People's Democratic Republic of Algeria

Ministry for Higher Education and Scientific Research

University of 20 août 1955 Skikda

Faculty of Sciences

Department of Computer Science

MASTER THESIS

Thesis submitted in partial fulfillment of the requirements to obtain Master's degree of computer science in Networks and Distributed Systems

Topic:

Prevention and Protection Against DDoS Attacks Using Machine Learning (Classification Algorithms)

Supervised by:

Pr. MAZOUZI SMAINE

Presented by:

NAFIR Khaoula

TALAA Imane Nour Allah

The committee:

Professor	Status
Rami Soumia	President
Bendjedou Zineb	Examinator
Pr Mazouzi Smaine	Supervisor

University Year: 2023/2024

Acknowledgment

First and foremost, we thank Allah, the greatest and most merciful, for giving us the strength to complete our studies. We thank our professor MAZOUZI Smaine, for guiding us through this thesis, and all our families for supporting us.

Dedication

This thesis is dedicated:

to my parents,

to my brother and my two sisters,

and to all my friends.

Khaoula

Dedication

This thesis is dedicated:

To my parents and my family for their constant encouragement and belief in me.

To my friends, for their companionship.

Imane

Abstract

In the current era of extensive digitalization across various aspects of human life, we are now faced with a complex intersection of diverse systems that regulate our daily routines. Regrettably, these systems are under constant threat from rapidly evolving attacks. As a consequence, safeguarding these systems poses a significant challenge for organizations, companies and individuals alike, as they all rely on common internet services. The focus of our thesis has been on the prevention and detection of Distributed Denial of Services (DDoS) attacks through the application of machine learning techniques, specifically utilizing classification algorithms like Random Forests, Decision Trees and AdaBoost. Our evaluation centered on the effectiveness of these methods in recognizing abnormal traffic patterns associated with DDoS attacks. While Random forests, which amalgamate multiple decision trees, exhibit robustness and efficiency, Decision Trees, despite their simplicity and speed, are susceptible to overfitting. Notably, AdaBoost, which enhances model performance by assigning weights to the errors of weak classifiers, merges as the most proficient in identifying DDoS attacks in our tests. The findings indicate that AdaBoost delivers superior accuracy compared to other algorithms.

Keywords: DDoS attacks, machine learning, classification algorithms, Random Forests, Decision Trees, AdaBoost.

Résumé

Avec la numérisation massive de la vie humaine dans tous les domaines ; Nous nous se trouvons aujourd'hui à l'intersection de plusieurs divers systèmes complexes qui régissent nos activités quotidiennes. Malheureusement, ces systèmes sont menacés par des attaques qui varient rapidement. La sécurisation de ces systèmes reste donc un enjeu majeur pour les institutions, les entreprises, voire même les particuliers ; car ils partagent tous, d'une manière ou d'une autre, les mêmes services Internet. Dans le cadre de la préparation de ce mémoire, nos efforts se sont concentrés sur la prévention et la détection des attaques par déni de service distribué (DDoS) en utilisant des techniques d'apprentissage automatique, notamment les algorithmes de classification tels que forêts aléatoires, les arbres de décision et AdaBoost. Ces méthodes ont été évaluées pour leur capacité à identifier des modèles de trafic anormaux associés aux attaques DDoS. Les forêts aléatoires, qui combinent plusieurs arbres de décision, sont robustes et performants. Les arbres de décision, malgré leur simplicité et leur rapidité, sont sujet d'une suradaptation. Quand AdaBoost, qui améliore les performances du modèle en pondérant les erreurs des classifieurs faibles, s'est révélé être le plus efficace pour détecter les attaques DDoS lors de nos tests. Les résultats montrent qu'AdaBoost offre des performances supérieures en termes de précision, surpassant les autres algorithmes.

Mots clés : Attaques DDoS, apprentissage automatique, algorithmes de classification, Forêt Aléatoire, Arbres de Décision, AdaBoost.

ملخص

مع الرقمنة الهائلة للحياة البشرية في جميع المجالات، نجد أنفسنا اليوم في مفترق طرق لعدة أنظمة معقدة تتحكم في كثير من أنشطتنا اليومية. لكن لسوء الحظ، تتعرض هذه الأنظمة بشكل متزايد وسريع لتهديدات أمنية تنمو وتتنوع بشكل كبير ومخيف. وبالتالي، فإن ضمان أمن هذه الأنظمة يظل تحديا كبيرا لكل مكونات البشرية كـ: المؤسسات، والشركات والمجتمعات وحتى الأفراد لأنهم يشتركون جميعاً في نفس خدمات الإنترنت.في إعداد هذه المذكرة ؛ انصب جهدنا حول اكتشاف ومنع الهجمات من نوع الحرمان من الخدمة (DoS) و تحديدا هجمات الحرمان الموزع للخدمة (DDoS) على مستوى الشبكات واسعة النطاق و التي تعد في حد ذاتها أنظمة معقدة و موزعة. و هذا باستخدام تقنيات التعلم الآلي، بما في ذلك خوارزميات التصنيف مثل الغابات العشوائية Random Forets و أشجار القرار (Decision Trees) و . AdaBoost فقد تمحورت دراستنا حول فعالية هذه الاساليب في التعرف على الحركات المرتبطة بهجمات . DDoS فالغابات العشوائية هي التي تجمع بين العديد من أشجار القرار، وهي بدورها قوية وفعالة. وبالرغم من بساطة وسرعة أشجار القرار، الا انها تخضع للتكيف المفرط. ولكن تبين أن AdaBoost هو الذي يعمل على تحسين أداء النموذج من خلال ترجيح أخطاء المصنفات الضعيفة، وهو الأكثر فعالية في إكتشاف هجمات DDoS في اختباراتنا. فقد أظهرت النتائج أن AdaBoost يوفر أداءً فائقًا من حيث الدقة، وهو بدوره متفوقًا على الخوارزميات الأخرى.

الكلمات الرئيسية : الأمن السيبراني , امن الشبكات ,الانظمة المعقدة , هجمات حجب الخدمة الموزعة (DDoS) ,التعلم الآلي , خوارزميات التصنيف ,الغابة العشوائية (Random Forest) ,شجرة القرار(Decision Tree), Adaboost .

Table of Contents

Tables of figures

List of tables

General introduction

In our digitally driven era, with the expansion and simplification of global personal and business connections, the Internet has surely emerged as the world's greatest public data network. Every day, there is an exponential increase in the amount of data traveling via business networks and the Internet. Computer security emerges as the cornerstone upon which our modern society is erected. Over time, the realm of information system security, also referred to as computer security, has escalated into a paramount concern for businesses, organizations, governments, nations, and individual users alike. This surge in concern stems from the persistent assaults experienced by users on their own computing devices and the escalating volume of data traversing business and internet networks each day. Protecting digital assets from an array of perils has transcended mere necessity; it has become an ongoing struggle at the forefront of technological advancement. This battle extends its reach from personal gadgets to global networks, impacting every corner of our interconnected digital world. The ability of any business to uphold a positive reputation among its users and peers is intrinsically linked to the security of its information systems. Even unsuccessful assaults inflict significant harm, eroding confidence and potentially driving consumers toward competing entities. Thus, the imperative for heightened safety measures remains paramount in addressing these evolving challenges and fostering trust within our interconnected digital landscape.

However, despite the implementation of the most advanced security mechanisms and technological advancements, the field of computer security remains fraught with obstacles and dangers. Cyberattacks persist as dynamic and relentless threats to the integrity and security of digital systems, orchestrated by hostile actors ranging from lone hackers to state-sponsored cyber warfare teams. These attacks manifest in myriad forms, all with the common objective of achieving malicious goals by exploiting vulnerabilities within computer systems and networks.

In this master's thesis, we are interested in a particular type of attacks against connected computer systems, namely denial of service, and in particular distributed denial of service (DDoS). We first present it in our literature review and then design machine learning-based models for DDoS detection.

The aim of such a study is to select the machine learning models that are most appropriate to the dataset we have. It is also a question of testing whether the use of machine learning meta-models, such as boosting, can significantly improve the detection results of such attacks.

Our master's thesis is organized as follows:

- In **Chapter 1**, we present the field of computer security, where we focus more on network attacks, including DoS and DDoS attacks.
- **Chapter 2** is devoted to the paradigm we have chosen to develop DDoS detection tools, namely machine learning.
- **Chapter 3** presents our contribution, namely the study of some machine learning models, including a meta-model for ensemble learning. The aim is to show the interest of the latter for an efficient DDoS detection.
- **Chapter 4** is reserved for the implementation of the studied models and the testing of these models on the learning base we have.
- A **general conclusion** summarizes our contribution and traces of potential perspectives to this work.

Chapter 01:

Cybersecurity & Attacks

1.1 Introduction

The significance of cybersecurity in today's digitally connected society cannot be emphasized. Malicious actors who want to take advantage of weaknesses in our networked systems constitute a growing threat as technology develops. Cybersecurity refers to the methods, tools, and procedures used to guard against damage, intrusion, and assaults on computers, networks, data, and applications.

Denial of Service (DoS) attacks, as well as their distributed version, DDoS, are common forms of cyberattacks that attempt to interfere with the operation of online services by flooding them with excessive traffic. These assaults, which pose serious risks to companies, organizations, and people alike, can be launched via botnets, amplification tactics, or other means. It is essential to comprehend these dangers to protect digital assets and preserve the integrity of online systems.

In this chapter, the fundamentals of cybersecurity are thoroughly examined, with a focus on the sneaky threats presented by Distributed Denial of Service (DDoS) and Denial of Service (DoS) assaults.

Section one : Cyber Security

1.2 Computer security:

1.2.1 Definition:

Computer security, also known as cybersecurity or information security, refers to the practice of protecting computer systems, networks, and data from unauthorized access, misuse, modification, or destruction. It encompasses a range of measures, technologies, and practices designed to safeguard digital assets and ensure the confidentiality, integrity, and availability of information.

1.2.2 Objectives:

Information systems are a vital organizational asset that should be safeguarded. Ensuring that an organization's hardware and software resources are only used within the intended scope is the goal of computer security.

Information systems security aims at the following objectives (C.A.I.D):

- **Confidentiality:** Information meant for specific individuals (permissions or rights ideas) can only be accessed by those who are permitted. Unwanted Access needs to be stopped.

- **Availability:** Throughout the designated times of usage, persistent and faultless access to information system resources is required. Ressources and services are routinely and swiftly available.
- **Authenticity:** Users are required to use an access code to verify their identity. It is not appropriate to combine identification with authentication: in the first scenario, the user is merely identified by his public ID, but in the second scenario, he has to provide a password or other secret element. The process of verifying the validity of an identifier is to compare it with a secret. This facilitates the management of resource access rights and preserves the integrity of exchange relationships.
- **Integrity:** the information must be accurate and unaltered by mistake, illegal activity, or malevolent intent. It is obvious that the factors taken into account ought to be precise and full. Typically, checksum or hash computation techniques are used to achieve this goal.

The following elements may also be considered parts of information systems security objectives:

- **Non-repudiation** (imputation): no user may dispute acts they have taken within the parameters of their permission actions, and no third party may claim credit for another user's actions.
- **Traceability**: is the assurance that attempts and actual access to the relevant parts are tracked down and that these traces are kept intact and functional.

1.2.3 General approach:

To safeguard IT systems from a range of risks and vulnerabilities, a thorough set of procedures, best practices, and safeguards must be put into place. according to the following steps:

- **Risk Assessment:** Determine important resources, possible dangers, and weaknesses in the system. Risk analyses, audits, and security assessments may fall under this category.
- **Defining Security Policies:** Create security rules and procedures that outline appropriate methods for handling incidents, gaining access, and using resources.

- **Access control and identity management:** To guarantee that only authorized users may access system resources, and implement robust identification, access rights, and authentication procedures.
- **Malware Protection**: To find and remove malware, install antivirus and anti-malware software.
- **Securing the Network:** To monitor and manage network traffic, configure firewalls, intrusion prevention systems (IPS), and intrusion detection systems (IDS). Utilize network security techniques as well, such as virtual private networks (VPNs) and data encryption.
- **Patch and update management:** Ensure that all software and systems have the most recent security updates installed regularly to address known vulnerabilities.
- **Data security:** When sensitive data is in transit or at rest, encrypt it. Establish data management procedures to control access and safeguard information from tampering, loss, or corruption.
- **Incident monitoring and detection:** To promptly identify and address security breaches, put in place log monitoring tools, security alerts, and incident management systems.
- **Security Awareness:** Inform users of possible hazards, recommended practices for IT security, and what to do in case of an incident.
- **Security Testing:** To assess the efficacy of security measures and find possible vulnerabilities, conduct frequent penetration testing, security audits, and incident simulation exercises.

Section Two: Attacks

1.3 Attacks:

Any computer connected to the Internet is vulnerable to several kinds of assaults. So, a cyberattack is a type of assault that aims to steal, modify, or destroy any important data stored on a computer or computer network. Any person or program that gains illegal access or usage can be the attacker.

Cyberattacks can be carried out either by individuals or by organizations. The goal of a cyberattack is to get access to a person's or a management's information system. Malicious code is used in cyberattacks, which modify computer data, code, or logic. This has disruptive

consequences, compromises data, and opens the door to cybercrimes including identity and information theft. we will cite some of the most famous attacks:

- **Malware :**

  Malware refers to malicious software designed to infiltrate, damage, or disrupt computer systems, networks, or devices, often without the consent or knowledge of the user. This umbrella term encompasses various types of malicious programs, including viruses, worms, Trojans, ransomware, spyware, adware, and rootkits. Malware can be used for a range of nefarious purposes, such as stealing sensitive information, compromising system integrity, facilitating unauthorized access, or conducting fraudulent activities. To protect against malware threats, it is essential to employ robust cybersecurity measures, including antivirus software, firewalls, regular software updates, and user education on safe computing practices.

- **Spoofing Attack:**

  is the act of disguising a communication or identity so that it appears to be associated with a trusted, authorized source. Spoofing attacks can take many forms, from the common email spoofing attacks that are deployed in phishing campaigns to caller ID spoofing attacks that are often used to commit fraud. Attackers may also target more technical elements of an organization's network, such as an IP address, domain name system (DNS) server, or Address Resolution Protocol (ARP) service, as part of a spoofing attack. [1]

- **Phishing:** is a form of social engineering attack aimed at deceiving users into divulging sensitive information, such as passwords or credit card details. These attacks can occur via email, text messages, or even telephone calls. One of the most popular phishing attacks is the Federal Trade Commission released this statement regarding phishing attempts during the 2018 World Cup in Russia. The scammer claimed the victim had won World Cup tickets through a lottery and asked for personal information in order to complete the prize claim process. [2]
- **Ransomware:** Malicious software, or ransomware, is a type of program that encrypts data on a victim's computer or network and prevents it from being accessed. After encrypting the data, the ransomware usually shows an alert requesting payment, frequently in cryptocurrency, for a decryption key that will unlock the contents. Attackers can use a variety of techniques to spread ransomware, such as phishing

emails, hacked websites, or taking advantage of holes in operating systems or software. Certain strains of ransomware can potentially infect many computers by moving laterally across a network.

The WannaCry ransomware attack, occurring in May 2017, is widely regarded as one of the most notorious cyber assaults in recent memory. It swiftly proliferated worldwide, affecting hundreds of thousands of computers across more than 150 countries by exploiting a security flaw in the Windows operating system dubbed Eternal Blue. Upon infiltration, WannaCry encrypted files stored on the compromised systems and demanded payment in Bitcoin in exchange for decryption keys. The attack wrought havoc on businesses, government entities, and essential infrastructure globally, underscoring the profound ramifications of ransomware attacks.

- **SQL Injection Attack:** is a method that enables the execution of SQL queries on database servers without requiring a legitimate session to be established or verifying the user's identity. This technique involves injecting executable code into the data of an SQL query. Rather than treating the injected code as regular data, the SQL server executes it, potentially granting the hacker access to unauthorized data.

  Heartland Payment Systems was the target of one of the most well-known SQL injection attacks in 2008. For thousands of companies, Heartland was a prominent payment processing provider that handled debit and credit card transactions.

We will concentrate on the network attacks DoS\DDoS:

1.3.1 Dos attack A Denial of Service (DoS) attack:

is a type of cyber-attack in which a malicious actor aims to make a computer or other device unavailable to its intended users by interrupting the normal operation of the device. DoS attacks typically work by overwhelming or saturating a targeted machine with requests until normal traffic can no longer be processed, resulting in denial of service for additional users. A DoS attack is characterized by the use of a single computer to launch the attack.

1.3.2 Distributed Denial of Service (DDoS) attack:

is one of the common attacks that is predominant in the cyber world. DDoS attack poses a serious threat to making a website, server, or network unavailable by overloading their capacity with a massive amount of traffic from multiple sources. The goal is to saturate the target server so that it can no longer respond to legitimate requests, making the service inaccessible. They are often carried out using botnets, which are remote-controlled and malware-infected computer networks. Botnets can consist of

thousands of computers, which makes them capable of generating a massive amount of traffic that can easily overwhelm a server. DDoS attacks are often carried out by hackers who seek to damage a company or service for financial, political, or mere malicious reasons. Cryptocurrency exchanges are vulnerable to DDoS attacks, as they are often the preferred targets of hackers seeking to steal funds or disrupt the market.

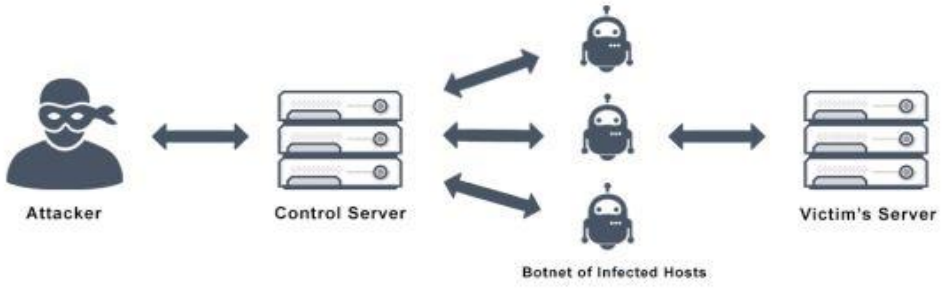

Figure 1: DDoS attack

One of the biggest DDoS assaults to date was launched on the content delivery network (CDN) provider Akamai in June 2021. Numerous well-known websites and online services, including significant financial institutions, airlines, and gambling platforms, were inaccessible due to the assault.

1.3.3 Types of DDoS Attacks:

Although the ultimate objective of a DDoS attack remains consistent in overpowering the system, the methods employed to achieve this goal can vary. There are three primary categories of DDoS attacks:

- Volume-Based or Volumetric Attacks
- Protocol Attacks
- Application-Layer Attack [3]

1.3.3.1 Volume-Based or Volumetric Attacks:

This form of attack is designed to manipulate the entire bandwidth that exists between the victim and the wider internet. An instance of a volume-based attack is DNS amplification. In this situation, the attacker falsifies the target's address and proceeds to send a DNS name lookup request to an accessible DNS server using the falsified address. When the DNS server sends

the DNS record response, it mistakenly directs it to the target, causing the target to receive an amplified version of the attacker's initially insignificant query.

Volume-based DDoS attacks include:

- **UDP floods:** a type of denial-of-service attack where a large number of User Datagram Protocol (UDP) packets are sent to a targeted server with the intention of overwhelming its capacity to process and respond. The firewall safeguarding the targeted server may also become overwhelmed as result of a UDP flood, leading to a denial of service for legitimate traffic.
- **ICMP flood attacks :**
  occur when malicious actors inundate the server with falsified ICMP packets originating from a large number of source IPs. This type of attack can overwhelm server resources, resulting in the inability to handle legitimate requests, potentially causing the server to crash or significantly impacting its performance. These attacks can be directed towards particular servers or launched randomly, ultimately causing a depletion of bandwidth resources.
- **Ping floods:** Attackers flood the server with spoofed ping packets from a huge set of source IPs. It is an evolution of the ICMP flood attacks. The attacker’s objective is to flood the server until it goes offline. The biggest downside from this attack for website owners is that it can be difficult to detect, and mistaken for legitimate traffic.

1.3.3.2 Protocol Attacks: The foundation of the internet lies in protocols, which facilitate the transfer of data from one point to another. DDoS attacks, which target vulnerabilities in Layers 3 and 4 protocol stacks, aim to overwhelm server resources and network hardware during the processing phase. Consequently, these attacks lead to service disruptions. The objective of such attacks is to exploit your network stack by inundating it with an excessive number of packets or overwhelming your network ports with an excessive amount of bandwidth. Protocol-based DDoS attacks include :

- **The SYN Flood attack** involves the exploitation of vulnerabilities in the TCP connection three-way handshake, which is the communication process between the client, host, and server. By sending spoofed SYN packets to the targeted server, attackers aim to exhaust the server's table memory connection, ultimately leading to a complete shutdown of the service.

- **The Ping of Death** refers to the act of sending harmful pings to a server, where the IP protocols are manipulated by attackers. Although this type of attack was prevalent in the 1990s, it is important to note that even in the present day, there are variations of Ping of Death attacks that can be directed towards applications or hardware. The consequence of such an attack is the server experiencing a reboot or a complete crash. This highlights the significance of not underestimating a Denial of Service (DoS) attack, as a single attacker has the potential to bring down an entire data center.

1.3.3.3 Application-Layer Attacks: These attacks also have the objective of depleting or overpowering the target's resources, yet they are challenging to identify as malicious. Commonly known as a Layer 7 DDoS attack, which refers to Layer 7 of the OSI model, an application-layer attack specifically targets the layer responsible for generating web pages in response to HTTP requests. During this type of attack, the attacker compels the victim's server to handle a workload beyond its normal capacity. An HTTP flood, categorized as an application-layer attack, resembles the act of continuously refreshing a web browser on multiple computers simultaneously. Consequently, the excessive influx of HTTP requests overwhelms the server, resulting in a DDoS situation.

1.3.4 Launching of DDoS Attacks:

There are four fundamental steps involved in initiating a DDoS attack. Firstly, the master attacker selects the agents that will carry out the attack. These agents are compromised machines with vulnerabilities that are exploited by the attacker. In the past, attackers manually gained control of these machines, but now, with advanced security attack tools, the identification of these machines has become automated. Secondly, the attacker compromises the agent machines by exploiting security weaknesses and planting attack code while ensuring that the code remains undetected. The compromised nodes, also known as agents or zombies, act as unwitting accomplices between the attacker and the victim. This strategy adds complexity to the DDoS attack, making it harder to trace back to the source due to the involvement of multiple machines and distributed routers. Without a robust defense mechanism, it is challenging for compromised agents to realize they are part of a DDoS attack system. Lastly, the attacker communicates with handlers to coordinate the attack schedule and upgrade agents using various protocols such as ICMP, TCP, or UDP. [4]

1.3.5 Detection of these attacks:

To effectively mitigate assaults, distributed denial of service (DDoS) attacks must be distinguished from regular network traffic using a technique called DDoS detection. A denial-of-service (DDoS) attack aims to prevent authorized users from accessing network services or applications by restricting their access. DDoS attack techniques come in a variety of forms and are constantly evolving in sophistication. Their shared objective, though, is to flood certain network resources with traffic or service requests from a large number of distinct sources, maybe hundreds of thousands or more. This basically means that identifying and banning a single IP address won't be enough to halt the attack. It is also extremely challenging to differentiate between attack and genuine user traffic when it is dispersed across so many places of origin due to the attack sources' vast dispersal.

1.3.6 Protection DDoS :

DDoS protection is a set of technologies and techniques that are used to protect networks and servers from DDoS attacks. DDoS protection solutions work by identifying and mitigating malicious traffic before it reaches the targeted server or network, and use methods such as traffic filtering, traffic redirection, traffic shaping, botnet tracking, machine learning, and cloud-based protection. It's important to have a comprehensive DDoS protection strategy in place to minimize the risk of a successful attack and minimize the impact on your business.

- **Traffic filtering:** DDoS protection solutions use traffic filtering techniques to identify and block malicious traffic. This can include techniques such as rate limiting, IP blocking, and packet inspection.
- **Traffic redirection:** DDoS protection solutions have the capability to redirect traffic to a separate network or server specifically designed to handle and filter out malicious traffic. This can be achieved through the utilization of a scrubbing center, a cloud-based service, or a specialized DDoS protection appliance.
- **Traffic shaping:** DDoS protection solutions can shape traffic in order to minimize its impact on the targeted server or network. This can involve implementing techniques such as traffic prioritization, traffic throttling, and traffic splitting.
- **Botnet tracking:** In addition to the above, DDoS protection solutions are equipped to track and block botnets, which are networks of compromised computers that can be exploited to launch DDoS attacks.

- **Machine learning:** DDoS protection solutions can use machine learning algorithms to identify and block malicious traffic, even if it is coming from a new or unknown source.
- **Cloud-based protection:** Cloud-based DDoS protection is a service provided by some cloud providers and third-party companies that use cloud-based resources to protect against DDoS attacks.

1.3.7 Measures to Defend Against DDoS Attacks

by combining these measures and staying vigilant, we can enhance their defenses against DDoS attacks and minimize their impact on operations and services.

- **DDoS protection services** are available through cloud providers and cybersecurity firms. These services detect and prevent DDoS attacks by redirecting traffic, filtering out harmful data, and ensuring legitimate traffic reaches its destination.
- **Creating a scalable and resilient infrastructure** is crucial for mitigating the effects of DDoS attacks. Cloud-based systems can adjust resources to manage higher traffic volumes during an attack.
- Utilize traffic analysis tools and anomaly detection systems to recognize abnormal traffic patterns and differentiate between legitimate and malicious requests.
- Rate Limiting: Implement rate-limiting mechanisms on your network to restrict the number of requests from a single source. This can help prevent the rapid accumulation of malicious traffic.
- **Content Delivery Networks (CDNs):** CDNs can help distribute traffic across multiple servers and data centers, minimizing the impact of a DDoS attack on any single server.
- **Web Application Firewalls (WAFs):** Deploy WAFs to protect against application layer attacks. WAFs can filter out malicious requests and payloads before they reach the application.
- **Anycast Routing:** Anycast routing directs incoming traffic to the nearest server in a network of distributed servers. This can help distribute the load and absorb the impact of a DDoS attack.
- **Incident Response Plan:** Develop a comprehensive incident response plan that outlines the steps to take in the event of a DDoS attack. This plan

should include communication strategies, roles and responsibilities, and procedures for bringing services back online.

- **Regular Updates and Patching:** Keep all software and systems up to date with the latest security patches. Many DDoS attacks exploit vulnerabilities that could be mitigated through proper patch management.
- **Traffic Analysis Tools:** Implement real-time traffic analysis tools that can identify traffic anomalies and automatically trigger mitigation measures.
- **ISP Collaboration:** Collaborate with Internet Service Providers (ISPs) to implement traffic filtering and redirection strategies that can help mitigate DDoS attacks closer to their source. [5]

1.3.8 DDoS attack patterns statistics:

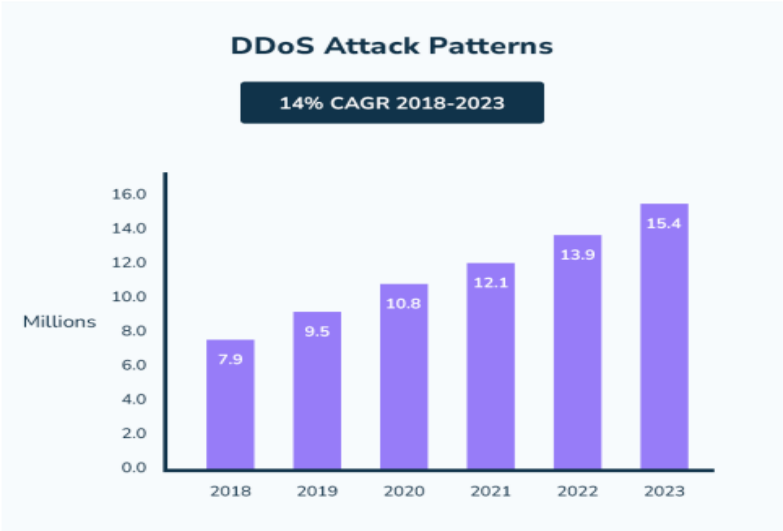

Figure 2: DDoS attack patterns

Cisco predicted that the number of DDoS attacks globally per year would double from 7.9 million in 2018 to 15.4 million in 2023. This is based on historical data on the number of attacks per year up to 2020 and projections for a further three years,

- There has been an 807% increase in DDoS attacks in the nine years to 2022. Quarterly incidents rose from ~325,000 in Q1 2013 to ~2.9 million in Q1 2022.
- NetScout analysis suggests there were ~13 million attacks in 2022; a new high benchmark for attack frequency.

- In 2022, there was a 74% YoY increase in the number of DDoS attacks.

- Initial projections suggest further increases in the DDoS incident rate for 2023. Lumen Technologies mitigated more than 8,600 DDoS attacks in Q1 - a 40% YoY increase, and the second busiest quarter in two years.
- Q1 2023 saw a 47% surge in attacks compared to the same period in 2022. [6]

Rise in Israel-Palestine and Taiwan-Related DDoS Attacks:

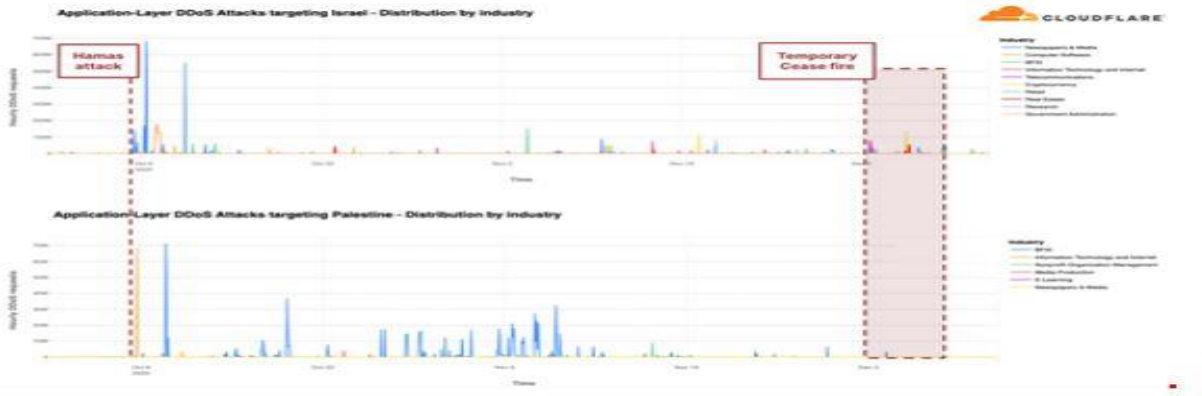

Figure 3: Largest HTTP DDoS attacks

• An increase in DDoS attack activity related to the Israel-Hamas war, with DDoS attack traffic targeting Israeli websites growing by 27% quarter-over-quarter, and by p 1,126% quarter-over-quarter for the traffic targeting Palestinian websites.

• An increase in DDoS attacks targeting Taiwan-related websites, with Q4 2023 registering a 3,370% growth compared to Q4 202. [7]

1.3.9 Conclusion:

In this chapter, we introduced our field of interest, namely computer security and cybersecurity. We showed the most widespread attacks against computer systems and networks, including the DDoS attack for which we reserved a particular interest, since it is the object of our research. In the next chapter, we present the machine learning paradigm that we adopt for the study and testing of models and meta-models for DDoS detection and that we present in chapters 3 and 4 of this thesis.

Chapter 02:

Machine Learning

2.1 Introduction

The recent major advancements in cybersecurity operations and technology can be attributed to data science. Data science, also known as machine learning techniques, is the study of actual occurrences using data. Its flexibility, scalability, and adaptability make it useful in various fields, including social networks, cloud technologies, online banking, mobile environments, and smart grids. In this chapter, we will talk first about artificial intelligence and we will delve into the core principles of machine learning, encompassing definitions, and categories, with a particular emphasis on deep learning and its functions.

2.2 Artificial intelligence (AI):

2.2.1 Definition: The term artificial intelligence (AI) describes the process of creating computer systems that can carry out operations that normally call for human intellect. Among these activities might include problem-solving, pattern recognition, interpreting natural language, and experience-based learning. AI systems are capable of mimicking human cognitive processes such as perception, learning, reasoning, problem-solving, and language comprehension.

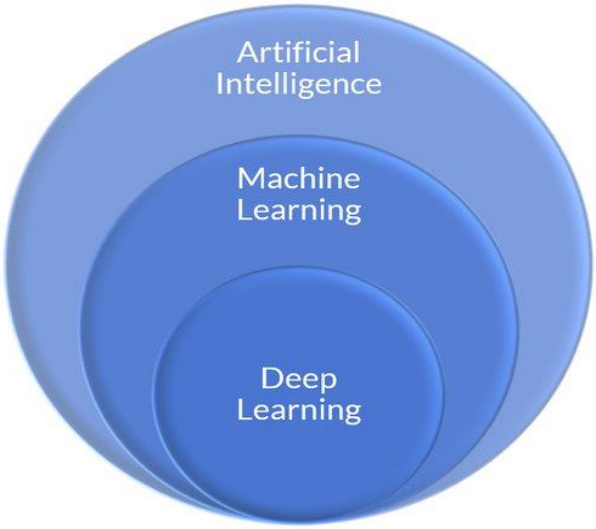

Figure 4: Diagram of decomposition of the field of artificial intelligence

2.2.2 Historical:

Artificial Intelligence (AI) boasts a rich history defined by notable milestones and phases of intensive research and advancement. The origins of AI can be traced back to the mid-20th

century, notably marked by the introduction of the term “artificial intelligence” during the 1956 Dartmouth Conference, credited to John McCarthy. Early AI pioneers like Marvin Minsky, Allen Newell, and Herbert Simon explored symbolic or rule-based approaches to problem-solving and reasoning, laying the groundwork for the field.

The 1950s and 1960s witnessed a fervent pursuit of creating programs capable of emulating human-like intelligence. Significant progress during this era included the development of groundbreaking programs such as the Logic Theorist (1955) and the General Problem Solver (1957), showcasing AI’s potential in logical reasoning and problem-solving tasks.

However, the 1970s and 1980s brought about a slowdown in AI advancements, often referred to as the “AI winter”. Funding for AI research dwindled, and enthusiasm waned due to challenges in achieving substantial breakthroughs and realizing the promised capabilities of AI systems.

The resurgence of AI occurred in the 1990s, propelled by advancements in computing power, algorithms, and data accessibility. Machine learning, particularly neural networks, gained prominence for their capacity to learn from data. Milestones like IBM’s Deep Blue defeating

Chess champion Garry Kasparov in 1997 underscored AI’s potential in complex problem-solving.

The 21st century witnessed unprecedented progress in AI, driven by breakthroughs in deep learning, reinforcement learning, and natural language processing. AI technologies have become ubiquitous across diverse industries, revolutionizing sectors such as healthcare, finance, transportation, and entertainment.

Contemporary AI research is focused on addressing challenges like data privacy, algorithmic, bias, and ethical considerations. Ongoing efforts aim to enhance the capabilities of AI systems, refine algorithms, and ensure responsible and ethical AI development and deployment.

2.2.3 Types of Artificiel intelligence :

- **Weak or specialized AI:** Specialized AI, another name for weak AI, is created to carry out a certain function. For example, Netflix uses algorithms to assess your watching preferences in order to propose shows and movies to you. Similar to this, Apple's Siri and Amazon's Alexa are voice assistants that can listen to voice requests and carry out basic tasks.

- **Strong AI:** Strong AI encompasses two main categories: artificial general intelligence (AGI) and artificial superintelligence (ASI). AGI, also known as general AI, represents a theoretical form of AI where machines would possess intelligence on par with humans, equipped with autonomous consciousness capable of problem-solving, learning, and future planning. ASI, alternatively referred to as superintelligence, is speculated to surpass human intelligence and capabilities. While strong AI remains largely theoretical and lacks practical implementations to date, AI researchers continue to explore its development actively.

2.3 Machine learning (ML):

Within the expansive domain of artificial intelligence, machine learning emerges as a fundamental methodology, empowering machines to acquire knowledge autonomously, thereby reducing reliance on pre-programmed instructions. Unlike traditional programming paradigms, machine learning leverages algorithms and statistical models to enable computers to learn from data iteratively, enhancing their performance over time.

Central to the functioning of machine learning is the incessant influx of data, which serves as the cornerstone for evaluating, training, and refining the learning algorithms. This continuous feedback loop allows machines to adapt and evolve their behaviors in response to changing circumstances, without the need for explicit human intervention.

Furthermore, the symbiotic relationship between machine learning and big data is undeniable. Big data, characterized by vast volumes, high velocity, and diverse varieties of data, provides the fuel that propels machine learning forward. The sheer scale and complexity of big data necessitate advanced analytical techniques like machine learning to derive meaningful insights and extract valuable knowledge.

Big data and machine learning are essentially closely related, with both enhancing and supporting each other. Big data depends on machine learning to realize its full potential by gleaning useful insights and patterns from the deluge of data, while machine learning uses big data to fuel innovation and discovery. The future of artificial intelligence and data-driven decision-making will surely be greatly influenced by the synergy between machine learning and big data as technology advances.

2.3.1 Techniques of Machine Learning:

2.3.1.1 Supervised Learning: The algorithm is trained on a labeled dataset, where each input example is paired with the corresponding correct output. The algorithm learns to map inputs to outputs based on the examples provided during training.

In data mining, supervised learning is divided into two distinct problem types: [8]

- **Classification:** A classification problem occurs when the output variable represents a distinct category.
- **Regression:** is employed to analyze the connection between independent and dependent variables.

2.3.1.2 Unsupervised Learning: Unsupervised learning involves training algorithms on unlabeled data. The goal is to find hidden patterns or structures in the data without explicit guidance. Also, the unsupervised learning model is classified into different categories of algorithms, such as:

- **Clustering:** is an unsupervised learning method, that arranges data points into clusters according to similarities or patterns, without requiring predetermined labels. Its utility spans tasks like data exploration, pattern recognition, and anomaly detection.
- **Association:** Through the examination of primary attributes within the data, an unsupervised learning model has the capability to anticipate additional attributes with which they are frequently linked.
- Anomaly detection involves using the model to pinpoint data outliers. For example, banks detect fraudulent activities by examining unusual purchasing patterns exhibited by customers such as instances where a card is used in vastly different locations within a short timeframe, prompting further investigation by the bank. [9]

2.3.2 Supervised learning algorithms: In machine learning, supervised learning techniques are employed when the model is trained on a labeled dataset, which means that every input data point has a matching target label. The following are some instances of algorithms for supervised learning: [10]

- **Support vector machines (SVM):** is a supervised machine learning model utilized primarily for classification purposes. It is a prominent member of the kernel method algorithms and is widely recognized for its effectiveness in separating data points within a dataset by identifying an optimal hyperplane.

So, the goal of an SVM is to identify the hyperplane that maximizes the margin—that is, the separation between the nearest data points and the hyperplane. The SVM provides superior generalization to unknown data by searching for the hyperplane with the highest margin, as opposed to merely identifying any hyperplane that divides samples.

SVMs employ the kernel technique, which entails projecting the data into a higher-dimensional space where it may be linearly separable, or the insertion of soft margins, which allow for some flexibility in data categorization and error tolerance. This method allows SVMs to function properly even in situations when a hyperplane cannot be used to divide data in the original space. [11]

- **Decision trees:**

  Decision trees are widely acknowledged as one of the most popular methods for representing classifiers. Researchers from various fields, including statistics, machine learning, pattern recognition, and data mining, have explored the process of growing decision trees from available data. In essence, a decision tree is a supervised machine-learning technique used for categorization or prediction by analyzing responses to a series of questions. [12]While it falls under the realm of supervised learning, where training and testing are done with pre-categorized data, decision trees may not always provide definitive answers. Instead, they offer options for data scientists to make informed decisions based on their judgment, emulating human thinking and enabling relatively easy understanding and interpretation of outcomes. [13]

- **Random forest:**

  Random forests mitigate the overfitting problem of decision trees by creating a diverse set of trees with slight variations. This approach ensures robust predictions by combining multiple trees that exhibit acceptable performance while having unique characteristics from one another.

  The randomness in the tree-building process guarantees diversity among the trees, which helps prevent overfitting. This randomness is evident in two main aspects: the random selection of data points for building each tree and the random selection of features at each split.

  Random forests are part of the ensemble learning technique known as "bagging" or "bootstrap aggregating." This method involves creating multiple instances of the same

model (decision trees), training each tree on a random subset of the data, and then combining their predictions to enhance accuracy.

- **Linear Regression:** is a simple method for supervised learning. It is particularly useful for predicting quantitative responses. Although it may appear less exciting compared to some more modern statistical learning approaches, linear regression is still a valuable and commonly used method. Moreover, it provides a solid foundation for new approaches that are seen as generalizations or extensions of linear regression.
- **K-nearest neighbor:** The KNN method, which is often referred to as K-nearest neighbor, is a non-parametric technique that groups data points according to their closeness and correlation with other accessible data.

Semi-supervised Learning: This approach combines elements of both supervised and unsupervised learning. The algorithm is trained on a dataset that contains a small amount of labeled data along with a larger amount of unlabeled data. [14]

2.2 Deep Learning:

2.2.1 Definition:

Deep learning is a form of machine learning and artificial intelligence (AI), replicates the cognitive processes through which humans acquire specific knowledge. Deep learning models can learn classification tasks and identify patterns in diverse data types such as photos, text, audio, and more. Moreover, it enables the automation of tasks that traditionally require human intelligence, such as image description and audio transcription. In the realm of data science, which encompasses statistics and predictive modeling, deep learning plays a crucial role. It proves highly advantageous for data scientists who face the challenge of collecting, analyzing, and interpreting vast amounts of data, as it accelerates and simplifies this process. While the human brain relies on millions of interconnected neurons to acquire knowledge, deep learning employs neural networks composed of multiple layers of software nodes that collaborate harmoniously. These deep learning models are trained using extensive sets of labeled data and neural network architectures. [15]

2.4.3 Application of deep learning:

Deep learning a subset within the realm of machine learning, has emerged as a revolutionary technology with broad applicability across numerous sectors. Its innate capacity to glean

intricate patterns from vast datasets has propelled significant progress in fields ranging from computer vision and natural language processing (NLP) to healthcare, finance, and beyond.

In the domain of computer vision, deep learning methodologies have brought about transformative capabilities in tasks like object detection, image classification, and facial recognition. These advancements are foundational to the development of autonomous vehicles, surveillance systems, and advanced medical diagnostic tools. Depp learning models exhibit remarkable proficiency in interpreting complex visual data, enabling machines to perceive and comprehend their surroundings with ever-increasing precision.

Likewise, within natural language (NLP), deep learning algorithms excel in endeavors such as sentiment analysis, machine translation, and text generation. These models possess the ability to process and decipher human language in a manner closely resembling human cognition. Consequently, the power applications such as virtual assistants, language translation services, and sentiment analysis tools crucial for social media monitoring and customer feedback assessment.

Deep learning's influence extends significantly into the realm if healthcare, where it plays a pivotal role in medical image analysis, disease diagnosis, and drug discovery. By scrutinizing extensive medical datasets, including images from modalities like MRI and CT scans, deep learning models aid healthcare professionals in diagnosing ailments, devising treatment strategies, and predicting patient prognoses with unprecedented accuracy. Moreover, these techniques expedite drug discovery processes, fostering the development of novel treatments and therapies.

In the financial domain, deep learning techniques are leveraged for algorithmic trading, fraud detection, and credit risk assessment. Analyzing streams of financial data, deep learning models identify lucrative trading opportunities, detect fraudulent activities, and evaluate creditworthiness

More effectively than conventional methodologies. These applications empower financial institutions to make data-driven decisions, mitigate risks, and elevate customer experiences.

Recommendation systems, powered by deep learning algorithms, personalize user interactions by furnishing tailored suggestions for products, content, and services. By analyzing user preferences and behavior, these systems deliver pertinent recommendations that enhance user engagement and satisfaction.

Technologies related to deep learning have transformed gaming, robotics, biometrics, and remote sensing. They improve autonomy, let intelligent agents communicate with their surroundings, and offer safe ways to authenticate. These technologies also support disaster

response, land use classification, and environmental monitoring. Deep learning will continue to influence technology and society in the future as long as research and development are conducted, solving difficult problems and spurring creativity.

2.4.4 Natural Language Processing:

Neural networks, inspired by the structure and operation principles of the nervous system and the brain of animals and humans, are information processing systems. Comprised of interconnected neurons, these networks communicate through activation signals. Similar to the human brain, neural networks can generate output patterns when presented with input patterns. They have become a prominent area of research and development in machine learning, offering solutions to real-world problems. Neural networks possess unique characteristics and capabilities that set them apart from other technologies. They have been successfully applied to various tasks, including human handwriting recognition, reading typewritten text, modeling complex systems, and predicting loan outcomes. There are two distinct types of neural networks: [16]

- Artificiel neural network
- Deep neural network

2.4.5 Artificial neural network :

When the term "neural networks" is mentioned, we often envision a network consisting of interconnected nodes, similar to the structure of the human brain. Typically, this network comprises numerous complex organizations that are interconnected. The neural organization serves as the backbone of deep learning, functioning as a computational model that is based on the architecture and capabilities of biological neural networks. It can be likened to an artificial human nervous system, which receives and processes information in terms of computer science. Neural networks mainly 3 layers :

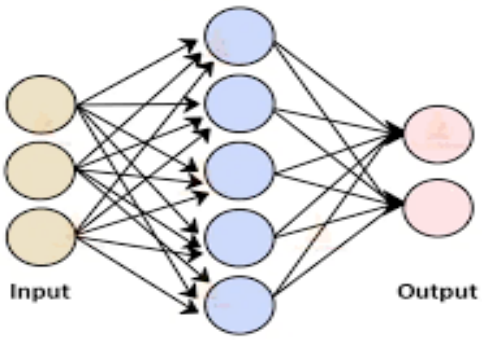

Figure 05: Simple Neural Network (1 hidden layer)

1. The input layer is responsible for communicating with external factors such as the climate and receiving inputs for the system. It manages all information sources before transferring them to the hidden layer for processing.

2. Within the hidden layer, a variety of neurons undergo an activation function. This layer serves as a bridge between the information and output layers, playing a crucial role in processing data sources.

3. The output layer of the neural network gathers and presents data in the intended manner. It is designed to showcase the output to the external environment as intended. [17]

2.4.5.2 Applications of ANN:

A neural network is going to play a vital role in the future. Nowadays, we are producing billions of data each day, thus there is a necessity for the neural network to analyze and develop the relationships between the data. There can be no doubt about the need for neural networks in every field coming in the future. Although we are discussing here a few important fields where the concept of the neural network is widely used. This field in data mining by business organizations to effectively extract valuable information from raw data. Data mining, also known as knowledge discovery and information retrieval, involves analyzing hidden patterns and useful information from raw data. In the field of Forecasting, predictions are typically based on either quantitative analysis, qualitative analysis, or a combination of both. Quantitative forecasting is often referred to as objective analysis, while qualitative forecasting is known as judgmental or subjective analysis. There is often a tension between these two approaches. Quantitative forecasts, which are commonly favored by operations, tend to be developed using a detailed perspective, whereas judgment-based forecasts, typically preferred by the marketing team, are approached from a broader viewpoint. For example, a key

marketing objective may be to ensure an ample supply, while the operations team's focus is on minimizing inventory. These two approaches aim to understand how forecasting errors occur and to explore the potential of utilizing artificial intelligence techniques. In Image compression, the goal of neural network data compression is to replicate the original image by storing, encrypting, and compressing it once again. we can use picture compression neural networks to reduce the amount of our data and the neural network plays a crucial role in route detection. It helps in identifying routes during travel by creating road intersections. Google Maps is a prime example of this technology. Moreover, neural networks have various applications in the medical, banking, and robotics sectors.

2.4.6 Deep neural network:

A deep neural network (DNN) is an artificial neural network (ANN) that consists of multiple layers positioned between the input and output layers. DNNs possess the remarkable ability to perform intricate tasks such as comprehending images, recognizing sounds, and understanding text. These networks are specifically designed to emulate the way humans think and learn, utilizing deep learning techniques in conjunction with artificial neural networks. The application of deep neural networks has proven to be highly successful in various domains, including image classification, language translation, and speech recognition. Notably, they excel at solving pattern recognition problems autonomously, without the need for human intervention. Moreover, DNNs have been effectively employed in early-stage diabetic retinopathy detection, microscope image analysis, and addressing the challenge of generalization in deep learning. [18]

2.4.6.1 The architecture of DNN:

Deep neural networks (DNNs) are popular in image and computer vision due to their high accuracy and ability to extract complex features. However, they require high computational complexity and memory. Advanced processing units enable the execution of computationally intensive DNNs for real-world applications like image classification, speech recognition, and natural language processing. Image recognition is crucial for tasks like tagging, searching, and guidance. [19]

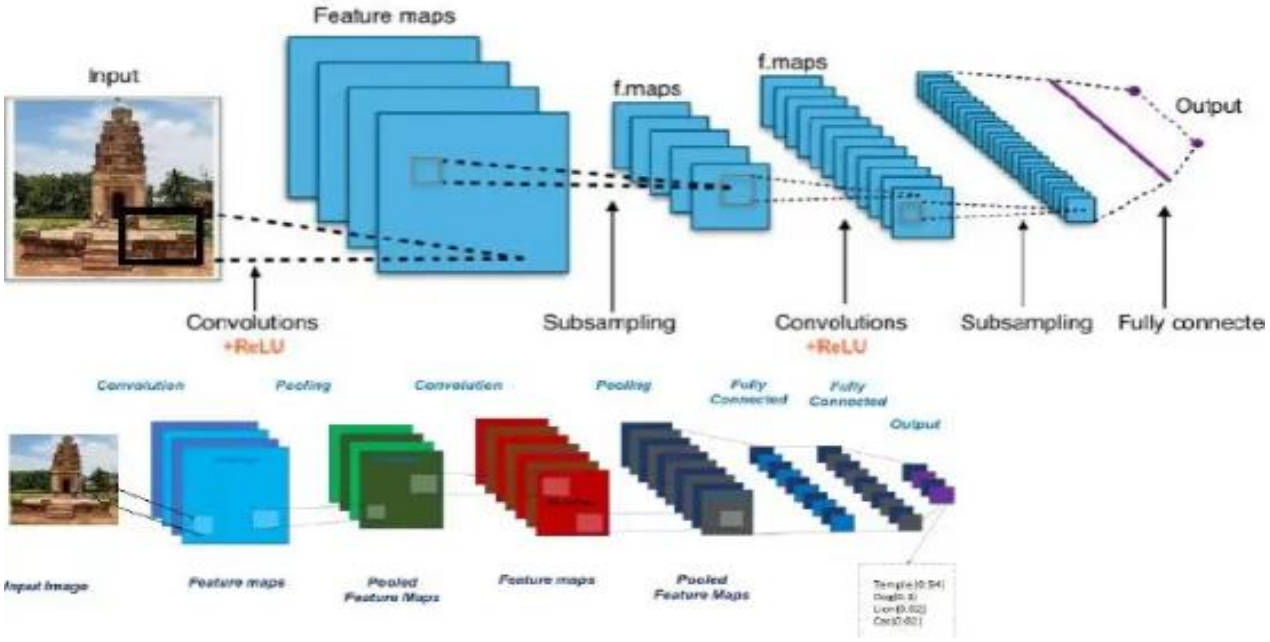

Figure 06: Architecture of DNN typically a CNN

2.4.6.2 Uses of DNN:

Deep Neural Networks (DNNs) are crucial in various industries, including image recognition, natural language processing, finance, recommendation systems, cybersecurity, and gaming. They can learn complex patterns from data, enhancing security, sentiment analysis, and user experience. In medical diagnostics, DNNs enable accurate disease detection. In Natural Language Processing, they improve sentiment analysis, machine translation, chatbots, spam detection, and human-computer communication. In cybersecurity, DNNs detect malware, intrusions, and DDoS attacks, allowing for prompt response and mitigation efforts to protect data assets and network infrastructure.

2.4.7 Convolutional Neural Network (CNN):

A convolutional neural network is a type of deep neural network widely used in deep learning for analyzing visual imagery. Unlike traditional neural networks that rely on matrix multiplications, Convent's employ a unique technique known as Convolution. In mathematics, convolution is a mathematical operation that combines two functions to create a third function, which represents the modification of one function by the other. [20]

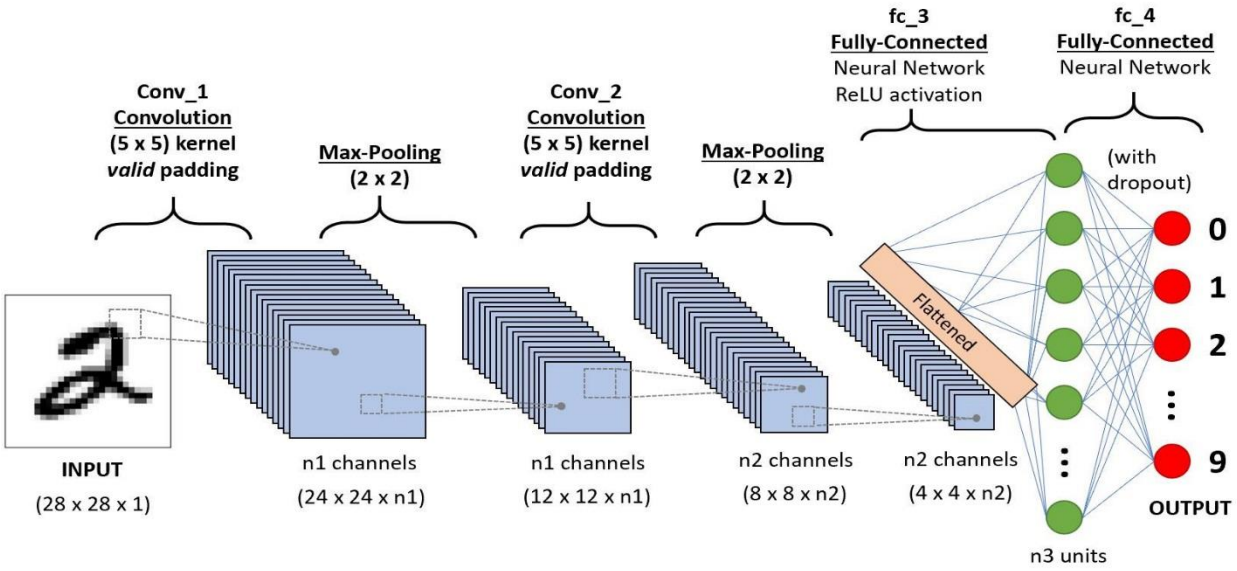

Figure 07: General architecture of deep convolution neural network with different types of layers.

The convolution layer in a CNN or ConvNet analyzes the fundus image and generates a feature map to predict the probability value for each image feature class. It extracts high-level features like edges and boundaries, using features dimensionality reduction and increasing features. The pooling layer processes the extracted features, selecting essential ones for dimensionality reduction and reducing noise information. The number of layers is increased based on the complexity of the fundus image. The fully connected input layer transforms the output into a singular vector, assigning a weight value to the input for precise label identification. The fully connected output layer calculates the probability value for determining the specific class of the fundus image. Softmax classification techniques are used to analyze low-level image features. [21]

2.4.8 Recurrent neural network (RNN):

A recurrent neural network (RNN) is an extension of a conventional feedforward neural network, which can handle a variable-length sequence input. The reason that RNN can handle time series is that RNN has a recurrent hidden state whose activation at each time is dependent on that of the previous time. Long short-term memory units (LSTMs) are one type of RNN, which makes each recurrent unit adaptively capture dependencies of different time scales. LSTMs have cell and forget gates to modulate the flow of information. [22]

The **figure 09** illustrates the architecture of RNN:

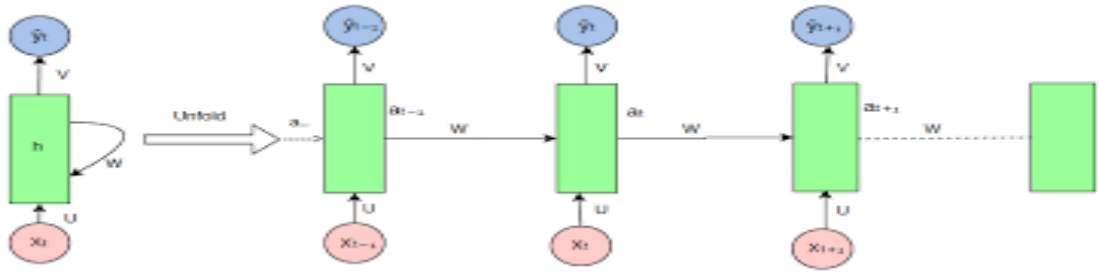

Figure 08: RNN Architecture

2.4.8.1 Types of RNN:

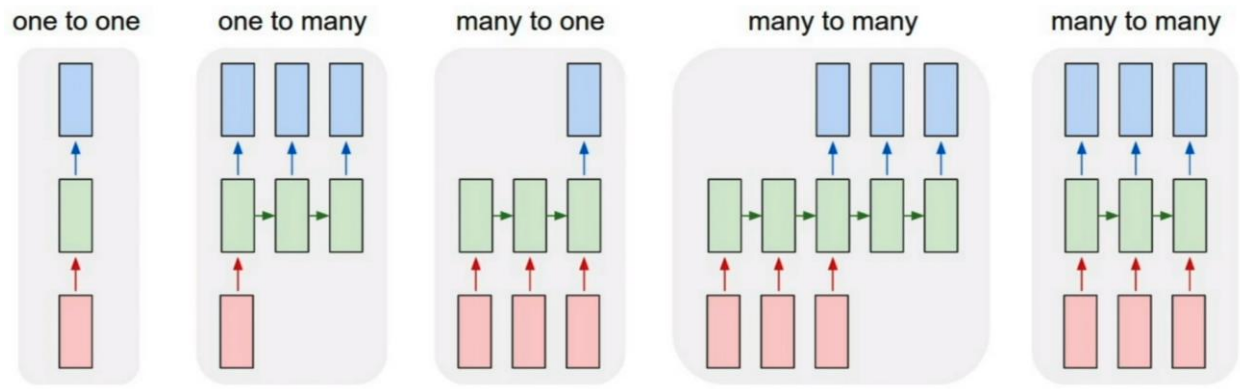

Figure 09: Types of RNN

- **One-to-one :**

  This refers to the traditional feed forward neural network structure, where there is one input and one expected output.

- **One to-many :**

  This can be likened to the process of image captioning. A single image is provided as a fixed size input, and the output can consist of words or sentences of varying lengths.

- **Many-to-one :**

This type of architecture is commonly used for sentiment classification. The input is typically a sequence of words or even paragraphs, and the output can be a continuous regression value representing the likelihood of a positive sentiment.

- **Many-to-many :**

This model is well-suited for tasks such as machine translation, similar to what we observe with Google Translate. The input can be an English sentence of varying length, and the output will be the same sentence translated into a different language, also with variable length. Additionally, the many-to-many model can be applied to video classification at the frame level. Each frame of a video is fed into the neural network, and an output is generated. However, since frames are often interdependent, it is crucial for the network to propagate its hidden state from one frame to the next. [23]

2.4.8.3 Uses of RNN:

Recurrent Neural Networks (RNNs) are instrumental in various domains due to their adeptness at handling sequential data and modeling temporal dependencies. Their applications span a wide range of fields, including Natural Language Processing (NLP), where they are utilized for tasks like sentiment analysis, language translation, and named entity recognition. In NLP, RNNs analyze the sentiment of text data from sources like social media posts, facilitate language translation through systems like Google Translate, and identify entities such as people and locations within text. Additionally, RNNs play a crucial role in Speech Recognition, enabling the conversion of speech signals into text for applications like virtual assistants. Moreover, RNNs are pivotal in Time Series Prediction, forecasting stock prices and weather conditions based on historical data. They also excel in Sequence Generation tasks, generating text, music, or images, and in Handwriting Recognition, enabling applications like optical character recognition (OCR). Furthermore, RNNs are instrumental in Gesture Recognition, Robotics, Biomedical Signal Analysis, Video Analysis, Music Composition, and Machine Translation, showcasing their versatility and applicability across diverse domains.

2.4.9 Long Short-Term Memory (LSTM):

The simple RNN repeating modules have a basic structure with a single tanh layer. RNN's simple structure suffers from short memory, where it struggles to retain previous time step information in larger sequential data. These problems can easily be solved by long short-term memory (LSTM), which is an advanced variant of Recurrent Neural Networks that is capable of remembering long periods of information. [24] The overall architecture of an LSTM network is depicted in **Figure11**.

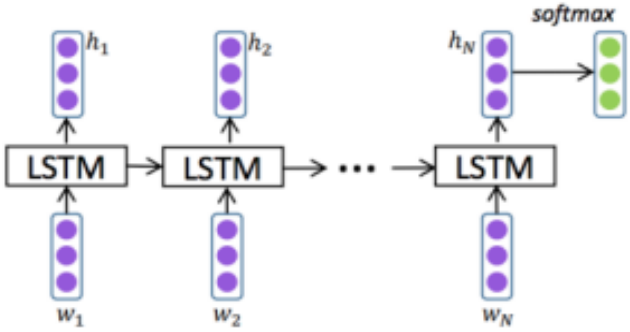

Figure 10: The high- level architecture of LSTM model

2.4.10 Reinforcement learning: is a machine learning (ML) method that teaches software how to make decisions to achieve the best possible outcomes. This technique draws inspiration from the experimental learning process used by humans to achieve their goals. Actions taken by the software that contribute to achieving the objective are reinforced, while those that hinder progress are disregarded.

RL algorithms operate on the principle of reward and punishment when analyzing data. They learn from feedback received after each action and identify the best strategies to achieve the desired outcomes. These algorithms can also delay gratification, understanding that the best long-term strategy may require short-term compromises or temporary setbacks. [25]

2.4.10.1 Reinforcement learning use cases: Application of reinforcement learning (RL) may be made in a wide range of real-world scenarios. They are a few examples:

- **Marketing Personalization:** Recommender systems are one example of an application where reinforcement learning (RL) is used to adapt suggestions to specific users based on their interactions, resulting in more personalized experiences. For instance, an application may show a user advert depending on certain demographic data. The program learns which advertisements to show on the user's smartphone to maximize product sales with each encounter with an advertisement.
- **Optimization Challenges:** Traditional optimization techniques assess and contrast potential solutions according to predetermined standards to find solutions. In reinforcement learning, learning relies on interactions to identify the optimal answers gradually, or as near to a solution as feasible.
- For instance, RL is used by a cloud expenditure optimization system to monitor shifting resource needs and choose the best instance kinds, numbers, and configurations. It bases

its judgments on variables including spending, consumption, and the state and availability of cloud infrastructure.

- **Financial predictions:** Financial market dynamics are intricate and involve statistical characteristics that fluctuate over time. By monitoring market movements and accounting for transaction costs, RL algorithms can maximize long-term profits.
- An algorithm may, for instance, study the trends and laws of the stock market before putting strategies to the test and documenting the results. It formulates a profit-maximizing plan and dynamically generates a value function.

2.4.10.2 The Benefits of Reinforcement Learning:

One of the many benefits of reinforcement learning (RL) is that it may perform well in contexts that are complicated and have a lot of rules and relationships. RL algorithms discover novel ways to maximize outcomes by quickly adapting to ever-changing settings. They can learn on their own and use human feedback to build systems that adjust to preferences, knowledge, and human corrections, so they don't need human engagement. RL is best suited for scenarios where decisions have long-term effects since it concentrates on optimizing long-term benefits. For instance, RL can maximize both long-term expenses and energy efficiency.

2.4.11 Conclusion

We have devoted this chapter to the most used and cited machine learning models in the literature of the field. These are the supervised, the unsupervised and those based on reinforcement models. We have deferred the details of decision tree models and ensemble learning meta-models to Chapter 3, because they represent the models on which our study is based. We have devoted a significant part to models based on neural networks including deep neural networks for their importance in the development of AI applications based on machine learning.

Chapter 03:

ML Models for DoS and DDoS detection

3.1 Introduction

The detection of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks presents a major challenge in the field of cybersecurity. These attacks, which aim to overwhelm target servers with excessive traffic, can result in service interruptions, financial losses, and damage to the reputation of businesses. Faced with this persistent threat, artificial intelligence (AI) models have emerged as essential tools for detecting and mitigating the effects of DoS/DDoS attacks.

AI models for DoS/DDoS detection leverage machine learning and data analysis capabilities to identify malicious patterns in network traffic. These models can be classified into two main categories: signature-based models and anomaly-based models. Signature-based models search for specific characteristics of known attacks in network traffic, while anomaly-based models detect aberrant or deviant behaviors that do not match normal traffic patterns.

In this chapter, we will explore the AI models for DoS/DDoS detection for which we are interested. We will discuss the advantages and limitations of each model, as well as the challenges associated with implementing these models in real-world environments. Finally, we will examine future research prospects in this field, highlighting emerging trends and innovation opportunities to enhance network resilience against DoS/DDoS attacks.

3.2 Overview of ensemble models in machine learning:

In machine learning, ensemble approaches combine the predictions of several base learners or weak learners to produce a single unified prediction. These strategies aim to improve performance, generality, and robustness by combining predictions from many models. Ensemble models are a prominent and commonly used machine learning approach. Simply said, they combine numerous simple classifiers to generate a strong classifier. These classifiers may be merged using methods like as bagging, boosting, stacking, or voting, with each giving a unique strategy for combining individual predictions to increase overall performance. We will describe the main techniques of ensemble learning which are bagging, boosting, gradient boosting, and random forest.

3.2.1 Bagging:

Bagging (Bootstrap Aggregating) is an ensemble approach that divides a dataset into n samples with replacement, which is also known as bootstrap sampling. Each of the n samples is then utilized to train an independent machine-learning model. After training,

the outputs of all of these different models are integrated into a single result via a process called voting. This enables Bagging to use a variety of predictions from several models to get a more robust and accurate final prediction. [26]

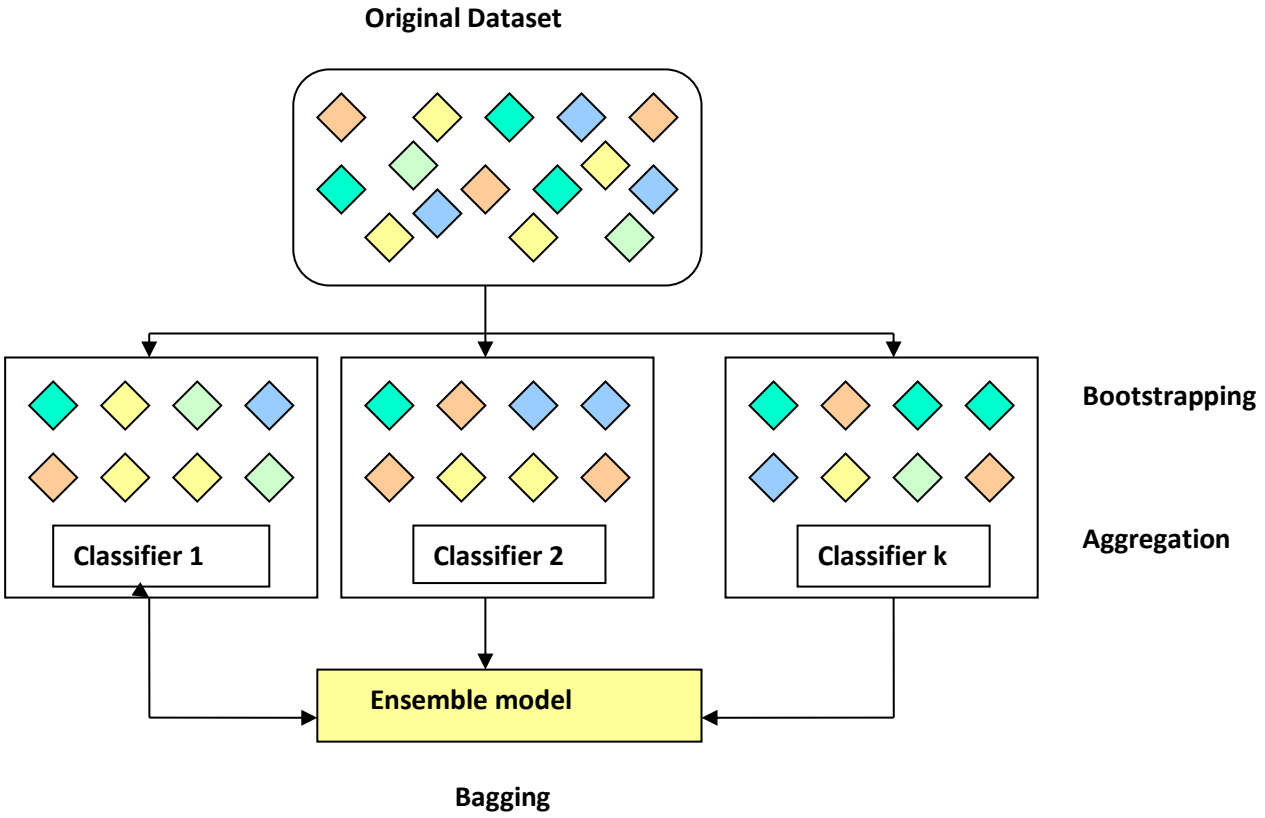

Figure 11: Ensemble learning by Bagging

3.2.2 Boosting:

Boosting is a strong machine-learning strategy that involves training a sequence of weak learners, such as decision trees, on different copies of the data. Each weak learner prioritizes cases that were misclassified by earlier models, giving them greater weights. The final forecast is then formed by integrating all of the weak learners' guesses, which is frequently done via weighted voting or averaging. The purpose of boosting is to improve model performance by repeatedly correcting mistakes committed by previous models and focusing on difficult cases. It is well known for its capacity to eliminate bias and enhance forecast accuracy, making it a popular choice for a variety of applications. Some well-known boosting algorithms are AdaBoost, Gradient Boosting Machines (GBM), Boost, LightGBM, and CatBoost. **[27]**

3.2.3 Gradient Boosting:

This boosting technique is mainly used with decision trees (it is then called Gradient Tree Boosting). the fundamental concept is to combine numerous classifiers but create them iteratively. These mini-classifiers are often basic and parameterized functions, most notably decision trees, with each parameter representing the branch split criterion. The final super-classifier is a weighted combination of these mini-classifiers. **[28]**

3.2.4 XGBoost uses decision trees with boosted gradients to increase speed and performance.

It is strongly dependent on the computing speed and performance of the target model. Model training is sequential, implementing gradient-boosted machines sluggish.

3.2.5 Stacking:

This is another ensemble method; it's referred to as a layered generalization.

This strategy works by allowing a training algorithm to combine predictions from multiple similar learning algorithms. Stacking has been effectively applied to regression, density estimation, distance learning, and classification. It can also be used to determine the error rate while bagging. **[29]**

3.2.6 LightGBM :

LightGBM is a tree-based gradient boosting variant similar to XGBoost, released on October 17, 2016, as part of Microsoft's Distributed Machine Learning Toolkit (DMTK) project. It is designed to be fast and distributed, resulting in faster training speed and low memory usage, with support for GPU and parallel learning, and the ability to handle large datasets. **[30]**

3.3 Models used :

3.3.1 Decision Trees:

3.3.1.1 Definition:

Decision trees are a non-parametric supervised learning approach that is used for both classification and regression applications. They have a hierarchical tree-like structure with a root node, branches, internal nodes, and leaf nodes. Each internal node represents an attribute decision, with each branch representing the outcome and each leaf node denoting a class label (in classification) or a continuous value (in regression). These trees are built recursively by selecting the appropriate characteristic to split the data on at each step, usually based on factors like information gain or impurity. This recursive construction process produces a

flowchart-like representation that is simple to analyze and understand, making decision trees ideal for exploratory analysis and decision-making in a variety of domains. [31]

3.3.1.2 Types of decision trees:

There are two kinds of decision trees: classification trees and regression trees. They are then divided into separate algorithms and joined together using various nodes and branches. It is vitally necessary that you select one that is directly relevant to the objective of your decision tree.

- **Classification trees:** Decision trees are commonly utilized to determine whether an event has occurred or not, typically resulting in a binary outcome of "yes" or "no". This decision-making approach finds widespread application across various real-world scenarios.

  The is an example of decision trees in classification tasks:

  Example: Homeownership based on age and income

  In a classification tree, the date set splits according to its variables. There are two variables, age, and income, that determine whether or not someone buys a house. If training data tells us that 70 % of people over age 30 bought a house, then the data gets split there, with age becoming the first node in the tree. This split makes the data 80% “pure”. The second node then addresses income from there.
- **Regression trees:** Regression trees are used when the target variable is continuous, which means the predicted outcome is a real number. These trees anticipate a continuous value for each input, allowing for real-number-based analyses.

3.3.1.3 Types of Decision Tree Algorithms:

3.3.1.3.1 ID3 (Iterative Dichotomiser 3):

The ID3 (Iterative Dichotomiser 3) algorithm, developed by Ross Quinlan in 1986, is a foundational approach for constructing decision trees in classification tasks. It employs a greedy, top-down strategy to build decision trees from given datasets. At each node, it selects the feature that maximizes information gain, calculated using entropy, to split the data. Entropy measures the uncertainty or disorder in a dataset, and information gain quantifies the reduction in entropy achieved by splitting the data based on a particular feature. ID3 recursively divides the dataset based on chosen attributes until either all examples in a node belong to the same class or there are no more attributes to divide on.

Despite its effectiveness, ID3 has drawbacks such as overfitting and the inability to handle continuous attributes directly. To mitigate these issues, other algorithms like C4.5 and CART have been developed.

The key principles of ID3 include entropy calculation, information gain maximization, and recursive partitioning, which are fundamental to understanding decision tree algorithms in machine learning.

3.3.1.3.2 C4.5:

Ross Quinlan developed the C4.5 decision tree technique to improve the ID3 algorithm. In machine learning and data mining applications, it is a popular method for building decision trees. Certain limitations of the ID3 algorithm are addressed in C4.5, such as its inability to deal with continuous features and proclivity to overfit the training set.

Ross Quinlan created the decision tree method C4.5 as an improvement to the ID3 algorithm, which is widely used in machine learning and data mining applications. C4.5 solves ID3's weaknesses, such as its inability to handle continuous variables and proclivity to overfit training data. The gain ratio, a variation of information gain, addresses bias toward qualities with multiple values. C4.5 sorts the attributes first then select the midway between neighboring values as a possible split point. It then determines the maximum value for each split point. C4.5 also generates rules from the decision tree, enabling predictions based on new data. Despite its usefulness, C4.5 is vulnerable to noisy data and may not perform well on datasets with a large number of characteristics.

3.3.1.3.3 CART (Classification and Regression trees): are decision tree algorithms used in machine learning to perform classification and regression problems. They work by recursively partitioning the feature space into subsets, which results in a tree-like structure with each internal node representing a feature-based choice and each leaf node representing the predicted outcome. [32]

3.3.1.3.4 CHAID (Chi-Square Automatic Interaction Detection):

CHAID, developed in South Africa in 1975 and later published by Gordon V. Kass in 1980, is a decision tree technique employing adjusted significance testing methods like Bonferroni correction and Holm-Bonferroni testing. Originally known as XAID, it serves both predictions, akin to regression analysis, and classification purposes, while also detecting variable interactions. Its roots trace back to AID and THAID procedures of the 1960s and 1970s, extending from earlier research, including that by Belson in the 1950s.

CHAID finds practical application in various domains such as direct marketing for consumer group selection and analysis of their responses to different variables. Early uses also include medical and psychiatric research. Like other decision trees, CHAID provides highly interpretable visual outputs. However, its default multiway splits demand substantial sample sizes for reliable analysis, as small sample sizes can lead to overly fragmented respondent groups. [33]

3.3.1.3.5 MARS (Multivariate Adaptive Regression Splines):

MARS, an extension of CART, uses splines to capture nonlinear interactions between variables. As a regression procedure, MARS employs forward stepwise selection to construct a piecewise linear model. This model displays the output variable as a linear function of the input variables, but the slope of the linear function might change at different places in the input space, which are referred to as knots. MARS chooses and arranges knots based on data distribution and the requirement to capture nonlinearities. [34]

3.3.1.4 Applications of Decision Trees:

Decision trees are used in different disciplines because of their simplicity, interpretability, and effectiveness in handling classification and regression tasks. Here are a few common applications :

- **Customer Relationship Management (CRM) :**
  Decision trees are used in CRM to forecast customer behavior, such as whether a customer will leave, reply to a marketing campaign, or buy a product. They let organizations modify marketing strategies and customer interactions depending on projected outcomes.
- **Medical Diagnosis:** Decision trees help with medical diagnosis by analyzing patient data (symptoms, test results, medical history) and predicting diseases or disorders. They can aid doctors in identifying potential ailments and recommending appropriate therapies.
- **Fraud detection**: systems use decision trees to identify suspicious transactions or activity. They employ a variety of variables, including transaction amount, frequency, location, and user behavior, to identify possibly fraudulent activities. [35]

3.3.1.5 Advantages and disadvantages of decision trees:

Advantages :

- **Interpretability:** Decision trees are simple to comprehend and interpret, making them useful for explaining the reasoning behind predictions to non-technical stakeholders.
- **Feature selection:** Decision trees implicitly perform feature selection by selecting the most informative features for splitting, reducing the need for manual feature engineering.

Disadvantages :

- **Instability:** Minor differences in data might result in diverse tree architectures, making decision trees less stable than other methods. Random Forests and other ensemble approaches can help to mitigate this issue.
- **Limited Expressiveness:** Decision trees may fail to represent complicated relationships in data when compared to more flexible models such as neural networks or gradient-boosting machines. Ensemble approaches can help to increase their expressiveness.

3.3.2 Random Forest :

Random Forest is a popular machine-learning algorithm used for both classification and regression tasks. It operates as an ensemble learning technique by amalgamating numerous decision trees to generate a conclusive prediction. The fundamental concept behind the random forest algorithm involves creating a substantial number of decision trees and subsequently merging their predictions to attain a more precise and consistent outcome. Each decision tree within the forest is trained on a random subset of the data and a random subset of the features. This approach aids in mitigating overfitting issues and enhancing the model's ability to generalize. [36]

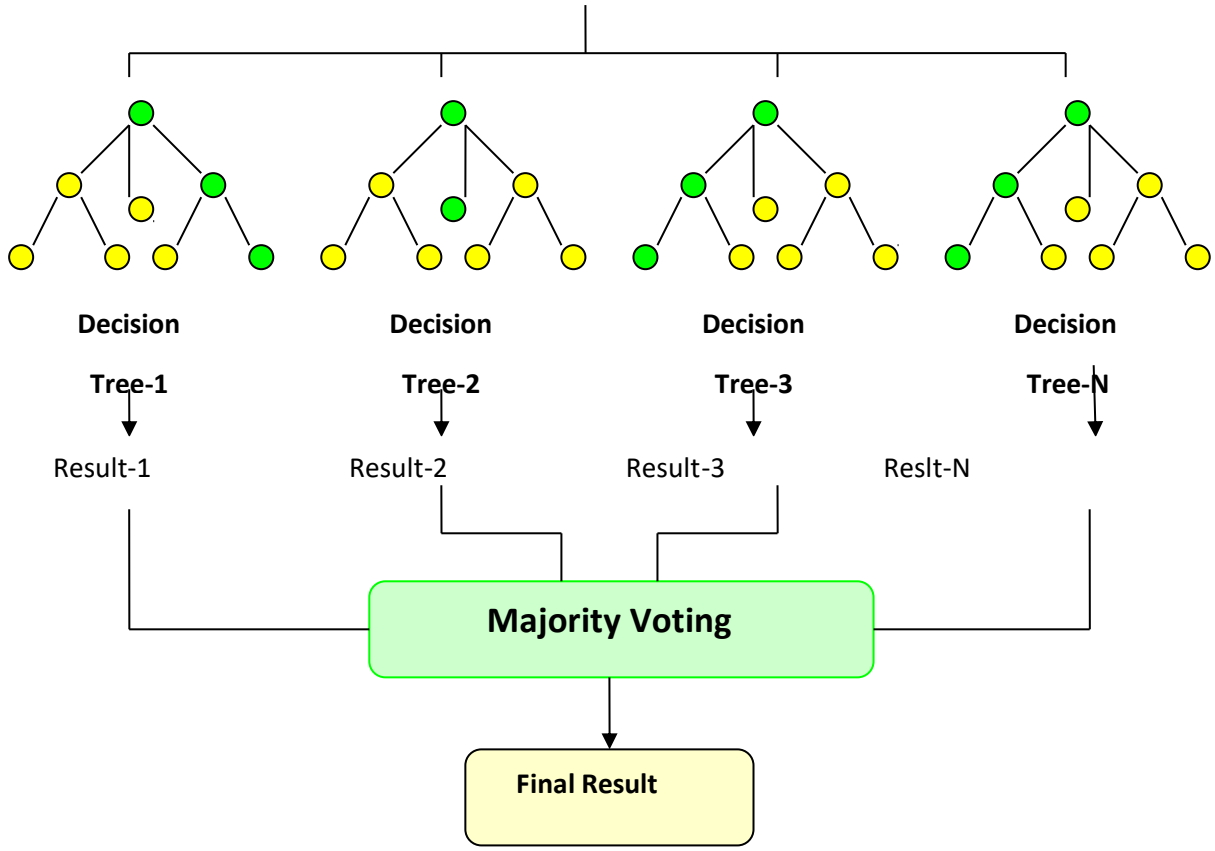

Figure 12: Random Forest meta-model

3.3.2.1 Random Forest in Classification and Regression:

Random forest shares similar hyperparameters with both decision trees and bagging classifiers. However, there is no need to combine a decision tree with a bagging classifier because the classifier class of random forest can be easily utilized. Additionally, random forest can handle regression tasks by utilizing the algorithm's regressor. When growing the trees in a random forest, additional randomness is introduced. Instead of searching for the most important feature when splitting a node, it searches for the best feature among a random subset of features. This leads to a wide diversity that generally results in a superior model. In a random forest classifier, only a random subset of features is considered by the algorithm for splitting a node. Furthermore, you can enhance the randomness of the trees by using random thresholds for each feature instead of searching for the best possible thresholds as a normal decision tree would do. [37]

3.3.2.2 Random Forest Algorithm:

Breiman proposed an integrated learning model in 2001, which utilizes the Decision Tree as the fundamental classifier known as Random Forest. This model generates multiple subsets of samples through the bootstrap method, constructs a Decision Tree for each subset, and then merges these Decision Trees to form a Random Forest. During the classification of a sample, the outcome is determined by a collective decision made by the Decision Trees. Scholars typically enhance the classifier's accuracy by starting from the classifier itself and minimizing the correlation between classifiers. In the Random Forest algorithm, the classification process involves a common error distribution among the base classifiers, resulting in a reduction of the classification impact. By considering the test characteristics and applying the rules of each randomly generated Decision Tree, the algorithm predicts and stores the expected outcome (target). The votes for each predicted outcome are then calculated, with the most voted outcome being considered as the final prediction from the Random Forest algorithm.

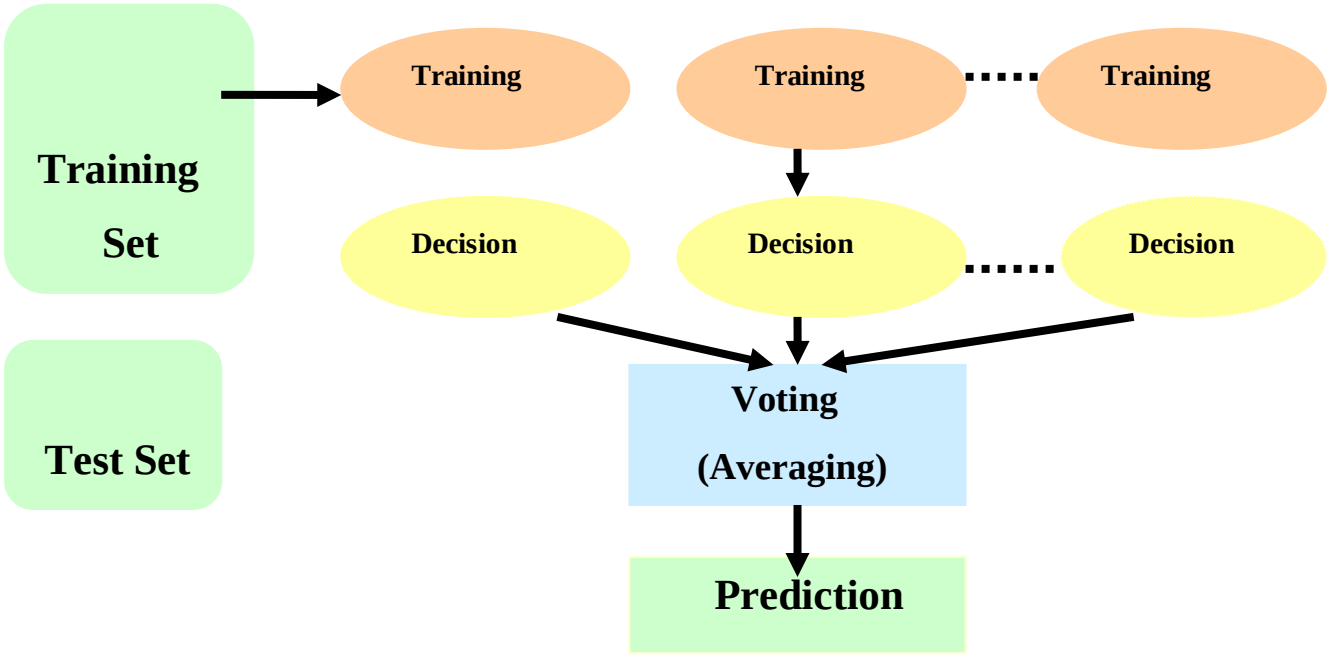

Figure 13: Diagram explains the working of the Random Forest meta-model

In the Random Forest algorithm, there are two steps:

One approach involves creating a Random Forest, while the other involves making predictions using the first step of the Random Forest classifier. The author initially presents the pseudo-code for building the Random Forest.

1. Select "K" features at random from the complete "m" features, where k << m.

2. Calculate the node "d" among the "K" features using the best split point.

3. Using the best division to divide the network into daughter nodes.

4. Repeat measures from 1 to 3 until the number of nodes 'n' has been reached.

5. Develop a forest to build the "n" number of trees by repeating steps 1 to 4 for "n" number of times.

After generating the Random Forest classifier in the subsequent step, we can proceed with making predictions. The pseudo-code for predicting with the Random Forest is as follows: It takes the test features and utilizes the rules from each randomly generated Decision Tree to predict the outcome and store the anticipated result (target). For each prediction objective, the votes are tallied. The final prediction from the Random Forest algorithm is determined by considering the highly voted predicted objective. [38]

3.3.2.3 Applications of Random Forest:

The random forest technique is a powerful machine-learning algorithm that has applications in a wide range of areas. In banking and the stock market, random forests are used to properly forecast stock values, assess credit risk, and detect fraudulent transactions. In the medical field, they are invaluable for identifying diseases, forecasting patient outcomes, and evaluating medical pictures to improve diagnostic accuracy.

Random forests boost e-commerce tremendously by improving recommendation systems, customer segmentation, and sales forecasting, allowing firms to better understand and serve their customers. Random forests play an important role in security and attack detection by identifying network intrusions, detecting malware, and evaluating suspicious activity to protect systems from cyber threats. Random forests' adaptability and durability make them an important tool for solving complicated issues and improving prediction performance in a wide range of industries. [39]

3.3.2.4 Advantages of Random Forest Algorithm:

- **High Accuracy:** By employing multiple decision trees, each trained on a unique subset of the data, Random Forest combines their predictions. This technique reduces the variability linked to individual trees, leading to more precise

predictions. It achieves this by averaging the predictions of these trees for regression tasks or by voting for classification tasks. When opting for an ensemble approach rather than a single decision tree model, the accuracy is generally improved.

- **Robustness to Noise**: Random Forest is known for its ability to effectively combine the predictions of multiple decision trees, making it resistant to noisy data. Since individual noisy data points are unlikely to impact all trees in the forest, the overall model performance is less likely to be affected. This robustness makes Random Forest a suitable choice for datasets with outliers or inherent noise.
- **Non-Parametric Nature**: The Random Forest approach is characterized by its non-parametric nature, as it does not make any assumptions about the data distribution at the root level or the relationship between the target variable and its characteristics. This adaptability allows Random Forest to be applied to diverse datasets and problem domains, enabling the identification of complex patterns without imposing strict constraints. [40]

3.3.2.5 Disadvantages of Random Forest Algorithm:

- **Computational Complexity:** Training a Random Forest model on a vast dataset or employing a significant number of trees in the forest can result in considerable computational costs. As each tree is trained individually, the process of aggregating their predictions demands substantial computing power. As a result, memory use may increase and training times may be extended, especially in systems with limited resources.
- **Memory Usage:** Random Forest models tend to use a lot of memory, particularly when working with big datasets or deeply rooted trees. The training data, feature splits, and leaf node predictions must all be stored in each decision tree in the forest. Memory utilization rises with the number of trees or the depth of the trees, which may cause memory limitations on some hardware systems.
- **Prediction Time**: Random Forest models can take longer to make predictions than certain other algorithms, even though they are more effective at training. This is particularly true for large datasets or models with a lot of trees. To get a final forecast, each observation must navigate through several decision trees in the

forest. This might lengthen the prediction time, especially for real-time or latency-sensitive applications. [41]

3.3.3 AdaBoost:

AdaBoost short for Adaptive Boosting is an ensemble learning used in machine learning for classification and regression problems. The main idea behind AdaBoost is to iteratively train the weak classifier on the training dataset with each successive classifier giving more weightage to the data points that are misclassified. The final AdaBoost model is decided by combining all the weak classifiers that have been used for training with the weightage given to the models according to their accuracies. The weak model which has the highest accuracy is given the highest weightage while the model which has the lowest accuracy is given a lower weightage. [42]

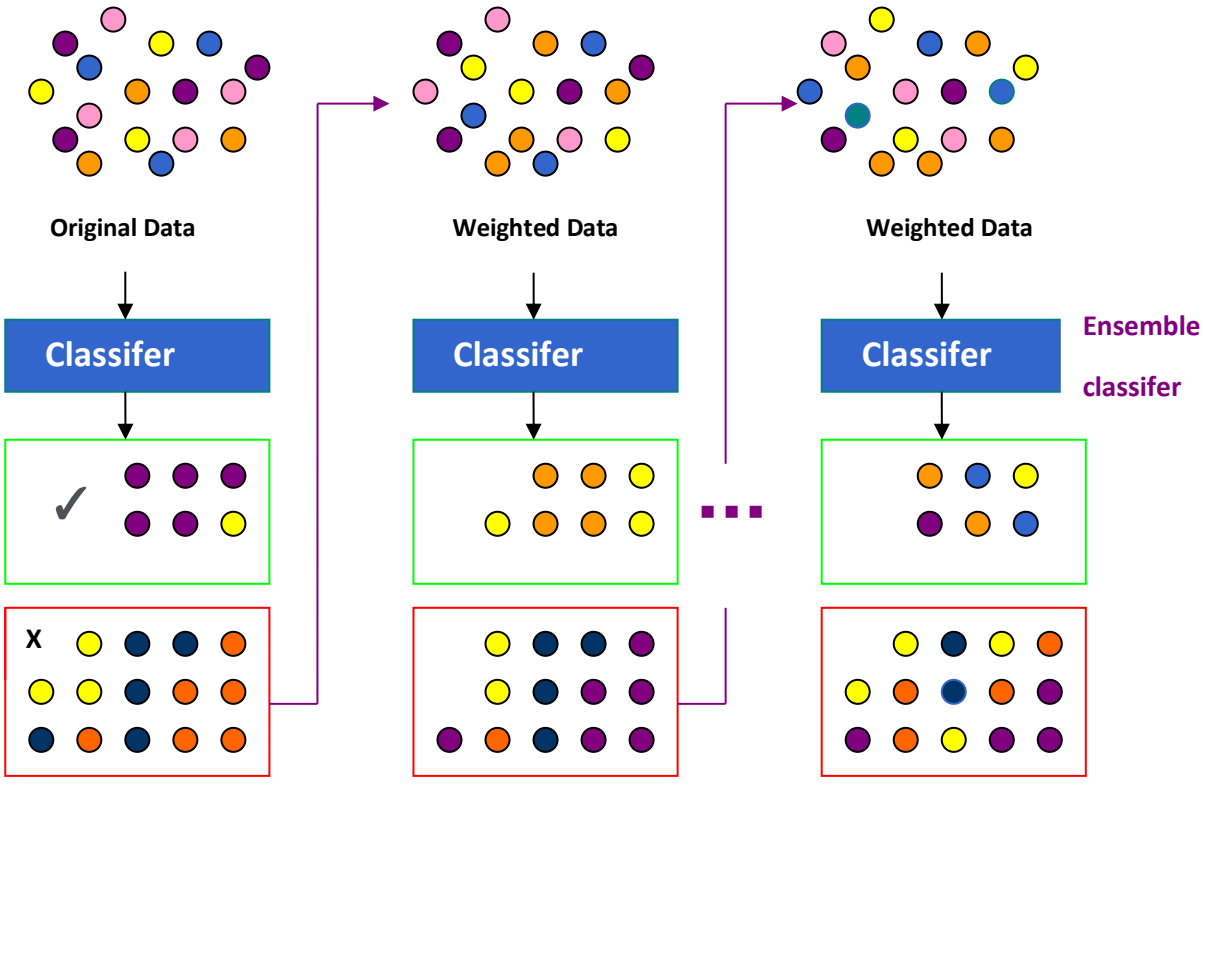

Figure 14: AdaBoost meta-model

3.3.3.1 How AdaBoost Works:

As usual with machine learning, the adaptation of (neural) weights plays a key role in modeling a learning process. The concatenation of different models allows AdaBoost to extend this principle by giving the individual classifiers different relevance for the overall prediction. This is known as soft voting. To achieve optimum weighting, a few individual steps must first be carried out. A prerequisite is the preparation of a suitable data set.

- **Establishing a fundamental algorithm :**
  it is essential to find an appropriate weak learner capable of making basic predictions regarding data points. Decision stumps, which are single-level decision trees, are widely used for this purpose. They divide the data set into distinct decision nodes. By comparing the Gini coefficients based on a single (binary) Feature, the most effective stump can be determined. This serves as the foundation for the ensemble model, which is then supplemented by additional algorithms.
- **Error computation :**
  Weightings are subsequently incorporated for every data point in the dataset to enable sampling and assess the accuracy of the classified target variable in each scenario. The inaccuracies identified through this method can then be assigned a greater weight for the following steps.
- **Classifier weighting :**
  AdaBoost can determine the relative importance of each classifier in the ensemble model by considering their error rates. The classifiers with higher accuracy or fewer errors have a greater influence on the overall prediction. This mathematical principle serves as a crucial foundation for the subsequent step.
- **Update sampling weights :**
  The initial equilibrium of the sample weights is now modified by considering the inaccurate data points from the first classifier. These erroneous predictions are given more weight in the following steps, while accurate predictions are assigned a lower weight. As a result, this adjustment enhances the training effect and progressively reduces the error function.
- **Iteration :**
  By employing a classifier that has been fine-tuned through error minimization and/or a dataset that has been refined using sample weights, it is possible to iterate the process multiple times. This iterative approach leads to the development of more accurate

models, which are eventually combined into an ensemble model, considering their respective weights or the principle of co-determination. AdaBoost can now utilize this methodology to classify new types of data, including previously unidentified data points. [43]

3.3.3.2 AdaBoost Algorithm:

Initialization

$D_1(i) = 1 / m, i = 1 ,..., m$

for $t = 1 ,...,T$:

$ht = L (LS, Dt)$

$\varepsilon t = \Sigma\ i : h(xi) \neq yi$ **Dt(i)**

$\alpha t = 1/2\ In\ (1-\varepsilon t/\ \varepsilon t)$

Pour i = 1 ..., m:

if $ht(xi) \neq yi$:

$\tilde{D}t+1\ (i) - \tilde{D}t(i)\ (1-\varepsilon t/\ \varepsilon t)$

Else , $\tilde{D}\ t+1(i)/(\Sigma i\ \tilde{D}t+1(i))$

End

$f\ T\ (x) = \Sigma\ (T ,t=1)\ \alpha t\ ht(x)$

$HT\ (x) = sign(fT\ (x))$ [44]

- **Initialize weights:** $D_1(i) = 1 / m, i = 1 ,...,m$
- D_1 is the distribution of weights assigned to the training examples at the first iteration (t=1).
- i is given an equal weight.
- m is the total number of training examples.
- $D_1(i) = 1 / m$ for each sample i, where m is the total number of samples.
- **Main Loop over Iterations t=1 to T**:
- $ht = L(LS, Dt)$: Train a weak classifier ht using a specified learning algorithm LS on the weighted data Dt .
- $\varepsilon t = \Sigma\ i : h(xi) \neq yi$ Dt(i) : Compute the weighted error εt where xi is a sample, ht(xi) is the prediction of ht for xi, and yi is the true label of xi
- $\alpha t = 1/2\ In\ (1-\varepsilon t/\ \varepsilon t)$: Calculate the weight αt assigned to the classifier ht based on its error rate εt.
- **Update Sample Weights** :
- For each training sample i=1 to m

- If the prediction of the weak classifier ht for sample xi(ht(xi))) is not equal to its true label yi (yi≠ht(xi)) , meaning it's misclassified:
- The weight $\tilde{D}t+1(i)$ for this misclassified sample is adjusted by multiplying the current weight $\tilde{D}t(i)$ by the factor $1-\varepsilon t/\varepsilon t$. This adjustment increases the weight of misclassified samples, making them more influential in the next round of training.
- Otherwise, if the prediction matches the true label:
- The weight $\tilde{D}t+1(i)$ for the correctly classified sample is adjusted differently. It's normalized by dividing it by the sum of all the updated weights $i\tilde{D}t+1(i)$. This normalization ensures that the sum of weights remains equal to 1, maintaining the validity of the probability distribution of weights.

- **Final Prediction fT(x) :**

- This equation calculates the final prediction fT(x) for a given input x by summing the predictions of all weak classifiers ht(x), each weighted by αt. The notation Σ(T,t=1)means summing over all iterations from t=1to T, where T is the total number of iterations.

- **Final Classification HT(x) :**

- Once we have fT(x), the final classification HT(x) for input x is determined by taking the sign of fT(x). If fT(x) is positive or zero, HT(x) is assigned a positive label (usually +1); if fT(x) is negative, HT(x) is assigned a negative label (usually -1).

Example:

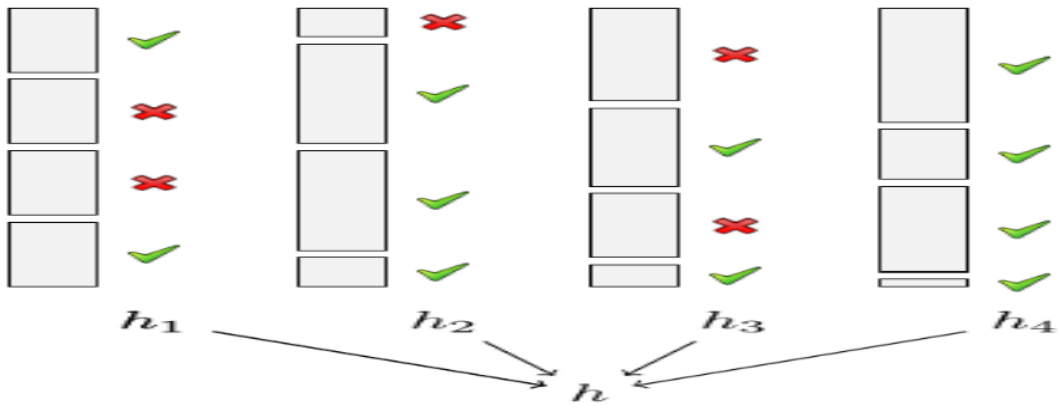

Figure15: Diagram explains how adaptive boosting works

The AI models are depicted as a collection of models {h1, h2, h3, h4} and their forecasts as a series of rectangles. The dimensions of the rectangles (forecasts) are directly related to their respective weights (i.e., their significance). h1 generates a set of forecasts. The discrepancies

between h1's predictions and the actual outcomes are assigned higher significance. Consequently, h2 should prioritize accurately forecasting the errors made by h1. (And the same principle applies to h3, h4...). In the end, 4 AI models correct each other's errors. The final ensemble classifier is called "h".

3.3.3.3 Applications of AdaBoost:

AdaBoost is a highly adaptable machine learning algorithm that has been widely utilized in various fields such as face detection, object recognition, and text classification. Its effectiveness shines through when faced with noisy data or datasets containing numerous irrelevant features. By concentrating on the most challenging samples through iterative processes and combining weak classifiers, AdaBoost boosts overall model accuracy and resilience. In the realm of face detection, AdaBoost is notably integrated into the Viola-Jones framework, significantly elevating real-time detection precision. When it comes to object recognition, AdaBoost aids in distinguishing between different objects in images by amplifying simple classifiers to create a more potent ensemble. In text classification, AdaBoost excels in tasks like spam detection and sentiment analysis by honing in on misclassified instances and refining the classifier. Additionally, in the field of security and attacks detection, AdaBoost is crucial for intrusion detection systems (IDS) and fraud detection. It enhances the identification of unauthorized access and anomalies in network traffic by iteratively focusing on subtle, misclassified patterns, improving the overall robustness of the detection system. Furthermore, AdaBoost extends its capabilities beyond classification to regression, where it reduces errors in predictive tasks, and clustering, where it aids in grouping similar data points by enhancing the assignment process through its iterative approach. This adaptability renders AdaBoost an invaluable tool across a wide array of machine learning applications, effectively managing complexity and enhancing predictive performance.

3.3.3.4 Advantages of AdaBoost Algorithm :

- **Versatile Application:** AdaBoost is adaptable across various machine-learning tasks, including classification and regression.
- **High Accuracy:** It often achieves superior accuracy compared to individual weak learners by prioritizing the improvement of misclassified instances in each iteration.
- **Computational Efficiency:** AdaBoost's computational efficiency is notable since it concentrates on enhancing performance by emphasizing misclassified instances**.**

- **Few Hyperparameters:** With minimal hyperparameters to tune, implementation is simplified, reducing the risk of overfitting. Effective with Class Imbalance: AdaBoost effectively manages class imbalances by focusing on misclassifications in each iteration. [45]

3.3.3.5 Disadvantages of the AdaBoost Algorithm:

- **Sensitivity to Noise:** It's vulnerable to noisy data and outliers, which can degrade its performance.
- **Slow Training:** Training can be sluggish, especially with numerous weak learners or features.
- **Potential for Overfitting:** Though less prone to overfitting than some algorithms, excessively weak learners can lead to overfitting.
- **Complex Final Model:** The model produced by AdaBoost can be intricate, posing challenges for interpretation compared to simpler models like decision trees. **[46]**

3.3.4 Comparative Analysis :

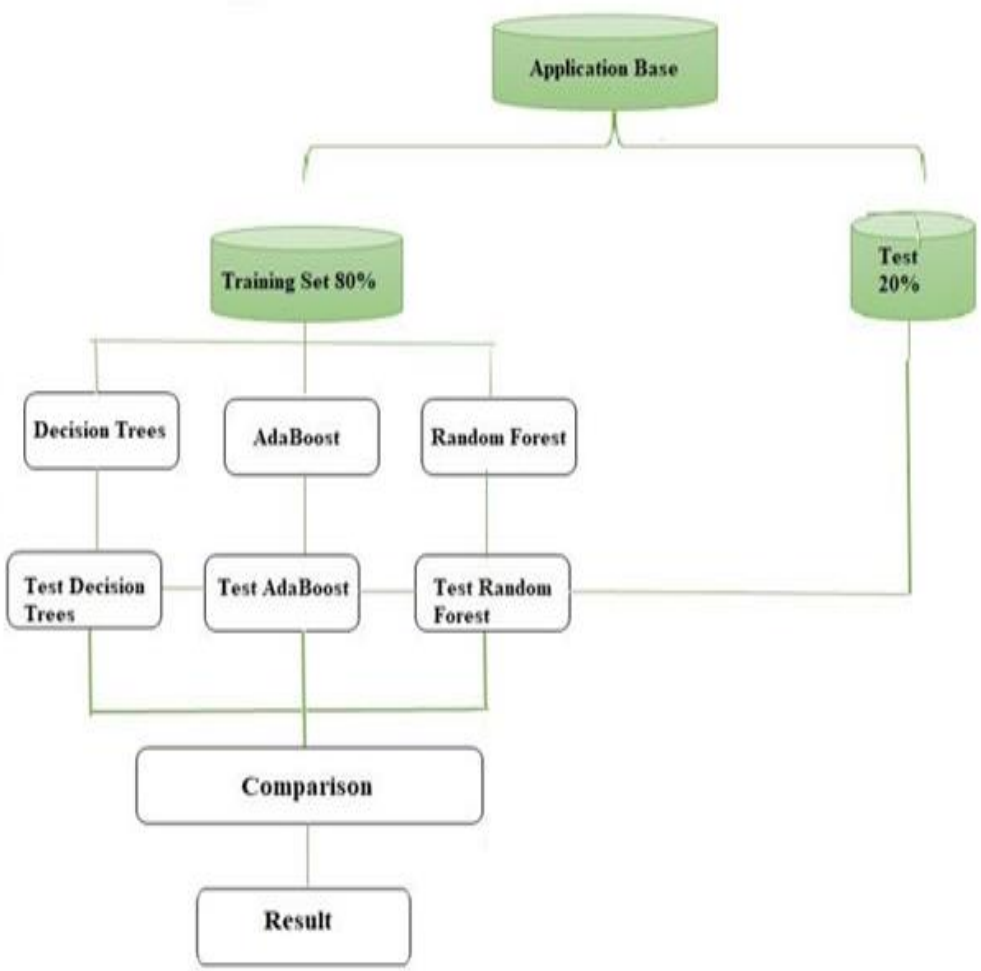

Figure 16: Diagram of the comparative study of models

This figure depicts the process of testing Distributed Denial of Service (DDoS) attacks with a Software-Defined Networking (SDN) dataset. [47]

- **Application Base :**
  - This SDN dataset contains information about network traffic, including DDoS attacks.
  - This Dataset has 104345 rows and 23 columns.

#	Features	Dtype
0	dt	Int64
1	switch	Int64
2	src	Object
3	dst	Object
4	pktcount	Int64
5	bytecount	Int64
6	dur	Int 64
7	dur_nsrc	Int64
8	tot_dur	Float64
9	flows	Int64
10	packetins	Int64
11	pktperflow	Int64
12	byte perflow	Int64
13	pktrate	Int64
14	pair flow	Int64
15	protocol	Object
16	port_no	Int64
17	tx_bytes	Int64
18	rx_bytes	Int64
19	tx_kbps	Int64
20	rx_kbps	Float64
21	tot_kbps	Float64
22	label	Int64

Table 01 : dataset (features)

- Target variable “label”: contains 1 (malicious) & 0 (benign)
- The dataset contains 3 categorical attributes + 20 numeric attributes
- Total number of DDoS attacks: 40784

Preparation and normalisation of the dataset:

- We start by loading it using a Python library (Pandas DataFrame) and then examine its basic structure and statistics. We visualize missing values using the Missingno library and then remove all rows containing null values.
- Next, we analyze the distribution of the target variable “label” by calculating the percentage of benign and malignant instances.
- We then separate the features into numerical and categorical types. Among numerical features, we further identify discrete and continuous features based on their number of unique values.
- After encoding, we normalize the features using functions to scale the data between 0 and 1.

- **Division into Training and Test Sets:**
  - To ensure the detection system's reliability, the dataset is separated into training and testing sets. A large chunk, 80%, is set out for training the detection algorithms, with the remaining 20% used as an independent test bed. This section enables robust model training and unbiased evaluation of performance.
- **Model Training :**
  - Three machine learning techniques are utilized to train the models for identifying DDoS attacks on the training set:
- **Decision Trees:** Used to determine if network communication is legitimate or malicious.
- **AdaBoost:** is a boosting technique that enhances accuracy by combining numerous weak classifiers into a strong classifier, making it appropriate for detecting network traffic anomalies.
- **Random Forest**: is an ensemble method that uses numerous decision trees to enhance the robustness and accuracy of DDoS attack detection.
- **Model Testing :**
  - Each trained model is then evaluated on the test set (20% of the data) to determine its capacity to detect DDoS attacks:
- **Comparison of Models :**
  - The models' performance is compared using parameters like as accuracy and precision to determine which model identifies DDoS assaults best in the SDN dataset.
- **Metric used :**

Accuracy is an evaluation metric for classification models. It is defined as the number of correct predictions divided by the total number of predictions.
Accuracy=(True positives+True negatives)/Total predictions

- **Results:** We will see the result of the three machine learning models Decision Trees, AdaBoost, and Random Forest that were trained and tested on a dataset of network traffic to detect DDoS attacks.

3.3.5 Conclusion:

In this chapter, we use three machine learning models Decision Trees, Random Forests, and AdaBoost to improve the detection of Distributed Denial of Service (DDoS) results. Each of these models brings unique benefits and qualities to the table, which substantially affects their ability to identify DDoS attacks. When comparing these models to measure their success in DDoS detection, multiple metrics are commonly used, including accuracy and computational efficiency. In the next chapter, we will look at the results of this comparison, throwing light on how each model performs against these criteria and providing insights into their relative strengths and shortcomings in DDoS detection scenarios.

Chapter 04:

Implementation & tests

4.1 Introduction:

In the current chapter, our focus will be on practically implementing and testing the distributed denial of service (DDoS) attack detection models that were previously explored in a theoretical manner. Specifically, we will provide a comprehensive explanation of how three machine learning models, namely decision trees, random forests, and AdaBoost, can be utilized to enhance the detection of DDoS attacks.

Python was chosen as our programming language due to its extensive libraries for machine learning, while Google Collaboratory was selected for its robust computational resources and collaborative features.

Next, we will describe the necessary steps for data preparation to train and evaluate the model. Additionally, we will compare the performance of these models in terms of accuracy and computational efficiency. This analysis will enable us to determine the most effective method for detecting DDoS attacks in various scenarios.

4.2 Python:

Python is a flexible and simple programming language defined as a very high-level, dynamic, object-oriented, general-purpose language that uses an interpreter and can be applied in a vast domain of applications. Supporting different programming styles, including structural and object-oriented, Python’s flexibility extends to its ability to use modular components from other languages; for example, you can write a program in C++ and import it into Python as a module. Conceived in the late 1980s by Guido van Rossum at CWI in the Netherlands, Python's implementation began in December 1989. It evolved from Python 2.0 in 2000, which introduced features like list comprehensions and garbage collection, to Python 3.0 in 2008, which addressed fundamental design flaws to enhance consistency and readability. Today, supported by a large, active community, Python is one of the most popular languages worldwide, renowned for its simplicity, versatility, and applications ranging from web development to scientific computing and artificial intelligence. [48]

Figure 17: Python

4.2.1 Python libraries for machine learning:

Python has a large ecosystem of machine-learning packages that support a wide range of activities, from data preprocessing to model deployment. These are the most popular Python packages used in our work:

Scikit-learn: is a Python library that supports conventional machine learning methods. It is built on top of two basic Python libraries: NumPy and SciPy. Scikit-learn is popular among machine learning developers because it supports both supervised and unsupervised learning techniques. This library can also be used for data analysis and mining purposes.

Keras: is a high-level neural networking API that works with TensorFlow, CNTK, and Theano libraries. It runs flawlessly on both CPU and GPU. It makes it very simple for Machine learning beginners to create a Neural Network.

Pandas: is a Python package mostly used for data analysis. Users must prepare the dataset before utilizing it to train machine learning. Pandas simplifies data extraction for developers because it was designed expressly for this purpose. It includes several tools in-depth data analysis, as well as high-level data structures.

Matplotlib: is a Python designed for data visualization. Developers use it to display data and patterns. It is a 2D charting toolkit that allows you to build 2D graphs and plots.

It includes the pyplot module, which is used to plot graphs and contains various features such as control line styles, font characteristics, axes formatting, and more. This library supports a variety of graphs and plots, including histograms, error charts, and bar charts. [49]

4.3 Google Collaboratory:

Google Colab has become an essential tool for data scientists and machine learning practitioners due to its effortless accessibility, collaborative features, robust computing resources, and seamless integration with Google services. As a cloud-based service on Jupyter

Notebooks, it enables users to develop deep learning applications in Python and provides a complimentary GPU processor, 12 GB of RAM, and over 100 GB of storage. Whether you are a student, an efficient environment to explore, develop, and share your data science and machine learning projects, with just a Google account. Embrace the power of Google Colab to unlock new possibilities in your data-driven journey. [50]

Figure 18: Google Collaboratory

4.3.1 How to use Google Colab:

- To begin working with Google Collaboratory Notebook, first log in to your Google account and then visit: https://colab.research.google.com which is the home page:

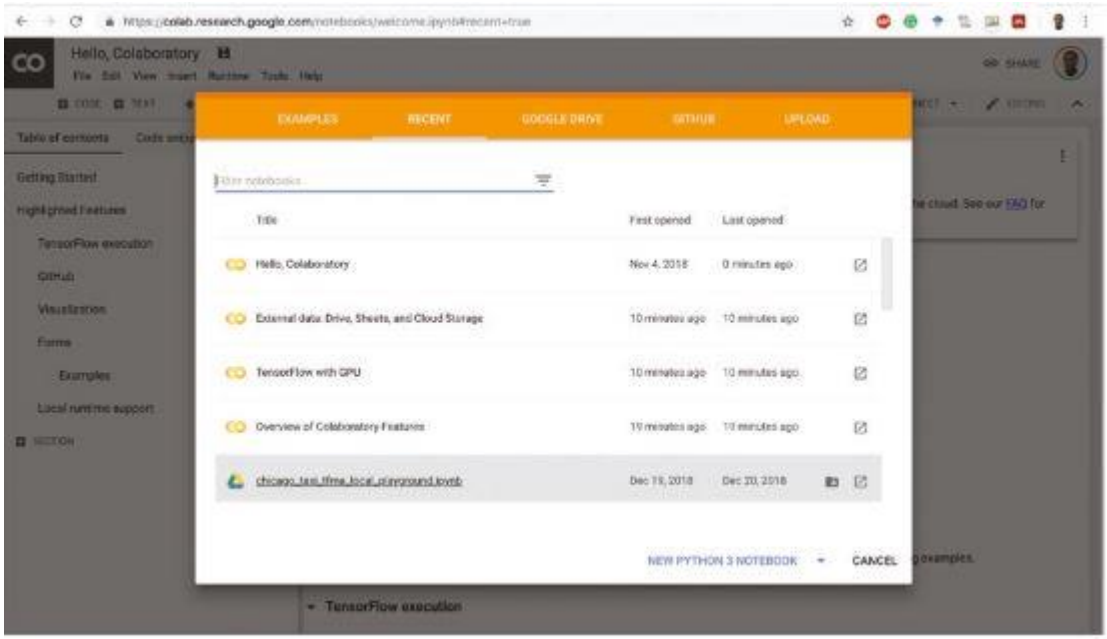

Figure 19: Home page

- Create a Collaboratory Notebook: you can create a new Jupyter Notebook by clicking Nouveau Notebook and you will find this page:

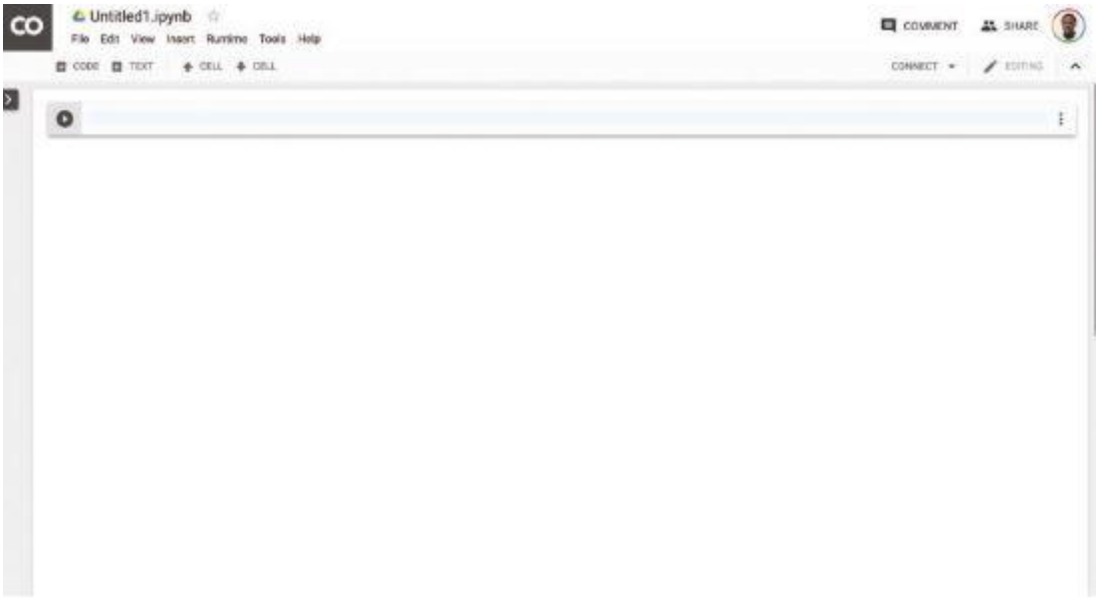

Figure 20: Python notebook

When you create a new notebook, it will generate a Jupyter notebook entitled Untitled1.ipynb and store it in your Google Drive in a folder called Colab Notebooks.

- **Storing Notebooks :**

  Notebooks on Colab are kept in Google Drive. They can be saved on GitHub or published as a GitHub Gist. They can also be downloaded to your local system. This figure represents the options for storing Jupyter notebooks running on Google Colab.

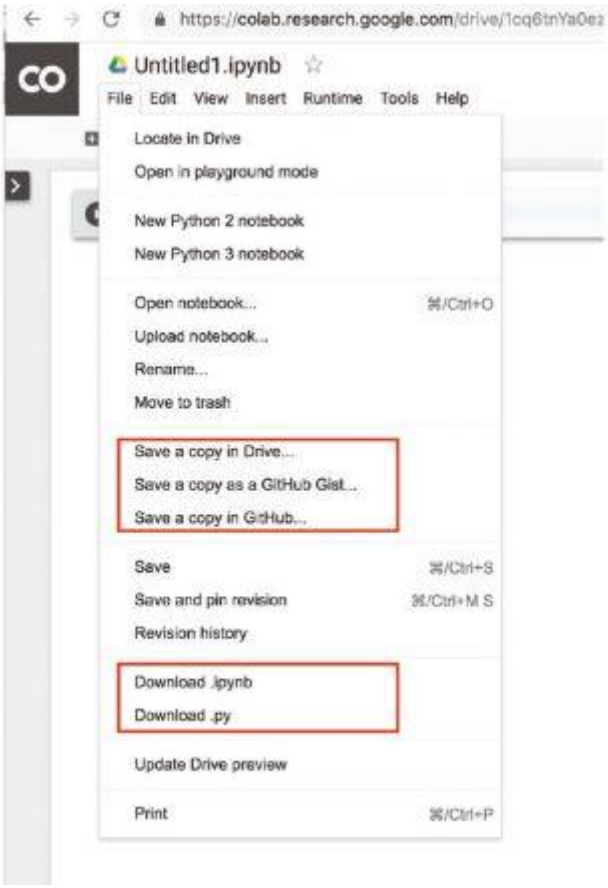

Figure 21: Storing Notebooks

4.4 Code snippets:

4.4.1 Loading the dataset:

```
# Load dataset
df = pd.read_csv('/content/drive/MyDrive/DDoS/SDN/dataset_sdn.csv')
print("This Dataset has {} rows and {} columns".format(df.shape[0], df.shape[1]))
df.info()
df.describe()
```

The provided code loads a dataset from a CSV file and displays its basic structure and summary statistics using the panda's module in Python:

- The code imports the panda's library for Python data manipulation and analysis. It loads a dataset from a specified file path by calling the pd.read_csv() method, which reads a CSV file and stores the contents in a pandas DataFrame. The file path identifies the dataset's location on Google Drive.

- The function determines the size and structure of a dataset by printing the number of rows and columns from the DataFrame's shape attribute. It then calls the info () method to return a summary of the DataFrame, highlighting any potential errors such as missing values or erroneous data types.
- The code uses the describe () method to provide descriptive statistics for the DataFrame, which provides insights into the data's distribution and variability and guides the following analytic stages.

4.4.2 Features separation:

```
# Feature separation
numerical_features = [feature for feature in df.columns if df[feature].dtypes != 'O']
print("The number of numerical features is",len(numerical_features),"and they are : \n",numerical_features)

categorical_features = [feature for feature in df.columns if df[feature].dtypes == 'O']
print("The number of categorical features is",len(categorical_features),"and they are : \n",categorical_features)

discrete_feature = [feature for feature in numerical_features if df[feature].nunique()<=15 and feature != 'label']
print("The number of discrete features is",len(discrete_feature),"and they are : \n",discrete_feature)

continuous_feature=[feature for feature in numerical_features if feature not in discrete_feature + ['label']]
print("The number of continuous_feature features is",len(continuous_feature),"and they are : \n",continuous_feature)
```

The code divides a dataset into numerical, categorical, discrete, and continuous features by selecting non-object and object data types, unique values (15 or less), and non-label columns. It also counts the features in each category and displays their names.

4.4.3 Separating input and output attributes:

```
# Separating input and output attributes
x = df.drop(['label'], axis=1)
y = df['label']
```

The code divides the input features (X) from the output labels (y) to properly format the data for training and assessing the model.

4.4.4 Train Test:

```
# Train-Test-Split
X_train, X_test, y_train, y_test = train_test_split(x, y,test_size=0.2)
print(X_train.shape, X_test.shape)
```

This code follows these steps:

- t splits the dataset (x and y) into training and testing sets.
- x is typically the feature data.

y is typically the target data (labels).

- The test_size=0.2 parameter specifies that 20% of the data should be reserved for testing, and the remaining 80% should be used for training.
- It then prints the shapes of the training and testing feature sets (X_train and X_test), which provides the dimensions of these arrays.

4.4.5 Importing models:

4.4.5.1 Decision Trees:

```
# Decision Tree Classifier
dt_clf = DecisionTreeClassifier(max_depth=5)
dt_clf.fit(X_train,y_train)
y_pred = dt_clf.predict(X_test)
accuracy = metrics.accuracy_score(y_test, y_pred)
Classifier_accuracy.append(accuracy*100)
print("Accuracy of Decision Tree Classifier : %.2f" % (accuracy*100) )
```

- **Initialization**: A Decision Tree Classifier is instantiated with a maximum tree depth of 5.
- **Training**: The model is trained using the fit method on the training dataset (X_train and y_train).
- **Prediction**: The trained model is then used to predict outcomes for the test dataset (X_test).
- **Evaluation**: The accuracy of the model is evaluated by comparing the predicted values (y_pred) with the true values (y_test) using metrics.accuracy_score.
- **Recording Results**: The accuracy percentage is appended to a list named Classifier_accuracy.
- **Output**: The accuracy of the classifier is printed to the console.

4.4.5.2 Random Forest:

```
# Random Forest
rf_clf = RandomForestClassifier(max_depth=2, random_state=0)
rf_clf.fit(X_train,y_train)
y_pred = rf_clf.predict(X_test)
accuracy = metrics.accuracy_score(y_test, y_pred)
Classifier_accuracy.append(accuracy*100)
print("Accuracy of Random Forest Classifier : %.2f" % (accuracy*100) )
```

- **Initialization**: A Random Forest Classifier is instantiated with a maximum tree depth of 2 and a fixed random state of 0 for reproducibility.
- **Training**: The model is trained using the fit method on the training dataset (X_train and y_train).
- **Prediction**: The trained model is then used to predict outcomes for the test dataset (X_test).
- **Evaluation**: The accuracy of the model is evaluated by comparing the predicted values (y_pred) with the true values (y_test) using metrics.accuracy_score.
- **Recording Results**: The accuracy percentage is appended to a list named Classifier_accuracy.
- **Output**: The accuracy of the classifier is printed to the console.

4.4.5.3 AdaBoost:

```
# AdaBoost Classifier
ab_clf = AdaBoostClassifier(n_estimators=100, random_state=0)
ab_clf.fit(X_train, y_train)
y_pred = ab_clf.predict(X_test)
accuracy = metrics.accuracy_score(y_test, y_pred)
Classifier_accuracy.append(accuracy*100)
print("Accuracy of AdaBoost Classifier : %.2f" % (accuracy*100) )
```

- **Initialization**: An AdaBoost Classifier is instantiated with 100 estimators and a fixed random state of 0 for reproducibility.
- **Training**: The model is trained using the fit method on the training dataset (X_train and y_train).
- **Prediction**: The trained model is then used to predict outcomes for the test dataset (X_test).
- **Evaluation**: The accuracy of the model is evaluated by comparing the predicted values (y_pred) with the true values (y_test) using metrics.accuracy_score.

- **Recording Results** : The accuracy percentage is appended to a list named Classifier_accuracy.
- **Output**: The accuracy of the classifier is printed to the console.

4.4.6 Comparison:

```
# Comparative analysis of models
Classifier_names = ["Decision Tree", "Random Forest", "AdaBoost"]
df_clf = pd.DataFrame()
df_clf['name'] = Classifier_names
df_clf['Accuracy'] = Classifier_accuracy
df_clf = df_clf.sort_values(by=['Accuracy'], ascending=False)
df_clf.head(3)
```

The DataFrame is generated by the code, containing classifier names and corresponding accuracies. The classifiers are then arranged in descending order based on accuracy, and the top three classifiers are presented.

4.4.7 Results and Discussion :

4.4.7.1 Results:

We implemented three ensemble learning models, AdaBoost, Random Forest, and Decision Trees. these models were trained and tested on a dataset to evaluate their performance detecting Distributed Denial of Service (DDoS) attacks.

- First, we tested the models with default parameters, which revealed that AdaBoost had the highest accuracy followed by Random Forest and Decision Trees.

Classifier	**Accuracy**
Decision Tree	**89,13%**
Random Forest	**97,45%**
AdaBoost	**99,95%**

Table 02: Classifier Accuracy with Default Parameters

The outcomes of our trials have been condensed in the table above. AdaBoost exhibited the greatest precision among the three models, achieving a remarkable 99.95% accuracy. Random Forests followed this at 97.45%, and Decision Trees at 89.13%.

- Then, we changed the parameter of max_depth for the two classifiers, while keeping the impact accuracy, we conducted experiments under three different max_depth configurations:
    - **The first case:** when the max_depth of decision trees= 2 and random forest = 2

Classifier	Accuracy
AdaBoost	99,94%
Random Forest	89,98%
Decision Trees	89,24%

Table 03: Classifier Accuracy with the same max_depth=2

- **The second case:** when the max_depth is Decision trees: max depth = 2 and Random Forest: max_depth=5:

Classifier	Accuracy
AdaBoost	99,92%
Random Forest	97,25%
Decision Trees	89,22%

Table 04: Classifier Accuracy with max_depth=5 for random forest and max_depth=2 for decision trees

- **The last case:** max_depth of Decision Tree = 5, Random Forest = 2

Classifier	Max_depth
AdaBoost	**99,95%**
Random Forest	**97,48%**
Decision trees	**89,54%**

Table 05: Classifier Accuracy with max_depth=2 for random forest and max_depth=5 for decision trees

- Finally, we also measured the prediction and test times for each classifier.

Classifier	Predict time (s)	Training time (s)
AdaBoost	**0,375713**	**20,349745**
Random Forest	**0,146402**	**8,45000**
Decision Tres	**0,003558**	**0,311440**

Table 06: Prediction time and Training time

4.4.8 Discussion:

We tested three machine learning models Random Forest, Decision Trees, and AdaBoost for DDoS attack detection and found that AdaBoost performs the best. AdaBoost is a highly accurate model that enhances weak classifiers by focusing on errors made by previous models. It assigns higher weights to misclassified instances, forcing subsequent classifiers to focus on harder cases. This iterative procedure produces a strong model that excels in recognizing intricate patterns within the data, making it more effective in identifying tiny differences in DDoS attacks. Although Decision Trees and Random Forests work well for many applications, they may not be as responsive to the intricate patterns of DDoS attacks. Random Forest, while highly accurate, relies on aggregating multiple decision trees to reduce overfitting but may miss finer details. Decision Trees, the simplest model, showed considerable accuracy but lagged behind ensemble methods due to overfitting in complex datasets like network traffic detection for DDoS attack detection. AdaBoost's greater accuracy and adaptability make it more effective than the other algorithms in this context.

4.5 Conclusion:

Throughout this chapter, we have delved into the practical implementation and testing of various machine learning models for detecting distributed denial of service (DDoS) attacks. Specifically, we have implemented and compared the performance of decision trees, random forests, and AdaBoost using Python libraries like Scikit-learn and Keras within a Google Collaboratory environment. Our experimental results clearly indicate that both the AdaBoost and Random Forest algorithms outperform decision trees significantly when it comes to detecting DDoS attacks. Random forests, with their ability to mitigate overfitting through the use of diverse trees trained on random subsets of data and features, have demonstrated enhanced robustness and accuracy. Similarly, AdaBoost, through its iterative improvement of

weak models and emphasis on misclassified instances, has exhibited excellent performance in identifying anomalous network traffic behaviors associated with DDoS attacks.

On the other hand, decision trees, despite being easy to interpret and quick to execute, have proven to be less effective in this particular context due to their tendency to overfit training data and lack of generalization across test sets.

In conclusion, we highly recommend the utilization of ensemble models such as Random Forest and AdaBoost for DDoS attack detection due to their superior accuracy and capability to handle complex data. This analysis underscores the significance of selecting appropriate algorithms based on specific detection requirements and the nature of the data to ensure effective protection against cyberattacks.

General Conclusion

To conclude, the increasing digitalization of our society has made IT security and network security crucial priorities. While traditional defense methods were effective against older attacks, they are no longer enough to combat modern threats like DDoS attacks. These attacks can result in financial losses, damage to reputation, and significant disruptions to businesses and organizations.

Fortunately, the emergence of advanced machine learning techniques provides a promising solution to prevent and defend against DDoS attacks. Classification algorithms such as AdaBoost, random forests, and decision trees have proven to be effective in detecting and mitigating these threats. Rigorous testing on real-world datasets has shown that these algorithms can recognize and respond to the complex attack patterns associated with DDoS attacks.

Specifically, our test results have demonstrated that AdaBoost, with its adaptive approach and focus on correcting classification errors, surpasses random forests and decision trees in terms of accuracy. This highlights the effectiveness of AdaBoost in identifying and neutralizing DDoS attacks, thereby contributing to the proactive protection of IT infrastructures.

In summary, the integration of machine learning into cybersecurity strategies represents a significant advancement in combating present and future threats. By utilizing advanced techniques like AdaBoost, random forests, and decision trees, we can enhance the resilience of our digital systems, safeguard our assets, and ensure the secure operation of critical services.

Bibliography

[1] R. W. Christoph Hopher. [Online]. Available: https://rvs.unibe.ch/teaching/cn%20applets/IP_Spoofing/IP%20Spoofing.pdf.

[2] D. Clavin. [Online]. Available: https://hempsteadny.gov/635/Famous-Phishing-Incidents-from-History .

[3] OVHcloud, "ovhcloud," [Online]. Available: https://www.ovhcloud.com/fr/security/anti-ddos/ddos-definition/.

[4] D. K. Bhattacharyya, ddos attacks evolution detection prevention reaction and tolerance, Boca Raton London New York: Taylor & Francis Group, an informa business, 27 avril .

[5] A. Huged, "Quora," 2023. [Online]. Available: https://fr.quora.com/.

[6] G. Smith, "stationx," 10 april 2024. [Online]. Available: https://www.stationx.net/ddos-statistics/.

[7] K. Poireault, "infosecurity-magazin," 2024. [Online]. Available: https://www.infosecurity-magazine.com/news/environmental-websites-ddos-surge/.

[8] "IBM," [Online]. Available: https://www.ibm.com/topics/supervised-learning.

[9] "tableau," [Online]. Available: https://www.tableau.com/learn/articles/top-machine-learning-methods.

[10] "DataCamp," [Online]. Available: https://www.datacamp.com/blog/supervised-machine-learning.

[11] D. S. G. S. P. R. a. V. Z. Vasilev, Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow., 2019.

[12] "Masters in Data Science," [Online].

[13] R. S. a. S. - Alsabti K., A Decision Tree Classifierfor Large Datasets, Conference on Knowledge Discovery and Data Mining, 1998.

[14] L. Rouvière, "Machine Learning," [Online]. Available: https://lrouviere.github.io/machine_learning/.

[15] [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network.

[16] "The NatureofCode," [Online]. Available: https://natureofcode.com/neural-networks/.

[17] S. V. P. S. B. Mayuri Thorat1, Artificial Neural Network: International Research Journal of Engineering and Technology.

[18] Q. jafeery, "linkedin," [Online]. Available: https://www.linkedin.com/pulse/deep-neural-network-qasim-jaffery.

[19] "Scribd," [Online]. Available: https://fr.scribd.com/user/721394809/shreyagoudar06.

[20] s. shah, "analyticsvidhya," [Online]. Available: definition: https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/.

[21] G. M. G. V. A. Shanthini, Deep Convolutional Neural Network for The Prognosis of Diabetic Retinopathy.

[22] J. Z. Feifan Cheng, "sciencedirect," [Online]. Available: https://www.sciencedirect.com/topics/chemical-engineering/recurrent-neural-network.

[23] "/machine-learning-notebook," [Online]. Available: https://calvinfeng.gitbook.io/machine-learning-notebook.

[24] U. P. M. (. M. FARHAD MORTEZAPOUR SHIRI1, *A Comprehensive Overview and Comparative Analysis on Deep.*

[25] "AWS," [Online]. Available: https://aws.amazon.com/fr/what-is/reinforcement-learning/#:~:text=Reinforcement%20learning%20(RL)%20is%20a,use%20to%20achieve%20their%20goals. .

[26] M. J. Alok Kumar, ensemble learning for Ai developers.

[27] Y. Freund and R. E. Schapire, Boosting foundations and algorithms.

[28] "CFI," [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/.

[29] "CFI," [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/.

[30] B. Quinto, "Next-Generation Machine Learning with Spark," Apress, Year: 2020, 2020.

[31] C. Staff, 2023. [Online]. Available: https://www.coursera.org/articles/decision-tree-machine-learning.

[32] J. F. Leo Breiman, "geeksforgeeks," 2022. [Online]. Available: https://www.geeksforgeeks.org/decision-tree-algorithms/.

[33] G. Ritschard, "geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/decision-tree-algorithms/#id3-iterative-dichotomiser-3 .

[34] J. H. Friedman, "doiorg," [Online]. Available: https://doi.org/10.1214/aos/1176347963.

[35] "geeksforgeeks," [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/decision-

tree/#:~:text=Decision%20trees%20are%20used%20for,and%20continuous%20variable%20decision%20trees..

[36] "medium," 2023. [Online]. Available: https://utsavdesai26.medium.com/mastering-random-forest-algorithm-a-step-by-step-learning-guide-f7abf2420a55.

[37] N. Donges, "builtin," [Online]. Available: https://builtin.com/data-science/random-forest-algorithm.

[38] G. Wanyama, 2020. [Online]. Available: https://dspace.ut.ee/items/c0a9c9f9-8f71-4e32-9143-4876570a9fd8 .

[39] N. Donges. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm#applications.

[40] "geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/what-are-the-advantages-and-disadvantages-of-random-forest/.

[41] "geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/what-are-the-advantages-and-disadvantages-of-random-forest/.

[42] "geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/implementing-the-adaboost-algorithm-from-scratch/.

[43] T. Filzinger, 2023. [Online]. Available: https://konfuzio.com/en/adaptive-boosting/.

[44] "MasterM-Parcoursrecherche".

[45] "quora," [Online]. Available: https://www.quora.com/What-are-the-pros-and-cons-of-AdaBoost-How-useful-is-it.

[46] "quora," [Online]. Available: https://www.quora.com/What-are-the-pros-and-cons-of-AdaBoost-How-useful-is-it.

[47] kaggle, "dataset sdn ddos," [Online]. Available: https://www.kaggle.com/datasets/aikenkazin/ddos-sdn-dataset.

[48] T. Software, TIOBE Programming Community Index Python.

[49] "free learning platform for better future," [Online]. Available: https://www.javatpoint.com/best-python-libraries-for-machine-learning#:~:text=Scikit%2Dlearn%20is%20a%20Python,supervised%20and%20unsupervised%20learning%20algorithms..

[50] N. T. [Online]. Available: - https://www.linkedin.com/pulse/google-colab-powerful-tool-machine-learning-nithinbharathi-t.

[51] [Online]. Available: https://rvs.unibe.ch/teaching/cn%20applets/IP_Spoofing/IP%20Spoofing.pdf.

[52] [Online]. Available: https://hempsteadny.gov/635/Famous-Phishing-Incidents-from-History.

[53] J. K. K. Dhruba Kumar Bhattacharyya, DDoS attacks: evolution, detection, prevention, reaction, and tolerance, 15 juil. 2016.

[54] [Online]. Available: https://www.ovhcloud.com/fr/security/anti-ddos/ddos-definition/.

[55] A. Hughes. [Online]. Available: https://fr.quora.com/.

[56] G. Smith, April 10, 2024 / By . [Online]. Available: https://www.stationx.net/ddos-statistics/.

[57] K. Poireault. [Online]. Available: https://www.infosecurity-magazine.com/news/environmental-websites-ddos-surge/.

[58] [Online]. Available: https://www.ibm.com/topics/supervised-learning.

[59] [Online]. Available: https://www.tableau.com/learn/articles/top-machine-learning-methods.

[60] [Online]. Available: https://www.datacamp.com/blog/supervised-machine-learning.

[61] [Online]. Available: https://www.ibm.com/topics/supervised-learning.

[62] D. S. G. S. P. R. a. V. Z. I. Vasilev, Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow., Packt Publishing Ltd, 2019.

[63] T. H. a. R. T. J. Friedman, The elements of statistical learning, Springer series in statistics New York, 2001.

[64] [Online]. Available: https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree/.

[65] R. S. a. S. V. C. - Alsabti K., A Decision Tree Classifierfor Large Datasets, Conference on Knowledge Discovery and Data Mining(KDD-98), August 1998.

[66] [Online]. Available: : chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://lrouviere.github.io/classif_sup/forets.pdf.

[67] [Online]. Available: definition: https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/.

[68] G. M. G. V. A. Shanthini, Deep Convolutional Neural Network for The Prognosis of Diabetic Retinopathy, 2022.

[69] [Online]. Available: https://aws.amazon.com/fr/what-is/reinforcement-learning/#:~:text=Reinforcement%20learning%20(RL)%20is%20a,use%20to%20achieve%20their%20goals. .

[70] [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network .

[71] S. V. P. S. B. A brief study Mayuri Thorat1, "Artificial Neural Network: International Research Journal of Engineering and Technology (IRJET)".

[72] D. D. K. M. G. G. Ravi Agnihotri, ": International Research Journal of Engineering and Technology (IRJET) Introduction of Neural Networks –A Trending Technology".

[73] [Online]. Available: https://fr.scribd.com/user/721394809/shreyagoudar06.

[74] J. Z. i. C. A. C. E. Feifan Cheng, 2019. [Online]. Available: https://www.sciencedirect.com/topics/chemical-engineering/recurrent-neural-network.

[75] S. Podel, 28 Augest 2023 . [Online]. Available: https://medium.com/.

[76] M. J. Alok Kumar, ensemble learning for Ai developers, 2020.

[77] Y. Freund and Schapire, Robert E, Boosting foundations and algorithms, 2012.

[78] [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/.

[79] [Online]. Available: https://www.coursera.org/articles/decision-tree-machine-learning.

[80] [Online]. Available: - https://www.geeksforgeeks.org/decision-tree-algorithms/#id3-iterative-dichotomiser-3.

[81] J. F. R. O. C. J. S. Leo Breiman, Classification and Regression Tree.

[82] [Online]. Available: https://www.geeksforgeeks.org/decision-tree-algorithms/#id3-iterative-dichotomiser-3.

[83] G. Ritschard, "CHAID and Earlier Supervised Tree Methods". Contemporary Issues in Exploratory Data Mining ., New York: Routledge: 48–74: the Behavioral Sciences, McArdle, J.J. And G. Ritschard (Eds), 2013.

[84] J. H. F. "Multivariate Adaptive Regression Splines." Ann. Statist. 19, 1991.

[85] [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/decision-tree/#:~:text=Decision%20trees%20are%20used%20for,and%20continuous%20variable%20decision%20trees..

[86] [Online]. Available: https://builtin.com/data-science/random-forest-algorithm#applications.

[87] [Online]. Available: https://utsavdesai26.medium.com/mastering-random-forest-algorithm-a-step-by-step-learning-guide-f7abf2420a55.

[88] [Online]. Available: https://dspace.ut.ee/items/c0a9c9f9-8f71-4e32-9143-4876570a9fd8 .

[89] [Online]. Available: Advantages of Random Forest Algorithm: https://www.geeksforgeeks.org/what-are-the-advantages-and-disadvantages-of-random-forest/.

[90] [Online]. Available: https://www.geeksforgeeks.org/implementing-the-adaboost-algorithm-from-scratch/.

[91] [Online]. Available: https://konfuzio.com/en/adaptive-boosting/.

[92] [Online]. Available: https://www.quora.com/What-are-the-pros-and-cons-of-AdaBoost-How-useful-is-it.

[93] [Online]. Available: - https://www.linkedin.com/pulse/google-colab-powerful-tool-machine-learning-nithinbharathi-t.

[94] TIOBE Programming Community Index Python, 2011.

[95] [Online]. Available: https://www.javatpoint.com/best-python-libraries-for-machine-learning#:~:text=Scikit%2Dlearn%20is%20a%20Python,supervised%20and%20unsupervised%20learning%20algorithms..

[96] jj, uuh, huju: jii, hui.

[97] "SpringerLink," 2023. [Online]. Available: https://link.springer.com/article/10.1007/s10462-023-10466-8.

[98] D. D. K. M. G. G. A Trending Technology. Ravi Agnihotri], International Research Journal of Engineering and Technology (IRJET) Introduction of Neural Networks.

[99] S. Podel, "medium," [Online]. Available: https://medium.com/.

[100] "geeksforgeeks," [Online]. Available: https://www.geeksforgeeks.org/what-are-the-advantages-and-disadvantages-of-random-forest/.