DDOS Detection Using Machine Learning Model

Faculty of Engineering and Technology, Jadavpur University in the fulfilment of the requirements for the degree of Master of Computer Science and Engineering

Submitted by

Sourav Bauri

Registration Number: 154141 of 2020-22

Class Roll Number: 002010502017

Examination Roll No: M4CSE22017

Under the Supervision of

Dr. Sarmistha Neogy

Professor, Dept. of Computer Science and Engineering

Jadavpur University

Dept. of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

June, 2022

Declaration of Originality &
Compliance of Academic Ethics

I hereby declare that this thesis contains literature survey and original research work done by me, as part of my MCSE studies. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

Name: SOURAV BAURI

Registration No: 154141 of 2020-22

Class Roll No: 002010502017

Examination Roll No: M4CSE22017

Thesis Report Title: DDOS detection using Machine learning model

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

_____

SOURAV BAURI

Department of Computer Science and Engineering Faculty of Engineering and Technology

Jadavpur University

To Whom It May Concern,

This is to certify that SOURAV BAURI, Registration Number: 154141 of 2020-22, Class Roll Number: 002010502017, Examination Roll Number: M4CSE22017, a student of MCSE, from the Department of Computer Science & Engineering, under the Faculty of Engineering and Technolog Jadavpur University has done a thesis report under my supervision, entitled as "DDOS detection using Machine Learning Model". The thesis is approved for submission towards the fulfilment of there requirements for the degree of Master of Computer Science and Engineering, from the Department of Computer Science & Engineering, Jadavpur University for the session 2020-22.

_____

Dr. Sarmistha Neogy
(Supervisor) Professor
Department of Computer Science and Engineering Jadavpur University

Countersigned

_____

Dr. Anupam Sinha
(Head of the Department)
Professor, Department of Computer Science and Engineering
Jadavpur University

_____

Dr. Chandan Mazumdar
Dean, Faculty of Engineering and Technology
Jadavpur University

Certificate of Approval

(Only in case the thesis report is approved)

The forgoing thesis is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve this thesis only for the purpose for which it is submitted.

_____ _____

Signature of the Examiner Signature of the Examiner

Date:_____ Date: _____

Acknowledgment

First of all, I would like to express my profound gratitude to my supervisor Prof. Sarmistha Neogy, Professor, Department of Computer Science and Engineering, Jadavpur University Kolkata for her outstanding and extraordinary guidance during my thesis work. I have gained immense benefit and knowledge while working under her guidance. I stress that without her constant motivation and devotion I would have failed to carry out this work within the stipulated time. I am immensely grateful and humbled by the facilities provided by her in order to ensure the successful completion of my thesis work.

I would also like to thank our Head of the Computer Science & Engineering department, Dr. Anupam Sinha for his excellent guidance and kind cooperation during the period of my study at Jadavpur University, Kolkata.

Let me take this opportunity to thank all the other faculties of the Department of Computer Science & Engineering who have directly or indirectly cooperated and supported me during the course of my study in this department. The thanks giving also extends to the staff and librarians of Jadavpur University who supported me in different activities related to my academic career during the course of study.

Finally, I would like to thank my parents, my friends and my acquaintances for their support and love in my life in order to pursue my life goals. Finally, I would congratulate myself for availing the opportunity to study at this renowned institution of Jadavpur University.

Regards,

Sourav Bauri

Department of Computer Science & Engineering

Master of Computer Science and Engineering

Jadavpur University

Abstract

The detection of anomaly traffic has become one of the principal directions in the field of network security intending to identify the attacks based on the specific deviations of the captured traffic. The cybercrime rate is increasing, capabilities of the cyber terrorists and hackers are growing at a higher rate. Today there is a requirement for the innovation and exploration for the mitigation of DDoS attacks. One of the most popular attacks in different layers of the network is Distributed Denial of Service (DDoS) a malicious try to interrupt regular traffic of a directed server, service, or network by irresistible to the target of its nearby infrastructure with anomalous flood traffic to the legitimate servers. An attacker usually targets for gaining access to virtual things like servers, applications, networks and sometimes targets particular transactions in an application. The detection of anomalous network traffic is one of the main challenging problems. which can harm a legitimate user very immensely. In simple language a huge number of false packets from different servers or different systems or software or bots etc sent by the hacker to make a traffic jam on the server. These false packets consist of the same features as the original, which is around 41. These Feature values are different from the original features value which are made by hackers or hacking tools. These false packets are sent to the server in a huge number at a time to make the server busy. Server get busy to responding those false packets. Legitimate users are unable to access the server. Our dataset name is KDDCUP99 which is available on the internet. To detect this type of attack basic Machine Learning Models are good to go. Our main aim in this paper is to make a combined machine Learning model using less features to get a higher accuracy than the previous paper model. Here we have been used Three Classification Machine Learning Algorithm KNN, Random Forest and Naive Biase.

Contents

Table of Contents

List of Figures

List of tables

Chapter 1

Introduction

1.1 Overview:

Now we are living in this era where the internet is rapidly growing, and IoT Devices are increasing day by day. At the same time there is a security concern. Hackers are always open to attack and steal our valuable information. One of the Powerful attack is DDoS Attack which stands for Distributed Denial Of Service. This is not a central attack, that mean it is distributed in nature. It is distributed across different sources, computers, bots etc. It is a malicious attempt to affect availability of a specific target system, such as a website or an application for its legitimate users. Attacker send large number of false packets or requests from different machine to the target website or application, these huge number of packets or requests overwhelms the target system or server due to which its performance decreases and stop responding to its legitimate users. There are Different types of DDoS Attack Like SYN Flood, ICMP Flood, HTTP Flood etc. To detect these types of DDoS Attack Machine Learning models are good enough. Now we are living in this era where the internet is rapidly growing, and IoT Devices are increasing day by day. At the same time there is a security concern. Hackers are always open to attack and steal our valuable information. One of the Powerful attack is DDoS Attack which stands for Distributed Denial Of Service. This is not a central attack, that mean it is distributed in nature. It is distributed across different sources, computers, bots etc. It is a malicious attempt to affect availability of a specific target system, such as a website or an application for its legitimate users. Attacker send large number of false packets or requests from different machine to the target website or application, these huge number of packets or requests overwhelms the target system or server due to which its performance decreases and stop responding to its legitimate users. There are Different types of DDoS Attack Like SYN Flood, ICMP Flood, HTTP Flood etc. To detect these types of DDoS Attack Machine Learning models are good enough. Since the first DoS attack was launched in 1974, DDoS attacks and other DoS attacks have remained among the most persistent and damaging cyber-attacks. These attacks reflect hackers' frustratingly high levels of tenacity and creativity—and create complex and dynamic challenges for anyone responsible for cyber security. The first-ever DoS attack occurred in 1974 courtesy of David Dennis—a 13-year-old student at

University High School, located across the street from the Computer-Based Education Research Laboratory (CERL) at the University of Illinois Urbana-Champaign. David recently learned about a new command that could be run on CERL's PLATO terminals. PLATO was one of the first computerized shared learning systems, and a forerunner of many future multi-user computing systems. Called "external" or "ext," the command was meant to allow for interaction with external devices connected to the terminals. However, when run on a terminal with no external devices attached it would cause the terminal to lock up—requiring a shutdown and power-on to regain functionality.

Curious to see what it would be like for a room full of users to be locked out at once, he wrote a program that would send the "ext" command to many PLATO terminals at the same time. Dennis went over to CERL and tested his program—, which succeeded in forcing all 31 users to power off at once. Eventually the acceptance of a remote "ext" command was switched off by default, fixing the problem.

During the mid to late 1990s, when Internet Relay Chat (IRC) was first becoming popular, some users fought for control of non-registered chat channels, where an administrative user would lose his or her powers if he or she logged off. This behavior led hackers to attempt to force users within a channel to all log out, so they could enter the channel alone and gain administrator privileges as the only user present. These "king of the hill" battles—in which users would attempt to take control of an IRC channel and hold it in the face of attacks from other hackers—were fought using very simple bandwidth-based DoS attacks and IRC chat floods. One of the first large-scale DDoS attacks occurred in August 1999, when a hacker used a tool called "Trinoo" to disable the University of Minnesota's computer network for more than two days. Trinoo consisted of a network of compromised machines called "Masters" and "Daemons," allowing an attacker to send a DoS instruction to a few Masters, which then forwarded instructions to the hundreds of Daemons to commence a UDP flood against the target IP address. The tool made no effort to hide the Daemons' IP addresses, so the owners of the attacking systems were contacted and had no idea that their systems had been compromised and were being used in a DDoS attack.

Other early tools include “Stacheldraht” (German for barbed wire), which could be remotely updated and support IP spoofing, along with “Shaft” and “Omega”, tools that could collect attack statistics from victims. Because hackers were able to get information about their DDoS attacks, they could better understand the effect of certain types of attacks, as well as receive notification when a DDoS attack was detected and stopped. Once hackers began to focus on DDOS attacks, DoS attacks attracted public

attention. The distributed nature of a DDoS attack makes it significantly more powerful, as well as harder to identify and block its source. With such a formidable weapon in their arsenals, hackers began to take on larger, more prominent targets using improved tools and methods.

By the new millennium, DDoS captured the public’s attention. In the year 2000, various businesses, financial institutions and government agencies were all brought down by DDoS attacks. Shortly after, DNS attacks began with all 13 of the Internet’s root domain name service (DNS) servers being attacked in 2002. DNS is an essential Internet service, as it translates host names in the form of uniform resource locators (URLs) into IP addresses.

In effect, DNS is a phonebook maintaining a master list of all Internet addresses and their corresponding URLs. Without DNS, users would not be able to efficiently navigate the Internet, as visiting a website or contacting a specific device would require knowledge of its IP address.
Therefore, this is clearly understandable that there is a great need of paying more attention towards creating better network security systems.

Any action that seeks to compromise the availability, confidentiality and integrity of a system is referred as attack [3]. The attacks are focused over the vulnerabilities of the user on the network by defying user access rights and gaining unauthorized access.

Hence, the Network Security has become a serious issue for all the network users around the world. To prevent our systems and data from theft and attacks beforehand, the

Intrusion Detection System (IDS) was developed.

Table 1:World internet usuage and population statistics

World Regions	Population	Internet Users Dec. 31, 2000	Internet Users Latest Data	Growth 2000-2015
Africa	1,158,353,014	4,514,400	318,633,889	6,958.2 %
Asia	4,032,654,624	114,304,000	1,405,121,036	1,129.3 %
Europe	827,566,464	105,096,093	582,441,059	454.2 %
Middle East	236,137,235	3,284,800	113,609,510	3,358.6 %
North America	357,172,209	108,096,800	310,322,257	187.1 %
Latin America / Caribbean	615,583,127	18,068,919	322,422,164	1,684.4 %
Oceania / Australia	37,157,120	7,620,480	26,789,942	251.6 %
WORLD TOTAL	**7,264,623,793**	**360,985,492**	**3,079,339,857**	**753.0 %**

Thesis Structure:

The thesis is organized as follows:

Chapter 1

Gives us an overview of what the main topic of the work is. It allows the reader to have a brief understanding of the technology domain of the work.

Chapter 2

Presents a detailed literature review of the past works done in the domain of DDOS attack detection those had already implemented .

Chapter 3

Here we present several types of computer attacks, discussed DDOS attack elaborately, Machine Learning and brief overview of the datasets used in our thesis.

Chapter 4

Presents the methodology and the Machine Learning Method used in the thesis elaborately.

Chapter 5

Discusses the results that have obtained from our existing domain of work.

Chapter 6

Presents the conclusion and the future work.

Chapter 7

Includes references that have been used throughout this thesis.

Chapter 2

Literature Survey

The concept of DDoS detection using machine learning came from the paper [1]. Where the author applied KNN, Naive Bayes, Random Forest algorithm. In paper [2] implemented a hybrid approach to detect DDoS packets, 14th features are selected among 41 on dataset NSL KDD. They use five feature selection techniques as Information Gain, Gain Ratio, Chi-squared, ReliefF, Symmetrical Uncertainty. And to get best features both five techniques are combined. Intrusion Detection System was proposed Using different Machine learning algorithm like KNN, SVM, MLP, DT in [2]. MLP is one of the neural network based algorithm, so it takes huge number of data to train and it takes huge time to process. Only one algorithm may not give the best accuracy or result so we taken different algorithm. KNN can deal with less data with good accuracy even with noisy data, but Decision Tree cannot handle noisy data. So we choose KNN as one of the algorithms. In literature, there are many applications in which decision tree based classification schemes are effectively used. They develop an intrusion detection system (IDS) based on machine learning [1]. They employ genetic algorithm (GA) along with Support Vector Machine (SVM) for automatically determining the appropriate set of features. The idea is then developed into fully functional IDS. Experiments of testing the IDS on the benchmark KDD CUP 99 datasets were presented. In this research, a cross layer protocol is designed to detect denial of service attack imposed on the network by malicious nodes [4]. The Denial of Service attack on sensor networks not only diminishes the network performance but also affects the reliability of the information. Swarm intelligence, an evolutionary algorithm is used in predicting the traffic patterns and detecting malicious nodes. This novel approach helps in keeping the network functional and self-sustaining by rerouting the information. The Sybil, worm-hole and jamming attacks are overcome using this protocol design with minimal resource exploitation. The performance of the network is evaluated based on the successful packet delivery, energy consumption and average percentage of threat detection. Bao et al have proposed network intrusion detection system based on support vector machine (SVM) [5].In this approach, Anomaly intrusion detection has combined with the misuse intrusion detection based on the support vector machine. Support vector machine helps to achieve higher detection accuracy to intrusion detection system when limited prior knowledge (small

sample) is available. Zhang et al have proposed a CH-SVM method for constructing the reduced training dataset based on the convex hull of original huge dataset to reduce the time and space cost without decay on accuracy [6]. The architecture's high-performance component is provided by a server selection framework which selects the "best server" to serve a request as well as allows for an efficient multiplexing of resources across the entire cluster grid [7]. Traditional approaches assume that minimizing network hop count minimizes client latency. In contrast, the proposed mechanism for server selection collects fine-grained server load and network latency measurements and forwards requests to the server that minimizes the total of estimated network and server delays. The architecture's DDoS resilience is provided via a combination of anomaly detection and scheduling based mitigation of DDoS attacks. In contrast to prior work, the suspicion mechanism assigns a continuous valued vs. binary suspicion measure to each client session, and the scheduler utilizes these values to determine if and when to schedule a session's requests. This research, they presented a literature on classification of available mechanisms for DDoS defense [8]. These defense mechanisms are used to prevent, detect, response and tolerate the DDoS attacks. It is well known that it is very difficult to stop the DDoS attack; therefore, it would be better to maximize the fault tolerance and quality of services under variety of intrusions and attacks. In their analysis, they will discuss the merits and demerits of each mechanism over others. In addition, this research provides better understanding of the DDoS attack problem and enables a security administrator to cope up against the DDoS threat. They proposed a simple algorithm to detect the DDOS attack[8].

Types Of Computer Attacks:

Attacks categorized in two types **Passive** and **Active.**

Passive Attacks –

In the passive attack, the attacker interrupts the connection to read and analyze the information but does not cause any damage as the attacker cannot update or modify the data, which is also known as eavesdropping. The passive attacker can’t cause a noise disturbance or error bits in the original message. The passive attack looks less harmful, but it is hard to detect as the individual is unaware of the attack, and damage can be severe if the right information is obtained, e.g., bank or credit card information, meeting papers, etc. Passive attacks can be interrupted by using encryption methods. That is why the passive attack focuses on prevention.

The passive attack can be used to gather information to launch a more adverse active attack. The passive attack does not result in the loss of the system assets. It threatens data confidentiality.

Types

- **Traffic analysis:** If we encrypt the message, the information is protected even if the attacker captured the message. He monitors communication traffic to collect information about identities, locations, length of the exchanged message, and to identify the pattern of the encryption used.
- **Release of message contents:** The attacker monitors the unprotected medium like a telephonic conversation or an email that contains sensitive data

Active Attacks –

An active attack refers to hacking as the attacker not only observes the data but also causes harm to the system and its resources by directly accessing the hardware on which the data resides. The active attacker tries to cause a noise disturbance in the data transmission by putting error bits in the transmission. In an active attack, the modification and loss of the data information threaten data availability and data integrity.

An active attack is easy to detect because the individual gets a notification about the attack when an unauthorized user tries to access the data illegally. In an active attack, the modification of information takes place that results in the loss and changes to the data information and infrastructure. An active attack emphasizes detection.

Types

- **Denial of service (DoS):** The attacker sends a large number of requests to slow down the server by which the authorized user cannot get a response from the server.

  The attacker accesses the stream by blocking the legal user. This attack has broadly discussed later.

- **Session replay:**

  A sequence of data units is captured and resent by the attackers.

- **Masquerade:**

The attacker uses a false identity and behaves like an authorized user by taking the privileged status; it grabs all the data.

- **Message modification:** Some portion of the message is altered, reordered, or delayed.

Key Differences

1. An active attack is a security incident that results in loss and changes to the data information and infrastructure. In contrast, the passive attack does not result in changes to the data information but planned to gather or use that information.
2. The active attack causes harm to the system and its resources; on the other hand, in the passive attack, the resources are not damaged.
3. In an active attack, the modification of the information occurs conversely in the passive attack; the modification of the information does not take place.
4. The active attack threatens the integrity and availability of data on the flip side; the passive attack threatens the confidentiality of data.
5. The active attack focuses on detection, while the passive attack focuses on prevention.
6. An active attack is easy to detect, while a passive attack is hard to detect.
7. In an active attack, the individual gets a notification about the attack, whereas, in a passive attack, the individual is unaware of the attack.
8. In an active attack, alterations and loss of the data occur; on the other hand, in a passive attack, the target is to gain information without any change in data.
9. In an active attack, the attacker tries to cause a disturbance in the data transmission, whereas in a passive attack, the attacker can't cause disturbance or error bits in the original message.

Viruses/Worms/Trojan horses: These are programs that replicate on host machines and propagate through a network.

Viruses: Viruses are programs that reproduce themselves by attaching them to other programs and infecting them. They can cause considerable damage (e.g., erase files on the hard disk) or they

may only do some harmless but annoying tricks (e.g., display some funny messages on the computer screen). Viruses typically need human interaction (e.g., trading files on a floppy or opening e-mail attachments) for replication and spreading to other computers.

Worms: Worms are self-replicating programs that aggressively spread through a network, by taking advantage of automatic packet sending and receiving features found on many computers. Worms can be organized into several categories like traditional worms, email worms, windows file sharing worms, hybrid worms [30,31,32].

Trojan horses: Trojan horses are defined as malicious, security-breaking programs that are disguised as something benign. For example, the user may download a file that looks like a free game, but when the program is executed, it may erase all the files on the computer. Victims typically download Trojan horses from an archive on the Internet or receive them via peer-to-peer file exchange. Some actual examples include Silk Rope and Saran Wrap.

Malware: Malware is a type of application that can perform a variety of malicious tasks. Some strains of malware are designed to create persistent access to a network, some are designed to spy on the user in order to obtain credentials or other valuable data, while some are simply designed to cause disruption.

Phishing: A Phishing attack is where the attacker tries to trick an unsuspecting victim into handing over valuable information, such as passwords, credit card details, intellectual property, and so on. Phishing attacks often arrive in the form of an email pretending to be from a legitimate organization, such as your bank, the tax department, or some other trusted entity. Phishing is probably the most common form of cyber-attack, largely because it is easy to carry-out, and surprisingly effective.

Man-in-the-middle attack (MITM): A man-in-the-middle attack (MITM) is where an attacker intercepts the communication between two parties in an attempt to spy on the victims, steal personal information or credentials, or perhaps alter the conversation in some way. MITM attacks are less common these days as most email and chat systems use end-to-end encryption which prevents third parties from tampering with the data that is transmitted across the network, regardless of whether the network is secure or not.

SQL injection: SQL injection is a type of attack which is specific to SQL databases. SQL databases uses SQL statements to query the data, and these statements are typically executed via a HTML form on a webpage. If the database permissions have not been set properly, the attacker may be able to exploit the HTML form to execute queries that will create, read, modify or delete the data stored in the database.

Distributed Denial-of-Service (DDoS) attack:

A DDoS attack is where an attacker essentially floods a target server with traffic in an attempt to disrupt, and perhaps even bring down the target. However, unlike traditional denial-of-service attacks, which most sophisticated firewalls can detect and respond to, a DDoS attack is able to leverage multiple compromised devices to bombard the target with traffic. Different DDOS Attacks are discussed below.

SYN-Flood:

The attacker sends a high volume of SYN packets to the targeted server, often with spoofed IP addresses. The server then responds to each one of the connection requests and leaves an open port ready to receive the response. While the server waits for the final ACK packet, which never arrives, the attacker continues to send more SYN packets. The arrival of each new SYN packet causes the server to temporarily maintain a new open port connection for a certain length of time, and once all the available ports have been utilized the server is unable to function normally. In networking, when a server is leaving a connection open but the machine on the other side of the connection is not, the connection is considered half-open. In this type of DDoS attack, the targeted server is continuously leaving open connections and waiting for each connection to timeout before the ports become available again. The result is that this type of attack can be considered a “half-open attack”.

ICMP-Flood:

An Internet Control Message Protocol (ICMP) flood DDoS attack, also known as a Ping flood attack, is a common Denial-of-Service (DoS) attack in which an attacker attempts to overwhelm a targeted device with ICMP echo-requests (pings). Normally, ICMP echo-request and echo-reply messages are used to ping a network device in order to diagnose the health and connectivity of the device and the connection between the sender and the device. By flooding the target with request

packets, the network is forced to respond with an equal number of reply packets. This causes the target to become inaccessible to normal traffic.

Ping Flood:

An evolved version of ICMP flood, this DDoS attack is also application specific. When a server receives a lot of spoofed Ping packets from a very large set of source IP it is being targeted by a Ping Flood attack. Such an attack's goal is to flood the target with ping packets until it goes offline. It is designed to consume all available bandwidth and resources in the network until it is completely drained out and shuts down. This type of DDoS attack is also not easy to detect as it can easily resemble legitimate traffic.

IP Null Attack:

Packets contain IPv4 headers which carry information about which Transport Protocol is being used. When attackers set the value of this field to zero, these packets can bypass security measures designed to scan TCP, IP, and ICMP. When the target server tries to put process these packets, it will eventually exhaust its resources and reboot.

CharGEN Flood:

It is a very old protocol which can be exploited to execute amplified attacks. A CharGEN amplification attack is carried out by sending small packets carrying a spoofed IP of the target to internet enabled devices running CharGEN. These spoofed requests to such devices are then used to send UDP floods as responses from these devices to the target. Most internet-enabled printers, copiers etc., have this protocol enabled by default and can be used to execute a CharGEN attack. This can be used to flood a target with UDP packets on port 19. When the target tries to make sense of these requests, it will fail to do so. The server will eventually exhaust its resources and go offline or reboot.

SNMP Flood:

Like a CharGEN attack, SNMP can also be used for amplification attacks. SNMP is mainly used on network devices. SNMP amplification attack is carried out by sending small packets carrying a spoofed IP of the target to the internet enabled devices running SNMP. These spoofed requests to such devices are then used to send UDP floods as responses from these devices to the target.

However, amplification effect in SNMP can be greater when compared with CHARGEN and DNS attacks. When the target tries to make sense of this flood of requests, it will end up exhausting its resources and go offline or reboot.

NTP Flood:

The NTP protocol is another publicly accessible network protocol. The NTP amplification attack is also carried out by sending small packets carrying a spoofed IP of the target

to internet enabled devices running NTP. These spoofed requests to such devices are then used to send UDP floods as responses from these devices to the target. When the target tries to make sense of this flood of requests, it will end up exhausting its resources and go offline or reboot.

SSDP Flood:

SSDP enabled network devices that are also accessible to UPnP from the internet are an easy source for generating SSDP amplification floods. The SSDP amplification attack is also carried out by sending small packets carrying a spoofed IP of the target to devices. These spoofed requests to such devices are used to send UDP floods as responses from these devices to the target. When the target tries to make sense of this flood of requests, it will end up exhausting its resources and go offline or reboot.

Chapter 3

Machine Learning:

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

Types of Machine Learning Algorithms

1. Supervised Learning:

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher. Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.

- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Types of supervised Machine learning Algorithms:

- **Regression**

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

I. Linear Regression
II. Regression Trees
III. Non-Linear Regression
IV. Bayesian Linear Regression
V. Polynomial Regression

- **Classification**

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc. Below are some popular Classification algorithms which come under supervised learning:

I. Random Forest
II. Decision Trees
III. Logistic Regression
IV. Support vector Machines

Advantages of Supervised learning:

With the help of supervised learning, the model can predict the output on the basis of prior experiences.

In supervised learning, we can have an exact idea about the classes of objects.

Supervised learning model helps us to solve various real-world problems such as fraud detection, spam filtering, etc.

Disadvantages of supervised learning:

Supervised learning models are not suitable for handling the complex tasks. Supervised learning cannot predict the correct output if the test data is different from the training dataset. Training required lots of computation times. In supervised learning, we need enough knowledge about the classes of object.

2. Unsupervised Machine Learning:

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as: Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Types of Unsupervised Learning Algorithm:

- **Clustering:**

  Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

- **Association:**

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Below is the list of some popular unsupervised learning algorithms:

- K-means clustering
- KNN (k-nearest neighbors)
- Hierarchal clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis
- Independent Component Analysis
- Apriori algorithm
- Singular value decomposition

Below there are some supervised machine learning algorithm discussed which are being used in our project

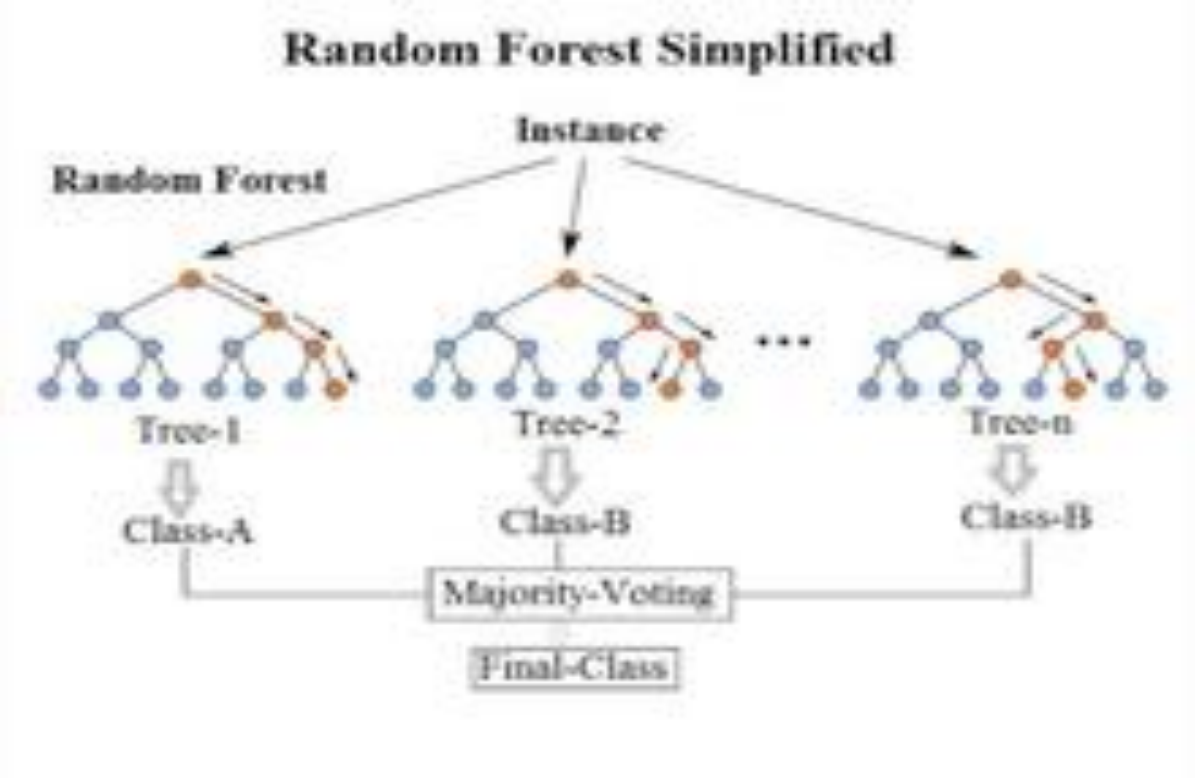
Some Supervised Machine Learning Algorithm Discussed in the below:

Random Forest:

The random forest classifier is a supervised learning algorithm which can be use for regression and classification problems. It is among the most popular machine learning algorithms due to its high flexibility and ease of implementation. it consists of multiple decision trees just as a forest has many trees. On top of that, it uses randomness to enhance its accuracy and combat overfitting, which can be a huge issue for such a sophisticated algorithm. These algorithms make decision trees based on a random selection of data samples and get predictions from every tree. After that, they select the best viable solution through votes. Assuming a dataset has “m” features, the random forest will randomly choose “k” features where $k < m$. Now, the algorithm will calculate the root node among the k features by picking a node that has the highest information gain.

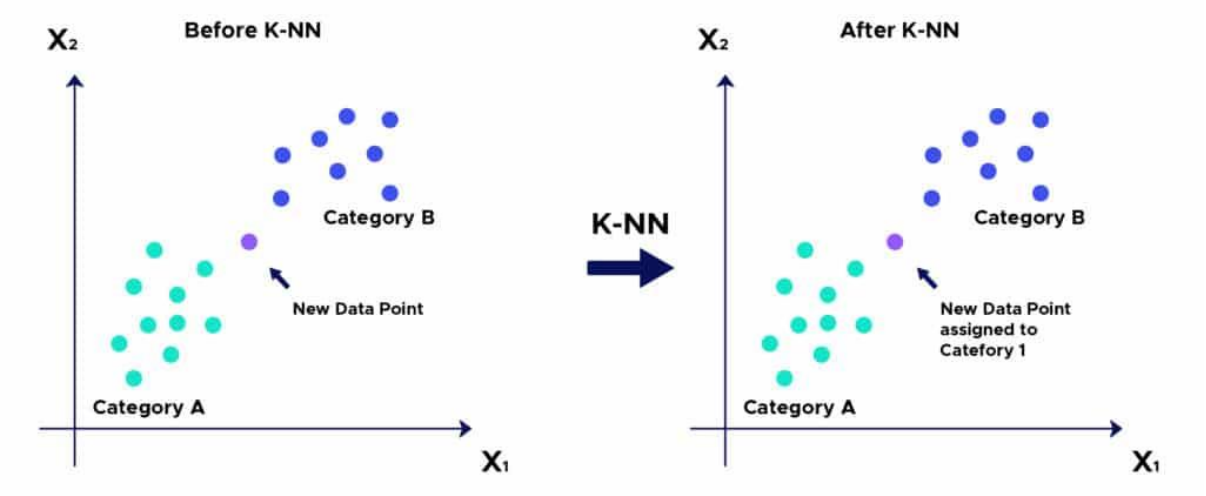
After that, the algorithm splits the node into child nodes and repeats this process “n” times. Now we

have a forest with n trees. Finally, we'll perform bootstrapping, ie, combine the results of all the decision trees present in the forest.



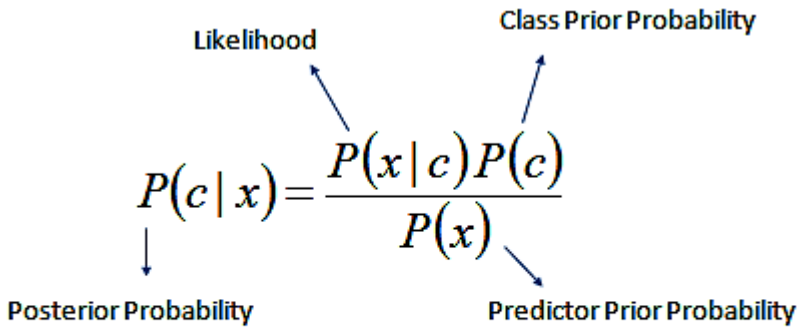
K-NN:

The K-Nearest Neighbour or the KNN algorithm is a machine learning algorithm based on the supervised learning model. The K-NN algorithm works by assuming that similar things exist close to each other. Hence, the K-NN algorithm utilises feature similarity between the new data points and the points in the training set (available cases) to predict the values of the new data points. In essence, the K-NN algorithm assigns a value to the latest data point based on how closely it resembles the points in the training set. K-NN algorithm finds application in both classification and regression problems but is mainly used for classification problems.



Naive Bayes Classifiers:

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Discussion Of Datasets:

Name of our dataset is **KDD CUP 99.**

KDD'99 data set was created by DARPA in 1999 by using recorded network traffic from 1998 dataset. It is being pre-processed into 41 features per network connection. Features in KDD'99 data

set are categorized into four groups i.e., Basic Features (#1 to #9), Content Features (#10 to #22), Time based traffic features (#23 to #31), and Host based traffic features (#32 to #41) as shown in Table 1. KDD'99 consists of 4,898,430 records that is larger than other data sets. Many data mining techniques has been applied to the KDD'99 data set to detect intrusions in network traffic.

The KDD Cup 99 data set stems from DARPA/ MIT Lincoln Laborator y packet traces, and it is the most widely used data set for NIDS evaluation. This is a transformed version of the DARPA data containing 41 features that are considered suitable for machine-learning classification algorithms.
The data set can be obtained as three partitions: a full training set, a 10% version of the training set, and a test set (https:// kdd.ics.uci.edu/databases/kddcup99 /kddcup99.html). Aside from the four categories of attacks within the DARPA data set, 17 new attacks were added in the test data.

Because it is considered a large data set for most machine-learning algorithms, many researchers prefer to use sampled data. Moreover, the records duplication in both training and test data can produce biased results for frequent instances, and overcoming such issues led to the generation of the NSL_KDD data set.

The NSL_KDD, from the Information Security Center of Excellence (ISCX), University of New Brunswick (UNB), is another distilled version of the KDD Cup 99 data sets. It was created in 2009 with the main aim of resolving the issue of redundant records found in KDD Cup 99 data sets,8 in which the ratio of duplicate records in the training and testing data was reported as being 78% and 75%, respectively.

Such huge redundancy may cause learning algorithms to produce biased evaluation results and, in turn, prevent them from learning infrequent records. After cleaning and resampling, the resultant data set consisted of 125,973 and 22,544 records for training and testing, respectively. This is derived from the original 4,900,000and 2,000,000 records, respectively, in the KDD Cup 99 data sets.

Table 2: KDD Cup'99 Data set Features List with Description

Attribute Number	Features	Description
1	Duration	Length of the time duration of the connection
2	Protocol type	Protocol used
3	Service	Service used by destination network
4	flag	Status of the connection (Error or Normal)
5	Src_bytes	Number of data bytes transferred from source to destination
6	dst_bytes	Number of data bytes transferred from destination to source
7	land	If source and destination port no. and IP addresses are same then it will set as 1 otherwise 0
8	Wrong_fragment	Total number of wrong fragments in a connection
9	urgent	Number of urgent packets (these packets with urgent bit activated)
10	hot	Number of 'hot' indicators means entering in a system directory
11	Num_failed_logins	Number of failed login attempts
12	Logged_in	Shows login status (1- successful login, 0- otherwise)
13	Num_compromised	Number of compromised conditions
14	Root_shell	Shows root shell status (1-if root shell obtained otherwise 0)

15	Su_attempted	Set as 1 if 'su root' command used otherwise set as 0
16	Num_root	Number of operations performed as root
17	Num_file_creations	Number of file creation operations
18	Num_shells	Number of shell prompts in a connection
19	Num_access_files	Number of operations on access control files
20	Num_outbound_cmds	Number of outbound commands in a ftp session
21	Is_host_login	If login as root or admin then this set as 1 otherwise 0
22	Is_guest_login	Set as 1 if login as guest otherwise 0
23	count	Number of connections to the same destination host
24	Srv_count	Number of connection to the same service (port number)
25	Serror_rate	Percentage of connections that have activated flag (#4) s0,s1,s2 or s3, among the connections aggregated in count (#23)
26	Srv_serror_rate	Percentage of connection that have activated flag (#4) s0,s1,s2 or s3, among the connections aggregated in srv count (#24)
27	Rerror_rate	Percentage of connections that have activated flag (#4) REJ, among the connections aggregated in count (#23)

28	Srv_rerror_rate	Percentage of connections that have activated flag (#4) REJ, among the connections aggregated in srv count (#24)
29	Same_srv_rate	Percentage of connections that were to the same services, among the connections aggregated in count (#23)
30	Diff_srv_rate	Percentage of connections that were to the different services, among the connections aggregated in count (#23)
31	Srv_diff_host_rate	Percentage of connections that were to different destination machines among the connections aggregated in srv count (#24)
32	Dst_host_count	Number of connections having the same destination host IP address
33	Dst_host_srv_count	Number of connections having same port number
34	Dst_host_same_srv_rate	Percentage of connections that were to the same service among the connections aggregated in dst host count (#32)
35	Dst_host_diff_srv_rate	Percentage of connections that were to different service among the connections aggregated in dst host count (#32)
36	Dst_host_same_src_port_rate	Percentage of connections that were to the same source port among the connections aggregated in dst host srv count (#33)

37	Dst_host_srv_diff_host_rate	Percentage of connections that were to the different destination machines among the connections aggregated in dst host srv count (#33)3
38	Dst_host_serror_rate	Percentage of connections that have activated flag (#4) s0,s1,s2 or s3, among the connections aggregated in dst host count (#32)
39	Dst_host_srv_serror_rate	Percentage of connections that have activated flag (#4) s0,s1,s2 or s3, among the connections aggregated in dst host srv count (#33)
40	Dst_host_rerror_rate	Percentage of connections that have activated flag (#4) REJ, among the connections aggregated in dst host count (#32)
41	dst host srv rerror rate	Percentage of connections that have activated flag (#4) REJ, among the connections aggregated in dst host srv count (#32)
42	label	Attack class label

Chapter 4

Methodology

Technology Used:

Python:

Python is an interpreted general-purpose high-level programming language. Python is design philosophy emphasizes code readability with its notable use of significant indentation. Its object-oriented and language constructs approaches focus to help programmers write clear, concise and logical code for small and large-scale projects **[**45]. Python is dynamically written and garbage collected. It supports many programming paradigms, structured (especially procedural), object-oriented, and functional programming. Python is often described as a "battery-packed" language due to its extensive standard library **[21]**. Python was conceived in the late 1980s [46] by Guido van Rossum at Centrum Wiskunde & Infromatica (CWI) in the Netherlands as a successor to ABC programming language, which was inspired by SETL **[22]**. Since then, it has constantly topped in the chart of most uses programming languages. Since last three decades in the industry have made Python a mature language. Python has a large community around the global that provides immense support for budding developers.

There are some famous Python libraries for machine learning which have been used in our thesis.

NumPy: NumPy is an open-source numerical Python library. NumPy contains a multi-dimensional array and matrix data structures. It can be utilized to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.

Pandas: Pandas is mainly used for data analysis. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

Scikit-learn: Scikit-learn is the most useful library for machine learning in Python. The sklearn library

contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

TensorFlow: It is an open-source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation.

Keras: Keras is a powerful and easy-to-use free open-source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Seaborn: Seaborn is a library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

Google Collab:

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that we create can be simultaneously edited by our team members - just the way we edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in our notebook.

As a programmer, we can perform the following using Google Colab.

- Write and execute code in Python
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

Data Preprocessing:

First of all need to connect google collabe with google drive where our data is stored and then read the data using following piece of code

```
[2]  from google.colab import drive
     drive.mount('/content/drive')
     path="/content/drive/MyDrive/kddcup99_csv.csv"
     df=pd.read_csv(path)

     Mounted at /content/drive
```

Figure 1 connect google collabe with drive and read the csv file

Shape of dataset: 494042 rows x 42 columns including label. Checking the shape is done by following piece of codes.

```
[3]  df.shape

     (494020, 42)
```

Figure 2 Checking the shape of the data

The first step in data preprocessing was to remove the rows with values infinity or nan in one of its columns.

```
df.isnull().sum()
```

```
duration                0
protocol_type           0
service                 0
flag                    0
src_bytes               0
dst_bytes               0
land                    0
wrong_fragment          0
urgent                  0
hot                     0
num_failed_logins       0
logged_in               0
lnum_compromised        0
lroot_shell             0
lsu_attempted           0
lnum_root               0
lnum_file_creations     0
lnum_shells             0
lnum_access_files       0
lnum_outbound_cmds      0
is_host_login           0
is_guest_login          0
```

Figure 3 checking any null values present in each column or not

Above code is to check any null values are present or not. There is no any null values present in our dataset.

Now let see how our dataset is look like before encoding.

```
df
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	1.0
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	1.0
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	1.0
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	1.0
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
494015	0	tcp	http	SF	310	1881	0	0	0	0	...	255	1.0
494016	0	tcp	http	SF	282	2286	0	0	0	0	...	255	1.0
494017	0	tcp	http	SF	203	1200	0	0	0	0	...	255	1.0
494018	0	tcp	http	SF	291	1200	0	0	0	0	...	255	1.0
494019	0	tcp	http	SF	219	1234	0	0	0	0	...	255	1.0

Figure 4 how our data look like before encoding

Now we have to replace categorical data with some values. From the above figure we can see there are three features which has categorical values. We have to replace these categorical into values by using Label Encoding technic. By the following code.

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df.protocol_type=le.fit_transform(df.protocol_type)
df.service=le.fit_transform(df.service)
df.flag=le.fit_transform(df.flag)
```

Figure 5 LabelEncoding Technic

```
df
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate
0	0	1	22	9	181	5450	0	0	0	0	...	9	1.0
1	0	1	22	9	239	486	0	0	0	0	...	19	1.0
2	0	1	22	9	235	1337	0	0	0	0	...	29	1.0
3	0	1	22	9	219	1337	0	0	0	0	...	39	1.0
4	0	1	22	9	217	2032	0	0	0	0	...	49	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
494015	0	1	22	9	310	1881	0	0	0	0	...	255	1.0
494016	0	1	22	9	282	2286	0	0	0	0	...	255	1.0
494017	0	1	22	9	203	1200	0	0	0	0	...	255	1.0
494018	0	1	22	9	291	1200	0	0	0	0	...	255	1.0

Figure 6 our data after Label encoding

```
new_df=df['label'].value_counts().rename_axis('sub_cat_val').reset_index(name='count')
new_df.head()
```

	sub_cat_val	count
0	smurf	280790
1	neptune	107201
2	normal	97277
3	back	2203
4	satan	1589

```
our_labels=['smurf','neptune','normal','back','satan']
our_values=[280790,107201,97277,2203,1589]
ex=[0.3,0.4,0.4,0,0.8]
plt.pie(our_values,labels=our_labels,explode=ex,autopct='%1.2f%%')
plt.show()
```

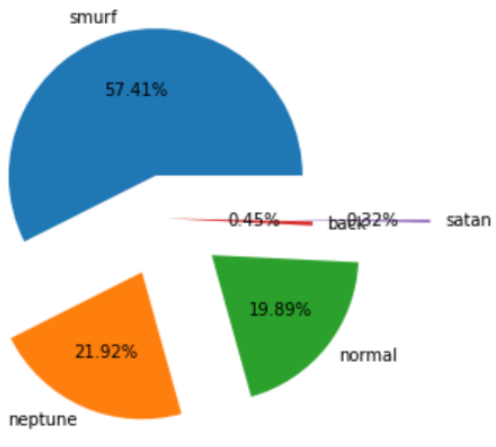


Figure 7 Share of different types of labels present in the dataset

Splitting the data into train and test part:

Importing train_test_split module from sklearn. model selection is required to perform training and testing the data.

Here we generate and separate the training and testing data to be used in our model. The dataset is divided into two parts, x for the input variables, and y for the output generation. The input and output data are now divided into training and testing data using the train_test_split method. 25% data are for testing and remaining 75% data are for training the model.

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(df.drop(labels=['label'],axis=1),df['label'],test_size=.25,random_state=0)
```

Figure 8 Splitting the data into train and test part

Now we are using two different Feature selection technics to extract best 6 features.

Mutual information and Chisqure technic

For each feature selection we wil fit three machine learning model KNN, Naïve Biase and Random Forest.

MUTUAL INFORMATION

Mutual information between two random variables is a non-negative value, which measures the dependency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency. Mutual information between two random variable would be zero if and only if they are completely independent otherwise mutual information between them is symmetric and non-negative.
Mutual information between two random variables X and Y can be stated as follows

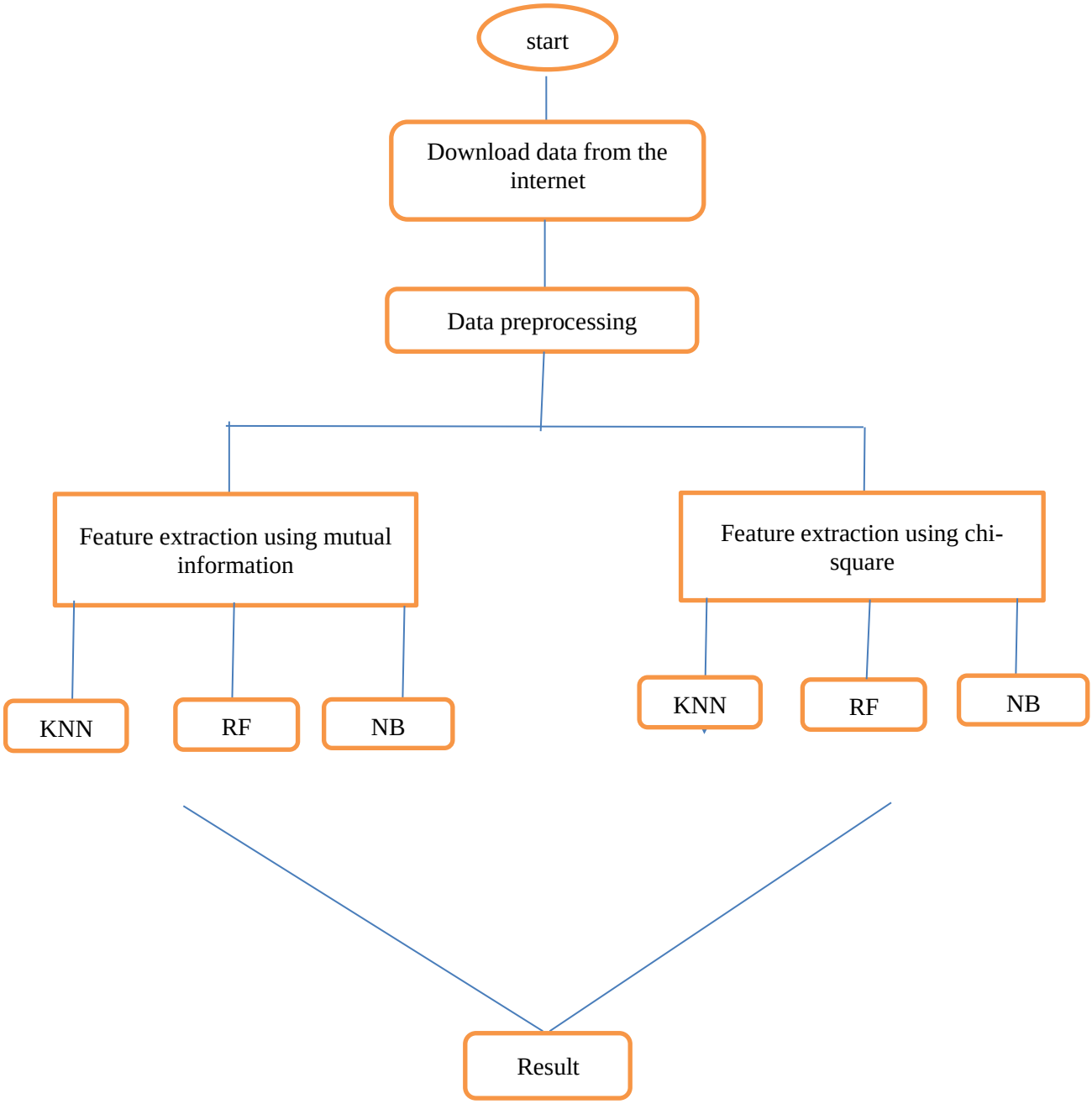
$$I(X;Y) = H(X) - H(X|Y)$$

$$= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$$

CHI-SQUARE

Chi-square test is a technique to determine the relationship between the categorical variables. The chi-square value is calculated between each feature and the target variable, and the desired number of features with the best chi-square value is selected.

Our Method Structure



```
[27] from sklearn.feature_selection import chi2
     f_p_value=chi2(x_train,y_train)
```

Figure 9 importing chisqure and calculating p-value

```
sel=SelectPercentile(chi2,percentile=15).fit(x_train,y_train)
x_train.columns[sel.get_support()]
```

```
Index(['duration', 'src_bytes', 'dst_bytes', 'count', 'srv_count',
       'dst_host_srv_count'],
      dtype='object')
```

```
x_train_chi=sel.transform(x_train)
x_test_chi=sel.transform(x_test)
```

Figure 10 top 6 best features extracted

In the above code top 6 features extracted into x_train_chi and x_test_chi for training the model and testing Respectively.

Now to train the model we will call fit function with parameter our our selected feature and label which are 75% of overall dataset.

```
[9] from sklearn.naive_bayes import GaussianNB
    gnb = GaussianNB()
    gnb.fit(x_train_chi,y_train)
```

```
GaussianNB()
```

Figure 11 training using Gussian naïve biase model

Accuracy can be measure by calling score function

```
[10] gnb.score(x_test_chi,y_test)
```

```
0.9341727055584794
```

Figure 12 Accuracy of the model

Similarly for Random Forest model we will train the model in same way by calling fit function by the help of below code.

```
from sklearn.ensemble import RandomForestClassifier
Rclf=RandomForestClassifier()
Rclf.fit(x_train_chi,y_train)
```

```
RandomForestClassifier()
```

Figure 13 Training Random Forest Model

```
[12] Rclf.score(x_test_chi,y_test)
```

```
0.9984939880976479
```

Figure 14 Accuracy of the model

Now the last model is our K Nearest Neighbours

```
[13] from sklearn.neighbors import KNeighborsClassifier
     model=KNeighborsClassifier(n_neighbors=481)
     model.fit(x_train_chi,y_train)
```

```
KNeighborsClassifier(n_neighbors=481)
```

Figure 15 Trainning KNN model

```
model.score(x_test_chi,y_test)
```

```
0.991878871300757
```

Figure 16 Accuracy of the model

If we use Mutual Information as a feature selection technic then apply all our 3 models then we get below result. Four important feature we extracted like ‘service’, ‘src_bytes’, ‘count’, ‘dst_host_same_src_port_rate’.

Gaussian Naive Bayes MODEL

```
[29]
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(x_train_mi,y_train)
```

```
GaussianNB()
```

```
[30] gnb.score(x_test_mi,y_test)
```

```
0.9509250637626007
```

RANDOM FOREST CLASSIFICATION MODEL

```
from sklearn.ensemble import RandomForestClassifier
Rclf=RandomForestClassifier()
Rclf.fit(x_train_mi,y_train)
```

```
RandomForestClassifier()
```

```
Rclf.score(x_test_mi,y_test)
```

```
0.9989959920650986
```

K-Nearest Neighbours MODEL

```
from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier(n_neighbors=481)
model.fit(x_train_mi,y_train)
```

```
KNeighborsClassifier(n_neighbors=481)
```

```
[34] model.score(x_test_mi,y_test)
```

```
0.9935387231286182
```

Chapter 5

Results

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$\text{Accuracy} = \text{Number of correct predictions Total number of predictions}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

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

Where *TP* = True Positives,

TN = True Negatives,

FP = False Positives,

FN = False Negatives.

Comparison of the accuracy of three models with chi-square and mutual-information

	model	Mutual_Info	Chi_square
0	GNB	95.00	93.4
1	RF	99.80	99.8
2	KNN	99.35	99.1

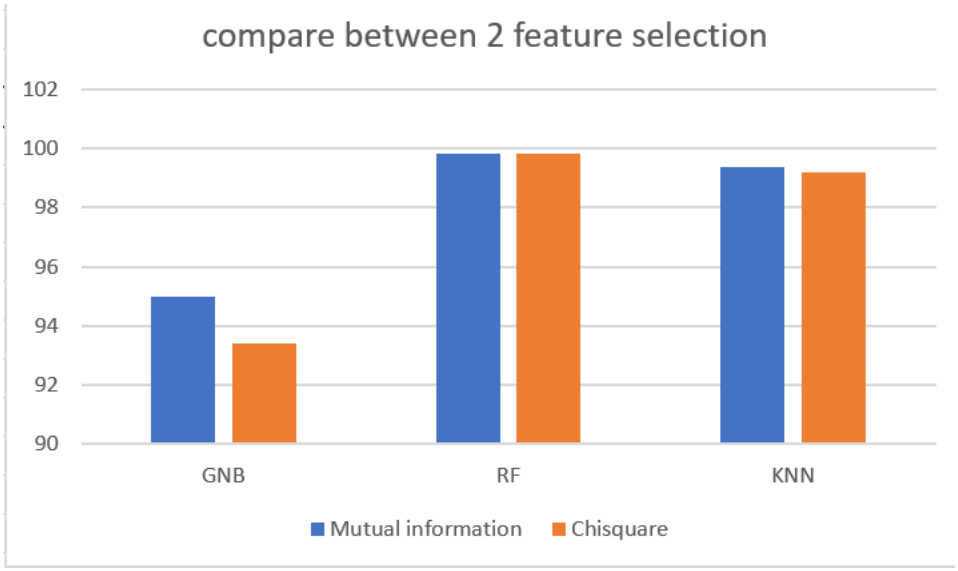


Figure 17 comparison of diff ML and diff feature selection technic

So we will consider Mutual information feature selection technic because it is giving more accuracy in GNB model and also in KNN model as compare to chi2 feature selection technic.

Now we will compare the accuracy of our model with the accuracy of the previous work[16].

Compare with previous model

	model	Previous model accuracy	Our model accuracy
0	GNB	93.95	95.0
1	KNN	96.42	99.3

We are getting better accuracy than the previous work [16].

Figure 18 comparison between previous and our model

In paper [16] they are getting accuracy 93.95 for GNB model whereas accuracy of our model is 95 which is a great difference and in case of KNN model their accuracy is 96.42 and our accuracy is 99.3 again we are getting better result.

Chapter 6

Conclusion and scope for future work

Distributed Deniel of Service attack is one of the most frequent attack and now a days it is a major concern of legitimate users. Detection of this issue is challenging.

To detect from a given dataset it is very important to select most appropriate and important feature. Feature selection technic plays a vital role to detect this attack.

In most of the paper they used normal feature to detect this attack and some of the paper [17] used their hybrid feature selection technics like combine more than one feature to detect this attack.

The difference between their and our work is that we selected only two feature selection technic Mutual Information and Chi-square and we got two different set of best feature for two different feature selection technics respectively and applied different Machine Learning model.

And for the first feature selection technic we are getting better accuracy for GNB and KNN as compare to second feature selection technic.

So we can conclude that mutual information is giving more appropriate feature than chi-square feature selection technic.

Now if we consider Mutual information feature selection technic then our accuracy is far better than the previous work.

For GNB model previous accuracy is 93.95 and for KNN is 96.42
For GNB model our accuracy is 95 and for KNN 99.3 which is far better than the previous [16].

So we can come to this conclusion that our model is far better than the previous model [16].
And in future it can be implemented and evaluate its performance in real world scenarios.

So our future goal is to real world deployment of the proposed approach.
.

Chapter 7

References:

[1] A. Alshamrani, A. Chowdhary, S. Pisharody, D. Lu, and D. Huang, “A defense system for defeating ddos attacks in sdn based networks,” and cooperative reinforcement learning.” Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 196–207.

[2] C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," in IEEE Access, vol. 5, pp. 21954-21961, 2017.
[3] T. Shon, Y. Kim, C. Lee, and J. Moon, “A machine learning framework for network anomaly detection using svm and ga,” Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, pp. 176–183, 2005.

[4] C. Manikopoulos and S. Papavassiliou, “Network intrusion and fault detection: a statistical anomaly approach,” IEEE Communications Magazine, vol.40, no.10, pp.76–82, 2002.

[5] S. Seufert and D. O’Brien, “Machine learning for automatic defence against distributed denial of service attacks,” 2007 IEEE International Conference on Communications, pp.1217–1222, 2007.

[6] Smys, S. "Ddos Attack Detection in Telecommunication Network using Machine Learning." Journal of Ubiquitous Computing and Communication Technologies (UCCT) 1, no. 01 (2019): 33-44.

[7] N. Zhang, F. Jaafar and Y. Malik, "Low-Rate DoS Attack Detection Using PSD Based Entropy and Machine Learning," 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, 2019, pp. 59-62.

[8] Y. Feng, H. Akiyama, L. Lu and K. Sakurai, “Feature Selection for Machine Learning-Based Early Detection of Distributed Cyber Attacks,” 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Athens, 2018, pp. 173-180.

[9] S. Yadav and S. Subramanian, "Detection of Application Layer DDoS attack by feature learning using Stacked Auto Encoder," 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), New Delhi, 2016, pp. 361-366.

[10] D. Chamou et al., "Intrusion Detection System Based on Network Traffic Using Deep Neural Networks," 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, 2019, pp. 1-6.

[11] Jin Kim, Nara Shin, S. Y. Jo and Sang Hyun Kim, "Method of intrusion detection using deep neural network," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, 2017, pp. 313-316.

[12] X. Yuan, C. Li and X. Li, "Deep Defense: Identifying DDoS Attack via Deep Learning," 2017 IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, 2017, pp. 1-8.

[13] D. Migault et al., "A Framework for Enabling Security Services Collaboration Across Multiple Domains," 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, 2017, pp. 999- 1010.

[14] Y. SU, X. MENG, Q. MENG and X. HAN, "DDoS Attack Detection Algorithm Based on Hybrid Traffic Prediction Model," 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Qingdao, 2018, pp. 1-5.

[15] P. Khuphiran, P. Leelaprute, P. Uthayopas, K. Ichikawa and W. Watanakeesuntorn, "Performance Comparison of Machine Learning Models for DDoS Attacks Detection," 2018 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2018, pp. 1-4.

[16] P. Amit V Kachavimath, Shubhangeni Vijay Nazare, Sheetal S Akki, “Distributed Denial of Service Attack Detection using Naïve Bayes and K-Nearest Neighbor for Network Forensics,” d International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020) IEEE Xplore Part Number: CFP20K58-ART; ISBN: 978-1-7281-4167-1

[17] Suman Nandi, Santanu Phadikar, Koushik Majumder, “Detection of DDoS Attack and Classification Using a Hybrid Approach,” 2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP) 978-1-7281-6708-4/20/$31.00 ©2020 IEEE 10.1109/ISEA-ISAP49340.2020.234999

[18] N. Weaver, V. Paxson, S. Staniford and R. Cunningham, A Taxonomy of Computer Worms, In Proceedings of the The Workshop on Rapid Malcode (WORM 2003), held in conjunction with the10th ACM Conference on Computer and Communications Security, Washington, DC, October 27, 2003.

[19] D. Denning, An Intrusion-Detection Model, IEEE Transactions on Software Engineering, vol. 13, 2, pp. 222-232, 1987.
[20] H. Debar, M. Dacier and A. Wespi, Towards a Taxonomy of Intrusion Detection Systems, Computer Networks, vol. 31, 8, pp. 805-822, 1999.

[21] Kuhlman, Dave. “A Python Book: Beginning Python, Advanced Python and Python Excercises”.

[22] Van Rossum, Guido (29 August 2000). “SETL (was: Lukewarm about range literals)”.

[23] T. Radwan, M. A. Azer, and N. Abdelbaki, "Cloud computing security: challenges and future trends," International Journal of Computer Applications in Technology. Vol. 55, no. 2, pp. 158-172, 2017.

[24] O. Osanaiye, K. K. R. Choo, and M. Dlodlo, "Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework," Journal of Network and Computer Applications, vol. 67, pp. 147-165, 2016.

[25] M. Yusof, F. Mohd, and M. Drais, "Detection and defense algorithms of different types of ddos attacks," International Journal of Engineering and Technology, vol. 9, no. 5, pp. 410, 2017.

[26] T. Siva and E. S. P. Krishna, "Controlling various network based ADoS attacks in cloud computing environment: by using port hopping technique," Int. J. Eng. Trends Technol, vol. 4, no. 5, pp. 2099-2104, 2013.

[27] N. Bharot, P. Verma, S. Sharma, and V. Suraparaju, "Distributed Denialof-Service Attack Detection and Mitigation Using Feature Selection and Intensive Care Request Processing Unit," Arabian Journal for Science and Engineering, vol. 43, no. 2, pp. 959-967, 2018.

[28] A. Rawashdeh, M. Alkasassbeh, and M. Al-Hawawreh, "An anomalybased approach for DDoS attack detection in cloud environment," International Journal of Computer Applications in Technology, vol. 57, no 4, pp. 312-324, 2018.

[29] V. Kumar and H. Sharma, "Detection and Analysis of DDoS Attack at Application Layer Using Naïve Bayes Classifier," Journal of Computer Engineering & Technology, vol. 9, no. 3, pp. 208-217, 2018.

[30] K. J. Singh, K. Johnson, and T. De, "Efficient Classification of DDoS Attacks Using an Ensemble Feature Selection Algorithm," Journal of Intelligent Systems, vol. 29, no. 1, pp. 71-83, 2017.

[31] T. V. Sindia, and J. P. M. Dhas, "SBS-SDN based Solution for Preventing DDoS Attack in Cloud Computing Environment," vol. 12, pp. 3593-3599, 2006.

[32] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," International Journal of Engineering Research & Technology (IJERT), vol. 2, no.12, pp. 1848-1853, 2013.

[33] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009. Accessed on: Nov. 2, 2019. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html

[34] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification

algorithms," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 6, pp. 446-452, 2015.

[35] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," Procedia Technology, vol. 4, pp. 119- 128, 2012.

[36] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," Journal of King Saud University-Computer and Information Sciences, vol. 29, no. 4, pp. 462-472, 2017.

[37] H. P. Vinutha and B. Poornima, "An ensemble classifier approach on different feature selection methods for intrusion detection," Information systems design and intelligent applications, Springer, Singapore, pp. 442-451, 2018.

[38] O. Osanaiye, H. Cai, K. K. R. Choo, A. Dehghantanha, Z. Xu, M. Dlodlo, "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing," EURASIP Journal on Wireless Communications and Networking, vol. 2016, no. 1, pp. 130, 2016.

[39] A. Harbola, J. Harbola, and K. S. Vaisla, "Improved intrusion detection in DDoS applying feature selection using rank & score of attributes in KDD-99 data set," In: 2014 International Conference on Computational Intelligence and Communication Networks, IEEE, 2014, pp. 840-845.

[40] H. Nkiama, S. Z. M. Said, and M. Saidu, "A Subset Feature Elimination Mechanism for Intrusion Detection System," International Journal of Advanced Computer Science and Applications, vol. 7, no. 4, pp. 148- 157, 2016.

[41] M. Darwish, A. Ouda, and L. F. Capretz, "Cloud-based DDoS attacks and defences," In: International Conference on Information Society (iSociety 2013), IEEE, 2013, pp. 67-71.

[42] A. Rajalakshmi, R. Vinodhini, and K. F. Bibi, "Data Discretization Technique Using WEKA Tool," International Journal of Science, Engineering and Computer Technology, vol. 6, no. 8, pp. 293, 2016.

[43] O. Osanaiye, K. K. R. Choo, and M. Dlodlo, "Analysing feature selection and classification techniques for DDoS detection in cloud," in Proceedings of Southern Africa Telecommunication, 2016.

[44] M. Alkasassbeh, "An empirical evaluation for the intrusion detection features based on machine learning and feature selection methods," arXiv preprint arXiv:1712.09623, 2017

[45] D.H. Park, H.K. Kim, I.Y. Choi, J.K. Kim, (2012). A literature review and classification of recommender systems research. Expert Syst Appl, 39 (11) (2012), pp. 10059-10072

[46]“About Python”. Python Software Foundation. Retrieved 24 april 2012., second section “Fans of Python use the phrase “batteries included” to describe the standard library, which covers everything from asynchronous processing to zip files”.