People's Democratic Republic of Algeria

Ministry of Higher Education and

scientific research

University of 20 Aout 1955 -skikda-

Faculty of Sciences

Computer Science department

Final dissertation for graduation On Master

in Computer Science – Specialty: Network and Distributed System

Theme :

DDOS ATTACKS DETECTION IN IOT USING CLASSIFICATION ALGORITHM

Realized by :

- Gabli Nadjet

Framed by :

- Dr.Cheikh Mohamed

Promotion : 2021 - 2022

Thanks

Praise be to God that by His grace good deeds are done, after God's help and grace, I thank all my family members, my father is the greatest man in the world, Mr. Gabli Moussa, my first supporter, my mother, Mrs Henday Hada the wonderful woman.ever , my sisters, Karima, Wadad and her three daughters, Soujod, Aseel and Sirin, and all the members of my family, especially my cousin Jamila and Tota and the wonderful lady Halima their mother who always What encourages me with a kind word and all my friends, especially Bushra Merdasi, Mona Talib, Marwa Abdel-Baqi. Djihane Boutraa and Maria DJamai for support.

I thank Dr. Sheikh Mohammed, the wonderful and most wonderful professor, for the guidance, assistance and encouragement. Thanks to his advice and guidance, I was able to complete this work.

I would like to thank all the employees of the Hajar-Soud Company in Annaba for their support, especially Mrs. Sassi Marwa, the wonderful woman.

Abstract

Distributed denial-of-service (DDoS) attack, is one of the most common IoT attacks . With the rapid development of computer and communication technology, the harm of DDoS attack is becoming more and more serious. Therefore, the research on DDoS attack detection becomes more important. In view of this, in this research we will select the best DDoS attack detection supervised techniques from the seven techniques K_Nearest_Neighbors (K-NN), super vector machine (SVM), naïve bayes (NB), decision tree (REPTree), random forest (RF) , decision tree(J48) and Multilayer Perceptron (MLP) ,using the CICDDoS2019 dataset and deep learning software WEKA tool for implementing the ML .The experimental results shown that the proposed DDoS attack detection method based on machine learning has a good detection rate for the current popular DDoS attack. And the random forest classification algorithm is the best CICDDoS2019 dataset classification algorithm with 99.99% detection.

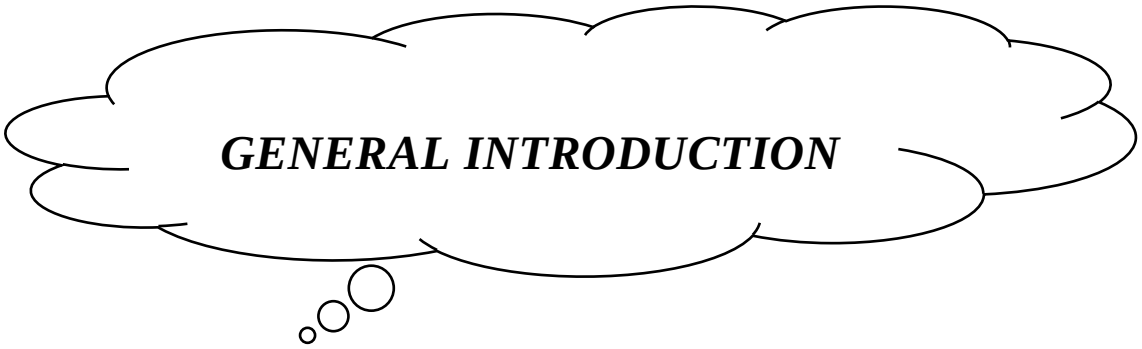
الملخص :

يعد هجوم رفض الخدمة الموزع (DDoS) أحد أكثر هجمات إنترنت الأشياء شيوعًا. مع التطور السريع لتكنولوجيا الكمبيوتر والاتصالات ، أصبح ضرر هجوم DDoS أكثر خطورة. لذلك ، يصبح البحث حول اكتشاف هجوم DDoS أكثر أهمية. في ضوء ذلك ، سنختار في هذا البحث أفضل التقنيات الخاضعة للإشراف لاكتشاف هجمات DDoS من التقنيات السبع K_Nearest_Neighbours (K-NN) ، وآلة المتجهات الفائقة (SVM) ، والخلجان الساذجة (NB) ، وشجرة القرار (REPTree) ، والغابات العشوائية (RF) وشجرة القرار (J48) و Multilayer Perceptron (MLP) ، باستخدام مجموعة بيانات CICDDoS2019 وأداة WEKA لتطبيق ML. أظهرت النتائج التجريبية أن طريقة اكتشاف هجوم DDoS المقترحة على أساس التعلم الآلي لها معدل اكتشاف جيد لهجوم DDoS الشعبي الحالي. وخوارزمية تصنيف الغابة العشوائية هي أفضل خوارزمية تصنيف مجموعة بيانات CICDDoS2019 مع كشف بنسبة 99.99٪.

Table of Contents

Figures List

Table List


GENERAL INTRODUCTION

General Introduction

In recent years, IoT has emerged as a promising technological solution for providing connectivity to myriads of heterogeneous devices across the globe. IoT can help us to access, control and manage these devices to get various functionalities in multiple application scenarios like smart home, smart healthcare, etc.that makes human life easy. The need to use (IoT) in our lives makes this field expands every day without stopping. Which would let everything connected to the internet exposure to penetration. As the need for (IoT) devices grows, the horizon of malicious abuse expands . Where it is exposed to many attacks, one of the most famous and most dangerous is DDos attack.

Distributed Denial of Service (DDoS) attacks have been reported as the most common attacks on IoT devices and network. A DoS attack is a malicious attempt done by an attacker using a single source to make a service or network resources inaccessible to legitimate users. When a DoS attack is launched using multiple distributed sources, it is called a DDoS attack .During the second half of 2021, cybercriminals launched approximately 4.4 million Distributed Denial of Service (DDoS) attacks, bringing the total number of DDoS attacks in 2021 to 9.75 million, a NETSCOUT report reveals. These attacks represent a 3% decrease from the record number set during the height of the pandemic but continue at a pace that's 14% above pre-pandemic levels. [1]

Our work revolves around this domain, which consists of securing a network against the DDoS attack in IoT using intrusion detection.

Problematic:

The traditional firewalls, IDS cannot defend against the complex DDoS attacks , as most of them filter the normal and suspicious traffic based upon the static predefined rules. However, the IDS that filter the intrusive attempts using artificial intelligence (AI) techniques are more reliable and effective as compared to the static predefined rules, so how to select the best classifier for DDoS attack in IoT using machine learning algorithm?

Objective of our work:

The goal of this project is to select the best classifier using machine learning algorithms to detect the DDoS attack in IoT.

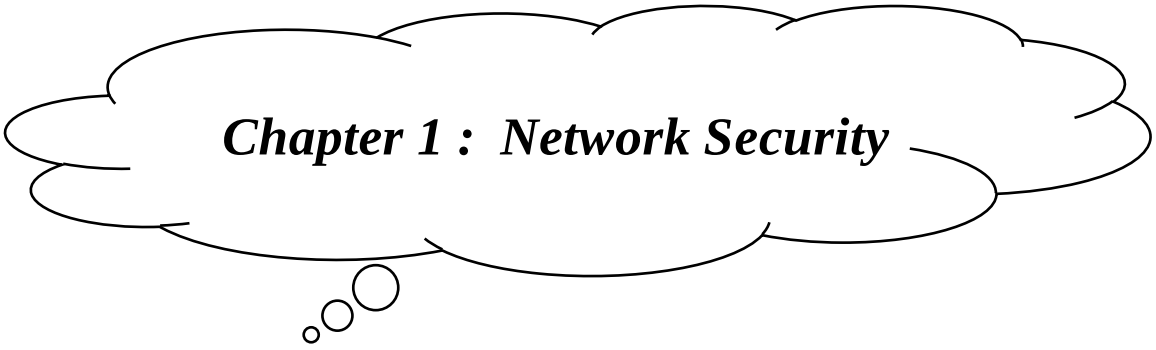
Memory organization : this memoir contains 4 chapter

Chapter 1: is a descriptive chapter for network security, on which we will define threats, malicious software and a security policy as well as the main security mechanisms.

Chapter 2: In the second chapter, we explained the intrusion detection system, its types and how it works, in addition to intrusion detection techniques and how to classify.

Chapter 3: In the third chapter, we explained in detail about the DDoS attack, its types, the best classification algorithms to classify it, and then the proposed method.

Chapter 4: In the fourth chapter ,we implemented our proposed methodology starting with a description of the CICDDOS2019 database used. Then, we will applied the seven machine learning techniques on the selected 5 types of DDoS to choose the best classifier to detect DDoS attack in IoT.


Chapter 1 : Network Security

1- Introduction

Since the inception of networked computers, network security has been a concern. Before the 90s, networks were relatively uncommon and the general public was not made-up of heavy internet users. with more and more sensitive information being placed on networks, it would grow in importance.to day a lot of incidences of a security breach happen, These security breaches could result in monetary losses of a large degree. Investment in proper security should be a priority for large organizations as well as common users. in this chapter we will descriptive the network security, on which we will define threats, malicious software and a security policy as well as the main security mechanisms.

2- IT Security

2-1- Definition:

IT security is the protection of information and especially the processing of information. IT security is intended to prevent the manipulation of data and systems by unauthorized third parties. The meaning behind this is that socio-technical systems, i.e. people and technology, within companies / organizations and their data are protected against damage and threats. This does not only mean information and data, but also physical data centers or cloud services[2].

2-2- IT Security objective :

Information has become more and more valuable over the last few years. Therefore it is all the more important to protect it. Information security is defined by the three IT protection goals of availability, integrity and confidentiality. These must be maintained. In addition, there are other parts to be added: Authenticity, accountability, non-repudiation and reliability[2].

- **Confidentiality of Information :** The confidentiality of IT Security means that data is only accessible to certain authorized persons. This means that access rights must also be assigned. Another central point in the confidentiality of information is the transport of data. This should always be encrypted, symmetrically or asymmetrically. This means that unauthorized persons cannot access the contents.
- **Information Integrity :** The integrity of the information should be seen, that the contents and data are always complete and correct. So the systems must also work together for their own benefit. In order to be able to use data, they must not be changed by means of a sales or processing operation. For this reason, it is also important to note that there is no

possibility for the authoritative Dritte to have (part of) the data available. As it is only possible to make a mistake, it has to be proven that this art of manipulation can be prevented, that the safety can be improved and that it can be used.

- **Availability of Information :** Ensuring the availability of the respective information means that data processing within the systems runs smoothly. The data must be able to be retrieved correctly at the desired time. This means that the computer systems must be protected against failures. This is why there are also load tests to check the limits, so that business operations are maintained in any case.

2-3- Why Security?:

- The Internet was initially designed for connectivity
- Trust assumed
- We do more with the Internet nowadays
- Security protocols are added on top of the TCP/IP
- Fundamental aspects of information must be protected
- Confidential data
- Employee information
- Business models
- Protect identity and resources
- We can't keep ourselves isolated from the Internet
- Most business communications are done online
- We provide online services
- We get services from third-party organizations online

2-4- Types of Security :

Depending on which experts you ask, there may be three or six or even more different types of IT security. Each security expert has their own categorizations. Furthermore, as networks continue to expand with the cloud and other new technologies, more types of IT security will emerge. However, for the most part, there are three broad types of IT security: Network, End-Point, and Internet security (the cybersecurity subcategory). The other various types of IT security can usually fall under the umbrella of these three types[3].

- **Computer Security:** is the protection of computer systems and networks from information disclosure, theft of or damage to their hardware, software, or electronic data,

as well as from the disruption or misdirection of the services they provide. The field is becoming increasingly significant due to the continuously expanding reliance on computer systems, the Internet and wireless network standards such as Bluetooth and Wi-Fi, and due to the growth of "smart" devices, including smartphones, televisions, and the various devices that constitute the IoT. Cybersecurity is also one of the significant challenges in the contemporary world, due to its complexity, both in terms of political usage and technology. Its primary goal is to ensure the system's dependability, integrity, and data privacy.

- **Network Security:** Network security refers to the interaction between various devices on a network. This includes the hardware and the software. Network security, according to SANS Institute, strives to protect the underlying networking infrastructure from unauthorized access, misuse, malfunction, modification, destruction, or improper disclosure, thereby creating a secure platform for computers, users, and programs to perform their permitted critical functions within a secure environment [4].
- **Internet Security:**. Internet security is a term that describes security for activities and transactions made over the internet. It's a particular component of the larger ideas of cybersecurity and computer security, involving topics including browser security, online behavior and network security[5].

2-5- Security Mechanism :

- **Encipherment :** This security mechanism deals with hiding and covering of data which helps data to become confidential. It is achieved by applying mathematical calculations or algorithms which reconstruct information into not readable form. It is achieved by two famous techniques named Cryptography and Encipherment. Level of data encryption is dependent on the algorithm used for Encipherment.
- **Access Control :**This mechanism is used to stop unattended access to data which you are sending. It can be achieved by various techniques such as applying passwords, using firewall, or just by adding PIN to data.
- **Notarization :** This security mechanism involves use of trusted third party in communication. It acts as mediator between sender and receiver so that if any chance of conflict is reduced. This mediator keeps record of requests made by sender to receiver for later denied.

- **Data Integrity :** This security mechanism is used by appending value to data to which is created by data itself. It is similar to sending packet of information known to both sending and receiving parties and checked before and after data is received. When this packet or data which is appended is checked and is the same while sending and receiving data integrity is maintained.
- **Authentication Exchange :** This security mechanism deals with identity to be known in communication. This is achieved at the TCP/IP layer where two-way handshaking mechanism is used to ensure data is sent or not
- **Bit stuffing :** This security mechanism is used to add some extra bits into data which is being transmitted. It helps data to be checked at the receiving end and is achieved by Even parity or Odd Parity.
- **Digital Signature :** This security mechanism is achieved by adding digital data that is not visible to eyes. It is form of electronic signature which is added by sender which is checked by receiver electronically. This mechanism is used to preserve data which is not more confidential but sender's identity is to be notified[6].

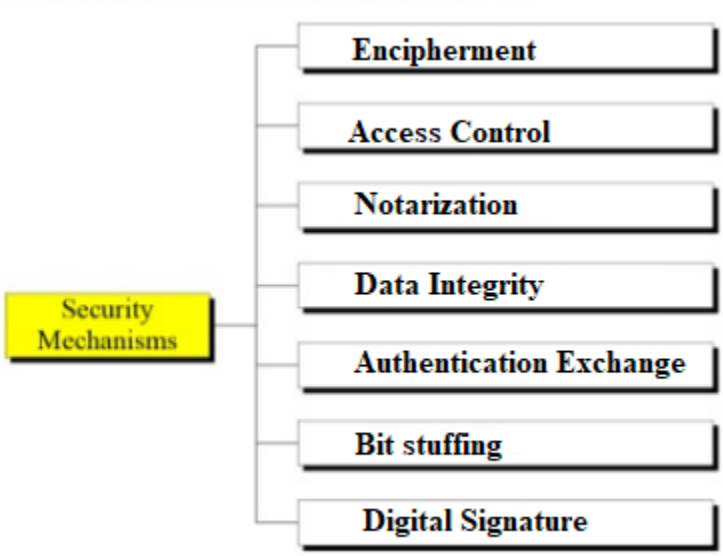


Figure 1.1: Types of Security Mechanism

2-6- Security Issue:

A security issue is any unmitigated risk or vulnerability in your system that hackers can use to do damage to systems or data. This includes vulnerabilities in the servers and software connecting your business to customers, as well as your business processes and people. A vulnerability that hasn't been exploited is simply a vulnerability that hasn't been exploited yet. Web security problems should be addressed as soon as they are discovered, and effort should be put into finding them because exploit attempts are inevitable [7]

- **Code Injection (Remote Code Execution):** To attempt a code injection, an attacker will search for places your application accepts user input – such as a contact form, data-entry field, or search box. Then, through experimentation, the hacker learns what various requests and field content will do. For example, if your site's search function places terms into a database query, they will attempt to inject other database commands into search terms. Alternatively, if your code pulls functions from other locations or files, they will attempt to manipulate those locations and inject malicious functions.
- **Cross-Site Scripting (XSS) Attack:** JavaScript and other browser-side scripting methods are commonly used to dynamically update page content with external information such as a social media feed, current market information, or revenue-generating advertisements. Hackers use XSS to attack your customers by using your site as a vehicle to distribute malware or unsolicited advertisements. As a result, your company's reputation can be tarnished, and you can lose customer trust.
- **Phishing:** Phishing is the attempt of acquiring sensitive information such as usernames, passwords, and credit card details directly from users by deceiving the users.Phishing is typically carried out by email spoofing or instant messaging, and it often directs users to enter details at a fake website whose "look" and "feel" are almost identical to the legitimate one. phishing can be classified as a form of social engineering.
- **Spoofing:** Spoofing is an act of masquerading as a valid entity through falsification of data (such as an IP address or username), in order to gain access to information or resources that one is otherwise unauthorized to obtain. There are several types of spoofing, including:
  - Email spoofing, is where an attacker forges the sending (*From*, or source) address of an email.

- IP address spoofing, where an attacker alters the source IP address in a network packet to hide their identity or impersonate another computing system.
- MAC spoofing, where an attacker modifies the Media Access Control (MAC) address of their network interface controller to obscure their identity, or to pose as another.
- Biometric spoofing, where an attacker produces a fake biometric sample to pose as another user

3- Network Security

3-1- Definition :

Network security is a broad term that encompasses a multitude of technologies, devices, and processes. In its simplest expression, it is a set of rules and configurations intended to protect the integrity, confidentiality and accessibility of computer networks and data, using software and hardware technologies. Every business, regardless of size, industry or infrastructure, needs some level of network security solutions to protect against the ever-growing cyber threat landscape.

Today's network architecture is complex, facing an ever-changing threat environment and attackers constantly looking for vulnerabilities to exploit. These vulnerabilities can exist in a wide variety of domains, including devices, data, applications, users, and locations. This is why many network security management tools and applications are used today to address security threats and vulnerabilities, as well as maintain regulatory compliance. When all it takes is a few minutes of downtime to cause widespread disruption and massive damage to an organization's bottom line and reputation, it's critical that these safeguards are in place [8]

3-2- History Of Network Security:

Recent interest in security was fueled by the crime committed by Kevin Mitnick the largest computer-related crime in U.S. history . Since then, information security came into the spotlight. Due to the evolution of information that is made available through the internet, information security is also required to evolve. Internet protocols in the past were not developed to secure themselves. security protocols are not implemented. This leaves the internet open to attacks.

The birth of the interne takes place in 1969 when Advanced Research Projects Agency Network (ARPANet) is commissioned by the department of defense (DOD) for research in networking. timeline can be started as far back as the 1930s. Polish cryptographers created an enigma machine in 1918 that converted plain messages to encrypted text. In 1930, Alan Turing, a

brilliant mathematician broke the code for the Enigma. Securing communications was essential in World War II.

In the 1960s, the term “hacker”. The Department of Defense began the ARPANet. This paves the way for the creation of the carrier network known today as the Internet. During the 1970s, the Telnet protocol was developed. This opened the door for public use of data networks that were originally restricted to government contractors and academic researcher.

During the 1980s, the hackers and crimes relating to computers were beginning to emerge. The Computer Fraud and Abuse Act of 1986 was created because of Ian Murphy’s crime of stealing information from military computers.

In the 1990s, Internet became public and the security concerns increased tremendously. On any day, there are approximately 225 major incidences of a security breach . These security breaches could also result in monetary losses of a large degree. Investment in proper security should be a priority for large organizations as well as common users [9].

3-3- How does network security work :

Network security revolves around two main processes: authentication and authorization. Authentication is the practice of checking the user’s identity before granting access to a system. A network can verify the user in three different ways:

- One-factor authentication: A user types in a username and password to log in.
- Two-factor authentication: A user provides a username and a password, but the network requires further verification. Systems typically ask for something that the user possesses, such as a security token, a credit card, or a mobile phone.
- Three-factor authentication: A network requires two-factor authentication plus biometric recognition such as a voice, fingerprint, or retinal scan. This and other security techniques are used to secure data centers.[10]

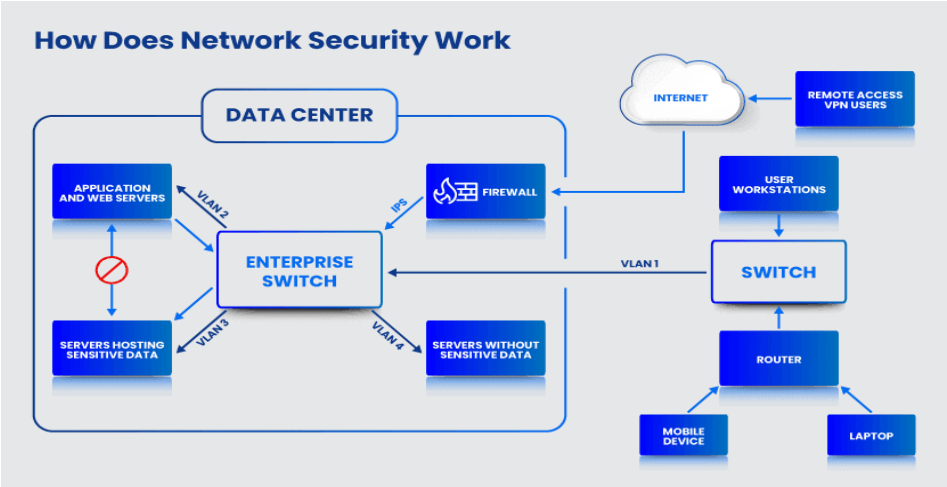


Figure1.2:network security work

3-4- Network Security Controls: Network security consists of three different controls: physical, technical and administrative.

- **Physical Network Security:** Physical security controls are designed to prevent unauthorized personnel from gaining physical access to network components such as routers, wiring bays, etc. Controlled access, through locks, biometric authentication and other devices, is essential for any organization.
- **Technical Network Security**: Technical security controls protect data that is stored on the network or that travels through, enters or leaves the network. The protection is two-fold: it must protect data and systems from unauthorized personnel, and it must also protect against malicious employee activity.
- **Administrative network security**: Administrative security controls consist of security policies and processes that control user behavior, including how users are authenticated, their level of access, and also how members of the IT department implement security controls. infrastructure changes [11].

3-5- Security problem:

- **Virus:** A virus is a malicious, downloadable file that can lay dormant that replicates itself by changing other computer programs with its own code. Once it spreads those files are infected and can spread from one computer to another, and/or corrupt or destroy network data.

- **Worms:** Can slow down computer networks by eating up bandwidth as well as the slow the efficiency of your computer to process data. A worm is a standalone malware that can propagate and work independently of other files, where a virus needs a host program to spread.
- **Trojan:** A trojan is a backdoor program that creates an entryway for malicious users to access the computer system by using what looks like a real program, but quickly turns out to be harmful. A trojan virus can delete files, activate other malware hidden on your computer network, such as a virus and steal valuable data.
- **Spyware:** Much like its name, spyware is a computer virus that gathers information about a person or organization without their express knowledge and may send the information gathered to a third party without the consumer's consent.[12]

3-6- Ways to secure a network: The security of a network is the security of the elements that compose it, there are several mechanisms and security devices, among them:

- **Firewall protection:** A firewall is either a software program or a hardware device that prevents unauthorized users from accessing your network, stopping suspicious traffic from entering while allowing legitimate traffic to flow through. There are several types of firewalls with different levels of security, ranging from simple packet-filtering firewalls to proxy servers to complex, next-generation firewalls that use AI and machine learning to compare and analyze information as it tries to come through.
- **Intrusion detection and prevention:** Intrusion detection and prevention systems (IDPS) can be deployed directly behind a firewall to provide a second layer of defense against dangerous actors. Usually working in tandem with its predecessor, the more passive intrusion defense system (IDS), an IDPS stands between the source address and its destination, creating an extra stop for traffic before it can enter a network. An advanced IDPS can even use machine learning and AI to instantly analyze incoming data and trigger an automated process – such as sounding an alarm, blocking traffic from the source, or resetting the connection – if it detects suspicious activity.
- **Network access control (NAC):** Standing at the frontline of defense, network access control does just that: it controls access to your network. Most often used for "endpoint health checks," NAC can screen an endpoint device, like a laptop or smart phone, to ensure it has adequate anti-virus protection, an appropriate system-update level, and the

correct configuration before it can enter. NAC can also be programmed for "role-based access," in which the user's access is restricted based on their profile so that, once inside the network, they can only access approved files or data.

- **Cloud security:** Cloud security protects online resources – such as sensitive data, applications, virtualized IPs, and services – from leakage, loss, or theft. Keeping cloud-based systems secure requires sound security policies as well as the layering of such security methods as firewall architecture, access controls, Virtual Private Networks (VPNs), data encryption or masking, threat-intelligence software, and disaster recovery programs.
- **Virtual Private Networks (VPNs):** A virtual private network (VPN) is software that protects a user's identity by encrypting their data and masking their IP address and location. When someone is using a VPN, they are no longer connecting directly to the internet but to a secure server which then connects to the internet on their behalf. VPNs are routinely used in businesses and are increasingly necessary for individuals, especially those who use public WIFI in coffeeshops or airports. VPNs can protect users from hackers, who could steal anything from emails and photos to credit card numbers to a user's identity.

3-7- Suggested solutions : Protecting your company is a must. Here are 5 security measures to implement.

- **Bolster Access Control :** Access control is an important part of security. Weak access control leaves your data and systems susceptible to unauthorized access.Boost access control measures by using a strong password system. You should have a mix of uppercase and lower case letters, numbers, and special characters. Also, always reset all default passwords.Finally, create a strong access control policy.
- **Keep All Software Updated:** As pesky as those update alerts can be, they are vital to your network's health.From anti-virus software to computer operating systems, ensure your software is updated. When a new version of software is released, the version usually includes fixes for security vulnerabilities.Manual software updates can be time-consuming. Use automatic software updates for as many programs as possible.
- **Standardize Software :** Keep your systems protecting by standardizing software. Ensure that users cannot install software onto the system without approval.Not knowing what

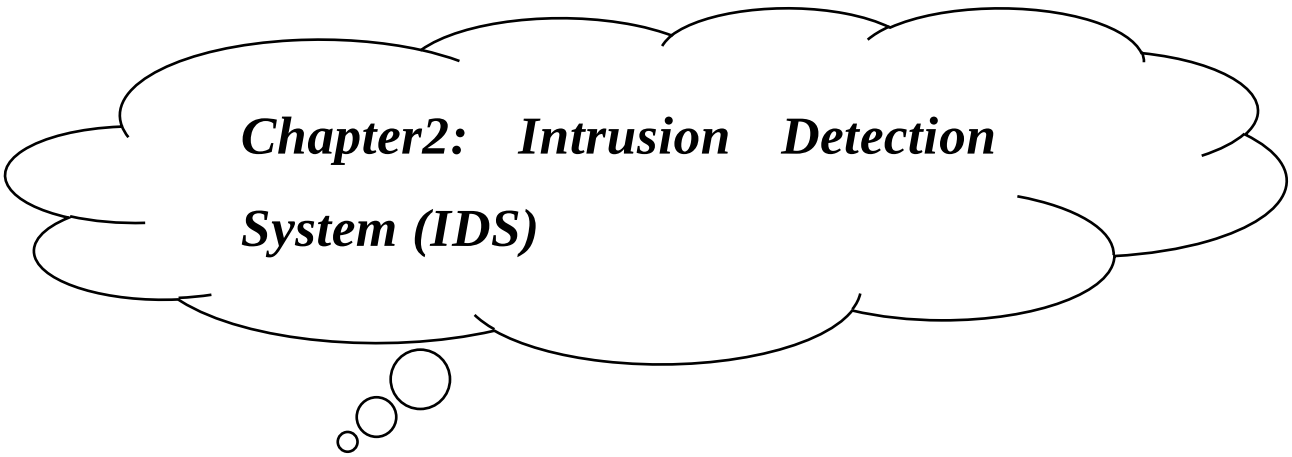
software is on your network is a huge security vulnerability. Make sure that all computers use the same:

- Operating system
- Browser
- Media player
- Plugins

- **Use Network Protection Measures :** Protecting your network is crucial. To keep your network and its traffic secured:
  - Install a firewall
  - Ensure proper access controls
  - Use IDS/IPS to track potential packet floods
  - Use network segmentation
  - Use a virtual private network (VPN)
  - Conduct proper maintenance

4- Conclusion :

In this chapter, we have presented an overview of computer security in general and network security, the types of security and the means of securing a network and the importance of setting up a security policy in order to remedy this. constant threats to a computer network. These threats generally manifest themselves in the form of computer attacks which we have illustrated in order to show the intensity of the danger. Finally we have proposed some existing solutions to protect and reduce the risks, in the next chapter we will present the Intrusion Detection System as a means of protection against those attacks.


Chapter2: Intrusion Detection System (IDS)

1- Introduction :

Information systems and networks are constantly exposed to cyber attacks. Firewalls and antiviruses to repel all these attacks are clearly not enough .An IDS (Intrusion Detection System) is a set of software and/or hardware components whose main function is to detect and analyze abnormal or suspect activities on the target analyzed (a network or a host). It thus makes it possible to have knowledge of successful attempts as stranded of intrusions.in this chapter i will present one of system for protect network from attack Intrusion detection system .

2- Comparison with firewalls

The main difference being that firewall performs actions such as blocking and filtering of traffic while an IPS/IDS detects and alert a system administrator or prevent the attack as per configuration.A firewall allows traffic based on a set of rules configured. It relies on the source, the destination addresses, and the ports. A firewall can deny any traffic that does not meet the specific criteria.

IDS is a passive device which watches packets of data traversing the network, comparing with signature patterns and setting off an alarm on detection on suspicious activity. On the contrary, IPS is an active device working in inline mode and prevent the attacks by blocking it [13].

- **Firewall:**
  - Firewall is a network security device that filters incoming and outgoing network traffic based on predetermined rules
  - Filters traffic based on IP address and port numbers
  - Layer 3 mode or transparent mode
  - Inline at the Perimeter of Network
  - Should be 1st Line of defense
  - Block the traffic
- **IPS:**
  - IPS is a device that inspects traffic, detects it, classifies and then proactively stops malicious traffic from attack.
  - inspects real time traffic and looks for traffic patterns or signatures of attack and then prevents the attacks on detection.
  - Inline mode , generally being in layer 2.

- Should be placed after the Firewall device in network.
- Preventing the traffic on Detection of anomaly.

- **IDS:**
  - An intrusion detection system (IDS) is a device or software application that monitors a traffic for malicious activity or policy violations and sends alert on detection.
  - Detects real time traffic and looks for traffic patterns or signatures of attack and them generates alerts
  - Inline or as end host (via span) for monitoring and detection
  - Should be placed after firewall
  - Alerts/alarms on detection of anomaly

3- Intrusion Detection system :

3-1- Definition :

An intrusion detection system (IDS) is a device or software application that monitors a network for malicious activity or policy violations. Any malicious activity or violation is typically reported or collected centrally using a security information and event management system. “Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices”. IDS technologies tackle security problems through event gathering, logging, detection and prevention .

An IDS contains various modules to monitor and decide whether activities are malicious or non-malicious in an efficient manner. The modules are a system to monitor, data collection and pre-processing, intrusion recognition engine and a reporting/response unit. The monitoring unit(s) collect samples of raw audit data (such as system logs, network packets, etc.) and pass them to the pre-processing stage. This unit analyses data and extracts features in such a way that the detection engine can later examine them. The intrusion recognition system searches the treated data for any sign of malicious conduct. Finally, based on detection engine outcomes, the alarm/response unit takes action. Figure 2.1 shows an IDS organization and the relations between these components. The solid lines refer to data/control flow while dashed lines refer to the response to intrusive

activities[14]

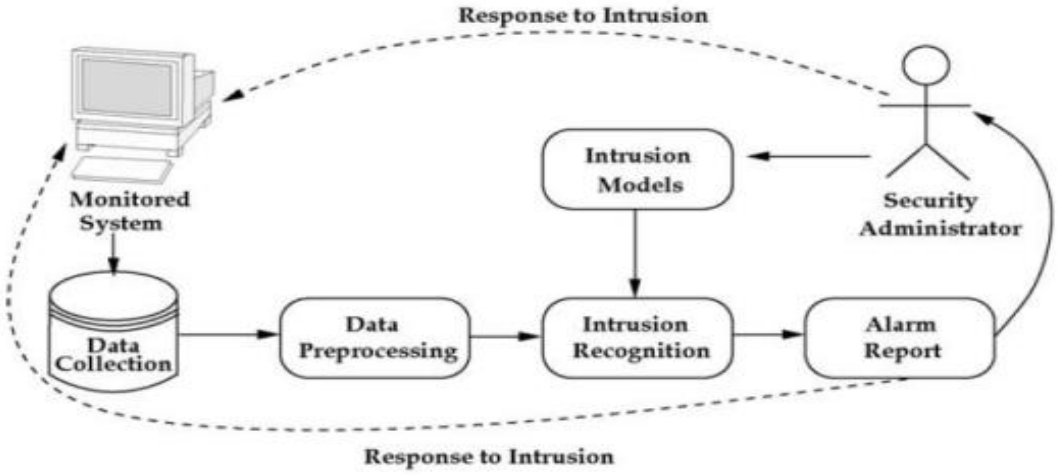


Figure2.1: IDS organization.

3-2- History :

The first paper about IDSs is dated on 26th February 1980 and was written by James P. Anderson. The topic was "Computer Security Threat: Monitoring and Surveillance"-James Anderson split penetrations in external and internal ones based on whether the user had access permission to the computer or not. The main targets of the security audit trail were these:

- Data could be obtained from different resources of the system.
- To avoid internal attacks, unusual behavior of users of resources should be detected
- Enough data should be obtained in order for the security administrators to find the problem.
- The security audit trail should be able to recognize the attacker strategy.

Anderson was focused on the collection of records that showed abnormal use of the system, such as use outside of time, abnormal frequency of use, abnormal patterns of reference to programs or data. He also alerted about the problem of the legitimate user that maybe has access to confidential data; it would be very difficult to record a feasible trail to detect some misuse. His paper was the first based on host intrusion detection and IDS in general.

The next works on the IDS field were mostly done by SRI International, a non-profit research institute created in the 40s in order to research and develop for government agencies. In 1983 Dr. Dorothy Denning started a project to analyze audit trails in the government. The system would take records of the computer activities of all the users in order to detect some possible misuse. Later the Intrusion Detection Expert System (IDES) was developed between 1984 and

1986 by Denning and Peter Neumann. The project was originally launched by the Navy of U.S. Everything was based on the study of user profiles that would shed some light in case of abuse or misuse. Rules of "normal" activities and user profiles were compared to detect possible misuse. IDES was a hybrid system since it used both misuse statistics and security rules.

Meanwhile, other advances on IDS were done, like the Haystack project for the US Air Force by Lawrence Livermore Labs. The system was developed in C and SQL and the principle was always the same: Focus on "bad" traffic and comparison with patterns to get a possible abuse. The system was turned into the Distributed Intrusion Detection System (DIDS), which improved the first project by also using traces from the servers; that was the reason of it being named Distributed. In 1989, the company called Haystack Labs was launched with the apparition of Stalker, an improved Haystack original system.

Both SRI International and Haystack helped in the development and improvement of host-based intrusion detection systems.

The first IDS that monitored network traffic was the Network System Monitor (NSM) and was developed in the California University to work on an UNIX station of Sun. The mode of use was very close to the IDS of today.

- All traffic was captured even if it was not directed to the system.
- Network packets were obtained.
- The protocol was identified in order to get the necessary data
- Data was inspected and compared with statistics and rules so the abuses or misuse could be noticed.

All that work was done by Todd Heberlein. With the Haystack project the IDS field changed and permitted the beginning of commercial applications. Finally the market was opened for the companies to show all their technology.

The first commercial product was developed by Haystack Labs, the already named Stalker, host based. Also the Air force Cryptologic Center came out with the Automated Security Measurement System and its work group made a spin-off to launch to the market a hardware and software mixed product in the field of network intrusion detection in 1994, known as Netranger.

During the last decade numerous vendors (Cisco, Internet Security Systems, Enterasys… have been continuously emerging due to the exponential growth of the internet and every company right now knows that a good shield for its computer-network system must be present and updated.

For a home user there also exists IDS products. A wide range of offerings can be found, from costly products to very good open source ones, like Snort that will be studied in depth in the following chapters.[15]

3-3- Important of IDS:

An IDS enables you to enhance the security of your network devices and valuable network data by pinpointing suspicious network traffic and bringing it to your attention. Your network needs strong security to protect existing information and transfers of internal and external network data. Cyberattacks are increasing in sophistication and regularity, so it's important to have a comprehensive and adaptable intrusion detection system.

Along with increasing network security, an intrusion detection system can help you organize critical network data. Your network generates tons of information every day through regular operations, and an intrusion detection system can help you differentiate the necessary activity from the less important information. By helping you determine which data you should pay attention to, an intrusion detection system can spare you from combing through thousands of system logs for critical information. This can save you time, reduce manual effort, and minimize human error when it comes to intrusion detection.

Gaining detailed, accurate visibility of network activity through an IDS can also help you demonstrate compliance. Intrusion prevention systems are built to detect, organize, and alert on inbound and outbound network traffic in depth, pinpointing the most critical information. By filtering through network traffic, an intrusion detection system could give you a leg up when it comes to determining the compliance of your network and its devices.

An IDS is made to optimize intrusion detection and prevention by filtering through traffic flow. This can save you time, energy, and resources while spotting suspicious activity before it turns into a full-blown threat. An IDS also provides increased visibility into network traffic, which can help you fend off and catch malicious activity, determine compliance status, and improve overall network performance. The more your IDS catches and understands malicious activity on your network, the more it can adapt to increasingly sophisticated attacks.[16]

3-4- The operation of an IDS:

an IDS is designed to observe network traffic and match traffic patterns to known attacks. Through this method, sometimes called pattern correlation, an intrusion prevention system could determine if unusual activity is a cyberattack. Once suspicious or malicious activity is discovered, an intrusion detection system will send an alarm to specified technicians or IT administrators. IDS alarms enable you to quickly begin troubleshooting and identify root sources of issues, or discover and stop harmful agents in their tracks.

Intrusion detection systems primarily use two key intrusion detection methods: signature-based intrusion detection and anomaly-based intrusion detection.

- Signature-based intrusion detection is designed to detect possible threats by comparing given network traffic and log data to existing attack patterns. These patterns are called sequences (hence the name) and could include byte sequences, known as malicious instruction sequences. Signature-based detection enables you to accurately detect and identify possible known attacks.
- Anomaly-based intrusion detection is the opposite it's designed to pinpoint unknown attacks, such as new malware, and adapt to them on the fly using machine learning.

Machine learning techniques enable an intrusion detection system (IDS) to create baselines of trustworthy activity—known as a trust model—then compare new behavior to verified trust models. False alarms can occur when using an anomaly-based IDS, since previously unknown yet legitimate network traffic could be falsely identified as malicious activity.

Hybrid intrusion detection systems use signature-based and anomaly-based intrusion detection to increase the scope of your intrusion prevention system. This enables you to identify as many threats as possible. A comprehensive intrusion detection system (IDS) can understand the evasion techniques cybercriminals use to trick an intrusion prevention system into thinking there isn't an attack taking place. These techniques could include fragmentation, low-bandwidth attacks, pattern change evasion, address spoofing or proxying, and more[16].

3-5- IDS Types:

3-5-1- Approaches of Intrusion Detection System: Approaches of intrusion detection system are:

- **Misuse Detection:** Misuse detection is also called signature-based or rulebased detection . In this detection approach, user's activities are compared with the attackers' known behaviors, to penetrate a system or network. In misuse detection, gathered information is analyzed and compared with large databases for attack signatures .

  Advantage is Misuse or signature-based detection is useful, because its detection rate is high and false alarm rate is low for known attacks.

- **Anomaly Detection:** In anomaly detection approach, activities that are varied from already established patterns for users or groups of users are identified . In this approach, profiles may be established for normal behavior of users, which comes from the statistics of data of users. When detection is performed, profile is compared with the actual users' data. If the threshold value is above or greater than the offset, the user's behavior is considered normal, and it is considered that he has no intention of attack. While, if the threshold value is less than the offset, user's behavior is considered abnormal and attack can occur . It means that it builds a baseline of what is normal. Normal behavior of network should be known before its implementation

  Advantage: Anomaly detection can detect unknown attacks easily, but its misjudgment rate is high . It can also detect previous unknown threats

3-5-2- Classification of Intrusion Detection Systems:

- **Network-based IDS (NIDS):** Network-based IDS are standalone hardware appliances which include network intrusion detection capabilities . They are mostly deployed on strategic point in network infrastructure such as at a boundary between networks, virtual private network servers, remote access servers, and wireless networks . NIDS monitors network traffic going through particular network segments or devices . It can capture and analyze data to detect known attacks or illegal activities or analyze network and application protocol activity to identify anomalous and suspicious activity by traffic scanning . NIDS can also be referred as "packetsniffers", because it captures and collect the data in the form of internet packets passing through communication mediums

- **Host Intrusion Detection (HIDS)**: In Host-Based IDS, the characteristics of a single host are monitored and the events of that host are observed for any malicious activity. They can monitor network traffic, logs, processes, operations performed by applications, file access and modification, and any configuration change in system . The deployment of HIDS is usually done on critical hosts. Critical host includes servers or systems that are publicly accessible and have some sensitive information. They are placed on one server or workstation, where data is collected from different resources and machine analyze the data locally.
- **Hybrid Intrusion Detection** :This technology is also called Network Node Intrusion Detection and it integrates the two technologies NIDS and HIDS for a single node. Literally, it means that a host intrusion detection system is extended by network intrusion detection capabilities in order to enable it to identify a wide set of attack patterns[17].

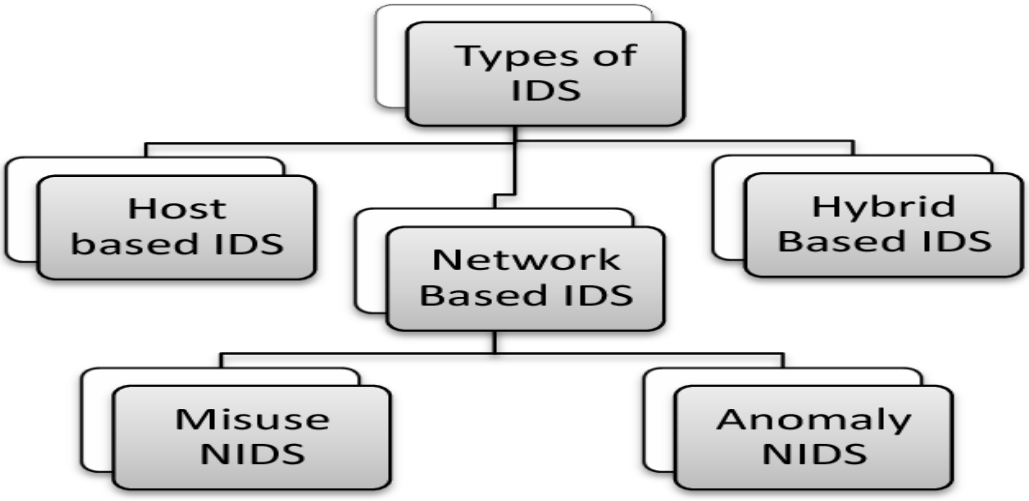


Figure2.2: Types of IDS.

3-6- Detection Technologies :

Furthermore, detection mechanisms were improved and this way, several approaches besides the two original mechanisms, namely misuse detection and anomaly detection (Mukherjee et al. (1994)) have been established, although some of them are just derived approaches. Basically, we distinguish the following five detection methodologies (Pilgermann (2003); NSS (2002)):

- **Pattern matching :** Most popular and most mature detection methodology, which attempts to detect malicious activities by searching for predefined patterns inside the network traffic.

- **Stateful pattern matching:** Extended version of pattern matching, which looks to overcome the problems with fragmented attacks by tracing the network traffic and reassembling fragmented pieces of network traffic
- **Protocol decoding:** Reassembles the network traffic and, with the knowledge about specifications of different network protocols, it is able to identify malicious behavior inside the traffic
- **Heuristic analysis:** used a kind of algorithmic logic, which the alarm decisions are based on. Often statistical evaluations of a special traffic type are used for these algorithms
- **Anomaly analysis:** Instead of searching for malicious behavior the normal behavior is being defined (using a learning phase or based on the setting of parameters) and significant deviations from this behavior are reported to be malicious behavior[18].

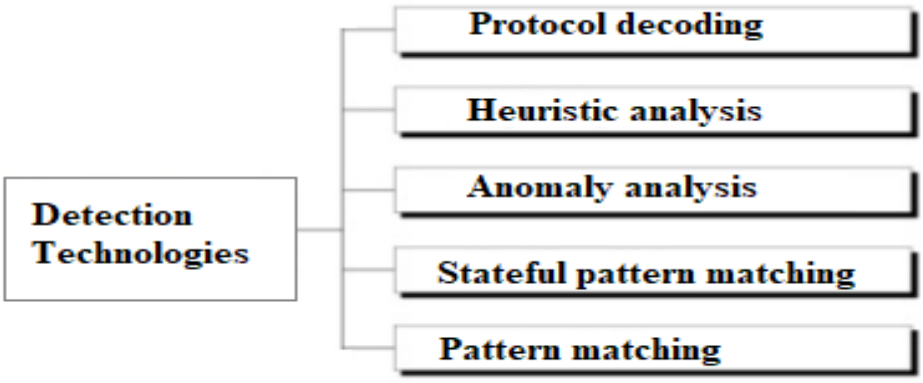


Figure2.3: Detection Technologies

3-7- IDS Classification:

There are several ways to classify IDSs depending on some criteria, such as information source, analysis type, type of response and detection time. These ones are the most common criteria and they will be explained in more detail in the following pages.

3-7-1- Information sources :

Information sources are one of the first issues to focus on when designing an intrusion detection system. These sources can be classified in many ways. With regard to the detection of intrusions, they are classified according to the location due to some IDSs analyzing network packages, captured from the network backbone or LAN segments

while other IDSs analyse events generated by the operating systems or application software for signs of intrusion.

3-7-1-1- NIDS (Network-based)

Most of the intrusion detection systems are Network-based. These IDSs detect attacks by capturing and analyzing network packets. Listening in a segment, a NIDS can monitor traffic affecting multiple hosts connected to that network segment, thus protecting these hosts.They are installed more frequently than those based on host because their configuration is general for the entire segment in which they operate The network-based IDSs are often formed by a set of sensors located at various points of the network. These sensors monitor traffic doing local analysis and reporting attacks carried out to the management console.

As sensors are limited to run the detection software, they can be more easily secured against attacks. Many of these sensors are designed to run in hidden mode, so it is more difficult for an attacker to determine their presence and location. There is the possibility of using a one-way cable for reception (sniffing cable), so the sensor can only receive data, preventing physically any outgoing signals. One equivalent option to the one-way cable is the use of a network tap (listening network device) a device similar to a network hub that enables to listen to communications without being detected .

- **Advantages**
  - A well-located IDS can monitor a large network as long as it has enough capacity to analyse the traffic in its totality.
  - The NIDSs have a small impact on the network, usually remaining passive and not interfering with normal operations of the latter.
  - NIDSs can be configured to be invisible to the network in order to increase the security against attacks.
- **Disadvantages**
  - The sensors not only analyse the headers of the packages, they also analyse their content, so they may have difficulties processing all packages in a large network or with much traffic and may fail to recognize attacks during periods of high traffic. Some vendors are trying to solve this problem by implementing IDSs completely in hardware, which makes them much faster.

- The network-based IDSs do not analyse the encrypted information. In environments where communication is encrypted it is unfeasible to examine the package contents and therefore unable to assess whether this is a package with malicious contents or not. This problem is increased when the organization uses encryption in the network layer (IPSec: Internet Protocol Security) between hosts, but can be solved with a more relaxed security policy (eg, IPSec in tunnel mode).
- The network-based IDSs do not know whether the attack was successful or not, the only thing known is that it was launched. This means that after a NIDS detects an attack, administrators must manually investigate every host attacked to determine if the attempt was successful or not.

3-7-1-2- HIDS (Host-based):

HIDS were the first type of IDSs developed and implemented. They run on the information acquired from inside a computer, such as audit files of the operating system. This allows the IDS to analyse actual activities with great precision, determining exactly which processes and users are involved in a particular attack within the operating system.

Like any intrusion detection system, HIDSs also report multiple false positives. Once the system is adjusted, the reduction of false positives is remarkable and then also this type of IDSs ignores very few attacks against the system.

In contrast to NIDSs, HIDSs can see the result of an attempted attack, as well as directly access and monitor data files and processes of the attacked system .

Although NIDSs have greater development and these days are more accepted, HIDS have certain advantages over them:

- **Advantages**
  - The host-based IDSs, having the ability to monitor local events of a host, can detect attacks that cannot be seen by a network- based IDS.
  - They can often operate in an environment in which network traffic travels encrypted, since the source of information is analysed before the data is encrypted on the origin host and / or after the data is decrypted on the destination host.
- **Disadvantages**

- Host-based IDSs are more costly (in time and money) to administer as they must be managed and configured at each monitored host. While the NIDSs have an IDS for multiple monitored systems, HIDSs have an IDS for each of them.
- If the analysis station is within the monitored host, the IDS can be disabled if an attack attains success on the machine
- They are not adequate for detecting attacks on an entire network (for example, port scans) since the IDS only analyses those network packets sent to it.

3-7-2- Analysis type :

There are two approaches to the analysis of events for detecting attacks: detection of signatures and detection of anomalies. The signature detection is the technique used by most commercial systems. The anomalies detection, in which the analysis looks for unusual patterns of activity, has been and remains under investigation. The detection of anomalies is used by a small number of IDSs .

3-7-2-1- Signature-based Detection :

Signature-based detectors analyse system activities looking for events matching a predefined pattern or signature that describes a well-known attack. They collect network traffic and then proceed to analyse it.

The analysis is based on a comparison of patterns (pattern matching). The system contains a database of attack patterns and will be looking for similarities with them and when a match is detected the warning will go off.

These systems are truly effective in detecting attacks but they generate a large number of false positives. Therefore it is necessary that the period in which they get regulated (tuning period) is as short as possible.

The proper operation of such a system depends not only on a good installation and configuration, but also on the fact that the database where the attack patterns are stored is updated.

- **Advantages**
  - Signature detectors are very effective in detecting attacks without generating a large number of false alarms.

- They can quickly and accurately diagnose the use of a specific attack technique. This can help those responsible for security to easily follow security problems and to prioritize corrective actions.

- **Disadvantages**

  - Signature detectors only detect the attacks they previously know, so they must be constantly updated with signatures of new attacks.
  - Many signature detectors are designed to use very tight patterns that prevent them from detecting variants of common attacks.

3-7-2-2- Anomaly-based Detection

The anomaly detection focuses on identifying unusual behavior in a host or a network. They operate assuming that the attacks are different from the normal activity. Anomaly detectors construct profiles representing the normal behavior of users, hosts or network connections.

These profiles are constructed from historical data collected during normal operation. The detectors collect data from the events and use a variety of measures to determine when the monitored activity deviates from normal activity. The measures and techniques used in the detection of anomalies include:

- Detecting a threshold on certain attributes of user behavior. Such behavior attributes may include the number of files accessed by a user in a given period of time, the number of unsuccessful attempts to enter the system, the amount of CPU used by a process, and so on. This level can be static or heuristic.
- Statistic measures, which can be parametric, where it is assumed that the distribution of the profiled attributes fits a certain pattern, or non - parametric, where the distribution of the profiled attributes is learnt from historical values observed over time.

- **Advantages**

  - The IDSs based on anomaly recognition detect unusual behavior. Thus they have the ability to detect attacks for which they have no specific knowledge.
  - Anomaly detectors produce information that is very useful to define new patterns for signature detection.

- **Disadvantages**

- The detection of anomalies produces a high number of false alarms due to the unpredictable behavior of users and networks.
- They require very hard training to characterize patterns of normal behavior.

3-7-3- Response:

Once the events have been analyzed and an attack has been detected, an IDS reacts .Responses can always be grouped into two categories: passive and active. The passive IDSs send reports to some others who will then take action on the matter, if it is appropriate. The active IDSs automatically launch replies to such attacks .

3-7-3-1- Passive Response:

In this type of IDS, the security manager or the system users are notified of what happened. It is also useful to alert the administrator of the site from which the attack was launched, but it is possible that the attacker can monitor the email of the organization or that he has used a false IP for the attack. In that case it would be useless to alert him.

3-7-3-2- Active Response:

The active responses are automatic actions that are taken when certain types of intrusions are detected. Two different categories can be set:

- Collection of additional information: It consist in incrementing the sensor's sensitivity level in order to obtain more clues of the possible attack (e.g. catching all packages from the source that launched the attack, during a certain period of time).
- Changing the environment: Another active response could be to stop the attack; For example, in the case of a TCP connection, the session can be closed by injecting TCP RST segments to the attacker and the victim, or filter the IP address of the intruder or the attacked port, to the access router or to the firewall in order to avoid future attacks.

3-7-4- Detection time :

Two main groups can be identified, those which detect intrusions in real time (in-line) and those which process audit data with some delay (off-line), that means not real time.

Some systems that have in-line detection can also carry out off- line detection over historic audit data. This type of systems combining both types of detection time is called hybrids [15].

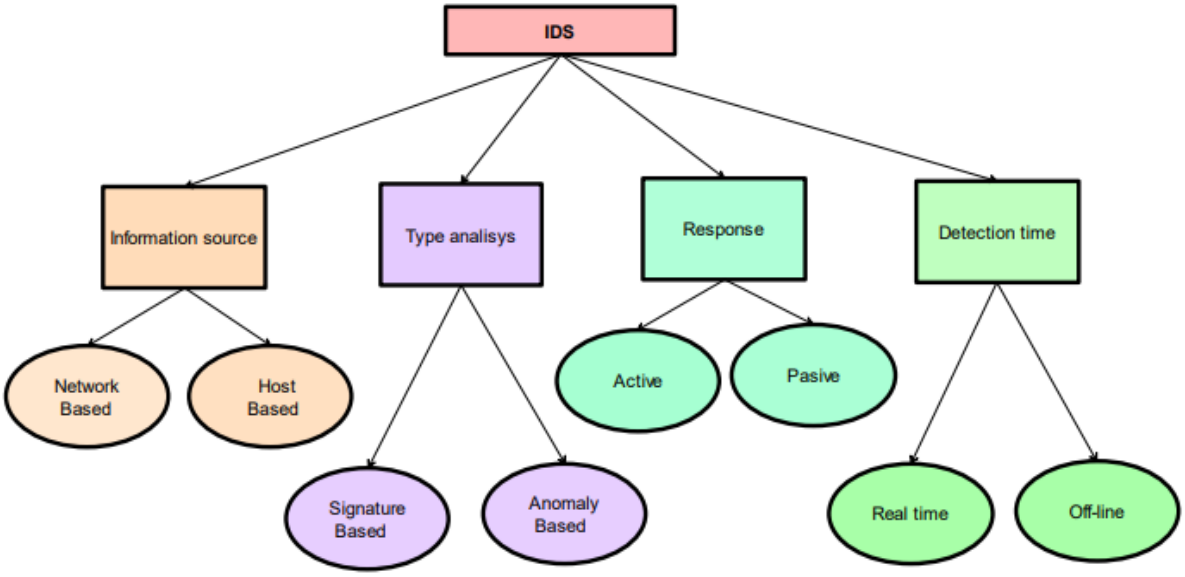


Figure 2.4: IDS classification

3-8- IDS Architectural Model :

Since the publication of Anderson's seminal paper, several intrusion detection systems have been invented. Today there exists a sufficient number of systems in the field for one to be able to form some sort of notion of a 'typical' intrusion detection system, and its constituent parts.

Any generalized architectural model of an intrusion detection system would contain at least the following elements:

- **Audit collection** Audit data must be collected on which to base intrusion detection decisions. Many different parts of the monitored system can be used as sources of data: keyboard input, command based logs, application based logs, etc. In most cases network activity or host-based security logs, or both, are used.
- **Audit storage** Typically, the audit data is stored somewhere, either indefinitely1 for later reference, or temporarily awaiting processing. The volume of data is often exceedingly large2 , making this is a crucial element in any intrusion detection system, and leading some researchers to view intrusion detection as a problem in audit data reduction .
- **Processing** The processing block is the heart of the intrusion detection system. It is here that one or many algorithms are executed to find evidence (with some degree of certainty) in the audit trail of suspicious behavior.

- **Configuration data** This is the state that affects the operation of the intrusion detection system as such; how and where to collect audit data, how to respond to intrusions, etc. This is thus the SSO's main means of controlling the intrusion detection system. This data can grow surprisingly large and complex in a real world intrusion detection installation. Furthermore, it is relatively sensitive, since access to this data would give the competent intruder information on which avenues of attack are likely to go undetected.
- **Reference data** The reference data storage stores information about known intrusion signatures—for misuse systems—or profiles of normal behaviour—for anomaly systems. In the latter case the processing element updates the profiles as new knowledge about the observed behaviour becomes available. This update is often performed at regular intervals in batches. Stored intrusion signatures are most often updated by the SSO, as and when new intrusion signatures become known.
- **Active/processing data** The processing element must frequently store intermediate results, for example information about partially fulfilled intrusion signatures. The space needed to store this active data can grow quite large.
- **Alarm** This part of the system handles all output from the system, whether it be an automated response to suspicious activity, or more commonly the notification of a SSO[19].

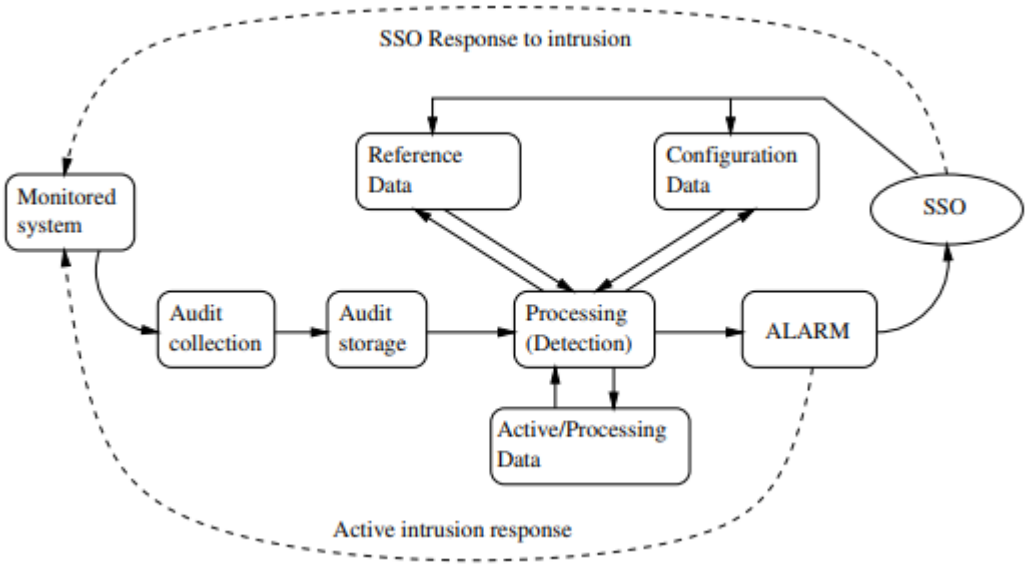


Figure2.5: IDS Architectural Model .

3-9- Where to place an IDS :

There are many ways to add the IDS tools to our network; each has its advantages and its disadvantages. The best choice would be a compromise between cost and desired properties, while maintaining a high level of benefits and a controlled number of disadvantages, all in accordance with the needs of the organization.

For this reason, the positions of the IDS within a network provide different characteristics. Then we will see different possibilities in the same network. We will suppose that we have a network where a firewall divides the Internet from the demilitarized zone (DMZDemilitarized Zone), and another one that divides the DMZ from the intranet of the organization as shown in the next figure. The DMZ is the area between the Internet and the internal network. It is used to provide public services without having to allow access to the private network of the organization. In this subnet are usually located the main services such as HTTP, DNS and other servers [15].

3-9-1- In front of the external firewall:

In this position the IDS will capture all the incoming and outgoing traffic of our network, so it will monitor the number and type of attacks against the organization infrastructure, and the external firewall. IDSs in this location should be configured with a low sensitivity since the number of false alarms here is high.

The main drawbacks of this location are that the IDSs can't detect attacks using in their communications some methods to hide information, such as encryption algorithms, and that in this location the traffic rate is usually so high that the IDSs can't monitor all the packages .

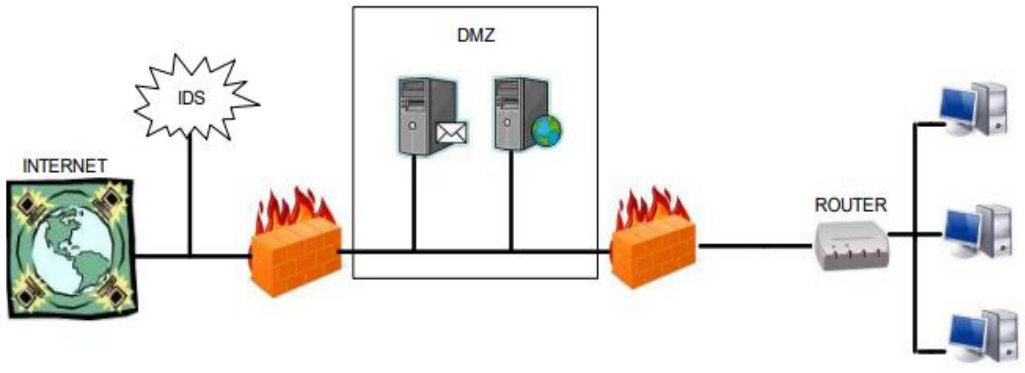


Figure2.6:place the IDS In front of the external firewall

3-9-2- Behind the external firewall

Another option is to place the IDS within the demilitarized zone, between the firewalls. Intrusions that successfully pass through the main firewall are monitored. Attacks against servers that provide public services in this subnet can be detected. The identification of the most common attacks enhances the main firewall configuration to be able to block them next time.

As in the previous case, the drawbacks are about the encrypted attacks, and the saturation of the IDS due to the high traffic rate. This area has fewer false alarms since at this point access only to our servers should be allowed .

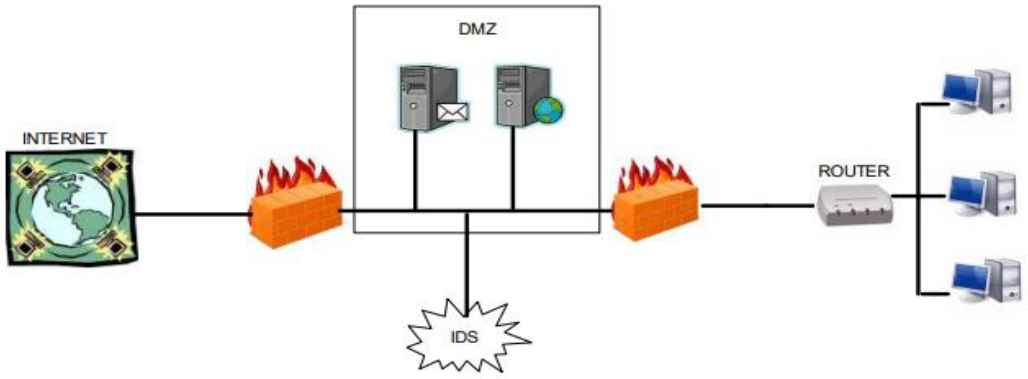


Figure2.7:place the IDS Behind the external firewall

3-9-3- Behind the second firewall

In this case the IDS is located between the second firewall and the internal network. It is not inside the internal network so it will not listen to any internal traffic. This IDS should be less powerful than those commented on before, as the volume of traffic is smaller at this point.

Any atypical traffic that comes up here must be considered hostile. At this point of the network fewer false alarms will occur, so any alarm from the IDS should be immediately studied.

This implementation makes these systems particularly vulnerable to attacks, not only from the outside but also inside their own infrastructure. It is vital to keep this in mind when implementing an intrusion detector in this location, in order to detect attacks produced from within the network itself, such as those launched by internal staff [15].

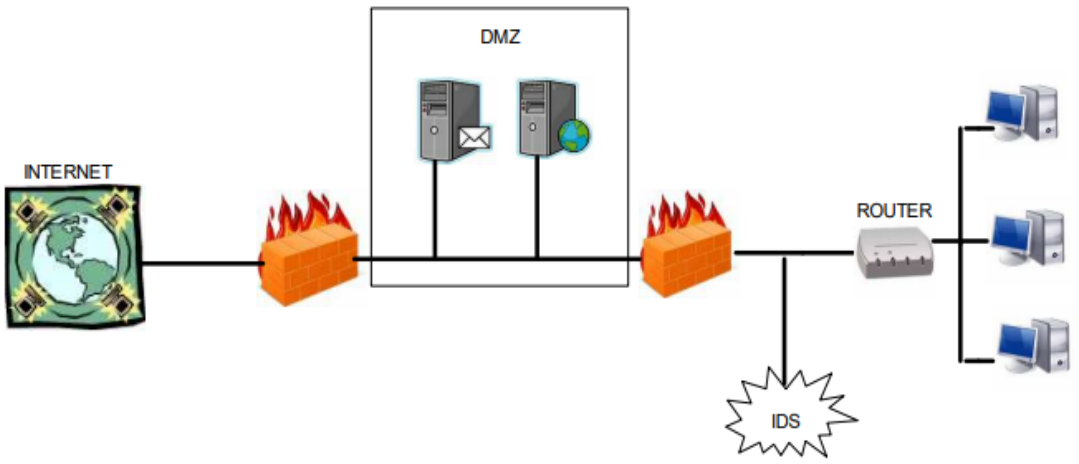


Figure2.8:place the IDS Behind the second firewall.

4- Related Work:

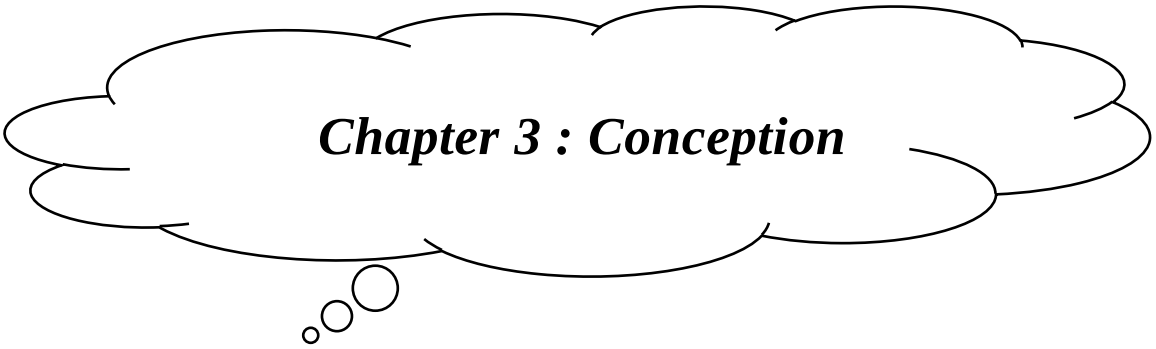
- The proposed methodology consists of four key steps which include: data acquisition, data cleaning, data conversion and attack pattern recognition. The first step of the proposed methodology is to acquire network traffic data. Once the data is acquired, it is preprocessed in order to get the refined data. During the preprocessing, we will perform two major steps, i.e., data cleaning and conversion of cleaned data into three channel images. The final step is to train and test the CNN model over the preprocessed data in order to evaluate the performance for detecting the DoS and DDoS attack patterns[20].
- the proposed framework reveals any strange movement by investigating the traffic and ordering this action as indicated by the ordinary exercises recorded for various iot gadgets. the data used in this study is considered as big data as it is extracted from several iot devices. therefore, there are many challenges that we will encounter with this type of data such as it is very large, variety, unstructured, can contain missing data and requires a new high-performance processing. the suggested framework divided into the following main phases as shown in the fig.1 below: dataset preparation, data preprocessing[21]
- In this study paper, we have considered the Naive Bayes(NB) , Bayes Network(BN), Radial Bias Function (RBF),Multi-Layer Perceptron (MLP),Back Propagation Network(BPN), Random Forest(RF), J48,Radial Basis Function Network (RBFN) classification techniques for this experiment. After several comparisons are made based on UCI data using above mentioned classification algorithms, we have observed that correctly classified instances

percentage is 100% for Random Forest and it is the highest accuracy rate compared to other classification algorithms[22].

- The intention and purpose of the thesis was to investigate how the neural network method would perform against the traditional/classic statistical machine learning methods in a document classification context applied with Swedish documents. Studying different methods in traditional/classic statistical area, the overall performance of the neural network is outperforming most of the classifiers. This is clearly shown from the result of the test accuracy. However, the SVM classifier is performing at the same high level of test accuracy and the other performance metrics measurements as well.[23]

5- Conclusion:

Most IDS are reliable, which explains that they are often integrated into safety solutions. The advantages they have in the face of other safety tools favor them.in This chapter we present the intrusion detection system ,the different between it and firewall and why is better also the Detection technologies and more ,in the next chapter we will talk about the classification methods and the proposed methodology.


Chapter 3 : Conception

1- Introduction :

Over the past few years, information technology has become widespread . Along with this rapid development, attacks on information have increased greatly and have become not only numerous and diverse but also sophisticated. With the growing complexity of attacks that exploited weaknesses in software and hardware systems and the advent of new ones, many solutions such as IDS , in this chapter we will present the conception and the methodology of our work.

2- DDOS Attack in IOT :

2-1- Definition :

A Distributed Denial of Service (DDoS) attack is a type of cyber-attack where there are many devices attacking a single server. This is usually done by overloading the server's connection and preventing it from receiving any more data. The devices that launch this type of attack can be computers, servers, or even personal gadgets such as smartphones, but they all have one thing in common: they must be connected to the internet to participate in the attack. DDoS attacks are so common that the largest company in this industry, Cloudflare, had to upgrade its service and add several new features and an extra layer of protection to cope with the growing demand. The overview of the DDoS attack is shown in figure 3.1[24].

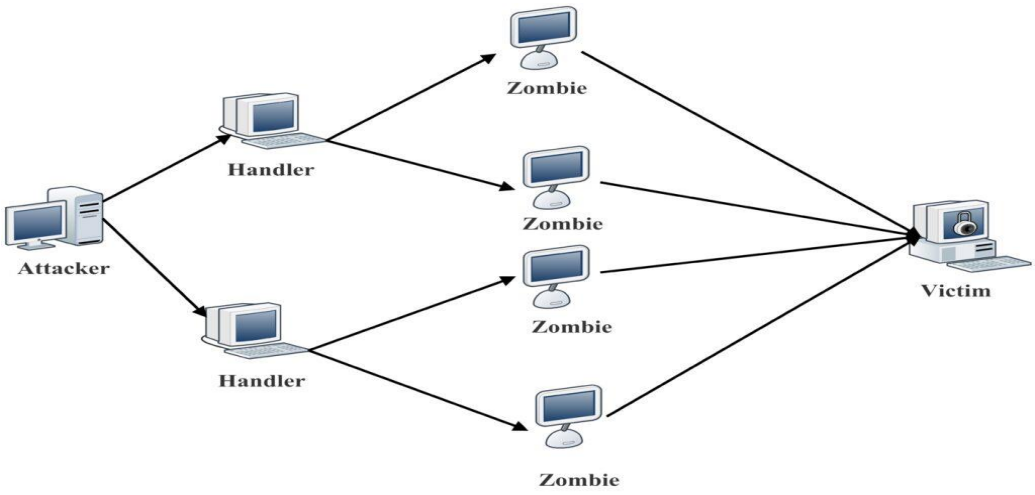


Figure 3.1: DDoS attack.

2-2- DDoS Types : we have select 5 attack NTP, UDP flood, SYN flood ,DNS flood ,NetBIOS attack.

- **NTP** : In NTP amplification attacks, the perpetrator exploits publically-accessible Network Time Protocol (NTP) servers to overwhelm a targeted server with UDP traffic. The attack is defined as an amplification assault because the query-to-response ratio in such scenarios

is anywhere between 1:20 and 1:200 or more. This means that any attacker that obtains a list of open NTP servers (e.g., by a using tool like Metasploit or data from the Open NTP Project) can easily generate a devastating high-bandwidth, high-volume DDoS attack.

- **UDP flood :** A UDP flood, by definition, is any DDoS attack that floods a target with User Datagram Protocol (UDP) packets. The goal of the attack is to flood random ports on a remote host. This causes the host to repeatedly check for the application listening at that port, and (when no application is found) reply with an ICMP 'Destination Unreachable' packet. This process saps host resources, which can ultimately lead to inaccessibility**.**
- **SYN Flood:** A SYN flood DDoS attack exploits a known weakness in the TCP connection sequence (the "three-way handshake"), wherein a SYN request to initiate a TCP connection with a host must be answered by a SYN-ACK response from that host, and then confirmed by an ACK response from the requester. In a SYN flood scenario, the requester sends multiple SYN requests, but either does not respond to the host's SYN-ACK response, or sends the SYN requests from a spoofed IP address. Either way, the host system continues to wait for acknowledgement for each of the requests, binding resources until no new connections can be made, and ultimately resulting in denial of service[25].
- **DNS Flood**: Domain Name Servers (DNS) are computer servers that translate website URLs into their IP addresses. For example, when you visit Facebook to view your friends' pages, you type Facebook[.]com into your browser. In effect, you're telling your computer to go to one of Facebook's IP addresses (Facebook has a huge number of them, considering the amount of traffic it has to accommodate). A Facebook IP address of this type is 66.220.144.0.DNS servers translate the name of the website you know into its real IP address.
- **NetBIOS** : NetBIOS is a data communication protocol with which external systems/applications can communicate over a local network. For example, it was used to gain remote access to shared folders on an internal network. The NetBIOS service is now out of date and is hardly used anymore. If the NetBIOS service is still enabled and is publicly accessible, NetBIOS can be abused in an amplification (D) DoS attack.Simply put, in an 'amplification attack' an amount of data is sent to your UDP port from a spoofed IP. Your VPS then sends a significantly larger amount of data back to the actual IP. This way, malicious parties can abuse your UDP port to perform a (D)DoS attack on the spoofed IP

address.For this reason, it is not permitted that the NetBIOS service is publicly accessible on a VPS at TransIP[26].

2-3- Why DDOS attacks more dangerous in IoT? :

These attacks are made more dangerous because perpetrators can now use the Internet of things (IoT) to make them more severe .They can do this by exploiting known vulnerabilities in-home devices like Wi-Fi routers, security cameras, and smart TVs. Assailants may utilize these susceptible devices to deliver a flood of traffic to select sites, disabling their servers. These attacks also pose a more serious threat to individuals whose devices are used in botnets. Many of these botnet victims do not even know their devices are being used in this way, leaving them vulnerable to identity theft or even physical harm[24].

3- Classification:

3-1- Definition:

Classification is the process of recognizing, understanding, and grouping ideas and objects into preset categories or "sub-populations." Using pre-categorized training datasets, machine learning programs use a variety of algorithms to classify future datasets into categories.

Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. One of the most common uses of classification is filtering emails into "spam" or "non-spam."

In short, classification is a form of "pattern recognition," with classification algorithms applied to the training data to find the same pattern (similar words or sentiments, number sequences, etc.) in future sets of data.Using classification algorithms, which we'll go into more detail about below, text analysis software can perform tasks like aspect-based sentiment analysis to categorize unstructured text by topic and polarity of opinion (positive, negative, neutral, and beyond)[31]. A Two-Step Process

- **Training:** describing a set of predetermined classes. Each sample is assumed to belong to a predefined class, as determined by the class label attribute. The set of tuples used for model construction: training set. The model is represented as classification rules, decision trees, or mathematical formula.
- **Classification:** for classifying future or unknown objects. Estimate accuracy of the model. The known label of test sample is compared with the classified result from the model.

Accuracy rate is the percentage of test set samples that are correctly classified by the model. Test set is independent of training set, otherwise over-fitting will occur.

3-2- Classification Methods :

The study of classification in statistics is vast, and there are several types of classification algorithms you can use depending on the dataset you're working with. Below are seven of the most common algorithms in machine learning.

Popular Classification Algorithms:

- Random Forest[29].
- Naive Bayes[30].
- K-Nearest Neighbors(KNN)[31].
- Decision Tree(REPTree) [29],[32].
- Support Vector Machines(SVM)[33].
- Multilayer Perceptron (MLP)[34],[35].
- Decision Tree-J48 [36].

3-2-1- Random Forest

Random Forest is an ensemble model where, multiple decision trees are combined to get a stronger model. The derived model will be more robust, accurate and handles overfitting better than constituent models.

- **Basic Theory :**

Random Forest have a set of decision trees ensemble with "bagging method" to obtain classification and regression outputs. In classification, it calculates the output using majority voting , whereas in regression, mean is calculated.

Random Forest comes up with a robust, accurate model that can handle large varieties of input data with binary, categorical, continuous features.

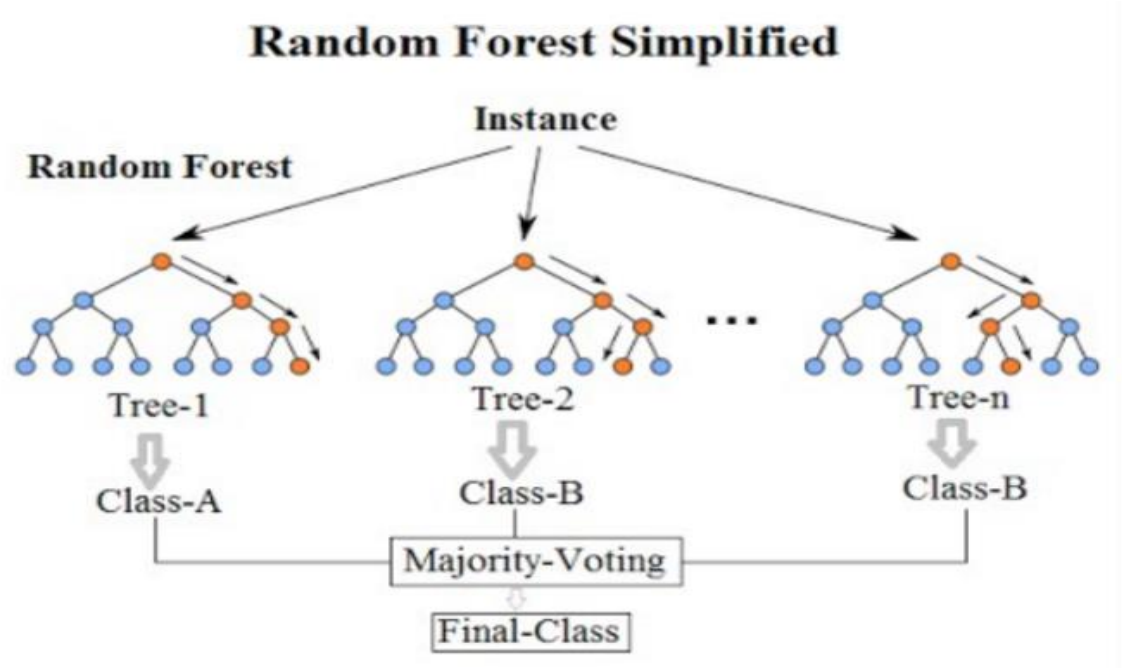


Figure3.2:Random Forest Simplified.

- **Advantages :**
  - Accurate and powerful model.
  - handles overfitting efficiently.
  - Supports implicit feature selection and derives feature importance.

3-2-2- Naive Bayes:

Naive Bayes is a classification algorithm. Traditionally it assumes that the input values are nominal, although it numerical inputs are supported by assuming a distribution. Naive Bayes uses a simple implementation of Bayes Theorem (hence naive) where the prior probability for each class is calculated from the training data and assumed to be independent of each other (technically called conditionally independent).

This is an unrealistic assumption because we expect the variables to interact and be dependent, although this assumption makes the probabilities fast and easy to calculate. Even under this unrealistic assumption, Naive Bayes has been shown to be a very effective classification algorithm [27].

Naive Bayes calculates the posterior probability for each class and makes a prediction for the class with the highest probability. As such, it supports both binary classification and multi-class classification problems. Naive Bayes calculates the possibility of whether a data point belongs

within a certain category or does not. In text analysis, it can be used to categorize words or phrases as belonging to a preset "tag" (classification) or not.

- **Basic Theory:**

Naive bayes model is based on Thomas Baye's bayes rule. The bayes rule can be stated as ,

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

In the above equation,

P(A|B) : (posterior probability) probability of event A to happen when event B is true.

P(A) ,P(B) : probability of event A and event B to happen.

P(B|A) : (likelihood) probability of event B to happen when event A is true.

The basic logic is to derive the probability of output label(Y) given the input X , from individual probabilities of features(Xi) given output label as Y, from the training data.

$$P(C_i|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n|C_i).P(C_i)}{P(x_1, x_2, \ldots, x_n)} \text{ for } 1 \leq i \leq k$$

$$P(C_i|x_1, x_2, \ldots, x_n) = \left(\prod_{j=1}^{j=n} P(x_j|C_i)\right) . \frac{P(C_i)}{P(x_1, x_2, \ldots, x_n)} \text{ for } 1 \leq i \leq k$$

Please notice in the above equation that naive bayes assumes all the features to be independent. The word "naive" itself is used to remind this. In case of multiple class labels, P(Ci|X) is calculated for each labels and label with maximum probability is chosen as the output.

There are few alternatives of naive bayes theorem as listed below :

- Gaussian : Gaussian naive bayes assumes features to be following Gaussian distribution.
- Multinomial : Multinomial naive bayes distribution is used when the data is assumed to be in multinomial distribution.

- **Advantage:**
  - It is easy and fast to predict the class of the test data set. It also performs well in multi-class prediction.

- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

3-2-3- K-nearest Neighbors :

K-nearest neighbors (k-NN) is a pattern recognition algorithm that uses training datasets to find the k closest relatives in future examples. When k-NN is used in classification, you calculate to place data within the category of its nearest neighbor. If k = 1, then it would be placed in the class nearest 1. K is classified by a plurality poll of its neighbors. The k-nearest neighbors algorithm supports both classification and regression. It is also called kNN for short.

It works by storing the entire training dataset and querying it to locate the k most similar training patterns when making a prediction. As such, there is no model other than the raw training dataset and the only computation performed is the querying of the training dataset when a prediction is requested. It is a simple algorithm, but one that does not assume very much about the problem other than that the distance between data instances is meaningful in making predictions. As such, it often achieves very good performance.

When making predictions on classification problems, KNN will take the mode (most common class) of the k most similar instances in the training dataset.[27]

- **Basic Theory :**

The basic logic behind KNN is to explore your neighborhood, assume the test datapoint to be similar to them and derive the output. In KNN, we look for k neighbors and come up with the prediction.

In case of KNN classification, a majority voting is applied over the k nearest datapoints whereas, in KNN regression, mean of k nearest datapoints is calculated as the output. As a rule of thumb, we selects odd numbers as k. KNN is a lazy learning model where the computations happens only runtime.

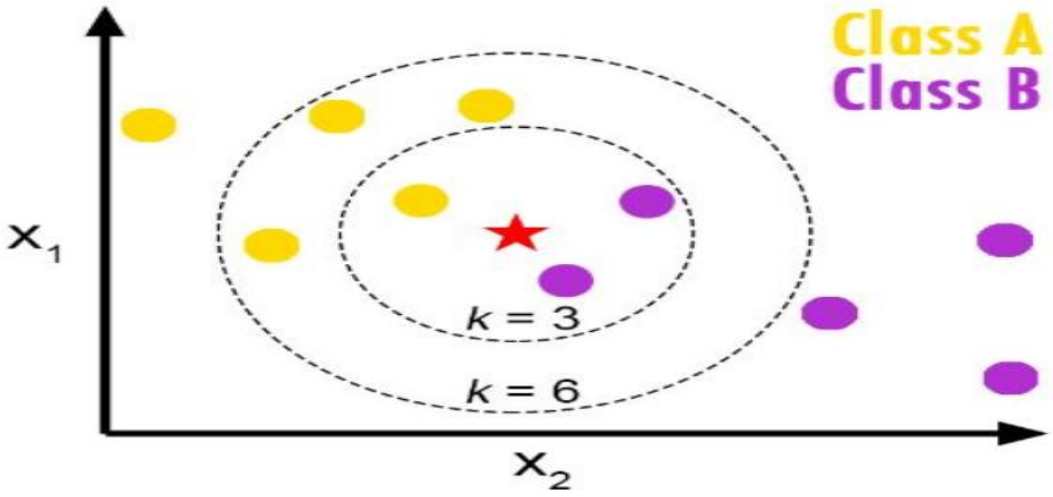


Figure3.3:KNN diagram.

In the above diagram yellow and violet points corresponds to Class A and Class B in training data. The red star, points to the testdata which is to be classified. when k = 3, we predict Class B as the output and when K=6, we predict Class A as the output [29]

- **Advantage:**
  - No Training Period: KNN is called Lazy Learner (Instance based learning). It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.
  - ince the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm.
  - KNN is very easy to implement. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)[37]

3-2-4- Decision Tree(REPTree):

A decision tree is a supervised learning algorithm that is perfect for classification problems, as it's able to order classes on a precise level. It works like a flow chart, separating data points into two similar categories at a time from the "tree trunk" to "branches," to "leaves," where the categories become more finitely similar. This creates categories within categories, allowing for organic classification with limited human supervision.

- **Basic Theory :**

Decision tree is derived from the independent variables, with each node having a condition over a feature. The nodes decides which node to navigate next based on the condition. Once the leaf node is reached, an output is predicted. The right sequence of conditions makes the tree efficient. entropy/Information gain are used as the criteria to select the conditions in nodes. A recursive, greedy based algorithm is used to derive the tree structure.

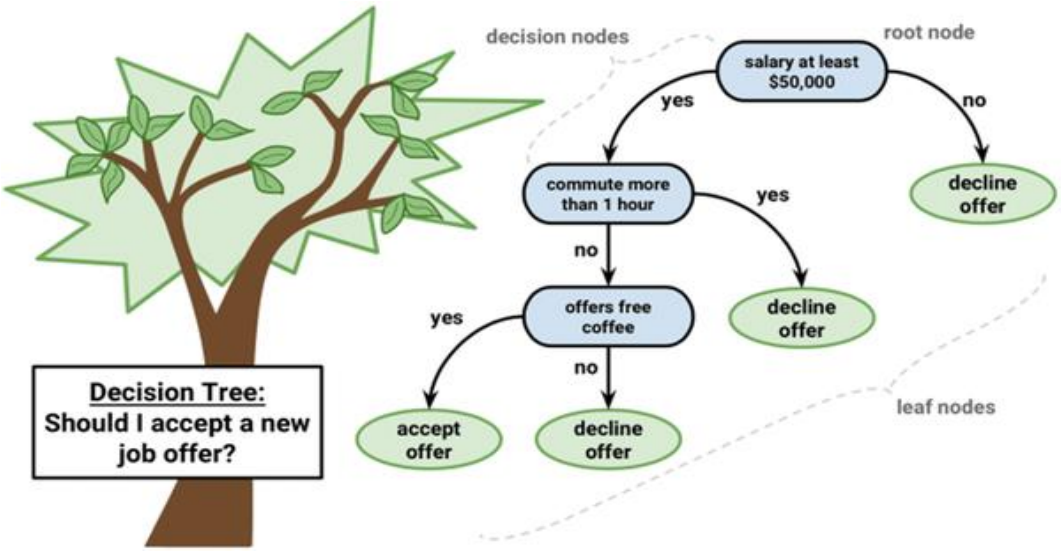


Figure3.4:Decision Tree diagram.

In the above diagram, we can see a tree with set of internal nodes(conditions) and leaf nodes with labels(decline/accept offer).

- **Algorithm to select conditions :**

for CART(classification and regression trees), we use gini index as the classification metric. It is a metric to calculate how well the datapoints are mixed together.

$$giniindex = 1 - \sum P_t^2$$

the attribute with maximum gini index is selected as the next condition, at every phase of creating the decision tree. When set is unequally mixed, gini score will be maximum.

For Iterative Dichotomiser 3 algorithm, we use entropy and information gain to select the next attribute. In the below equation, H(s) stands for entropy and IG(s) stands for Information gain. Information gain calculates the entropy difference of parent and child nodes. The attribute with maximum information gain is chosen as next internal node[29].

$$H(s) = -\sum P_c \cdot log(P_c)$$
$$IG(s) = H(s) - \sum_t P_t \cdot H(t)$$

- **Advantage:**
  - It can be used for both classification and regression problems: Decision trees can be used to predict both continuous and discrete values i.e. they work well in both regression and classification tasks.As decision trees are simple hence they require less effort for understanding an algorithm.
  - It can capture nonlinear relationships: They can be used to classify non-linearly separable data.
  - They are very fast and efficient compared to KNN and other classification algorithms.
  - The data type of decision tree can handle any type of data whether it is numerical or categorical, or boolean.
  - Normalization is not required in the Decision Tree.
  - The decision tree is one of the machine learning algorithms where we don't worry about its feature scaling. Another one is random forests. Those algorithms are scale-invariant.
  - It gives us and a good idea about the relative importance of attributes[28].

3-2-5- Support Vector Machines(SVM):

Support Vector Machines were developed for binary classification problems, although extensions to the technique have been made to support multi-class classification and regression problems. The algorithm is often referred to as SVM for short.SVM was developed for numerical input variables, although will automatically convert nominal values to numerical values. Input data is also normalized before being used.

SVM work by finding a line that best separates the data into the two groups. This is done using an optimization process that only considers those data instances in the training dataset that are closest to the line that best separates the classes. The instances are called support vectors, hence the name of the technique. In almost all problems of interest, a line cannot be drawn to neatly separate the classes, therefore a margin is added around the line to relax the constraint, allowing some instances to be misclassified but allowing a better result overall.

Finally, few datasets can be separated with just a straight line. Sometimes a line with curves or even polygonal regions need to be marked out. This is achieved with SVM by projecting the data into a higher dimensional space in order to draw the lines and make predictions. Different

kernels can be used to control the projection and the amount of flexibility in separating the classes. A support vector machine (SVM) uses algorithms to train and classify data within degrees of polarity, taking it to a degree beyond *X/Y* prediction. [27]

- **Basic Theory:**

Support Vector Machine is a supervised learning technique extensively used in text classification, image classification, bioinformatics etc.

In Linear SVM, the problem space must be linearly separable. A hyperplane is derived by the model, that maximizes the classification margin. The hyperplane will be an N-1 dimensional subspace if there are N features present. The boundary nodes in the feature space are called support vectors. Based on the their relative position, the maximum margin is derived and an optimal hyperplane is drawn in the midpoint.

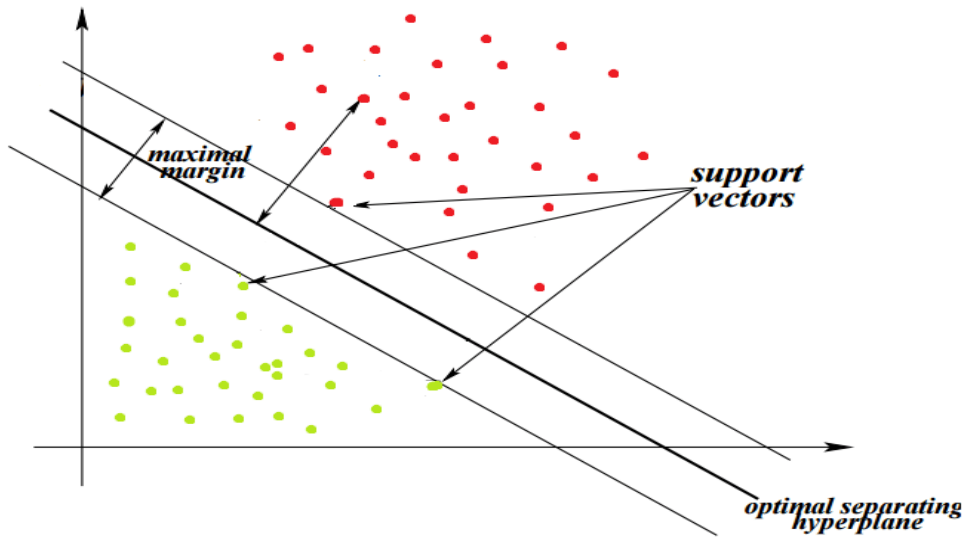


Figure3.5:SVM algorithm.

Value of margin(m) will be inversely proportional to ||w||, where w is the set of weight matrices. For maximizing margin, we'll have to minimize||w||. The optimization problem will be,

$$\text{Minimize} \quad \frac{||\vec{w}||^2}{2} \quad \text{where} \quad y_i\,(\vec{w}.\vec{x}+b) \geq 1 \text{ for any } i = 1,\ldots,n$$

The above optimization works well for fully linearly separable solutions. For handling outliers, we need a slack term as below. The second term uses the hinge loss to get slack variable.

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\sum_i \max(0, 1 - y_i(w^\intercal x_i + b))$$

C is the regularization parameter that balances the miss penalty and margin width. As the mathematical explanations are well above the scope of this story, I will not be explaining them in depth.

The basic logic is that, to minimize the cost function, w is forced to tune with maximum margin between the classes. C value will decide the level of regularization applied over the datasets. It decides the level (soft/hard) margin to be applied over the datasets. In short, C is the level of ignorance over outliers.

Non-linear SVM when the dataset is not linearly separable. A kernel function is used to derive a new hyperplane for all the training data. The distribution of labels in new hyperplane will be such that training data will be linearly separable. Later, a linear curve will classify the labels in the hyperplane. When the classification results are projected back to the feature space, we get a non linear solution.

The only change in the equation here is, the introduction of a new kernel function. The new equation will look like,

$$\text{Minimize } \frac{||\vec{w}||^2}{2} + C\sum_i \zeta_i \quad \text{where } y_i(\vec{w}.\phi(x_i) + b) \geq 1 - \zeta_i \text{ for all } 1 \leq i \leq n, \zeta_i \geq 0$$

The xi will be replaced by ϕ(xi) which will transform the dataset into the new hyperplane.[29]

- **Advantage :**
  - SVM works relatively well when there is a clear margin of separation between classes.
  - SVM is more effective in high dimensional spaces.
  - SVM is effective in cases where the number of dimensions is greater than the number of samples.
  - SVM is relatively memory efficient[38].

3-2-6- Multilayer Perceptron (MLP):

The Multilayer Perceptron was developed to tackle this limitation. It is a neural network where the mapping between inputs and output is non-linear.

A Multilayer Perceptron has input and output layers, and one or more **hidden layers** with many neurons stacked together. And while in the Perceptron the neuron must have an activation function that imposes a threshold, like ReLU or sigmoid, neurons in a Multilayer Perceptron can use any arbitrary activation function.

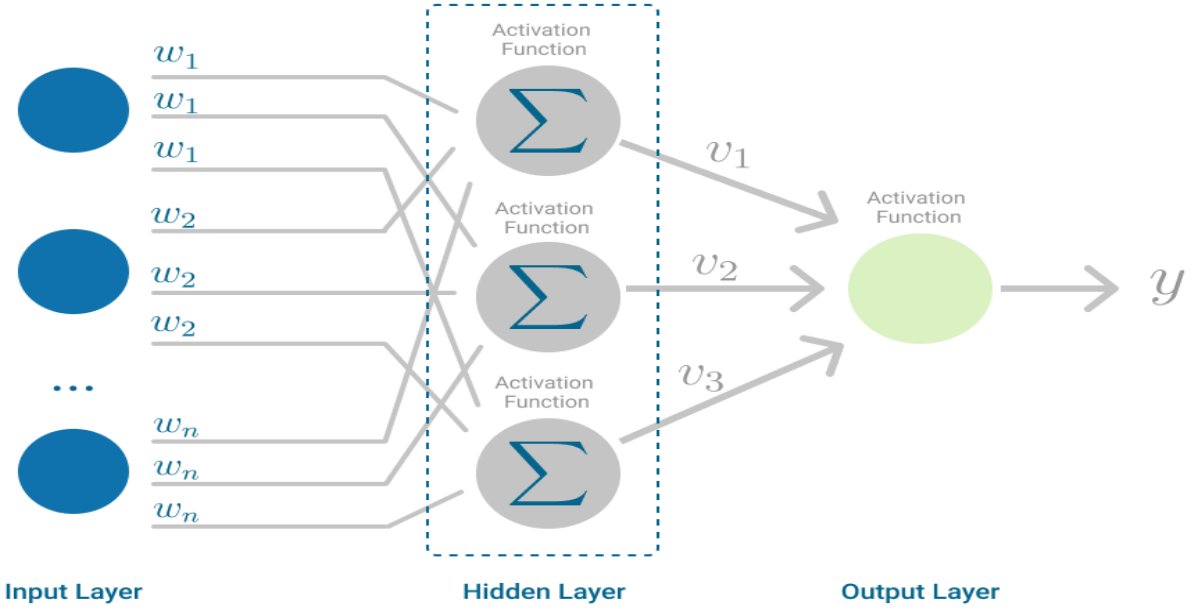


Figure3.6:MLP algorithm.

Multilayer Perceptron falls under the category of feedforward algorithms, because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer.

Each layer is feeding the next one with the result of their computation, their internal representation of the data. This goes all the way through the hidden layers to the output layer.[40]

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include [39]:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- One of the preferred techniques for gesture recognition.

- MLP/Neural networks do not make any assumption regarding the underlying probability density functions or other probabilistic information about the pattern classes under consideration in comparison to other probability based models
- They yield the required decision function directly via training. 5. A two layer backpropagation network with sufficient hidden nodes has been proven to be a universal approximator.

3-2-7- Decision Tree-J48:

The C4.5 algorithm is a classification algorithm which produces decision trees based on information theory. It is an extension of Ross Quinlan's earlier ID3 algorithm also known in Weka as J48, J standing for Java. The decision trees generated by C4.5 are used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

The J48 implementation of the C4.5 algorithm has many additional features including accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc. In the WEKA data mining tool, J48 is an open-source Java implementation of the C4.5 algorithm. J48 allows classification via either decision trees or rules generated from them.

This algorithm builds decision trees based on a set of training data in the same way the ID3 algorithm does, by using the concept of information entropy. The training data is a set S={s1, s2, …} of already classified samples. Each sample si consists of a p-dimensional vector (x1, i, x2, i, …, xp, i) where the xj represents the attribute values or features of the corresponding sample, as well as the class in which the sample falls. To gain the highest classification accuracy, the best attribute to split on is the attribute with the greatest information [40].

4- Proposed Methodology:

In this section, the steps of the proposed methodology for DDoS attack detection are discussed.

- In the first step, I extract the CICDDoS 2019 datasets .
- In the second step ,preprocessing the dataset.
- In the third step ,I apply machine learning techniques for the classification of DDoS attacks.
- Consult the result and select the best classifier.

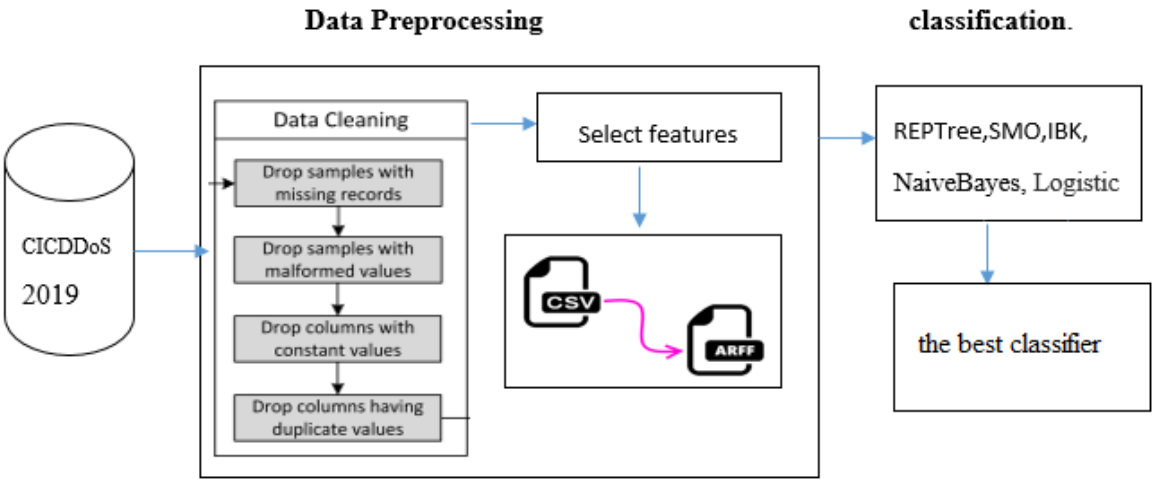


Figure 3.7 shows the workflow of the proposed methodology for DDoS attack detection. Each step of the proposed method is explained in the following subsections Figure3.7: Proposed Methodology For Detecting IoT DDoS Attacks using Machine learning

4-1- Dataset:

The CICDDoS2019 datasets are extracted from the respective websites http://205.174.165.80/CICDataset/CICDDoS2019/Dataset/ contains benign and the most up-to-date common DDoS attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter-V3 with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files).

In this dataset, we have different modern reflective DDoS attacks such as PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS and SNMP. Attacks were subsequently executed during this period. As Table III shows, we executed 12 DDoS attacks includes NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN and TFTP on the training day and 7 attacks including PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag and SYN in the testing day. The traffic volume for WebDDoS was so low and PortScan just has been executed in the testing day and will be unknown for evaluating the proposed model.

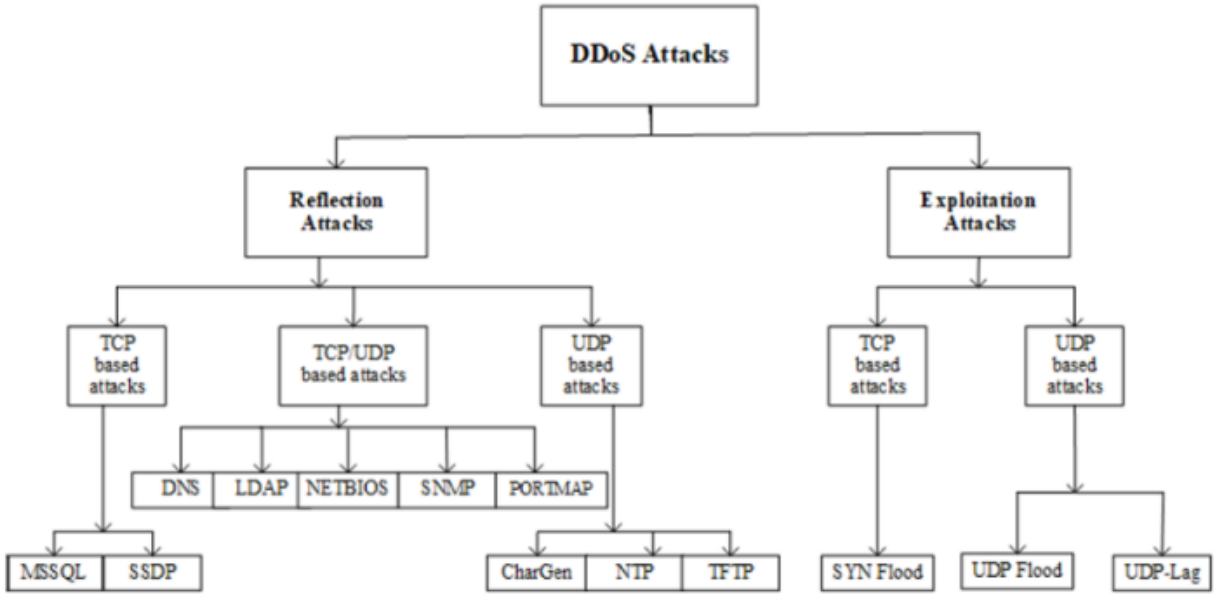


Figure3.8: DDoS attack Types.

4-2- Data Preprocessing : Once the data is acquired, the next stage is to preprocess the data in order to bring it in a refined form.

- **Data Cleaning:** The acquired network traffic data was in .csv format which includes more than 80 flow features. In order to better train
  - removed the unwanted features from the data set which are not useful for classifying the attack and normal traffic. As based upon these static features, one cannot decide whether a certain flowid, srcIP, etc., whenever found will always generate malicious or normal packets. That's why we dropped such unwanted features and excluded them from My training set.
  - we analyzed the whole dataset in order to deal with missing or malformed data. For this purpose, we first checked that which samples contain missing values, which samples have inadequate values like nan, -inf, +inf, etc. As we had a large number of samples in the dataset, so we dropped all those samples which comprise of missing or malformed values.

Flow ID	Bwd PSH Flags	Bwd Avg Bulk Rate
Source IP	Fwd URG Flags	SimillarHTTP
Source Port	Bwd URG Flags	Fwd PSH Flags
Destination IP	FIN Flag Count	Fwd Packet Length Max

Destination Port	PSH Flag Count	Fwd Packet Length Min
Protocol	ECE Flag Count	Fwd Packet Length Mean
Timestamp	Fwd Avg Bytes/Bulk	Fwd Packet Length Std
Max Packet Length	Fwd Avg Packets/Bulk	Bwd Packet Length Std
Packet Length Mean	Fwd Avg Bulk Rate	Min Packet Length
Packet Length Std	Bwd Avg Bytes/Bulk	min_seg_size_forward
Packet Length Variance	Bwd Avg Packets/Bulk	RST Flag Count
CWE Flag Count	Avg Fwd Segment Size	Fwd Header Length
Bwd Packet Length Min	Subflow Bwd Packets	Subflow Fwd Packets
Bwd Packet Length Mean	Subflow Bwd Bytes	Subflow Fwd Bytes
Init_Win_bytes_backward		

Table3.1:the droppted features.

After that, we figured out the features which were either duplicate or entirely had a constant value in case of all labels. Such constant features are not useful for discriminating the attack or normal traffic and may decrease the performance of the machine learning model, if included in the training set. Therefore, we also dropped constant features from the training set.

- **Select features:** , after the cleaning the data we left with 41 features which were unique and important.

1. Flow Duration 2. Total Fwd Packets 3. Total Backward Packets 4. Total Length of Fwd Packets 5. Total Length of Bwd Packets 6. Bwd Packet Length Max 7. Flow IAT Mean 8. Flow IAT Std 9. Flow IAT Max 10. Flow IAT Min 11. Fwd IAT Total 12. Fwd IAT Mean 13. Fwd IAT Std 14. Fwd IAT Max 15. Fwd IAT Min	16. Bwd IAT Total 17. Bwd Header Length 18. Fwd Packets/s 19. Bwd Packets/s 20. SYN Flag Count 21. ACK Flag Count 22. URG Flag Count 23. Down/Up Ratio 24. Average Packet Size 25. Avg Bwd Segment Size 26. Fwd Header Length1 27. Init_Win_bytes_forward 28. act_data_pkt_fwd 29. Active Mean 30. Active Max	31. Active Min 32. Active Std 33. Idle Std 34. Idle Max 35. Bwd IAT Mean 36. Bwd IAT Std 37. Bwd IAT Max 38. Bwd IAT Min 39. Idle Min 40. Idle Mean 41. Class

Table3.2:the selected features.

- **Convert data csv to arff :** to convert the csv table to arff do this steps
  - Open Weka. If you're working in Weka, you have a built-in tool that will convert your .CSV files to the .ARFF format. You'll usually find Weka in the Applications folder.
  - Click the Tools menu. It's in the menu bar at the top of the Weka window.
  - Click ArffViewer. This opens a blank window called ″ARFF-Viewer.″
  - Click the File menu. It's at the top of the ARFF-Viewer window.
  - Click Open. A file browser window will appear. And Navigate to the folder that contains the .CSV file.
  - Select CSV data files (*.csv) from the ″Files of Type″ menu. You should now see the .CSV file you need to convert in the window.
  - Select the .CSV and click Open. This opens the file in the viewer.
  - Click the File menu then Click Save As.
  - Name the file. The file name must end with ″.ARFF″ (e.g., mydata.ARFF).
  - Click Save. The .CSV file is now converted to the .ARFF format [41].

4-3- Attack Classification :

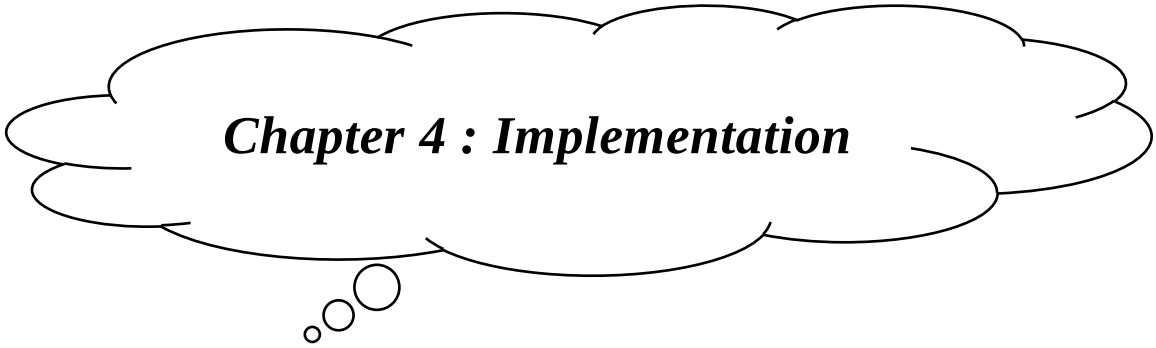
In the last step we classify the data using the best 5 machine learning algorithm ,after select the csv or arff file we select the Test Options there are four testing options as listed below :

- Training set
- Supplied test set
- Cross-validation
- Percentage split

Select the classifier finally checking the results.

5- Conclusion:

Classification of network attack is an important network security research. In this chapter we introduced the most popular attack classification, we starting by explaining the DDOS attack and its dangerous in IOT, then define what is a classification and its methods select the best 5 machine learning algorithm used to choose a better classifier to achieve the objective of this thesis. working on the CICDDOS2019 dataset. Then, we will present the steps to realize our proposed methodology.in the next chapter n the next chapter we will implement the the methode and consult the results.


Chapter 4 : Implementation

1- Introduction:

In this chapter, we will implement our proposed methodology starting with a description of the CICDDOS2019 database used. Then, we will apply the seven machine learning techniques on the selected 5 types of DDoS to choose the best classifier to detect DDoS attack in IoT. For each specific context, there is an optimal classifier according to the error rate criterion, but none is optimal in all cases. Tables make it possible to quickly assess the results and facilitate the choice of a good classifier in a given context.

2- Presentation of the CICDDOS-2019 database

CICDDoS2019 contains benign and the most up-to-date common DDoS attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter-V3 with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files).

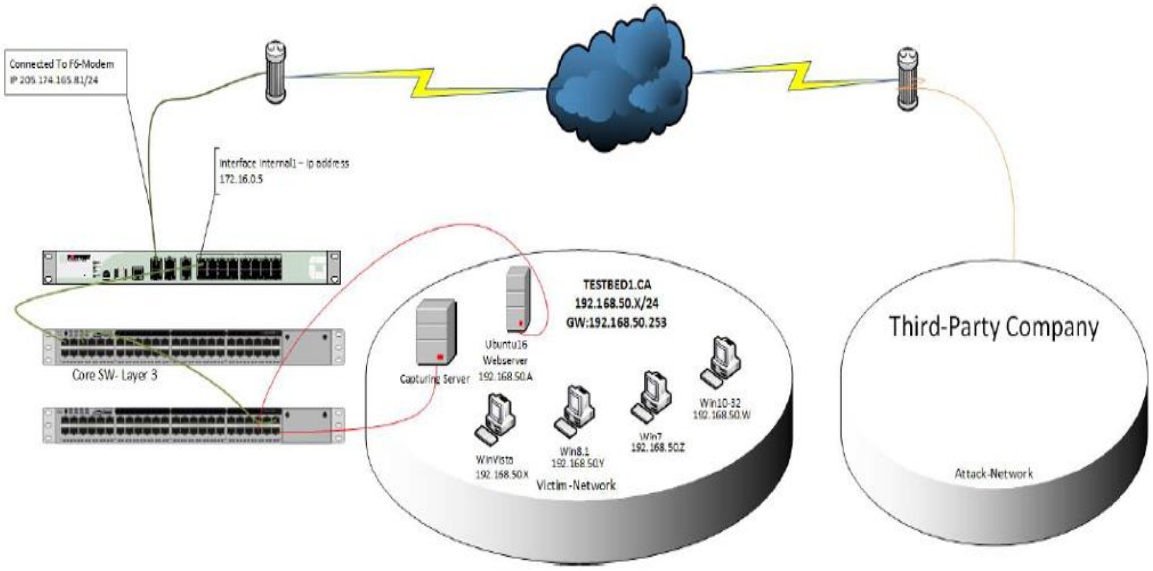


Figure 4.1:CICDDoS219 dataset .

Generating realistic background traffic was our top priority in building this dataset. We have used our proposed B-Profile system (Sharafaldin, et al. 2016) to profile the abstract behaviour of human interactions and generates naturalistic benign background traffic in the proposed testbed (Figure 2). For this dataset, we built the abstract behaviour of 25 users based on the HTTP, HTTPS, FTP, SSH and email protocols[43].

3- Environment WEKA :

3-1- Definition :

WEKA - an open source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems. What WEKA offers is summarized in the following diagram

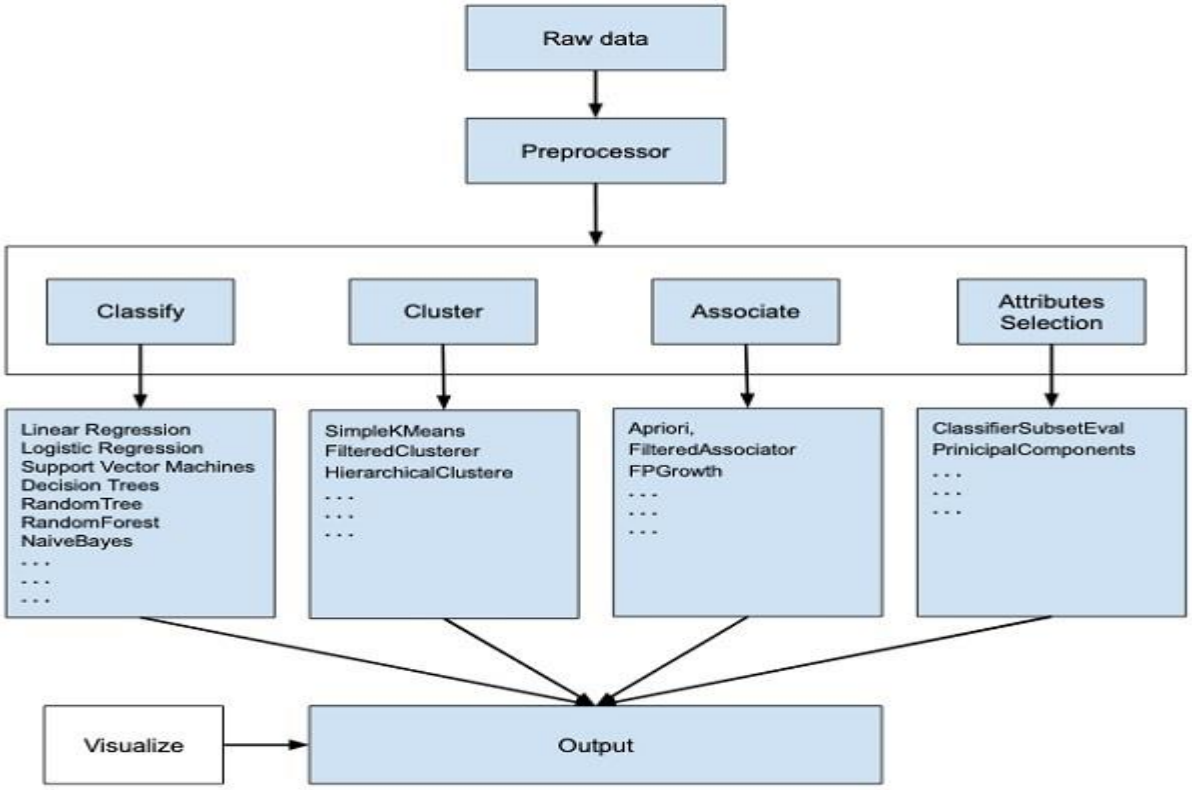


Figure 4.2:weka diagram.

If you observe the beginning of the flow of the image, you will understand that there are many stages in dealing with Big Data to make it suitable for machine learning

First, you will start with the raw data collected from the field. This data may contain several null values and irrelevant fields. You use the data preprocessing tools provided in WEKA to cleanse the data.

Then, you would save the preprocessed data in your local storage for applying ML algorithms.

Next, depending on the kind of ML model that you are trying to develop you would select one of the options such as Classify, Cluster, or Associate. The Attributes Selection allows the automatic selection of features to create a reduced dataset.

Note that under each category, WEKA provides the implementation of several algorithms. You would select an algorithm of your choice, set the desired parameters and run it on the dataset.

Then, WEKA would give you the statistical output of the model processing. It provides you a visualization tool to inspect the data. The various models can be applied on the same dataset. You can then compare the outputs of different models and select the best that meets your purpose.

Thus, the use of WEKA results in a quicker development of machine learning models on the whole. Now that we have seen what WEKA is and what it does, in the next chapter let us learn how to install WEKA on your local computer.[42]

3-2- Graphical User Interface Of WEKA :

- **Explorer :** An environment for exploring data with WEKA (the rest of this documentation deals with this application in more detail).
- **Experimenter:** An environment for performing experiments and conducting statistical tests between learning schemes.
- **KnowledgeFlow:** This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.
- **SimpleCLI:** Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

Figure 4.3:Weka GUI.

we will describe only the first component "Explorer".

- **Explorer:** The WEKA Explorer windows show different tabs starting with preprocess. Initially, the preprocess tab is active, as first the data set is preprocessed before applying algorithms to it and explored the dataset.The tabs are as follows:
  - **Preprocess:** Choose and modify the loaded data.
  - **Classify:** Apply training and testing algorithms to the data that will classify and regress the data.
  - **Cluster:** Form clusters from the data.
  - **Associate:** Mine out association rule for the data.
  - **Select attributes:** Attribute selection measures are applied.
  - **Visualize:** 2D representation of data is seen.
  - **Status Bar:** The bottommost section of the window shows the status bar. This section shows what is happening currently in the form of a message, such as a file is being loaded. Right-click on this, ***Memory information*** can be seen, and also ***Run garbage collector*** to free up space can be run.
  - **Log Button:** It stores a log of all actions in Weka with the timestamp. Logs are shown in a separate window when the Log button is clicked.

- **WEKA Bird Icon:** Present on the bottom right corner shows WEKA bird with represents the number of processes running concurrently (by x.). When the process is running the bird will move around.

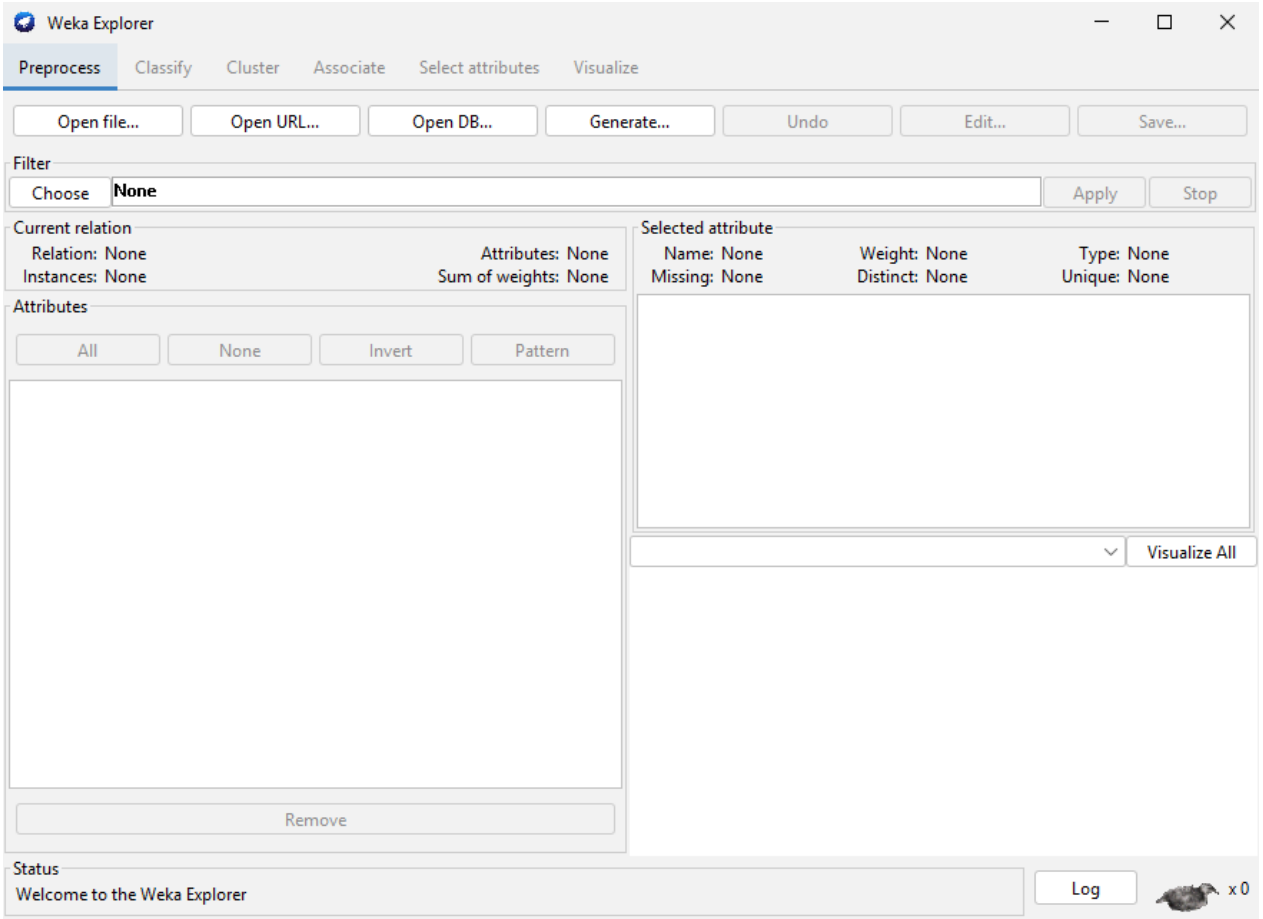


Figure 4.4:weka Explorer Interface.

4- Weka – Classifiers

4-1- **DNS flood attack:** We used 200,000 packets of DNS attack to apply the seven selected machine learning techniques on this attack.

4-1-1- RF:

- **Test mode: split 60.0% train, remainder test:**

Time taken to build model: 250.03 seconds

Time taken to test model on test split: 4.57 seconds

Correctly Classified Instances	79985	**99.9813 %**
Incorrectly Classified Instances	15	**0.0187 %**
Kappa statistic	0.9893	
Mean absolute error	0.0004	
Root mean squared error	0.012	
Relative absolute error		2.5757 %
Root relative squared error		12.7602 %
Total Number of Instances	80000	

Table4.1: split 60.0% train of DNS attack using Random forest

4-1-2- NB:

- **Test mode: split 60.0% train, remainder test:**

Time taken to build model: 5.36 seconds

Time taken to test model on test split: 8.17 seconds

Correctly Classified Instances	79607	**99.5088 %**
Incorrectly Classified Instances	393	**0.4913 %**
Kappa statistic	0.6839	
Mean absolute error	0.0049	
Root mean squared error	0.0701	
Relative absolute error		28.2129 %
Root relative squared error		74.627 %
Total Number of Instances	80000	

Table4.2: split 60.0% train of DNS attack using Naïve bayes.

4-1-3- KNN:

- **Test mode: split 60.0% train, remainder test:**

Time taken to build model: 40.85 seconds

Time taken to test model on test split: 6.15 seconds

Correctly Classified Instances	79733	**99.6663 %**
Incorrectly Classified Instances	267	**0.3337 %**
Kappa statistic	0.8202	
Mean absolute error	0.0033	

Root mean squared error	0.0578	
Relative absolute error		19.1675 %
Root relative squared error		61.5114 %
Total Number of Instances	80000	

Table4.3: split 60.0% train of DNS attack using KNN

4-1-4- REPTree :

- **Test mode: split 60.0% train, remainder test:**

Time taken to build model: 15.37 seconds

Time taken to test model on test split: 0.69 seconds

Correctly Classified Instances	79976	**99.97 %**
Incorrectly Classified Instances	24	**0.03 %**
Kappa statistic	0.9829	
Mean absolute error	0.0005	
Root mean squared error	0.0167	
Relative absolute error		2.6429 %
Root relative squared error		17.8064 %
Total Number of Instances	80000	

Table4.4: split 60.0% train of DNS attack using REPTree

4-1-5- SVM:

- **Test mode: split 60.0% train, remainder test:**

Time taken to build model: 20.18 seconds

Time taken to test model on test split: 2.77 seconds

Correctly Classified Instances	79733	**99.6663 %**
Incorrectly Classified Instances	267	**0.3337 %**
Kappa statistic	0.8202	
Mean absolute error	0.0033	
Root mean squared error	0.0578	
Relative absolute error		19.1675 %
Root relative squared error		61.5114 %
Total Number of Instances	80000	

Table4.5: split 60.0% train of DNS attack using SVM

4-1-6- MLP:

- **Test mode: split 60.0% train, remainder test:**

Time taken to build model: 5086.04 seconds

Time taken to test model on test split: 6.85 seconds

Correctly Classified Instances	79821	**99.7763 %**
Incorrectly Classified Instances	179	**0.2238 %**
Kappa statistic	0.8841	
Mean absolute error	0.0025	
Root mean squared error	0.037	
Relative absolute error		14.3592 %
Root relative squared error		39.4299 %
Total Number of Instances	80000	

Table4.6: split 60.0% train of DNS attack using MLP

4-1-7- J48:

- **Test mode: split 60.0% train, remainder test:**

Time taken to build model: 45.35 seconds

Time taken to test model on test split: 0.33 seconds

Correctly Classified Instances	79971	**99.9638 %**
Incorrectly Classified Instances	29	**0.0362 %**
Kappa statistic	0.9795	
Mean absolute error	0.0005	
Root mean squared error	0.0188	
Relative absolute error		2.7566 %
Root relative squared error		20.0393 %
Total Number of Instances	80000	

Table4.7: split 60.0% train of DNS attack using J48

4-2- NetBIOS: We used 200,000 packets of NetBIOS attack to apply the seven selected machine learning techniques on this attack .

4-2-1- RF:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 170.32 seconds

Time taken to test model on test split: 8.71 seconds

Correctly Classified Instances	79998	**99.9975 %**
Incorrectly Classified Instances	2	**0.0025 %**
Kappa statistic	0.9957	
Mean absolute error	0.0002	
Root mean squared error	0.0073	
Relative absolute error		3.5529 %
Root relative squared error		13.5055 %
Total Number of Instances	80000	

Table4.8: split 60.0% train of NetBIOS attack using RF

4-2-2- NB:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 4.4 seconds

Time taken to test model on test split: 9.7 seconds

Correctly Classified Instances	79881	**99.8512 %**
Incorrectly Classified Instances	119	**0.1487 %**
Kappa statistic	0.7801	
Mean absolute error	0.0015	
Root mean squared error	0.0386	
Relative absolute error		25.2867 %
Root relative squared error		71.7232 %
Total Number of Instances	80000	

Table4.9: split 60.0% train of NetBIOS attack using NB

4-2-3- KNN:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 89.08 seconds

Time taken to test model on test split: 0.88 seconds

Correctly Classified Instances	79942	**99.9275 %**
Incorrectly Classified Instances	58	**0.0725 %**
Kappa statistic	0.8813	
Mean absolute error	0.0007	
Root mean squared error	0.0269	
Relative absolute error		12.3246 %
Root relative squared error		50.0726 %
Total Number of Instances	80000	

Table4.9: split 60.0% train of NetBIOS attack using KNN

4-2-4- REPTree :

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 12.53 seconds

Time taken to test model on test split: 1.74 seconds

Correctly Classified Instances	79987	**99.9838 %**
Incorrectly Classified Instances	13	**0.0163 %**
Kappa statistic	0.9717	
Mean absolute error	0.0002	
Root mean squared error	0.116	
Relative absolute error		3.6665 %
Root relative squared error		21.6538 %
Total Number of Instances	80000	

Table4.9: split 60.0% train of NetBIOS attack using REPTree

4-2-5- SVM:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 77.56 seconds

Time taken to test model on test split: 2.9 seconds

Correctly Classified Instances	79942	**99.9275 %**
Incorrectly Classified Instances	58	**0.0725 %**
Kappa statistic	0.8813	
Mean absolute error	0.0007	

Root mean squared error	0.0269	
Relative absolute error		12.3246 %
Root relative squared error		50.0726 %
Total Number of Instances	80000	

Table4.10: split 60.0% train of NetBIOS attack using SVM

4-2-6- MLP:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 0.21 seconds

Time taken to test model on test split: 32802.48 seconds

Correctly Classified Instances	79989	**99.9862 %**
Incorrectly Classified Instances	11	**0.0138 %**
Kappa statistic	0.9763	
Mean absolute error	0.0001	
Root mean squared error	0.0117	
Relative absolute error		2.3483 %
Root relative squared error		21.8061 %
Total Number of Instances	80000	

Table4.11: split 60.0% train of NetBIOS attack using MLP

4-2-7- J48:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 31.37 seconds

Time taken to test model on test split: 4.24 seconds

Correctly Classified Instances	79989	**99.9862 %**
Incorrectly Classified Instances	11	0.0138 %
Kappa statistic	0.9763	
Mean absolute error	0.0002	
Root mean squared error	0.0115	
Relative absolute error		2.9977 %
Root relative squared error		21.4271 %

Total Number of Instances	80000	

Table4.12: split 60.0% train of NetBIOS attack using J48

4-3- NTP: We used 200,000 packets of NTP attack to apply the seven selected machine learning techniques on this attack .

4-3-1- RF:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 6.09 seconds

Time taken to test model on test split: 8.06 seconds

Correctly Classified Instances	79976	**99.97 %**
Incorrectly Classified Instances	24	**0.03 %**
Kappa statistic	0.9975	
Mean absolute error	0.0009	
Root mean squared error	0.0175	
Relative absolute error		0.7937 %
Root relative squared error		7.1273 %
Total Number of Instances	80000	

Table4.13: split 60.0% train of NTP attack using RF

4-3-2- NB:

- **Test mode: split 60.0% train, remainder test**

Correctly Classified Instances	76014	**95.0175 %**
Incorrectly Classified Instances	3986	**4.9825 %**
Kappa statistic	0.4336	
Mean absolute error	0.0498	
Root mean squared error	0.2232	
Relative absolute error		41.8966 %
Root relative squared error		91.1492 %
Total Number of Instances	80000	

Table4.14: split 60.0% train of NTP attack using NB

4-3-3- KNN:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 0.5 seconds

Time taken to test model on test split: 3688.9 seconds

Correctly Classified Instances	79890	**99.8625 %**
Incorrectly Classified Instances	110	**0.1375 %**
Kappa statistic	0.9885	
Mean absolute error	0.0014	
Root mean squared error	0.0371	
Relative absolute error		1.1712 %
Root relative squared error		15.1556 %
Total Number of Instances	80000	

Table4.15: split 60.0% train of NTP attack using KNN

4-3-4- REPTree :

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 13.78 seconds

Time taken to test model on test split: 2.19 seconds

Correctly Classified Instances	79921	**99.9013 %**
Incorrectly Classified Instances	79	**0.0988 %**
Kappa statistic	0.9918	
Mean absolute error	0.0015	
Root mean squared error	0.031	
Relative absolute error		1.2603 %
Root relative squared error		12.6642 %
Total Number of Instances	80000	

Table4.16: split 60.0% train of NTP attack using REPTree

4-3-5- SVM:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 678.7 seconds

Time taken to test model on test split: 0.61 seconds

Correctly Classified Instances	78297	**97.8713 %**
Incorrectly Classified Instances	1703	**2.1288 %**
Kappa statistic	0.8346	
Mean absolute error	0.0213	
Root mean squared error	0.1459	
Relative absolute error		17.9006 %
Root relative squared error		59.5898 %
Total Number of Instances	80000	

Table4.17: split 60.0% train of NTP attack using SVM

4-3-6- MLP :

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 10520.12 seconds

Time taken to test model on test split: 6.48 seconds

Correctly Classified Instances	79195	**98.9938 %**
Incorrectly Classified Instances	805	**1.0063 %**
Kappa statistic	0.9183	
Mean absolute error	0.0126	
Root mean squared error	0.0799	
Relative absolute error		10.6139 %
Root relative squared error		32.6522 %
Total Number of Instances	80000	

Table4.18: split 60.0% train of NTP attack using MLP

4-3-7- J48:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 58.45 seconds

Time taken to test model on test split: 1.44 seconds

Correctly Classified Instances	79969	**99.9613 %**
Incorrectly Classified Instances	31	**0.0387 %**

Kappa statistic	0.9968	
Mean absolute error	0.0006	
Root mean squared error	0.0196	
Relative absolute error		0.4756 %
Root relative squared error		8.0106 %
Total Number of Instances	80000	

Table4.19: split 60.0% train of NTP attack using J48

4-4- SYN: We used 200,000 packets of SYN attack to apply the seven selected machine learning techniques on this attack.

4-4-1- RF:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 13.6 seconds

Time taken to test model on test split: 3.19 seconds

Correctly Classified Instances	79992	**99.99 %**
Incorrectly Classified Instances	8	**0.01 %**
Kappa statistic	0.6363	
Mean absolute error	0.0001	
Root mean squared error	0.01	
Relative absolute error		34.7873 %
Root relative squared error		73.035 %
Total Number of Instances	80000	

Table4.20: split 60.0% train of SYN attack using RF

4-4-2- NB:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 6.12 seconds

Time taken to test model on test split: 14.42 seconds

Correctly Classified Instances	77522	**96.9025 %**
Incorrectly Classified Instances	2478	**3.0975 %**
Kappa statistic	0.0084	
Mean absolute error	0.0309	

Root mean squared error	0.1741	
Relative absolute error		10766.5379 %
Root relative squared error		1271.715 %
Total Number of Instances	80000	

Table4.21: split 60.0% train of SYN attack using NB

4-4-3- KNN:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 690.16 seconds

Time taken to test model on test split: 6.88 seconds

Correctly Classified Instances	78297	**97.8713 %**
Incorrectly Classified Instances	1703	**2.1288 %**
Kappa statistic	0.8346	
Mean absolute error	0.0213	
Root mean squared error	0.1459	
Relative absolute error		17.9006 %
Root relative squared error		59.5898 %
Total Number of Instances	80000	

Table4.22: split 60.0% train of SYN attack using KNN

4-4-4- RepTree :

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 10.45 seconds

Time taken to test model on test split: 4.7 seconds

Correctly Classified Instances	79990	**99.9875 %**
Incorrectly Classified Instances	10	**0.0125 %**
Kappa statistic	0.6153	
Mean absolute error	0.0001	
Root mean squared error	0.0111	
Relative absolute error		48.9916 %
Root relative squared error		80.9446 %
Total Number of Instances	80000	

Table4.23: split 60.0% train of SYN attack using REPTree

4-4-5- SVM:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 8.79 seconds

Time taken to test model on test split: 5.6 seconds

Correctly Classified Instances	79990	**99.9875 %**
Incorrectly Classified Instances	10	**0.0125 %**
Kappa statistic	0.6666	
Mean absolute error	0.0001	
Root mean squared error	0.0112	
Relative absolute error		43.4842 %
Root relative squared error		81.6556 %
Total Number of Instances	80000	

Table4.24: split 60.0% train of SYN attack using SVM

4-4-6- MLP:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 4521.73 seconds

Time taken to test model on test split: 6.5 seconds

Correctly Classified Instances	79991	**99.9887 %**
Incorrectly Classified Instances	9	**0.0112 %**
Kappa statistic	0.6086	
Mean absolute error	0.0001	
Root mean squared error	0.0101	
Relative absolute error		45.9792 %
Root relative squared error		73.5572 %
Total Number of Instances	80000	

Table4.25: split 60.0% train of SYN attack using MLP

4-4-7- J48:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 39.33 seconds

Time taken to test model on test split: 0.38 seconds

Correctly Classified Instances	79989	**99.9862 %**
Incorrectly Classified Instances	11	**0.0138 %**
Kappa statistic	0.5217	
Mean absolute error	0.0002	
Root mean squared error	0.0116	
Relative absolute error		56.2864 %
Root relative squared error		84.5401 %
Total Number of Instances	80000	

Table4.26: split 60.0% train of SYN attack using J48

4-5- UDP: We used 200,000 packets of UDP attack to apply the seven selected machine learning techniques on this attack .

4-5-1- RF:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 244.9 seconds

Time taken to test model on test split: 6.37 seconds

Correctly Classified Instances	79992	**99.99 %**
Incorrectly Classified Instances	8	**0.01 %**
Kappa statistic	0.9885	
Mean absolute error	0.0003	
Root mean squared error	0.0106	
Relative absolute error		3.6382 %
Root relative squared error		16.0203 %
Total Number of Instances	80000	

Table4.27: split 60.0% train of UDP attack using RF

4-5-2- NB:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 5.17 seconds

Time taken to test model on test split: 14.07 seconds

Correctly Classified Instances	79653	**99.5662 %**
Incorrectly Classified Instances	347	**0.4338 %**
Kappa statistic	0.593	
Mean absolute error	0.0043	
Root mean squared error	0.0657	
Relative absolute error		47.0438 %
Root relative squared error		99.2062 %
Total Number of Instances	80000	

Table4.28: split 60.0% train of UDP attack using NB

4-5-3- KNN:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 0.55 seconds

Time taken to test model on test split: 5009.76 seconds

Correctly Classified Instances	79986	**99.9825 %**
Incorrectly Classified Instances	14	**0.0175 %**
Kappa statistic	0.98	
Mean absolute error	0.0002	
Root mean squared error	0.0132	
Relative absolute error		1.9507 %
Root relative squared error		19.9865 %
Total Number of Instances	80000	

Table4.29: split 60.0% train of UDP attack using KNN

4-5-4- REPTree:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 13.25 seconds

Time taken to test model on test split: 1.16 seconds

Correctly Classified Instances	79977	**99.9712 %**
Incorrectly Classified Instances	23	**0.0288 %**
Kappa statistic	0.9669	
Mean absolute error	0.0004	
Root mean squared error	0.0167	
Relative absolute error		4.8899 %
Root relative squared error		25.1862 %
Total Number of Instances	80000	

Table4.30: split 60.0% train of UDP attack using REPTree

4-5-5- SVM:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 27.8 seconds

Time taken to test model on test split: 1.31 seconds

Correctly Classified Instances	79841	**99.8012 %**
Incorrectly Classified Instances	159	**0.1988 %**
Kappa statistic	0.7655	
Mean absolute error	0.002	
Root mean squared error	0.0446	
Relative absolute error		21.6251 %
Root relative squared error		67.3559 %
Total Number of Instances	80000	

Table4.31: split 60.0% train of UDP attack using SVM

4-5-6- MLP:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 5530.82 seconds

Time taken to test model on test split: 3.89 seconds

Correctly Classified Instances	79919	**99.8988 %**
Incorrectly Classified Instances	81	**0.1013 %**
Kappa statistic	0.8711	
Mean absolute error	0.0013	

Root mean squared error	0.028	
Relative absolute error		13.8464 %
Root relative squared error		42.2789 %
Total Number of Instances	80000	

Table4.32: split 60.0% train of UDP attack using MLP

4-5-7- J48:

- **Test mode: split 60.0% train, remainder test**

Time taken to build model: 58.78 seconds

Time taken to test model on test split: 2.74 seconds

Correctly Classified Instances	79982	**99.9775 %**
Incorrectly Classified Instances	18	**0.0225 %**
Kappa statistic	0.9741	
Mean absolute error	0.0003	
Root mean squared error	0.0149	
Relative absolute error		3.2244 %
Root relative squared error		22.4604 %
Total Number of Instances	80000	

Table4.33: split 60.0% train of UDP attack using J48

5- Results:

This study demonstrates the detecting execution of the seven supervised ML classifiers, which are K_Nearest_Neighbors (K-NN), support vector machine (SVM), naïve bayes (NB), decision tree (RepTree), random forest (RF) , decision tree(J48), Multilayer Perceptron (MLP).

The experiments in this study use a hardware specification of AMD A4-9120 RADEON R3,4COMPUTE CORES 2C+2G 2.20GHz processor, 4 GB RAM with the operating system Windows 10, 64 bit. In this research, the ML technique in WEKA tool is being tested for forecasting DDoS attacks. This study uses the WEKA version 3.9.6 tool for data pre-processing, categorization, regression, assembling, visualization and association rules. The Java code has been used for writing WEKA, and it is an open source tool established in New Zealand at the University of Waikato. All the algorithms that have been used are supported in WEKA. WEKA has a graphical user interface and a command-based interface which make it attractive to be used in this research.

It requires file formats such as CSV and ARFF. In machine learning, the dataset is required to train selected algorithms to gain knowledge.

In case of DNS attack the **Random Forest(RF)** was the best classifier ,when split 60.0% train, remainder test: Time taken to build model: 250.03 seconds, Time taken to test model on test split: 4.57 seconds, Correctly Classified Instances : **99.9813 %** and Incorrectly Classified Instances : **0.0187 % (see Table 4.1)**

In case of NetBIOS attack the **Random Forest(RF)** was the best classifier ,when split 60.0% train, remainder test: Time taken to build model: 170.32 seconds, Time taken to test model on test split: 8.71 seconds, **Correctly Classified Instances :99.9975 %** and **Incorrectly Classified Instances: 0.0025 % (see Table 4.8)**

In case of NTP attack the **Random Forest(RF)** was the best classifier ,when split 60.0% train, remainder test: Time taken to build model: 311.25 seconds, Time taken to test model on test split: 6.52 seconds, **Correctly Classified Instances :99.97 %** and **Incorrectly Classified Instances: 0.03 % (see Table4.13)**

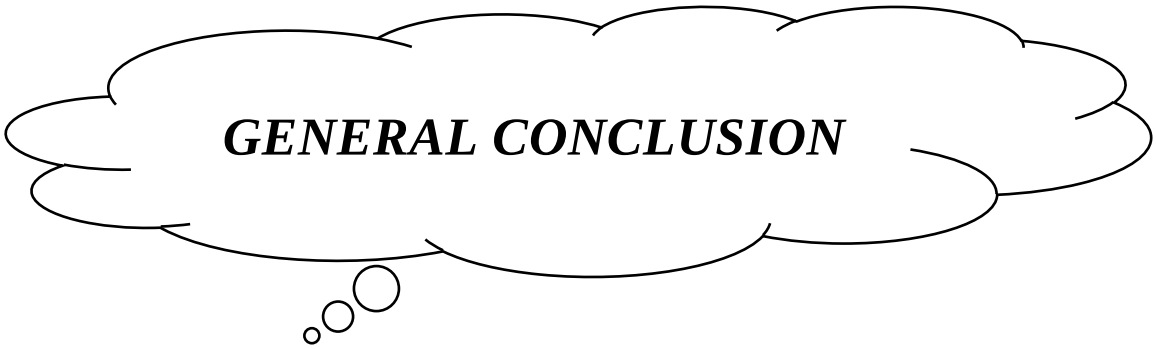
In case of SYN attack the **Random Forest(RF)** was the best classifier when split 60.0% train, remainder test the best classification when use Time taken to build model: 13.6 seconds, Time taken to test model on test split: 3.19 seconds, **Correctly Classified Instances :99.99 %** and **Incorrectly Classified Instances: 0.1 % (see Table4.20)**

In case of UDP attack the **Random Forest(RF)** was the best classifier when split 60.0% train, remainder test :Time taken to build model: 244.9 seconds, Time taken to test model on test split: 6.37 seconds, **Correctly Classified Instances :99.99 %** and **Incorrectly Classified Instances: 0.01 % (see Table4.27)**

	Random Forest(RF)	
	Incorrectly Classified Instances	Correctly Classified Instances
DNS	**0.0187 %**	**99.9813%**
NetBIOS	**0.0025 %**	**99.9975 %**
NTP	**0.03 %**	**99.97 %**
SYN	**0.1 %.**	**99.99 %**
UDP	**0.01 %.**	**99.99 %**

6- Conclusion:

In this chapter, we have applied various classification algorithms for intrusion detection in a CICDDoS2019 dataset through weka software platform, based on accuracy rate, random forest classification algorithm obtain the highest accuracy rate when comparing with other classification algorithms, the correctly classified instance of this algorithm is 99.99% from this accuracy rate, the results shown that the random forest classification algorithm is the best CICDDoS2019 dataset classification algorithm.


GENERAL CONCLUSION

General Conclusion:

IoT attacks represent a real risk which threatens networks and devices, one of the most common network attacks at present which is DDoS attack ,Despite the large number of traditional detection solutions that exist currently, DDoS attacks continue to grow in frequency, volume, and severity. which led us to try in this work, to brings an analysis of the problem so we make a comparative study of different types of classifiers, for shown the highest perform of machine learning algorithm selected the best seven algorithms K_Nearest_Neighbors (K-NN), support vector machine (SVM), Naïve Bayes (NB), decision tree (RepTree), random forest (RF) , decision tree(J48) and Multilayer Perceptron (MLP) using CICDDos2019 dataset ,working on WEKA deep learning software to better detect DDoS attack .

In this research, DDoS attacks are serious challenges to many areas of our life. This leads us to try to find a comprehensive intrusion detection system to decrease the number of attacks facing many sectors. This study has used CICDDoS2019, which is the newest and complete dataset accessible by Canadian Institute for Cybersecurity. The results shown that the random forest classification algorithm is the best CICDDoS2019 dataset classification algorithm.

Finally, since scientific research has no limits and there are possible improvements to perfect this model such as the use of combinations of classifiers to choose the others network parameters and also the realization of a multi-class classification which also gives the type of attack detected.

REFERENCES

References

[1] https://www.helpnetsecurity.com/2022/03/28/ddos-attacks-2021/#:~:text=During%20the%20second%20half%20of,million%2C%20a%20NETSCOUT%20report%20reveals.

[2] https://www.hornetsecurity.com/en/knowledge-base/it-security/?_adin=02021864894

[3] https://en.wikipedia.org/wiki/Computer_security#Robert_Morris_and_the_first_computer_worm

[4] https://blog.rsisecurity.com/what-are-the-different-types-of-it-security/

[5] https://www.kaspersky.com/resource-center/definitions/what-is-internet-security

[6] https://www.geeksforgeeks.org/types-of-security-mechanism/

[7] https://www.liquidweb.com/blog/most-common-web-security-problems/#:~:text=What%20is%20a%20Security%20Issue,your%20business%20processes%20and%20people.

[8] https://www.forcepoint.com/cyber-edu/network-security#:~:text=Network%20security%20is%20a%20broad,both%20software%20and%20hardware%20technologies.

[9] https://www.radware.com/resources/network_security_history.aspx#:~:text=Generally%2C%20the%20first%20firewall%20is,of%20a%20building%20or%20complex

[10] https://phoenixnap.com/blog/what-is-network-security

[11] https://www.forcepoint.com/fr/cyber-edu/network-security

[12] https://www.checkpoint.com/cyber-hub/network-security/what-is-network-security/#TypesofNetworkSecurity

[13] https://ipwithease.com/firewall-vs-ips-vs-ids/#:~:text=The%20main%20difference%20being%20that,a%20set%20of%20rules%20configured.

[14] H.Alyasiri,“ Developing Efficient and Effective Intrusion Detection System using Evolutionary Computation”,November 2018.[Online]. Available: https://etheses.whiterose.ac.uk/23699/1/Hasanen_Thesis_2018.pdf

[15] I.Porres Ruiz,M.del Mar Fernández de Ramón, “An Evaluation of current IDS”, March 28th 2008.[Online]. Available: https://www.diva-portal.org/smash/get/diva2:18049/FULLTEXT01.pdf

[16] https://logicalread.com/intrusion-detection-system/#.YjG-feiZPIV

[17] H.Ahmad Madni Uppal1 , M.Javed2 and M.J. Arshad ," An Overview of Intrusion Detection System (IDS) along with its Commonly Used Techniques and Classifications", February 2014.[Online]. Available: http://www.ijcst.org/Volume5/Issue2/p4_5_2.pdf

[18] https://www.researchgate.net/publication/338410162_PhD_Thesis_Inter-Organisational_Intrusion_Detection_System_Communication_to_implement_Network_Defence

[19] L.Leichtnam,"Detecting and visualizing anomalies in heterogeneous network events : Modeling events as graph structures and detecting communities and novelties with machine learning ",6 Oct 2021.[Online]. Available: https://tel.archives-ouvertes.fr/tel-03368501/document

[20] F.Hussain,S.Ghazanfar Abbas,M.Husnain,"IoT DoS and DDoS Attack Detection using ResNet",December 2020 .[Online]. Available: https://www.researchgate.net/publication/346614725_IoT_DoS_and_DDoS_Attack_Detection_using_ResNet

[21] A.Hamid Mohammed,A.Abdu Ibrahim,M.Hassan Aysa," IoT Ddos Attack Detection Using Machine Learning",November2020.[Online].Available:https://www.researchgate.net/publication/346082170_IoT_DDOS_ATTACK_DETECTION_USING_MACHINE_LEARNING

[22] K.Sanyal,D. Kumbhakar,S. Karforma," COMPARATIVE STUDY OF MACHINE LEARNING ALGORITHMS FOR EVALUATING STUDENT ACADEMIC PERFORMANCE",2020.[Online].Available:https://ejmcm.com/article_4039_14d3f4a17cd25a2b81ea34584ecfa161.pdf

[23] H.Moritz," A comparative study of machine learning algorithms for Document Classification" , 2020.[Online].Available:https://www.diva-portal.org/smash/get/diva2:1448204/FULLTEXT01.pdf

[24] https://insights2techinfo.com/how-iot-is-making-ddos-attacks-more-dangerous/

[25] https://www.imperva.com/learn/ddos/ddos-attacks/#:~:text=Includes%20SYN%20floods%2C%20fragmented%20packet,packets%20per%20second%20(Pps).

[26] https://www.transip.co.uk/knowledgebase/entry/1206-protecting-against-netbios-abuse/

[27] https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/

[28] https://www.educba.com/decision-tree-advantages-and-disadvantages/

[29] Rami J. Alzahrani, A.Alzahrani," Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic ",2021.[Online].Available:https://www.mdpi.com/2079-9292/10/23/2919/html

[30] I.Sofi1, A.Mahajan, V.Mansotra,“Machine Learning Techniques used for the Detection and Analysis of Modern Types of DDoS Attacks”,June 2017.[Online].Available:https://www.academia.edu/download/53811814/IRJET-V4I6200.pdf

[31] R.Astuti Nugrahaeni,K.Mutijarsa,” Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification”,09 March 2017.[Online].Available:https://ieeexplore.ieee.org/abstract/document/7873831

[32] Ö.Tonkal ,H.Polat,E.Başaran,Z.Cömert 1 andR.Kocaoğlu ,” Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking”, 21 May 2021.[Online].Available:https://www.mdpi.com/2079-9292/10/11/1227

[33] K.Muthamil Sudar; M. Beulah; P. Deepalakshmi; P. Nagaraj; P. Chinnasamy,” Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques”, 21 April 2021.[Online].Available:https://ieeexplore.ieee.org/abstract/document/9402517

[34] V.de MirandaRiosab,P.R.M.Ináciob,D.Magonic,Mário M.Freire,“Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms “,26 February 2021.[Online].Available:https://www.sciencedirect.com/science/article/abs/pii/S1389128620313633

[35] J.Arturo Pérez-Díaz,I.Amezcua Valdovinos,K.Raymond Choo,Dakai Zhu ,” A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning”,25 August 2020.[Online].Available:https://ieeexplore.ieee.org/abstract/document/9177002

[36] O.Rahman,M.Ali Gauhar Quraishi; C.Lung,” DDoS Attacks Detection and Mitigation in SDN Using Machine Learning” ,2019.[Online].Available:https://ieeexplore.ieee.org/abstract/document/8817237

[37] http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-knn.html

[38] https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107

[39] https://courses.media.mit.edu/2006fall/mas622j/Projects/manu-rita-MAS_Proj/MLP.pdf

[40] https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141

[41] https://www.wikihow.tech/Convert-CSV-to-ARFF

[42] https://www.tutorialspoint.com/weka/what_is_weka.htm

[43] https://www.unb.ca/cic/datasets/ddos-2019.html