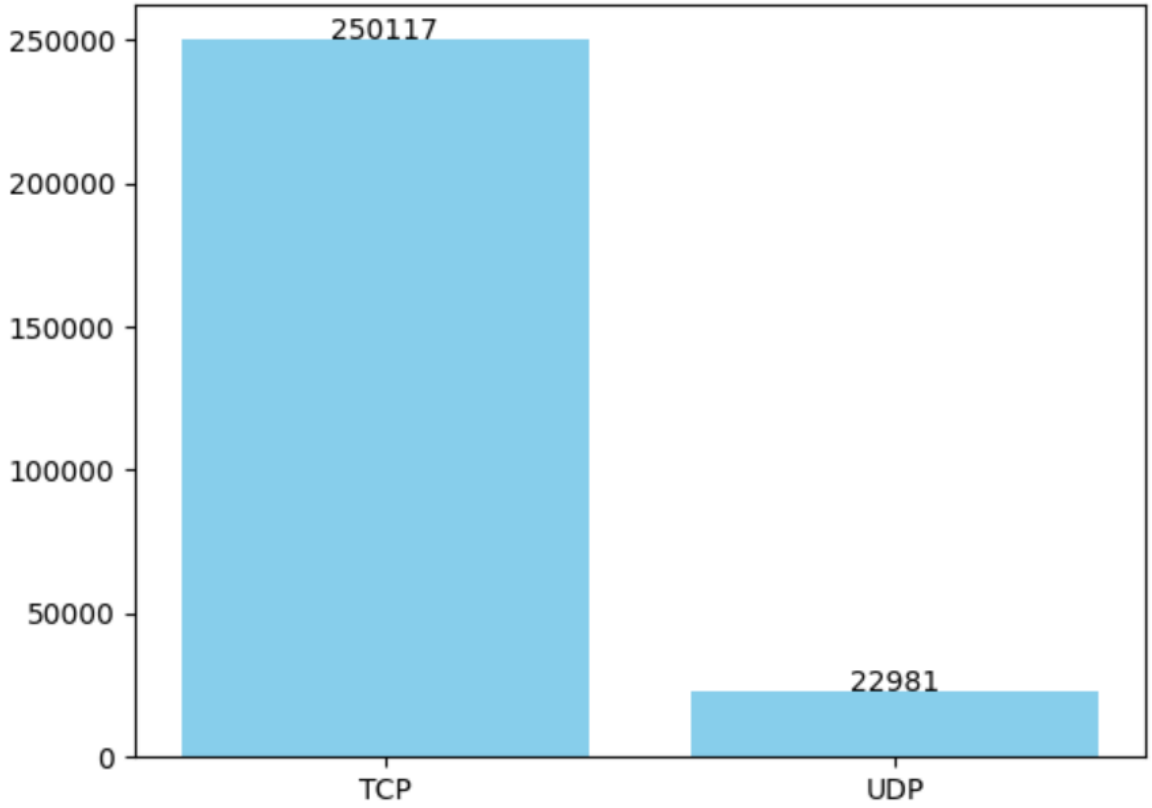

Fig 6.1: Bar graph of transport layer protocols

The amount of packets containing the various transport layer protocols are:

Transport Layer Protocols	Number of Packets	Percentage
Transmission Control Protocol (TCP)	250117	91.58%
User Datagram Protocol (UDP)	22981	8.42%

Table 6.1: Distribution of transport layer protocols

6.2 DISTRIBUTION OF APPLICATION LAYER PROTOCOLS

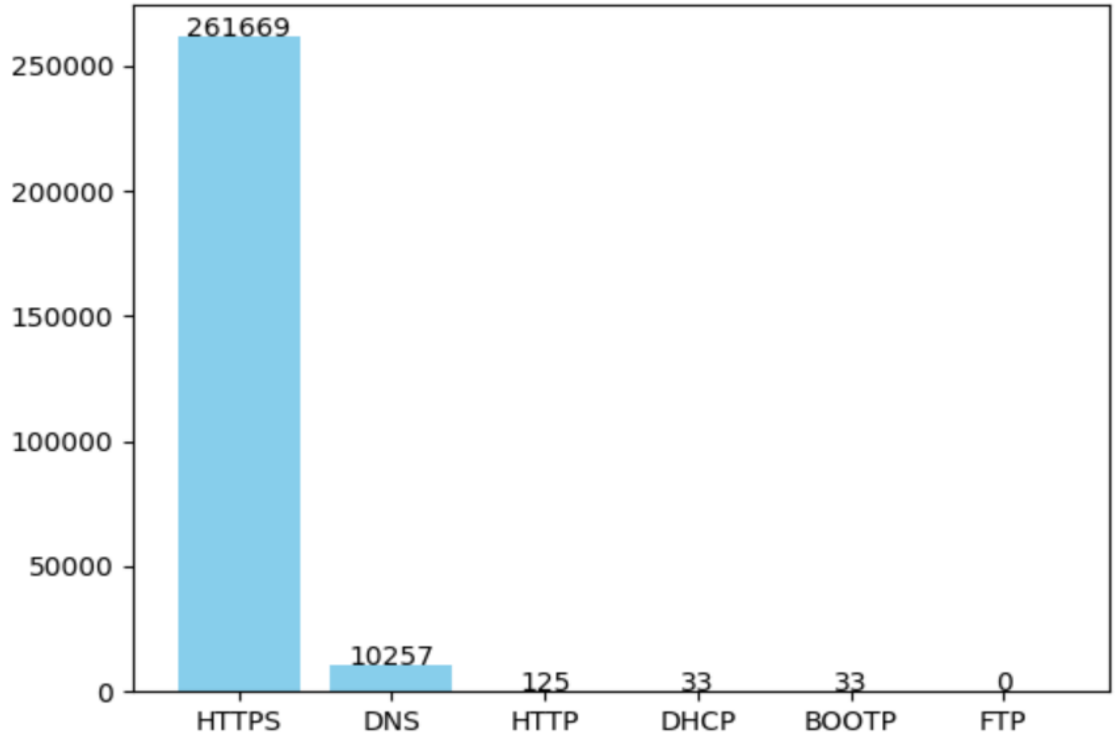

Fig 6.2: Bar graph of application layer protocols

The amount of packets containing the various application layer protocols are:

Application Layer Protocols	**Number of Packets**	**Percentage**
HTTPS	261669	96.16%
DNS	10257	3.77%
HTTP	125	0.05%
DHCP	33	0.01%
BOOTP	33	0.01%
FTP	0	0.00%

Table 6.2: Distribution of application layer protocols

6.3 DISTRIBUTION OF PACKETS FROM VARIOUS COUNTRIES

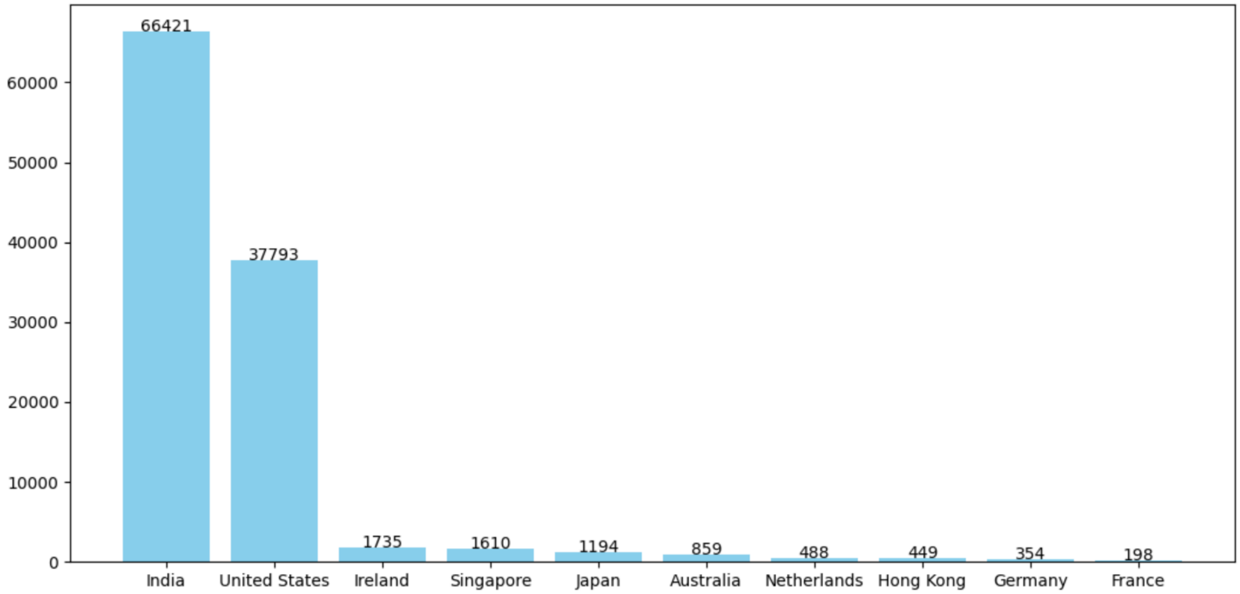

Fig 6.3: Bar graph of protocols from various countries

The amount of packets coming from various countries of the world are:

Countries	Number of Packets	Percentage
India	66421	59.78%
United States	37793	34.02%
Ireland	1735	1.56%
Singapore	1610	1.44%
Japan	1194	1.07%
Australia	859	0.77%
Netherlands	488	0.44%
Hong Kong	449	0.40%
Germany	354	0.32%
France	198	0.18%

Table 6.3: Distribution of protocols from different countries

6.4 NETWORK TRAFFIC MAP

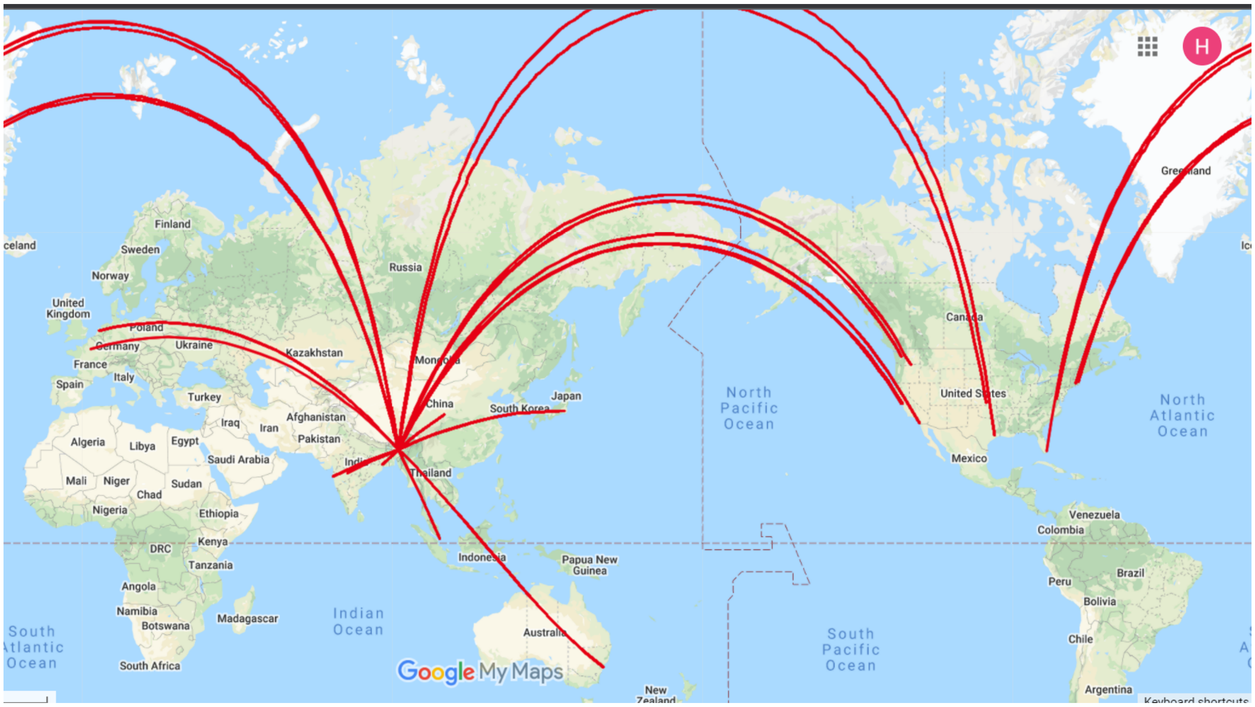

Fig 6.4: Map showing the distribution of network traffic

This map has been plotted by uploading a kml file in Google My Maps. The kml file contains the latitude and longitude corresponding to each IP address. This map gives a good idea about the amount of traffic coming from different parts of the world. However, Google My Maps does not allow to upload a large file. So, we had to restrict the number of packets to 2000.

6.5 ACCURACY OF LOGISTIC REGRESSION

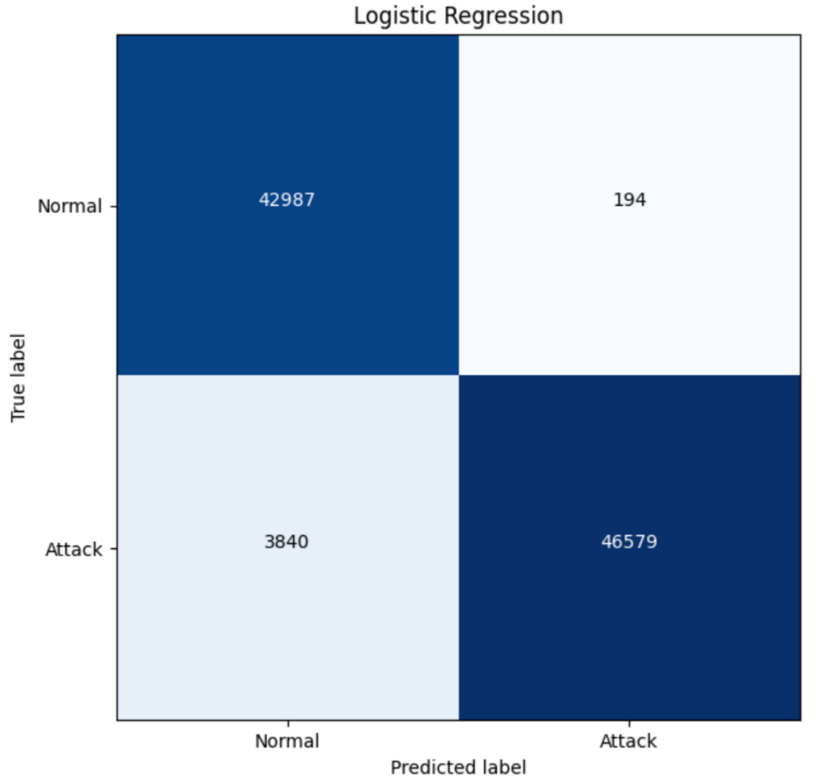

Fig 6.5: Confusion matrix of logistic regression

The results of the logistic regression are:

Parameter	**Value**
True Positive	46579
True Negative	42987
False Positive	194
False Negative	3840
Recall	0.9238
Precision	0.9958
Accuracy	0.9569
F1 Score	0.9902

Table 6.4: Various accuracy parameters of Logistic Regression

6.6 ACCURACY OF DECISION TREE CLASSIFIER

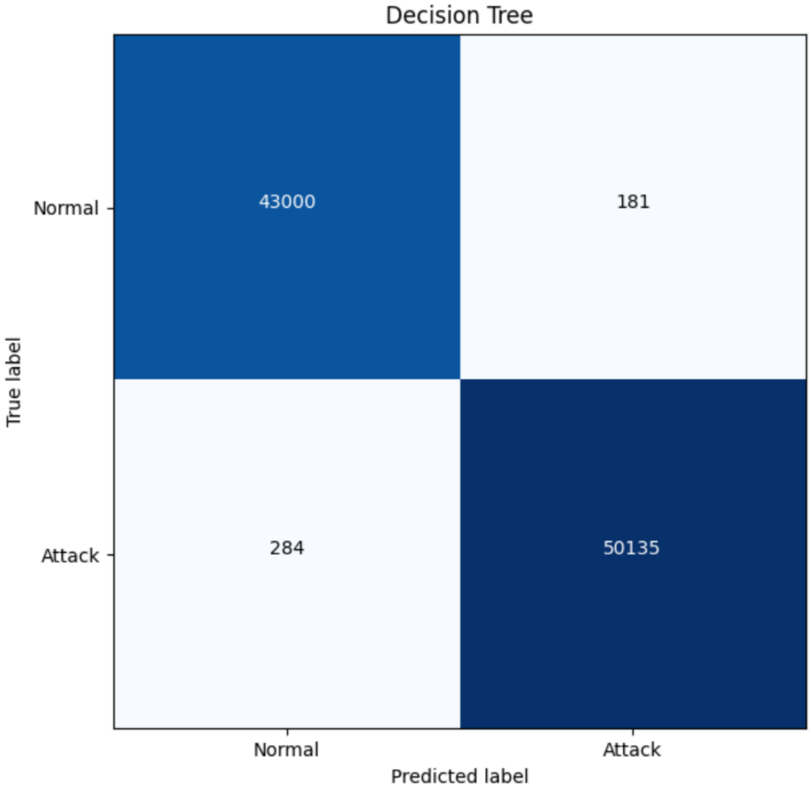

Fig 6.6: Confusion matrix of decision tree classifier

The results of the decision tree classifier are:

Parameter	**Value**
True Positive	50135
True Negative	43000
False Positive	181
False Negative	284
Recall	0.9943
Precision	0.9964
Accuracy	0.9951
F1 Score	0.9954

Table 6.5: Various accuracy parameters of Decision Tree Classifier

6.7 ACCURACY OF RANDOM FOREST CLASSIFIER

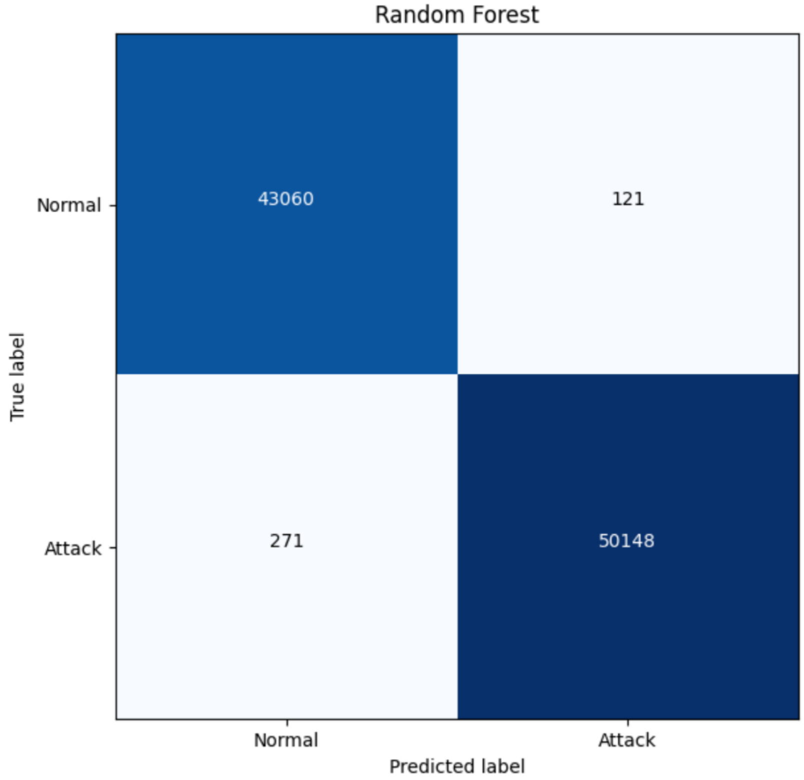

Fig 6.7: Confusion matrix of random forest classifier

The results of the random forest classifier are:

Parameter	**Value**
True Positive	50148
True Negative	43060
False Positive	121
False Negative	271
Recall	0.9946
Precision	0.9976
Accuracy	0.9958
F1 Score	0.9961

Table 6.6: Various accuracy parameters of Random Forest Classifier

6.8 ACCURACY OF K NEIGHBOURS CLASSIFIER

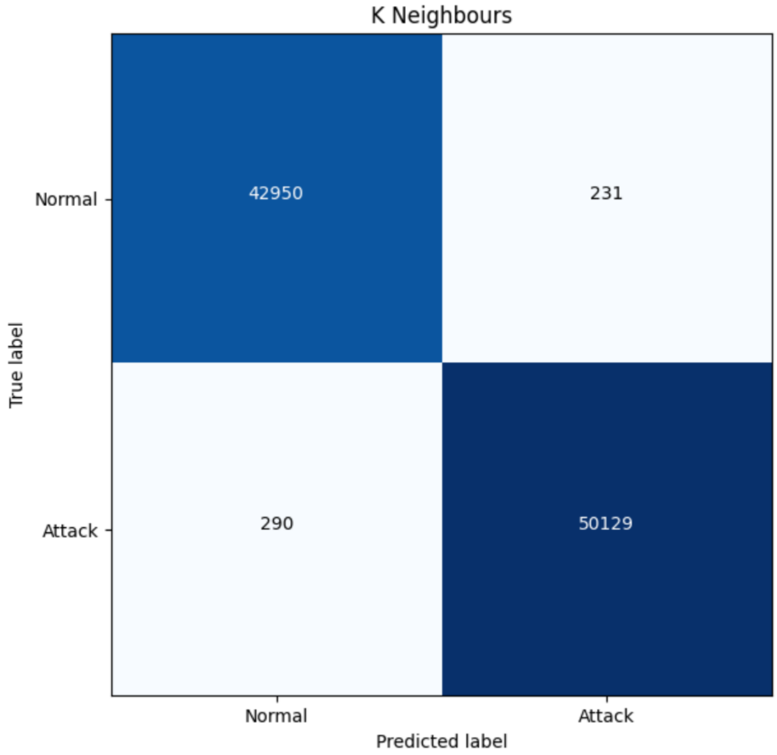

Fig 6.8: Confusion matrix of k neighbors regression

The results of the K Neighbours classifier are:

Parameter	**Value**
True Positive	50129
True Negative	42950
False Positive	231
False Negative	290
Recall	0.9942
Precision	0.9954
Accuracy	0.9944
F1 Score	0.9948

Table 6.7: Various accuracy parameters of K Neighbours Classifier

6.9 ACCURACY OF NAIVE BAYES CLASSIFIER

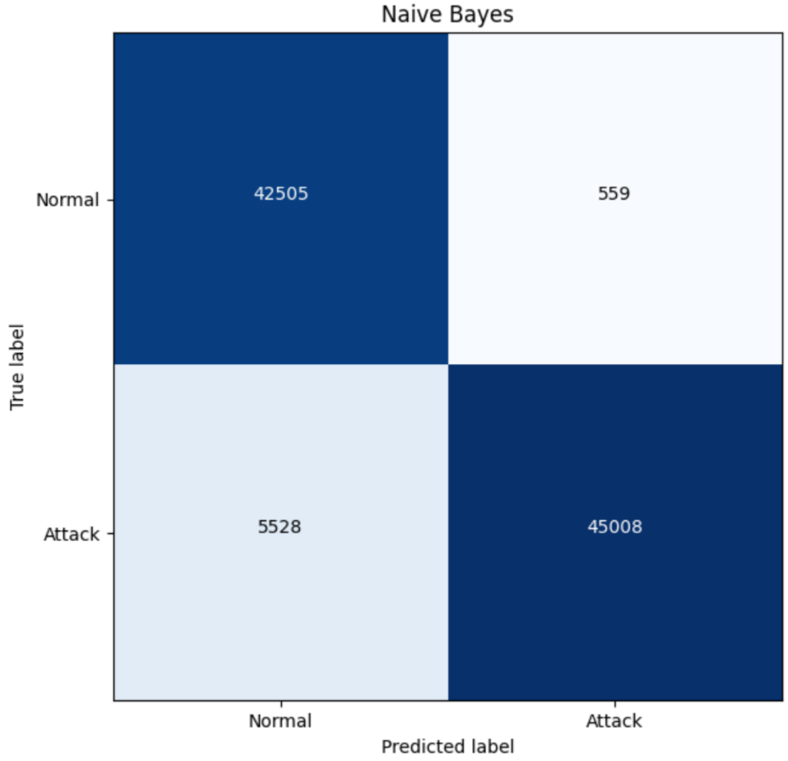

Fig 6.9: Confusion matrix of naive bayes classifier

The results of the Naive Bayes classifier are:

Parameter	**Value**
True Positive	45008
True Negative	42505
False Positive	559
False Negative	5528
Recall	0.8906
Precision	0.9877
Accuracy	0.9349
F1 Score	0.9367

Table 6.8: Various accuracy parameters of Naive Bayes Classifier

6.10 ACCURACY OF STANDARD VECTOR MACHINE

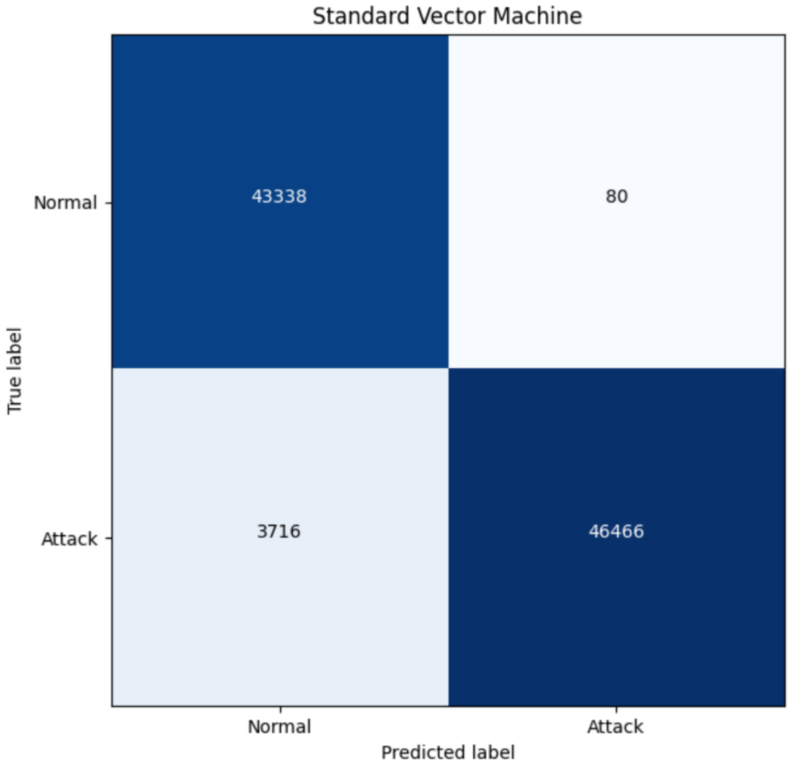

Fig 6.10: Confusion matrix of standard vector machine

The results of the Standard Vector Machine are:

Parameter	**Value**
True Positive	46466
True Negative	43338
False Positive	80
False Negative	3716
Recall	0.9259
Precision	0.9983
Accuracy	0.9432
F1 Score	0.9607

Table 6.9: Various accuracy parameters of Standard Vector Machine

6.11 RELATIVE IMPORTANCE OF THE FEATURES

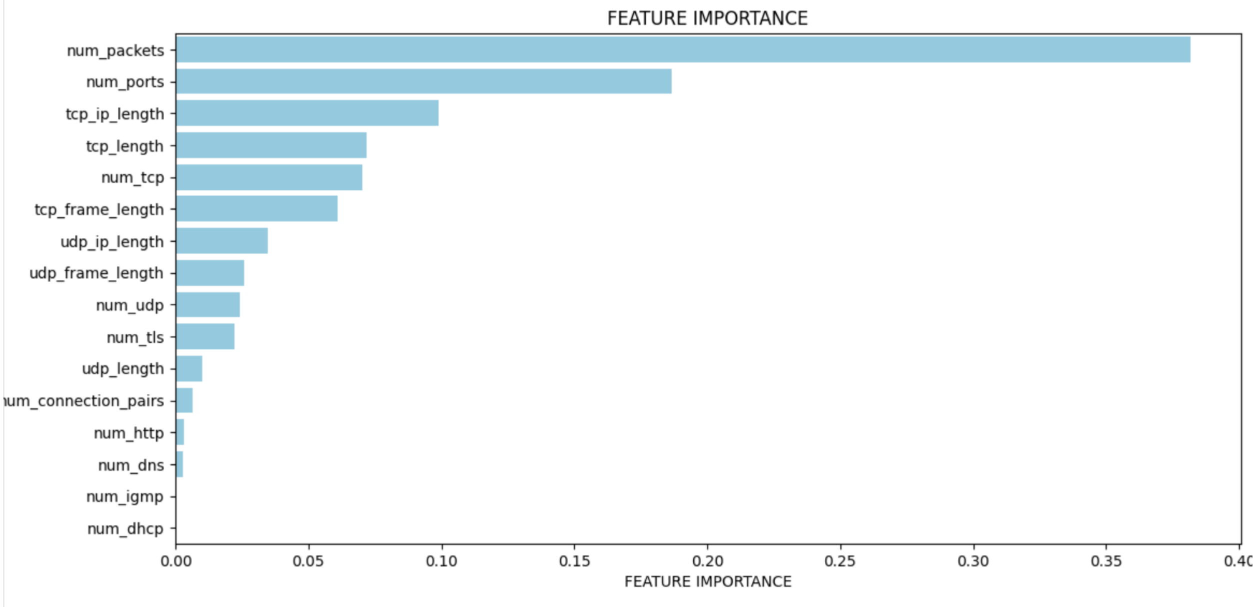

Fig 6.11: Bar graph of feature importance

Features	**Score**
tcp_frame_length	0.0608
tcp_ip_length	0.0989
tcp_length	0.0719
udp_frame_length	0.0258
udp_ip_length	0.0348
udp_length	0.0101
num_tls	0.0221
num_http	0.0031
num_dhcp	0.0001
num_dns	0.0031
num_tcp	0.0703

num_udp	0.0241
num_igmp	0.0001
num_connection_pairs	0.0061
num_ports	0.1867
num_packets	0.3816

Table 6.10: Importance of various features

From the results of feature importance, we can conclude that the feature num_packets is the most important or prominent feature that has been used by the classification algorithms to classify the data. It is not surprising because, in a DDoS attack, the number of packets at a given time interval will definitely be high as compared to normal traffic. However, there are also some other features like num_ports, tcp_length, udp_length who have also played a very substantial role in the classification. The feature importance of features like num_http, num_dns, num_igmp and num_dhcp are very low most probably because the number of HTTP, DNS, IGMP and DHCP packets in our datasets are very less in number.

7. CONCLUSION

In this project, we have tried to find solutions to mitigate Distributed Denial of Service (DDoS) attack. DDoS attack is one of the most prominent and costliest network security issue in the modern world. This attack keeps happening very frequently to some of the most popular website in the world and it leads to enormous loss financially. First, we have captured the packets flowing in a network and derived various statistics about the network traffic like the proportion of transport and application layer protocols, and the distribution of network traffic coming from different regions of the world. Then, we tried to simulate a DDoS attack in a lab environment and captured the traffic using wireshark. The most interesting and challenging part of this project was the feature extraction. In feature extraction, we have taken the packet capture files as input and created a time window such that it calculates the total count of the features at that particular time window. Calculating feature values by creating a time window gives a much better reflection of the network traffic in our dataset. Finally, we have classified the network traffic with various machine learning algorithms into either normal or attack traffic using the features we have extracted. We have used six of the most popular machine learning algorithms for binary classification. From the results, we have found that random forest classifier and k nearest neighbours classifiers gave the best result although random forest classifier was much faster than k nearest neighbours in terms of execution time. On the other hand, naive bayes classifier performed relatively poorly in the classification. One reason for this might be because naive bayes classifier assumes that the features are independent of each other which is not true in our case. Standard Vector Machine gave the least number of False Positive compared to other algorithms but the number of False Negative was a bit high. We have also created a feature importance graph using random forest classifier. The result of the feature importance graph showed that the number of packets in a time window is the most significant features through which the classification between normal and attack traffic is done.

8. FUTURE SCOPE

In this project, we have created a machine learning model which classifies normal and attack traffic with a fairly good accuracy. However, it will be even better if this model can be used in some real world server to test its ability to detect ddos attack. If more real world data of ddos attacks are generated, and given as input to our classification model, then it will make the model even more accurate. Further, the feature importance graph that we have generated can be used in various applications to get some interesting statistics about the correlations of the features with different kinds of network traffic.

Another future work that can be done on this topic is using unsupervised learning to classify the network traffic. In our project, we have used supervised learning to classify the network traffic by labelling the packets into either normal or attack traffic. So, it will be an interesting study to look at how an unsupervised learning algorithms work for this problem using the same features that we have extracted.

REFERENCES

[1] Jiahui Chen, Joe Breen, Jeff M. Phillips, Jacobus Van der Merwe, "Practical and configurable network traffic classification using probabilistic machine learning", *Cluster Computing*, 2021.

[2] T. P. Fowdur, B. N. Baulum, Y. Beeharry, "Performance analysis of network traffic capture tools and machine learning algorithms for the classification of applications states and anomalies", International Journal of Information Technology, 2020.

[3] Pranita Mane, Yash Parkar, Jaideep Patel, Viral Sanghavi, Amey Walanje, "Traffic Classification Using Machine Learning", SSRN Electronic Journal , 2019.

[4] Ons Aouedi, Kandaraj Piamrat, Benoît Parrein, "Performance evaluation of feature selection and tree-based algorithms for traffic classification", Communications Workshops (ICC Workshops) 2021 IEEE International Conference on, pp. 1-6, 2021.

[5] Yoga Durgadevi Goli, R Ambika, "Network Traffic Classification Techniques-A Review", *Computational Techniques Electronics and Mechanical Systems (CTEMS) 2018 International Conference*

[6] Y. N. Soe, P. I. Santosa, and R. Hartanto, “DDoS Attack Detection Based on Simple ANN with SMOTE for IoT Environment,” Proc. 2019 4th Int. Conf. Informatics Comput. ICIC 2019, pp. 0–4, 2019

[7] Zhong Fan and Ran Liu,” Investigation of Machine Learning Based Network Traffic Classification”, International Symposium on Wireless Communication Systems (ISWCS) 2017, pp1-6.

[8] T. Nguyen and G. Armitage, “A survey of techniques for Internet traffic classification using machine learning”, IEEE Communications Surveys & Tutorials, Vol. 10, No. 4, fourth quarter 2018, pp 56-76.

[9] Muhammad Shafiq, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, Nabin Kumar Karn, Foudil Abdessamia, “Network Traffic Classification Techniques and Comparative Analysis Using Machine Learning Algorithms”, 2nd IEEE International

[10] Fatih Ertam, Ilhan Firat Kilinçer, Orhan Yaman,”Intrusion Detection in Computer Networks via Machine Learning Algorithms”, International Artificial Intelligence and Data Processing Symposium (IDAP),2017,pp 1-4