الجمهورية الجزائرية الديمقراطية الشعبية

Republique Algerienne Democratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة الشهيد الشيخ العربي التبسي - تبسة

Université Echahid Cheikh Larbi Tébessi – Tébessa

Faculté des Sciences et de la Technologie

Département de d'Électronique et Télécommunications

MEMOIRE

Présenté pour l'obtention du **diplôme** de **Master Académique**

Filière : Télécommunications

Spécialité : Réseaux et Télécommunications

Par :

- **Aya ARAB**
- **Yamina SOUAHI**

THEME

AI in Cybersecurity: The Design of a Learning Model for the Detection of Zero-Day Attacks

Présenté et évalué, le 12 /06 / 2024, par le jury composé de :

Nom et Prénom		**Grade**	**Qualité**
M.	Abdallah MERAOUMIA	Prof.	Président
Mme.	Chaima AOUICHE	MCB	Rapporteuse
M.	Mohammed SAIGAA	MCB	Examinateur

Promotion : 2023/2024

Abstract

The proliferation and utilization of the Internet and mobile applications have expanded cyberspace, rendering it more susceptible to automated and persistent cyberattacks. Cybersecurity techniques provide enhancements in security measures to detect and counter these attacks. However, traditional security systems are no longer sufficient, as cybercriminals are adept at evading them. Attackers and malicious software evolve daily, necessitating the continuous development of Intrusion Detection Systems (IDS) to effectively combat these threats.

This study aims to propose a learning model that enable to evaluate the long-term performance of IDS and detect zero-day attacks using machine learning (ML) and deep learning (DL) algorithms. To achieve this, three progressive datasets have been selected: CIC-IDS 2017(used for training), CSE-CIC-IDS 2018 and CIC-DDoS2019 (used for testing), which are widely used to develop IDS systems. Further analysis was performed on these datasets and binary classification was conducted using decision tree (DT), random forest (RF), support vector machine (SVM) naïve bayes (NB), artificial neural network (ANN) and deep neural network (DNN). Our experiments achieved good performance results and demonstrated more resistant to overfitting using the training dataset. However, the performance significantly dropped on the testing dataset.

Keywords: Cyber Security, Zero-Day Attacks, Intrusion Detection system (IDS), Artificial Intelligence (AI), Machine Learning (ML), Deep learning (DL).

Résumé

La prolifération et l'utilisation d'Internet et des applications mobiles ont étendu le cyberespace, le rendant plus vulnérable aux cyberattaques automatisées et persistantes. Les techniques de cybersécurité apportent des améliorations dans les mesures de sécurité pour détecter et contrer ces attaques. Cependant, les systèmes de sécurité traditionnels ne sont plus suffisants, car les cybercriminels sont experts dans leur contournement. Les attaquants et les logiciels malveillants évoluent quotidiennement, nécessitant un développement continu des systèmes de détection d'intrusion (IDS) pour lutter efficacement contre ces menaces.

Cette étude vise à proposer un modèle d'apprentissage pour évaluer la performance à long terme des IDS et détecter les attaques zero-day en utilisant des algorithmes d'apprentissage automatique (ML) et d'apprentissage profond (DL). Pour ce faire, trois ensembles de données progressifs ont été sélectionnés : CIC-IDS 2017 (utilisé pour l'entraînement), et CSE-CIC-IDS 2018 et CIC-DDoS2019 (utilisés pour les tests), qui sont largement utilisés pour développer des systèmes IDS. Une analyse approfondie a été réalisée sur ces ensembles de données, et une classification binaire a été effectuée en utilisant l'arbre de décision (DT), la forêt aléatoire (RF), la machine à vecteurs de support (SVM), le Naïve Bayes (NB), le réseau de neurones artificiels (ANN) et le réseau de neurones profonds (DNN). Nos expériences ont obtenu de bons résultats de performance et ont montré une plus grande résistance au surapprentissage avec l'ensemble de données d'entraînement. Cependant, la performance a considérablement diminué sur les ensembles de données de test.

Mots clés : Cybersécurité, Attaques zero-day, Système de détection des intrusions (IDS), L’intelligence artificielle (AI), L’apprentissage automatique, L’apprentissage profond.

ملخص

ان انتشار واستخدام الإنترنت والتطبيقات المحمولة قد وسّع الفضاء الإلكتروني، مما جعله أكثر عرضة للهجمات الإلكترونية الآلية والمستمرة. تقدم تقنيات الأمن السيبراني تحسينات في إجراءات الأمان للكشف عن هذه الهجمات ومكافحتها. ومع ذلك، لم تعد الأنظمة الأمنية التقليدية كافية، حيث أصبح مجرمو الإنترنت ماهرين في التهرب منها. يتطور المهاجمون والبرمجيات الخبيثة يوميًا، مما يستدعي التطوير المستمر لأنظمة كشف التسلل (IDS) لمكافحة هذه التهديدات بفعالية.

تهدف هذه الدراسة إلى اقتراح نموذج تعليمي لتقييم الأداء طويل المدى لمعرفات الهوية ((IDSواكتشاف هجمات اليوم صفر باستخدام خوارزميات التعلم الآلي ((ML والتعلم العميق (DL) وللقيام بذلك، تم اختيار ثلاث مجموعات بيانات تقدمية: (CIC-IDS 2017المستخدمة للتدريب)، وCSE-CIC-IDS 2018 و (DDoS2019 (-CICالمستخدمة للاختبار)، والتي تستخدم على نطاق واسع لتطوير أنظمة IDS. تم إجراء تحليل موسع على مجموعات البيانات هذه، وتم إجراء تصنيف ثنائي باستخدام شجرة القرار (DT)والغابات العشوائية (RF)، وآلة ناقل الدعم (SVM)، وسذاجة بايز (NB)، والشبكة العصبية الاصطناعية (ANN) والشبكة العصبية العميقة (دي إن إن). حققت تجاربنا نتائج أداء جيدة وأظهرت مقاومة أكبر للتركيب الزائد مع مجموعة بيانات التدريب. ومع ذلك، انخفض الأداء بشكل ملحوظ في مجموعات بيانات الاختبار.

الكلمات الدالة: أمن الشبكات، نظام كشف التسلل، هجمات يوم الصفر، الذكاء الاصطناعي، التعلم الآلي، التعلم العميق

Acknowledgements

First and foremost, we would like to thank Allah, the Almighty, for granting us the strength, knowledge, and perseverance to complete this project. Without His guidance, none of this would have been possible.

We would like to express our deepest gratitude to our supervisor, « **Dr. Chaima Aouiche »**. *Her invaluable guidance, expertise, and unwavering support have been a beacon of light throughout the duration of this project. Her dedication to our success and her belief in our abilities have been truly inspiring. She has not only been a mentor but also a source of constant encouragement and motivation. Her thoughtful insights and patient assistance have played a crucial role in the successful completion of our work, and for that, we are eternally grateful.*

We also wish to express our sincere thanks to the members of the jury, ***« Professor. Abdallah Meraoumia »*** *and «* ***Dr. Mohammed Saigaa*** *», for their time and willingness to evaluate our work. We greatly appreciate their involvement and consideration.*

Lastly, we would like to acknowledge our families for their unwavering support and understanding during this challenging journey. Their love and encouragement have been a constant source of motivation for us. We are deeply thankful for their patience and sacrifices, which have allowed us to focus on and complete this project.

Thank you.

A. Aya & S. Yamina

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI: Artificial Intelligence

ANN: Artificial Neural Network

CIC: Canadian Institute for Cybersecurity

CPU: Central Processing Unit

CSV: Comma-Separated Values

CSE: Communications Security Establishment

DDoS: Distributed Denial of Service

DNN: Deep Neural Network

DoS: Denial of Service

DT: Decision Tree

FN: False Negative

FP: False Positive

GPU: Graphics Processing Unit

HIDS: Host-based Intrusion Detection System

IDS: Intrusion Detection Systems

MitM: Man-in-the-Middle

ML: Machine Learning

NIDS: Network-based Intrusion Detection System

R2L: Remote to Local

RF: Random Forest

SVM: Support Vector Machine

TN: True Negative

TP: True Positive

U2R: User to Root

VPN: Virtual Private Network

CHAPTER 1:

Introduction

1 Introduction

1.1 Background

Cybersecurity has become one of the most critical issues of our time, with the number and complexity of cyberattacks increasing at an alarming rate. Despite significant progress in developing advanced security technologies and personalized defense strategies, the battle against cyber threats remains ongoing. It is projected that cybercrime will result in global damages amounting to $10.5 trillion annually by 2025, underscoring the persistent and escalating nature of cybersecurity challenges.

Extensive research in cybersecurity has identified various types of malicious activities and attacks that significantly contribute to security breaches and data losses. Understanding these malicious activities is crucial as they provide clear guidelines for identifying and stopping cyber threats. Recognizing these malicious activities helps develop effective strategies to predict, detect, and neutralize cyber threats. This knowledge improves the overall security of digital systems by enabling more precise and targeted defense measures.

Identifying unknown attacks using separate datasets created at different times has been suggested as an effective way to understand and mitigate cyber threats. This approach provides a comprehensive view of the evolving nature of cyberattacks. By training models on one dataset and testing them on a newer dataset, researchers can evaluate the models' ability to detect new and unknown threats. This method is especially beneficial in cybersecurity studies where challenges such as data variety and temporal changes exist. Therefore, using multiple datasets created at different times helps obtain a robust set of insights into unknown attacks, as shown by several studies using this approach.

Using separate datasets for training and testing, rather than a single dataset, allows for a robust evaluation of intrusion detection systems (IDS) by exposing them to a variety of attack patterns and scenarios. This method ensures that the models can effectively detect new and unknown threats, enhancing the robustness and accuracy of IDS in identifying and neutralizing unknown cyber threats, thereby ensuring more reliable protection for digital infrastructures.

Numerous studies have identified many unknown attacks, but these identifications have typically been based on a single dataset. These studies have reported hundreds of such unknown attacks related to various types of cyberattacks. However, the total number of unknown attacks in different contexts is still underexplored and needs to be defined as they appear. Since cyberattacks undergo a complex dynamic evolution, identifying unknown attacks as they emerge is necessary to understand and decipher the precise mechanisms of cyberattack progression. This understanding is crucial for developing appropriate cybersecurity measures.

This study begins with a brief introduction to the problem of unknown attack prediction, followed by the motivation and main objectives of the research. The study concludes with a detailed structure of the dissertation, outlining the methodologies and approaches used to address the challenges in identifying and mitigating unknown cyber threats.

1.2 Previous Works, Motivation and Objectives

Unknown, new attacks detection, or in simpler terms, zero-day attacks, is still a very challenging task in networks and cybersecurity research. Typically, zero-day attacks are of unknown nature, with no matching patterns in the network, thus making it harder to detect and defend against. Hence, the problem of zero-day attacks identification is to find the set of the most plausible vulnerabilities that alter and damage the system behavior.

Despite the rapid progress in network analysis detection approaches and adoption of modern intrusion detection systems (IDS) to identify various types of attacks and intrusions, the ultimate goal of discovering the complete catalog of unknown attacks is still ignored. Anomaly based methods were often used by various studies to detect unknown attacks. With the adoption of machine learning in IDS, the identification of unknown attacks had become easier. For instance, Verma and ranga [1] adopted multiple machine learning models for IDS. Ferag et al [2] reviewed 40 works that implemented deep learning in IDS. Kilincer et al [3] used SVM, KNN, and DT, including multiple datasets to evaluate the performance of IDS. Hindy et al. [4] focused on the performance of ML based IDS on detecting unknown attacks. The study proposed an IDS to detect zero-day attacks with high recall rates while keeping the miss rate to a minimum. These few studies focused on older datasets. These studies were able to detect cyber-attacks, ignoring the long term performance of IDS. Based on this motivation, the objectives of this study are described as follows:

1. Evaluating the long-term performance of intrusion detection systems using machine learning (ML) and deep learning (DL) algorithms with separate datasets.
2. Propose an improved learning model and analysis framework based intrusion detection system that can detect zero-day attacks.
3. Reviewing current ML DL algorithms for detecting various cyberattacks, including zero-day attacks.

1.3 Organization of The Dissertation

This dissertation includes some theoretical and practical findings, consisting of a brief review of the fundamental concepts of cybersecurity, cyberattacks (**Chapter2**), and the most common ML and DL algorithms-based IDS used for zero-day attack detection (**Chapter3**). Besides, three up-to-date intrusion detection system datasets are expounded upon and analyzed to conduct our experiments (**Chapter4**).

CHAPTER 2:

An Overview of Cybersecurity

2 An Overview of Cybersecurity

2.1 Introduction

In today's digitally interconnected world, Internet usage has experienced exponential growth, with individuals and companies increasingly conducting daily transactions in cyberspace rather than in the physical realm. The coronavirus (COVID-19) pandemic has further accelerated this trend, ushering in new paradigms in corporate transactions, remote work culture, and online education. Consequently, the widespread adoption of digital environments has led to a surge in both the frequency and sophistication of cyber-attacks, rendering traditional systems obsolete. These attacks, including zero-day exploits which target vulnerabilities across hardware, software, and communication layers, pose significant challenges to cybersecurity.

The adoption of modern intrusion detection systems (IDS), and AI-based anomaly detection, holds promise in bolstering cybersecurity practices by identifying deviations from normal behavior within network traffic and system operations.

In this chapter, we delve into the realm of cybersecurity, elucidating its principle criteria and key fundamental concepts. We define cybersecurity and various types of cyber-attacks, including known and unknown threats such as zero-day attacks, and discuss associated strategies. We also explore mechanisms for enhancing security measures. Furthermore, we provide a comprehensive overview of intrusion detection system (IDS), detailing their definitions, operational principles, strategic placement, and types. Lastly, we address addressing network analysis detection systems, highlighting their significance in shaping modern cybersecurity practices.

2.2 Cybersecurity

2.2.1 Definition

There have been different definitions for cyber security, such as data security, information security, network security, and cyber security. When protecting data in a digital landscape, understanding these distinctions becomes crucial. Data security focuses on protecting digital data from unauthorized access, modification, or disclosure throughout its lifecycle. Information security extends its coverage to include both physical and electronic information, aiming to prevent unauthorized access, use, disclosure, modification, review, recording, or destruction. Network security, meanwhile, prioritizes maintaining the confidentiality, integrity, and accessibility of computer networks and transmitted data [5].

On the other hand, Cyber security, derived from the term 'cyber' which describes networks with infrastructure information systems, also referred to as 'virtual reality'. Cyber security constitutes a comprehensive strategy against digital threats, it encompasses the holistic practice of safeguarding computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks in the expansive cyberspace. Therefore, cyber security has a much wider application area that covers end-to-end information flows. Nowadays, the term “cybersecurity” is mainly being used [6] [7].

2.2.2 Cybersecurity fundamental principle:

Cybersecurity mainly involves three dimensions: Security principles, Information states and Countermeasures. The first dimension focuses on protecting information from attackers, known as the “principle of information security”. This principle includes the concepts of confidentiality, integrity, and availability (CIA) as depicted in figure 2-1 and explained below:

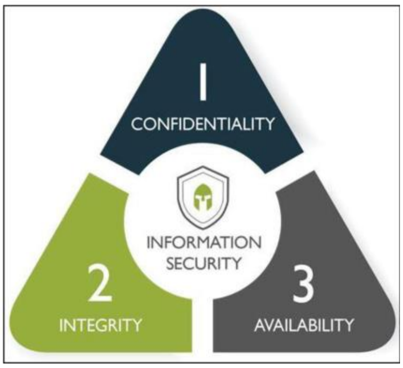


Figure 2-1: Security principles

- **Confidentiality:** Ensures that only authorized individuals possess access to exchanged information.

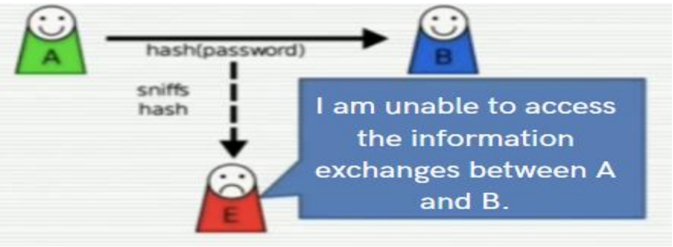

A
hash(password)
B
sniffs hash
I am unable to access the information exchanges between A and B.
E

Figure 2-2: Confidentiality principle

- **Integrity:** Ensures the accuracy and reliability of information, safeguarding it from unauthorized modification.

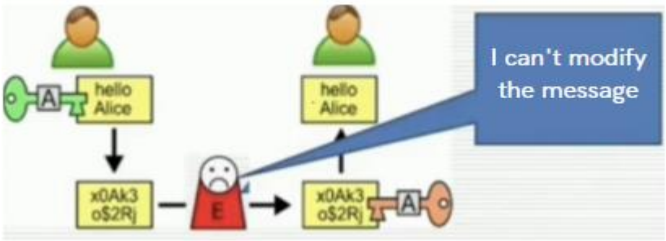


Figure 2-3: Integrity principle

- **Availability:** Guarantees secure access to all necessary data or services on the system for authorized individuals.

Figure 2-4: Availability

The constant changes in the cybersecurity landscape required a more detail approach. This evolution has led to the recognition of additional criteria beyond the traditional CIA triad, such as authentication (ensures identity verification and validates legitimacy for accessing specific company resources, such as data, software, etc) and non-repudiation (ensures that a transaction cannot be denied by preventing both the sender and the receiver from denying sending a message or receiving it). These additional properties enhance the security and reliability of the overall information security framework [8].

2.2.3 Cyberattacks

With the development of technology, accessing information has become easier and more widespread. However, this increased accessibility has also introduced significant challenges in maintaining information security. In today’s world, where information systems are integral to all aspects of life, the threat of cyberattacks has become increasingly prominent. They target a broad spectrum of areas, including daily life, government institutions, the economy, commerce, banks and hospitals.

Various definitions have been given to cyberattacks, and these attacks have been categorized in multiple ways. Below, we describe some common types of cyberattacks, classified into two categories that we believe represent the ideal categorization:

- **known Attacks**

Known cyberattacks refer to those that have been publicly disclosed, documented, and studied by cybersecurity experts and researchers. These attacks often exploit well-known vulnerabilities or use techniques that have been observed and analyzed in the past.

Known attacks are classified according to various criteria, such as the activities and targets of the attacker. Table 2-1 shows some of the most well-known types.

Type of Attack	Definition
Malware	Malicious software designed to cause damage or disrupt functionality [5].
Phishing	Fraudulent attempts to obtain sensitive information by impersonating a legitimate entity [6].
Ransomware	Malicious software that encrypts the victim's data and demands payment for decryption [6].
Espionage	The act of secretly gathering confidential information without the owner's consent [6].
Distributed Denial of Service (DDoS)	Attacks aiming to make a service unavailable by overwhelming it with traffic from multiple sources [6].
Man-in-the-Middle (MitM) Attack	An attacker intercepts and alters communication between two parties who believe they are directly communicating [7].
Password Attacks	Attempts to gain unauthorized access by cracking or guessing passwords [7].

Table 2-1: known attacks Types

Other extensive research studies has systematically classified all known cyberattacks into: (a) active, (b) passive or (c) into one of the following four categories [8].

- **Denial of Service (DoS):** Denial of Service attacks are a class of attacks aimed at disrupting, limiting, or preventing legitimate users from accessing network or computer services by overwhelming and saturating its computing or memory resources, thereby preventing legitimate user access to a machine. The goal of DoS attacks is generally to disrupt services rather than to steal information, but they can nonetheless cause significant disruptions in terms of time and resources.

- **User to Root (U2R) Attacks:** User to Root attacks are a class of attacks where an attacker starts with access to a normal user account on the system and then exploits a vulnerability to gain administrative access to the system.
- **Remote to Local (R2L) Attacks:** Remote to Local attacks are a class of attacks where the attacker deliberately sends data packets to a specific machine over a computer network. By exploiting existing security vulnerabilities on this machine, the attacker succeeds in illegally gaining privileged local access. These attacks can be carried out by exploiting vulnerabilities in operating systems or applications to access sensitive information or perform malicious actions.
- **Probe Attacks:** Probe attacks are cyberattacks aimed at acquiring information about the network or computer system by using probes to identify vulnerabilities or flaws. Probes are detection devices capable of analyzing the network, monitoring traffic, and identifying potential threats. These attacks can be used to map systems, identify weak points, and prepare for more advanced attacks.

- **Unknown Attacks**

As there are known attacks, there are also unknown attacks, which are very challenging to detect and pose serious problems for cybersecurity, because they are not recognized or identified by traditional detection methods. One of the most well-known unknown attacks is the zero-day attack.

- **Zero-day Attack definition:**

A zero-day attack is a new type of cyberattack that is unknown to the public and the cybersecurity community, hence the name "Zero-day." This type of attack can be delineated as an anomalous traffic pattern devoid of any corresponding signatures in malware or attack detection components within the network.

Zero-day attacks exploit vulnerabilities in software or security policies that have not yet been disclosed to the public. They are of unknown nature; attackers can use them in combination with other complex attacks to prevent themselves from being detected by the intrusion detection methods. These attacks can take on various forms, including polymorphic worms, viruses, Trojans, network attacks, and other malware. The primary objective of these attacks is to gain unauthorized access to or threaten a running system [9] [10] .

Bilge and Dumitras [11] conducted a study on the duration and prevalence of zero-day attacks. According to their findings, a typical zero-day attack persists for an average of 312 days before detection. It is discovered every 17 days, and it takes 15 days on average to develop a patch to fix it. In the illustrated timeline presented in Figure 2-5, the progression of a zero-day attack unfolds as follows: Initially, the developer uploads software that is accessible to the public. Following this, hackers identify zero-day vulnerabilities and exploit them to seize control of the system. Once these vulnerabilities are brought to the developer's attention, they initiate efforts to rectify the issues by addressing the bugs and creating patches.

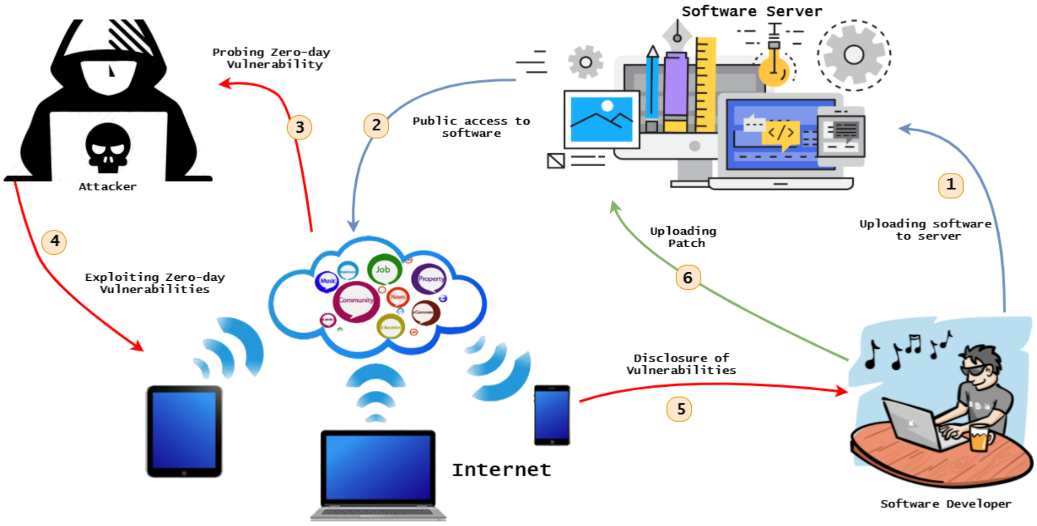
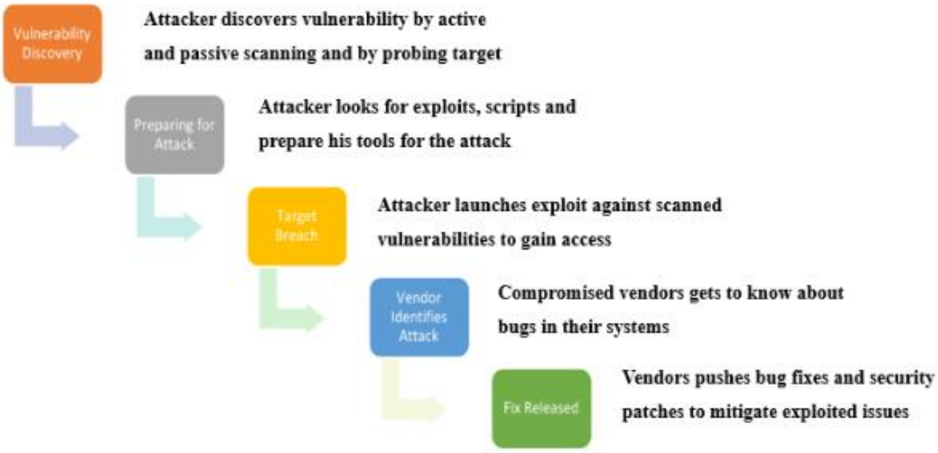

Software Server
Probing Zero-day Vulnerability
3
2
Public access to software
Attacker
1
Uploading Patch
Uploading software to server
4
6
Exploiting Zero-day Vulnerabilities
Disclosure of Vulnerabilities
5
Internet
Software Developer

Figure 2-5: Timeline of the Stages of a Zero-day Attack [16]

- **Zero-day attack detection strategy process:**

The schematic for a zero-day attack detection process is given in figure 2-6.



Attacker discovers vulnerability by active and passive scanning and by probing target

Attacker looks for exploits, scripts and prepare his tools for the attack

Attacker launches exploit against scanned vulnerabilities to gain access

Compromised vendors gets to know about bugs in their systems

Vendors pushes bug fixes and security patches to mitigate exploited issues

Figure 2-6: Zero-day Attack detection process

2.2.4 Cybersecurity Protection Mechanisms:

Some common types of countermeasures used in cybersecurity include:

- **Firewall**

These are software programs or devices that protect a computer or a network of computers from intrusions originating from a third-party network (notably the internet). They filter data packets exchanged with the network and block those that do not meet security requirements [12].

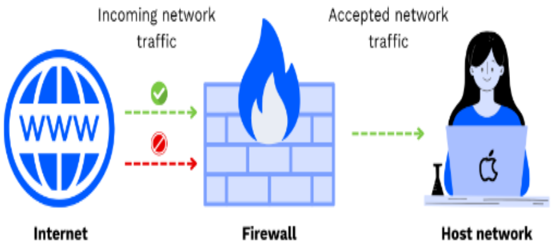


Figure 2-7: Firewall

- **Anti-Virus**

Antivirus software is capable of detecting malicious programs such as viruses and worms, destroying or quarantining them, and sometimes repairing infected files without damaging them [12].

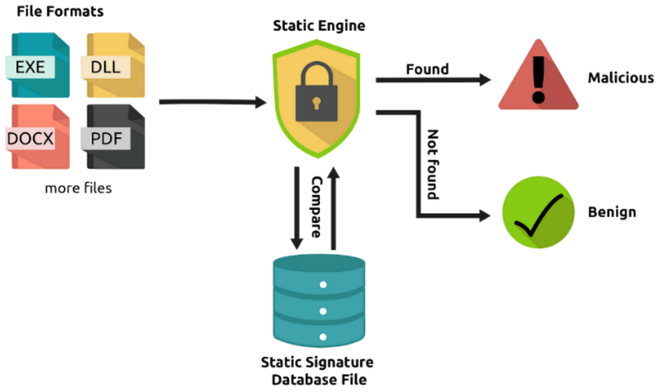


Figure 2-8: The Function of an Anti-Virus

- **Cryptography**

Cryptography is a set of encryption methods used to secure sensitive information from unauthorized access during transmission over networks, ensuring that only individuals with the designated keys can access encrypted data.

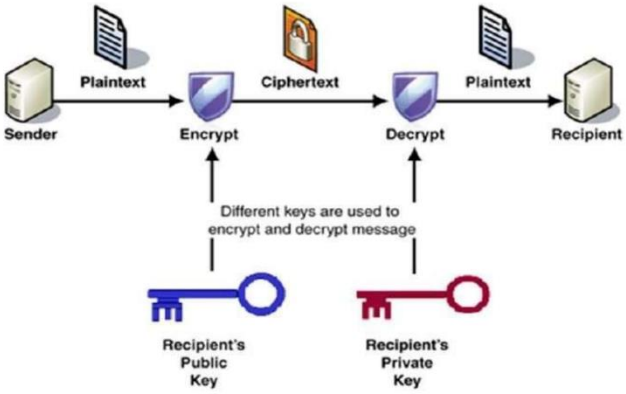


Figure 2-9: The Process of Encryption and Decryption in Cryptography

- **VPN**

A Virtual Private Network (VPN) is a secure connection between a device and a network, using the public Internet. It creates a virtual point-to-point connection through tunneling protocols, ensuring confidentiality, authentication, and integrity. The tunneling principle involves encapsulating data packets within another packet, sending them over the internet, and creating a secure and encrypted connection [19].

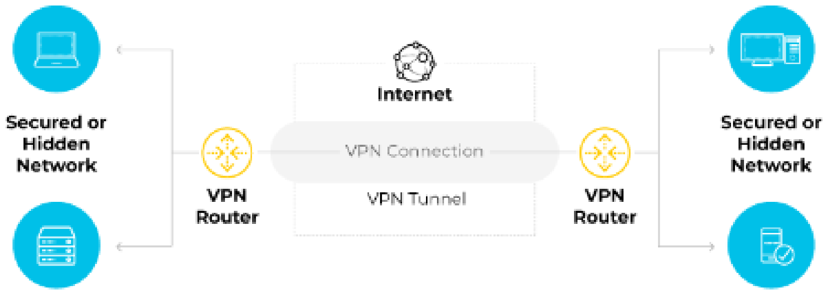


Figure 2-10: VPN Tunneling illustration for Secure Data Transmission

- **INTRUSION DETECTION SYSTEM (IDS)**

These tools are designed to monitor network traffic to detect and prevent abnormal behavior or potential threats within a network. It was one of the best solution of effective detection for unknown cyberattacks.

2.3 INTRUSION DETECTION SYSTEM (IDS)

2.3.1 Definition

An Intrusion Detection System (IDS) is a set of software and/or hardware components that closely monitor network activities with the aim of detecting any attempts to intrude into computer systems. The primary objective is to identify suspicious network traffic patterns and potentially dangerous computer behaviors that may evade detection by a traditional firewall. This monitoring is crucial to ensure optimal protection against threats that could compromise the availability, integrity, or confidentiality of computer systems [20].

2.3.2 Operating principles of IDS

Intrusion Detection Systems (IDS) seek out known attack characteristics or identify activities that do not adhere to a specific standard. They then inform administrators of these anomalies and potentially malicious actions so that they can be analyzed at the application and protocol layers [8].

Figure 2-11 illustrates how an IDS operates and performs various actions during intrusion detection.

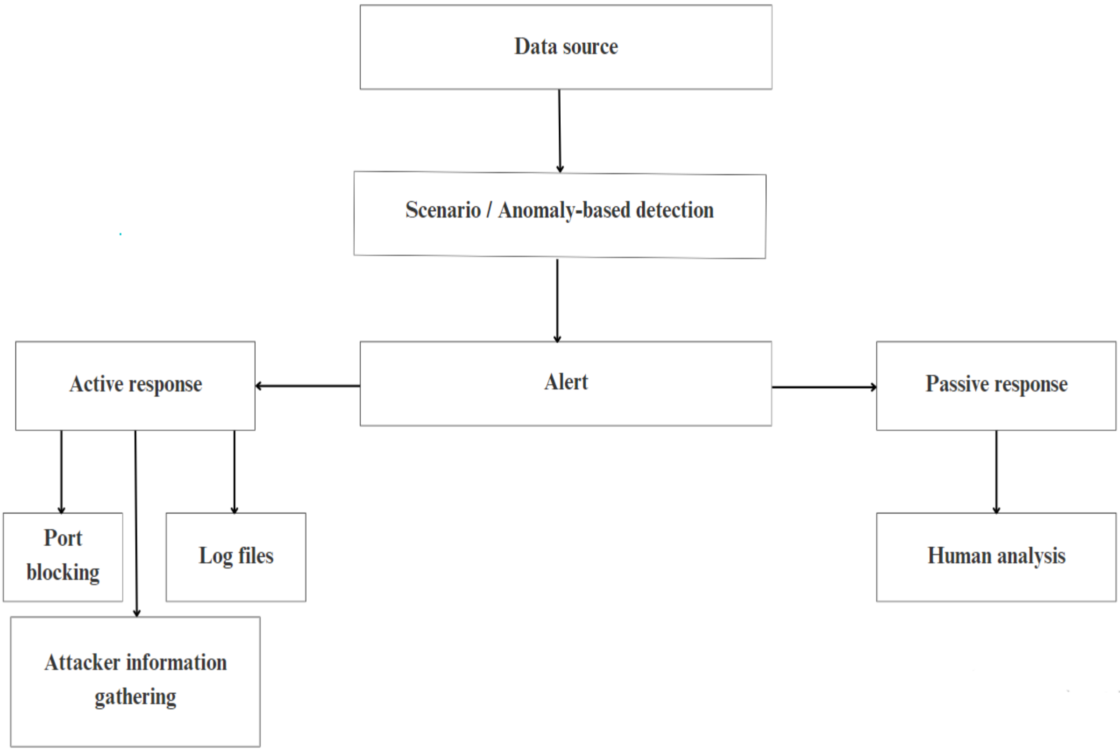


Figure 2-11: Operation of an Intrusion Detection Systems (IDS)

2.3.3 Strategic placement of IDS

It is crucial to properly position the intrusion detection system, which involves identifying the resources to be protected and what is most likely to be attacked. Therefore, it is necessary to carefully deploy it in the appropriate zone. There are several strategic locations where it is recommended to install an IDS [16].

Figure 2-12 shows a local network as well as the three possible positions for an IDS.

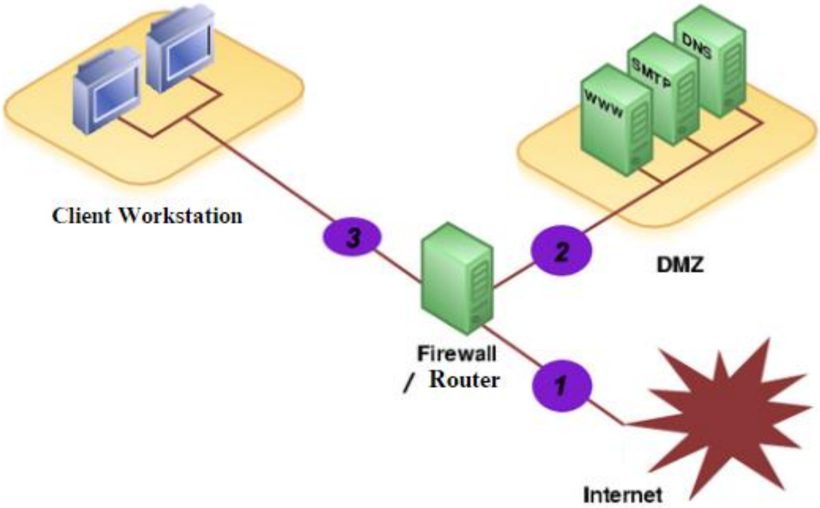


Figure 2-12: Strategic Placement of Intrusion Detection Systems (IDS)

- **Position (1):** When the IDS is positioned in this manner, its goal is to detect all frontal attacks coming from the outside towards the firewall. As a result, numerous alerts will be signaled, making log consultation more complicated.
- **Position (2):** Installing the IDS on the DMZ (Demilitarized Zone) allows it to detect attacks that have not been filtered by the firewall and require a certain level of operation. It will be easier to consult the logs here because benign attacks will not be recorded.
- **Position (3):** The objective of the IDS in this position is to signal internal attacks originating from the company's local network. It would be advisable to install it in this location, as 80% of attacks come from within. Additionally, if trojans are present in the IT park (due to careless internet browsing), they can be easily detected here for subsequent removal.

2.3.4 Types of IDS

Depending on the placement of the IDS within the network architecture to be monitored, we distinguish three types of IDS:

- **Host-Based Intrusion Detection Systems (HIDS)**

A Host-Based Intrusion Detection System (HIDS) is indeed an essential security tool that plays a crucial role in analyzing activities on a single host system to detect suspicious behavior or potential intrusions that could be related to malicious software such as worms, viruses, or Trojan horses. From an architectural perspective, HIDS is typically deployed on critical machines which are at risk of attacks and contain sensitive data for the organization. For instance, web servers and application servers can significantly benefit from protection by a HIDS. The HIDS collects data from a machine where an HIDS client is installed, then analyzes this information to monitor the system's operation and status, aiming to detect potential threats. Figure 2-13 illustrates an example of a HIDS architecture [15].

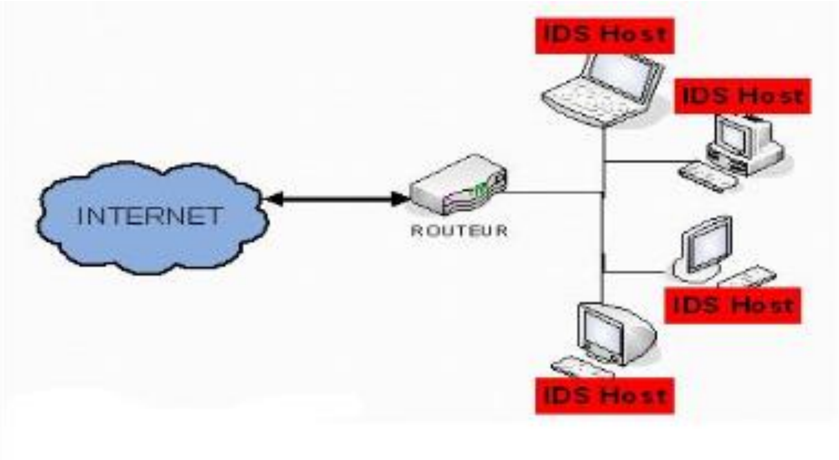


Figure 2-13: Example of a HIDS Architecture

- **Network- Based Intrusion Detection Systems (NIDS)**

A Network-Based Intrusion Detection System (NIDS) is a type of intrusion detection system designed to monitor network traffic for suspicious activities and issue alerts when such activities are detected. Architecturally, a NIDS is typically positioned on an isolated network, between the network entry points and the network terminals, to capture a copy of the traffic, specifically the packets circulating on the network. By strategically positioning the NIDS in this manner, it can effectively detect potential threats or policy violations and raise alerts when suspicious activities are identified. Figure 2-14 illustrates an example of a NIDS architecture [15].

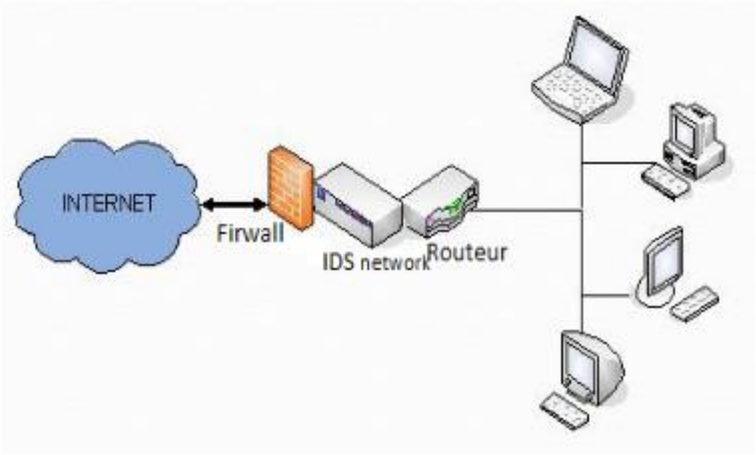


Figure 2-14: Example of a NIDS Architecture

- **Hybrid Intrusion Detection Systems**

Hybrid Intrusion Detection Systems are designed to combine the strengths of both Network-Based Intrusion Detection Systems (NIDS) and Host-Based Intrusion Detection Systems (HIDS) to provide a comprehensive and robust security solution. They are strategically placed and function as NIDS and/or HIDS depending on their locations. All these hybrid IDS then report alerts to a central machine that consolidates everything and links information from various sources. Thus, it's understood that hybrid IDS are based on a distributed architecture, where each component unifies its sending format. This facilitates communication and extracts more accurate alerts. Figure 2-15 illustrates an example of a Hybrid IDS architecture [15].

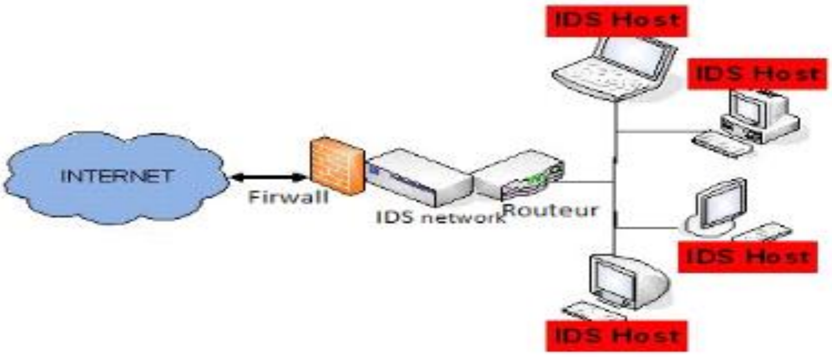


Figure 2-15: Example of an Hybrid IDS Architecture

2.3.5 Network analysis detection systems

Intrusion detection is based on this fundamental approaches :

- **Signature based IDS**

Signature-based Intrusion Detection Systems (SIDS) use pattern-matching techniques to identify known attacks. When an intrusion signature matches one listed in the signature database, an alarm is generated. Although these systems offer high accuracy in detecting known attacks, they cannot detect the attack if it is not found in the IDS database such as zero-day attacks. Despite this disadvantage, SIDS maintain their effectiveness through regular updates to their signature patterns [8].

- **Anomaly based IDS**

Anomaly-based Intrusion Detection Systems (AIDS) also called rule-based systems, have garnered the interest of researchers due to their ability to circumvent the limitations inherent in SIDS. These systems monitor all events on the network. An alarm is generated whenever there is deviation outside the specified rules, potentially indicating an intrusion. The biggest advantage of anomaly based intrusion detection systems is that they can detect unknown attacks such as the zero-day attacks without relying on predefined signatures. But from another perspective, it registers the largest values of false positive alarm, which means there is a large number of normal packets considered as attacks packets. However, effective deployment necessitates the formulation of a robust model delineating normal system behavior, achievable through methodologies encompassing machine learning, statistical analysis, or knowledge-based approaches [8].

- **Artificial Intelligence (AI) based IDS**

Effectively identifying cyberattacks is a critical challenge for network operators and managers, particularly in the rapidly evolving modern networks. To improve intrusion detection accuracy and defend against more attacks, including new and unknown zero-day attacks, many artificial intelligence (AI) techniques can be used to develop Intrusion Detection Systems (IDSs).

These techniques include regularization techniques, Machine Learning (ML), Deep Learning (DL) and ensemble methods, which are powerful tools for learning useful features from network traffic and predicting normal and abnormal activities based on the learned patterns [17]. These methods have been successfully employed by various researchers to detect different network anomalies, while some resist overfitting others suffer the most from overfitting.

The most common ML algorithms used for IDS are Decision Tree, Random Forest, Support Vector Machine (SVM), Naïve Bayes and K-Mean Clustering, The most common DL algorithms used for IDS are Artificial Neural Networks and Deep Neural Networks [18]. Figure 2-16 summarizes the classification of IDS systems from the detection perspective, highlighting the various AI methods employed for intrusion detection.

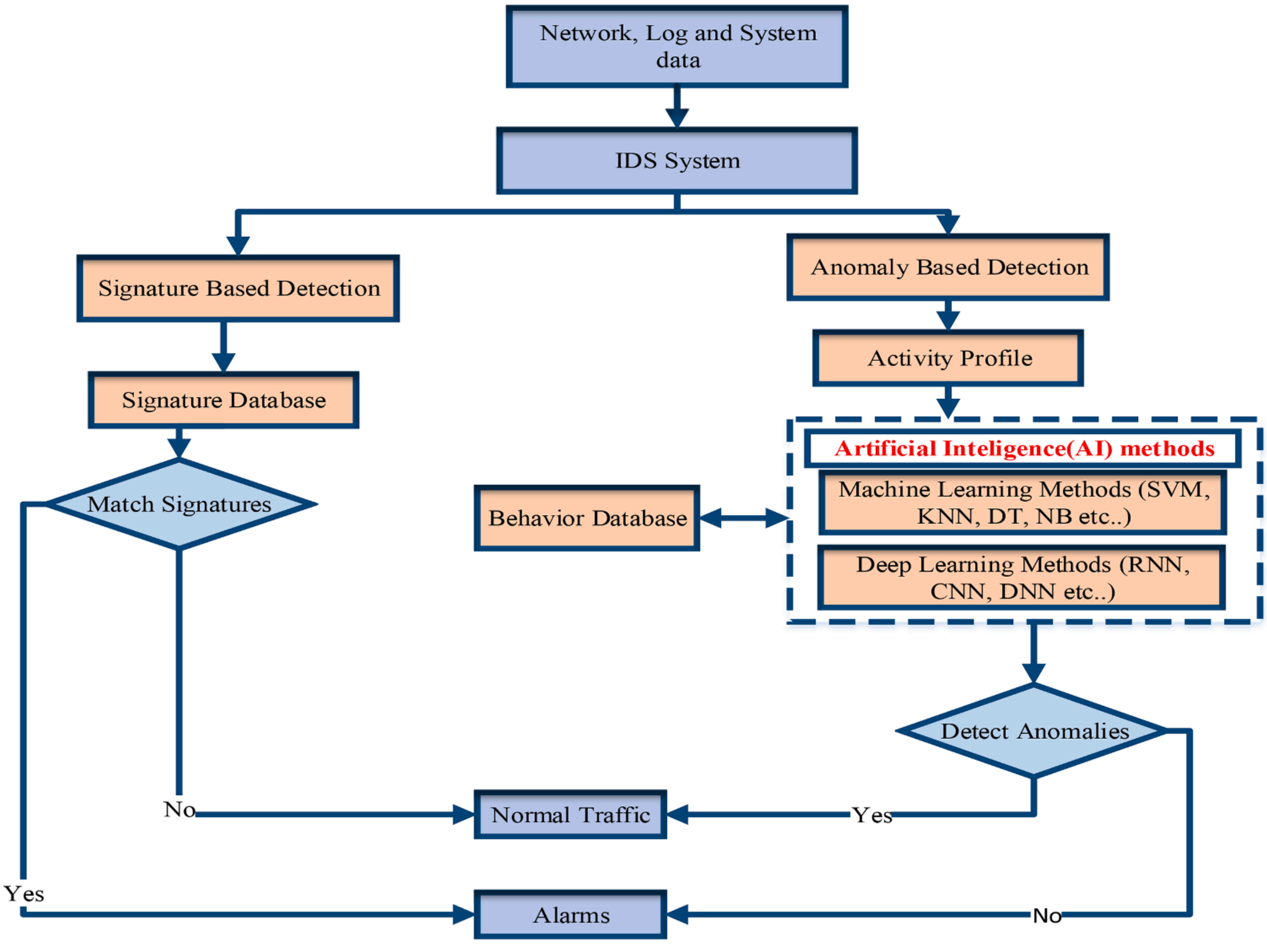

Network, Log and System data
IDS System
Signature Based Detection
Anomaly Based Detection
Signature Database
Activity Profile
Artificial Inteligence(AI) methods
Match Signatures
Behavior Database
Machine Learning Methods (SVM, KNN, DT, NB etc..)
Deep Learning Methods (RNN, CNN, DNN etc..)
Detect Anomalies
No
Normal Traffic
Yes
Yes
Alarms
No

Figure 2-16: Classification Schema of IDS System [22]

2.4 Conclusion

Despite their crucial role in network security, intrusion detection systems (IDS) face significant challenges such as packet loss, resource saturation, and the difficulty in identifying unknown attacks, including zero-day threats. These limitations reflect the broader landscape of cybersecurity threats. As cyberattacks proliferate, cybersecurity will become an increasingly critical concern.

Figure 2-17 resumes the diverse security solutions, both technical and non-technical, employed to counter these growing threats. We will focus on the application of AI in addressing few issues, emphasizing the interconnected nature of challenges and responses within the cybersecurity domain.

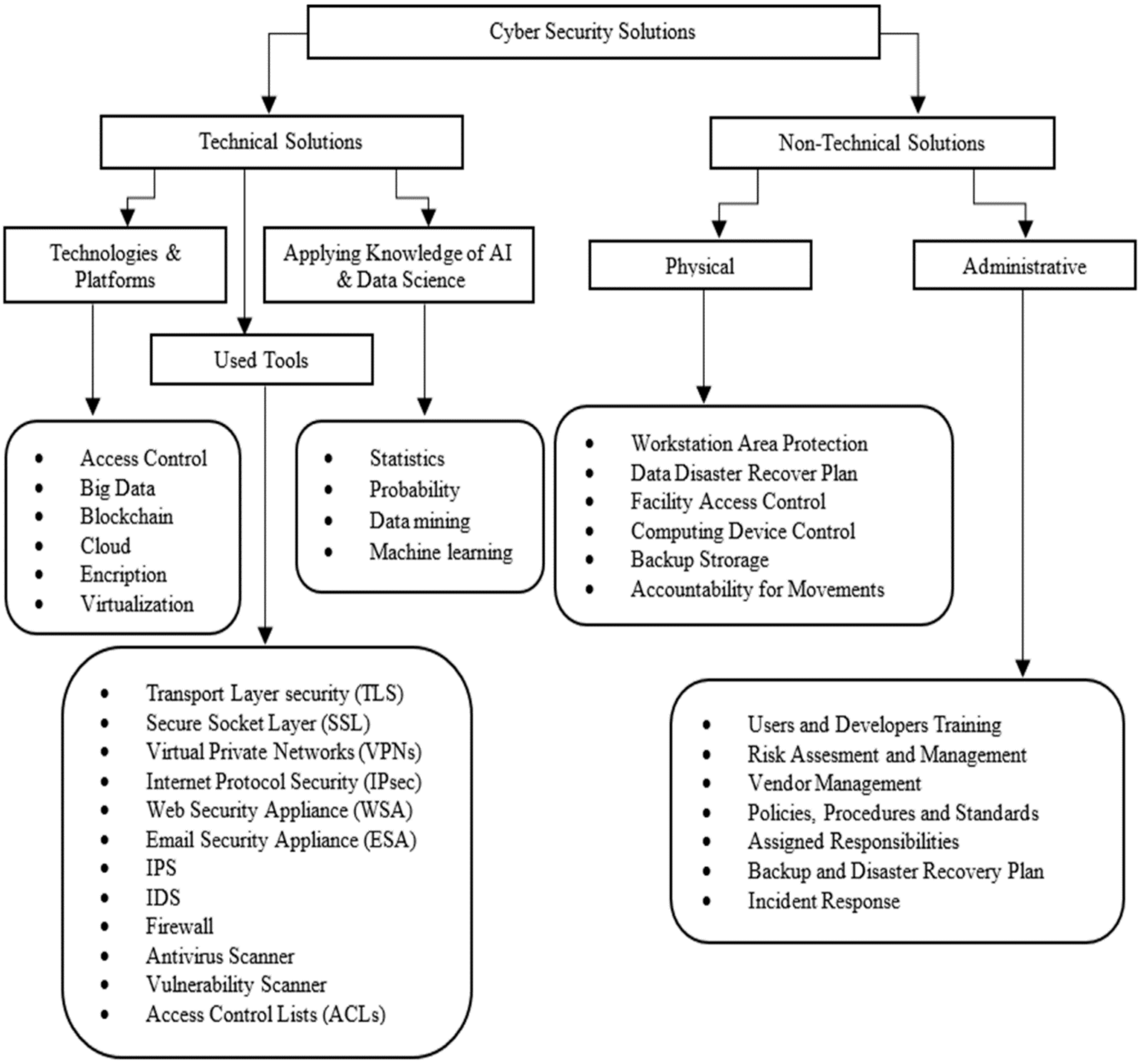

Cyber Security Solutions
Technical Solutions
Non-Technical Solutions
Technologies & Platforms
Applying Knowledge of AI & Data Science
Used Tools
Physical
Administrative
Access Control
Big Data
Blockchain
Cloud
Encription
Virtualization
Statistics
Probability
Data mining
Machine learning
Workstation Area Protection
Data Disaster Recover Plan
Facility Access Control
Computing Device Control
Backup Strorage
Accountability for Movements
Transport Layer security (TLS)
Secure Socket Layer (SSL)
Virtual Private Networks (VPNs)
Internet Protocol Security (IPsec)
Web Security Appliance (WSA)
Email Security Appliance (ESA)
IPS
IDS
Firewall
Antivirus Scanner
Vulnerability Scanner
Access Control Lists (ACLs)
Users and Developers Training
Risk Assesment and Management
Vendor Management
Policies, Procedures and Standards
Assigned Responsibilities
Backup and Disaster Recovery Plan
Incident Response

Figure 2-17: Technical and Non-technical Cybersecurity Solutions

CHAPTER 3:

Artificial Intelligence Vs Machine Learning Vs Deep Learning

3 Artificial Intelligence Vs Machine Learning Vs Deep Learning

3.1 Introduction

Cybersecurity and artificial intelligence (AI) are two growing technologies. The integration of Artificial Intelligence (AI) into cybersecurity has become indispensable, providing innovative solutions to combat the escalating threats in cyberspace. Through advanced algorithms and techniques, AI revolutionizes threat detection and response mechanisms, enabling organizations to protect their data and systems effectively. Machine learning, including supervised and unsupervised methods, plays a central role in analyzing vast amounts of data to identify patterns and anomalies indicative of cyber threats. Deep learning further enhances cybersecurity capabilities by extracting intricate features from raw data, enabling accurate detection of even the subtlest threats. By continuously adapting and learning from new threats, AI-powered cybersecurity systems bolster resilience and minimize the impact of cyberattacks, marking a significant advancement in cyber defense strategies.

The proliferation of security monitoring systems in all communication domains has led to the massive data generation. This data typically includes details on questionable activity occurring within apps and networks. By using AI approaches, models may be trained to look for zero-day exploits or undiscovered malware based on the characteristics and behavior of packets moving through networks, which will shorten the time it takes to find assaults.

In this chapter, we embark on an exploration of Artificial Intelligence (AI), including its milestones and definitions, Additionally, we cover the fundamental principles of Machine Learning, both supervised and unsupervised learning. Furthermore, our investigation extends to Deep Learning. Lastly, we conclude by the discussion of the critical role of AI in cybersecurity domain, underscoring its importance in the detection and mitigation of cyber threats, and its contribution to fortifying defense mechanisms against evolving security challenges.

3.2 Artificial Intelligence

3.2.1 Milestones

Artificial Intelligence (AI) has undergone significant transformations since its inception, becoming an integral part of modern technology. The journey of AI has been marked by several groundbreaking milestones that have shaped its development and applications. The timeline in Figure 3.1 represents pivotal moments where advancements in AI research and technology have led to significant achievements, pushing the boundaries of what machines can do.

3.2.2 Definition

Artificial Intelligence often abbreviated as AI has been defined in various ways by different sources, reflecting the evolving nature of the field and its diverse applications. Here are four notable definitions:

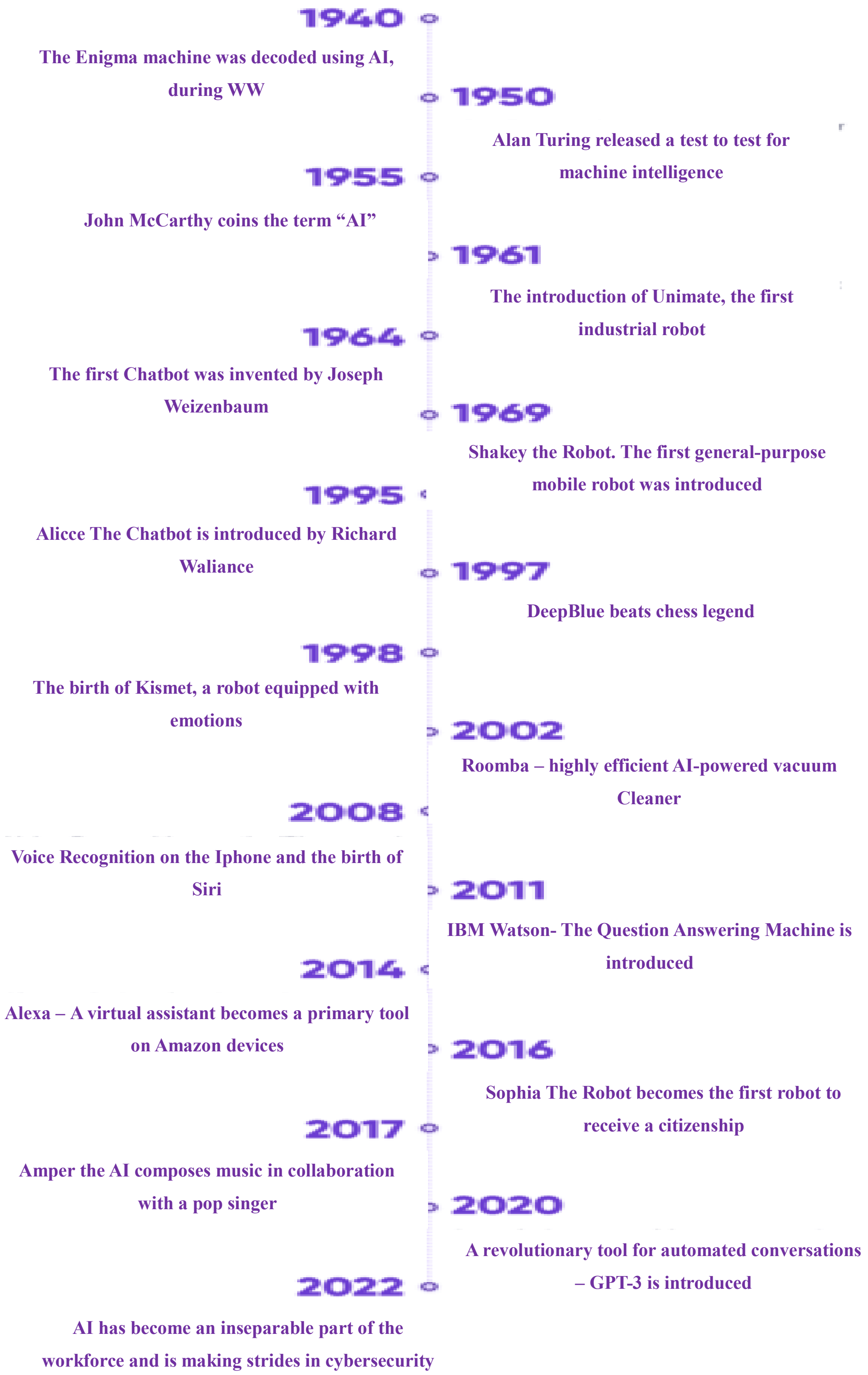
Marvin Lee Minsky, one of the creators of AI, defines it as "the construction of computer programs that perform tasks in a manner that is more satisfying to humans as they require high-level mental processes such as perceptual learning, memory organization, and critical reasoning "[24].

Merriam-Webster defines artificial intelligence as "a branch of computer science dealing with the simulation of intelligent behavior in computers". This definition focuses on the aspect of AI as a subset of computer science aimed at replicating intelligent behavior.

Intel describes AI as "a program that can sense, reason, act, and adapt". This definition emphasizes the dynamic and adaptive nature of AI systems.

Wikipedia states that the term "artificial intelligence" is applied when "a machine mimics 'cognitive' functions that humans associate with other human minds, such as 'learning' and 'problem-solving'".

Artificial Intelligence is a vast and rapidly evolving field that encompasses numerous subfields, each with its unique focus and potential. Machine Learning, Deep Learning, and Generative AI as clearly shown in Figure 3.2. These are the three key subfields that have made significant contributions to the field of AI and its applications in various domains, such as cybersecurity.



1940

The Enigma machine was decoded using AI, during WW

1950

Alan Turing released a test to test for machine intelligence

1955

John McCarthy coins the term “AI”

1961

The introduction of Unimate, the first industrial robot

1964

The first Chatbot was invented by Joseph Weizenbaum

1969

Shakey the Robot. The first general-purpose mobile robot was introduced

1995

Alicce The Chatbot is introduced by Richard Waliance

1997

DeepBlue beats chess legend

1998

The birth of Kismet, a robot equipped with emotions

2002

Roomba – highly efficient AI-powered vacuum Cleaner

2008

Voice Recognition on the Iphone and the birth of Siri

2011

IBM Watson- The Question Answering Machine is introduced

2014

Alexa – A virtual assistant becomes a primary tool on Amazon devices

2016

Sophia The Robot becomes the first robot to receive a citizenship

2017

Amper the AI composes music in collaboration with a pop singer

2020

A revolutionary tool for automated conversations – GPT-3 is introduced

2022

AI has become an inseparable part of the workforce and is making strides in cybersecurity

Figure 3-1: Artificial Intelligence milestones [23]

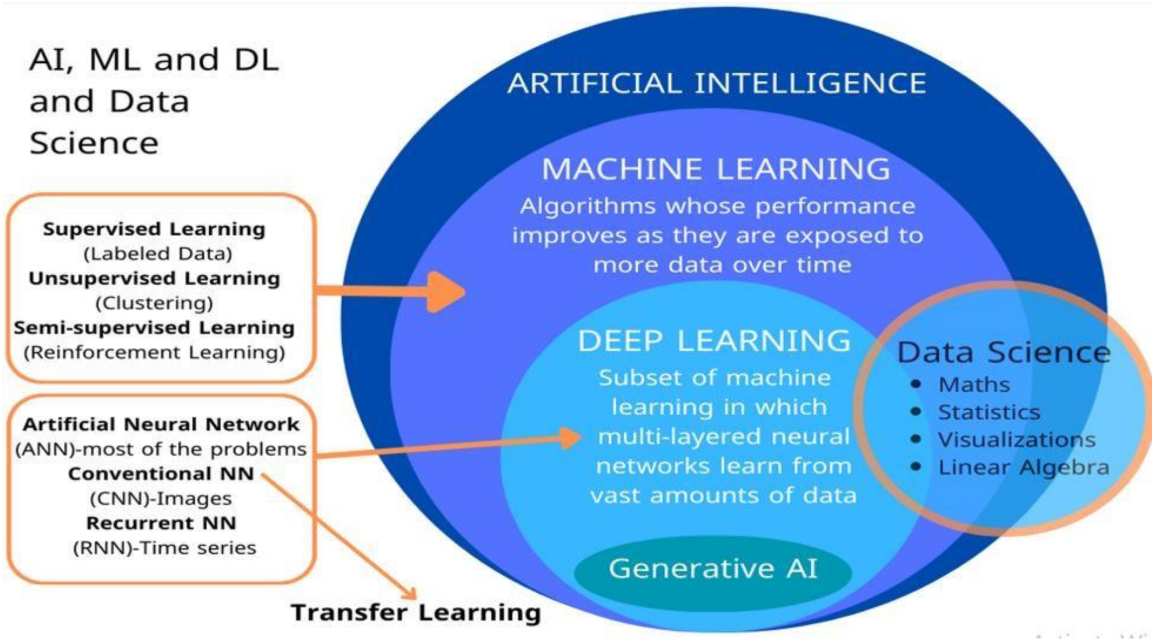

AI, ML and DL and Data Science
Supervised Learning (Labeled Data)
Unsupervised Learning (Clustering)
Semi-supervised Learning (Reinforcement Learning)
Artificial Neural Network (ANN)-most of the problems
Conventional NN (CNN)-Images
Recurrent NN (RNN)-Time series
Transfer Learning
ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
Algorithms whose performance improves as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multi-layered neural networks learn from vast amounts of data
Generative AI
Data Science
Maths
Statistics
Visualizations
Linear Algebra

Figure 3-2: A representation of Artificial Intelligence subfields

3.3 Machine Learning

Machine learning is an area of artificial intelligence that focuses on teaching machines to use algorithms to learn new things and make judgments based on data. ML is closely related to mathematical methods that enable data extraction, pattern recognition, and conclusion drawing. The two most significant applications of ML technology are regression and classification [25].

ML techniques are playing a vital role in numerous applications of cybersecurity for the early detection and prediction of different attacks such as spam classification, fraud detection, malware detection, phishing, dark web or deep web sites, and intrusion detection. ML techniques can address the scarcity of required personnel with expertise in these niche cybercrime detection technologies. ML is one of the possible solutions to act quickly against such attacks because ML can learn from experiences and respond to newer attacks on time [26].

Machine learning is generally classified into two main categories: "supervised learning" and "unsupervised learning".

3.3.1 Supervised Learning

Supervised learning is the most common form of machine learning. The process of supervised learning involves transforming one set of data into another. The program is trained on a predefined set of training examples, which enhances its ability to draw accurate conclusions when new data is encountered [27].

Various algorithms used for supervised ML classification include Decision Trees, Random Forests, Naïve Bayes, and the most famous one, Support Vector Machines (SVM).

- **Decision Trees**

A decision tree is a non-parametric supervised learning model used for both classification and regression tasks. It exhibits a hierarchical, tree-like organization consisting of a root node, branches, internal node, and leaf node as shown in Figure 3-3.

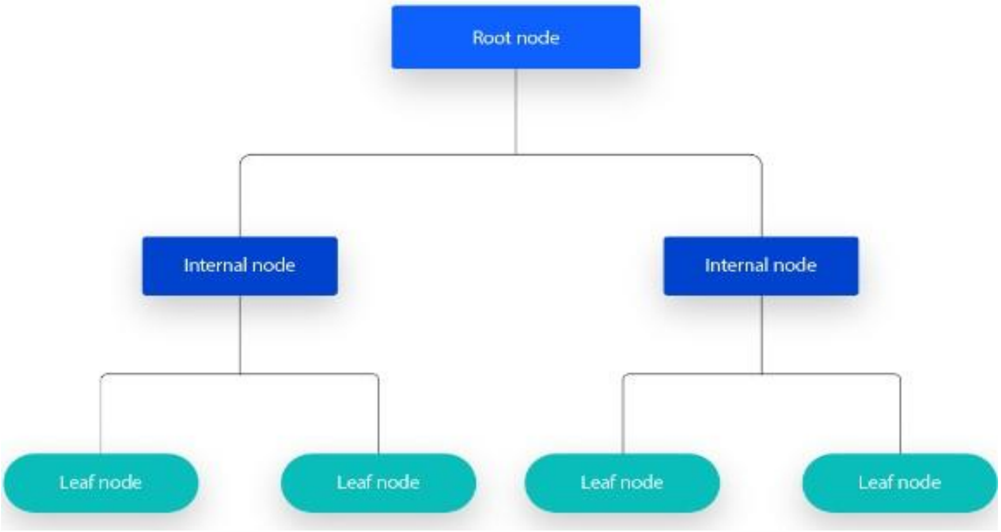


Figure 3-3: Decision Tree

As depicted in the above diagram, a decision tree begins with a root node, which has no incoming branches. The branches emanating from the root node led to internal nodes, also known as decision nodes. Based on the available features, both types of nodes conduct evaluations on homogeneous subsets, which are referred to as leaf nodes or terminal nodes. The Leaf nodes represent all possible outcomes within the dataset etc [28].

The example in figure 3-4 shows a decision tree used for classifying plant species based on two features: petal length and stem length. The tree starts at the root node with the question, "Is the petal length greater than 2.5 cm?" If the answer is "No," the plant is classified as Species B. If "Yes," it proceeds to a decision node that asks, "Is the stem length greater than 12 cm?" If this answer is "Yes," the plant is classified as Species A, and if "No," it is classified as Species C.

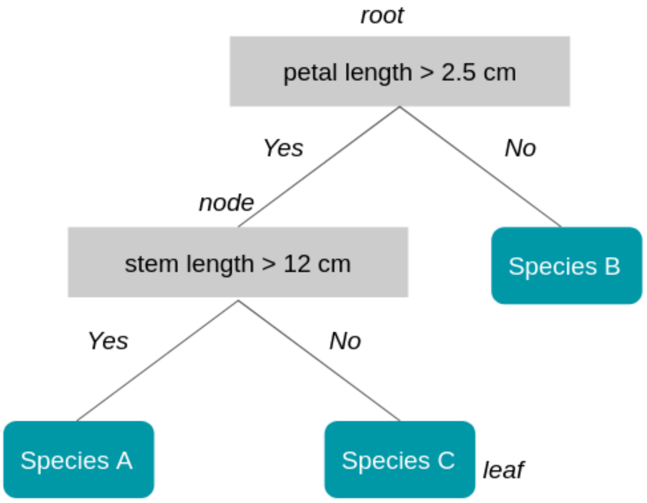


Figure 3-4: Example of a Decision Tree Algorithm

- **Random Forest**

Random forest is a commonly-used machine learning algorithm, trademarked by Leo Breiman and Adele Cutler, that combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems [29]. The figure 3-5 illustrates the method by which results from multiple decision trees are aggregated within the random forest to yield a final outcome.

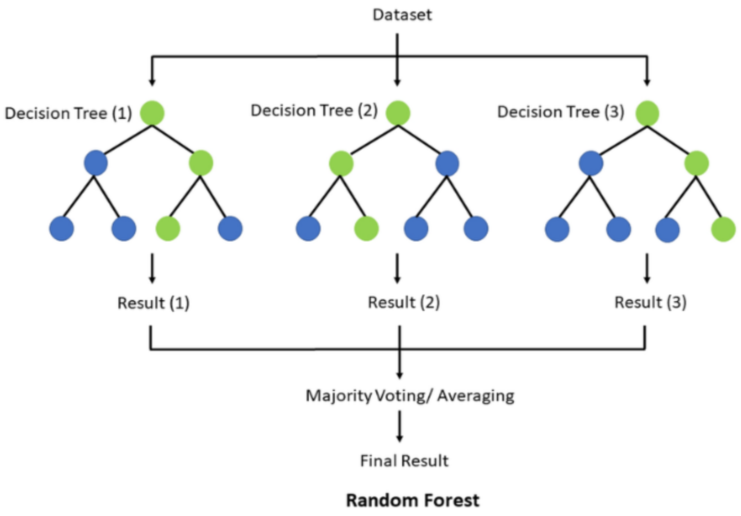


Figure 3-5: Random Forest

The example in figure 3.6 shows a Random Forest classifier using three decision trees to predict a field of study based on proficiency in Mathematics and Language. The final prediction, determined by majority vote, is "CS" (Computer Science) in this example, illustrating how Random Forest improves accuracy by aggregating multiple trees predictions.

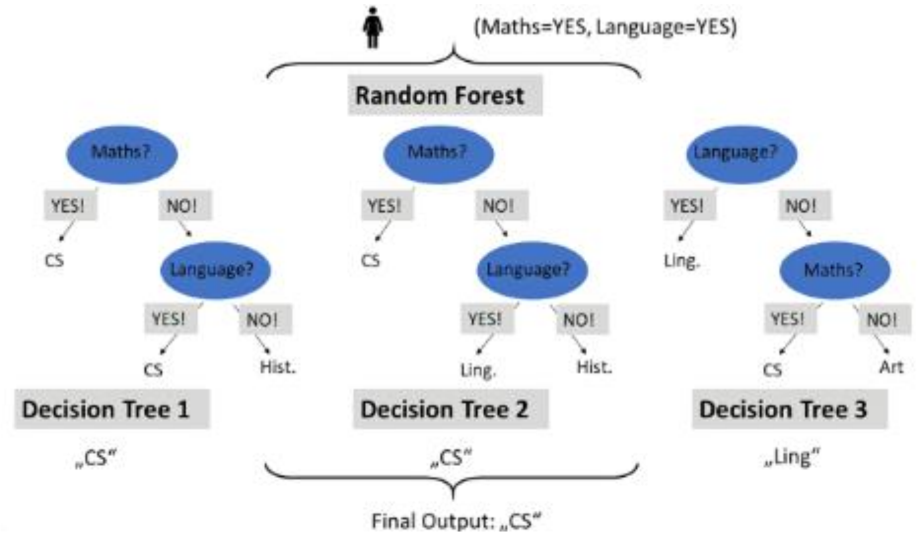


Figure 3-6: Example for Random Forest

- **Naïve Bayes**

Naive Bayes techniques are among the most well-known probabilistic models. They are primarily based on Bayes theorem (Bayes, 1963). Naive Bayes algorithms are frequently employed for document classification. The probability of each class among the examples can be estimated based on a document, and the document is assigned the most probable class. This process is referred to as "Prior probabilities."[30]. The Naive Bayes algorithm utilizes Bayes' theorem, which is expressed as:

$$P\left({}^{A}/_{B}\right) = \frac{P\left({}^{B}/_{A}\right) * P(A)}{P(B)} \qquad 3\text{-}1$$

- **Support Vector Machine (SVM)**

SVM[24] is a discriminative classifier defined by a splitting hyperplane. To classify intrusions linearly, Support Vector Machines (SVMs) employ a kernel function to translate the training data into a higher-dimensional space. SVMs are particularly useful when there are many attributes and few data points. They are well-known for their capacity to generalize. Many features in IDS datasets are redundant or have less of an impact on classifying data items into the appropriate categories. Consequently, when training SVMs, features selection should be taken into account. Multiple class classification is another application for SVM, and Figure 3-7. shows the typical SVM:

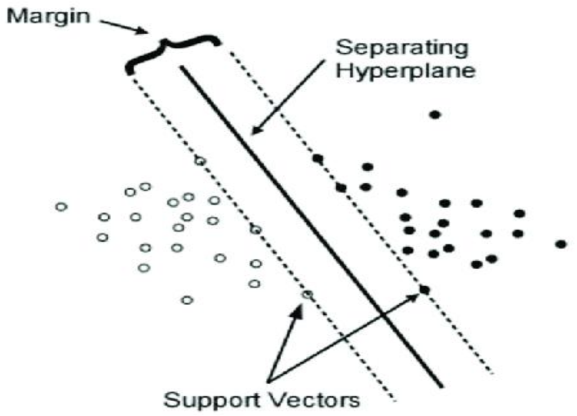


Figure 3-7: Support Vector Machine

3.3.2 Unsupervised Learning

Unsupervised learning is employed to discern inherent patterns within data without relying on predefined labels. This approach encompasses algorithms designed for clustering and association rule learning.

Clustering, a fundamental method in unsupervised learning, entails partitioning input data into distinct groups based on latent similarities not previously recognized. Commonly employed algorithms in unsupervised learning techniques include the k-means clustering algorithm, hierarchical clustering algorithm, Apriority algorithm, and principal component analysis. Clustering is a very practical Machine Learning algorithm for identifying groups of similar behaviors. This algorithm allows working on the data and classifying it. With a clustering algorithm, it's possible to automatically discover groups (or clusters) of clients who exhibit similar behaviors. This allows grouping audiences according to similar behaviors, easily identifying "free electrons" that do not belong to a group, and even discovering previously unknown behaviors. It thus becomes easier to personalize the service and therefore offer a quality experience to clients [32].

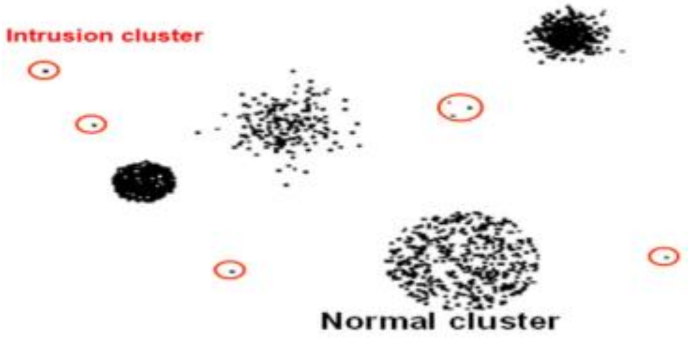


Figure 3-8: Using Clustering for Intrusion Detection [35]

3.4 Deep Learning

Deep learning (DL) has emerged as a pivotal area of interest for researchers across various domains, particularly in science and technology. Over the past decade, convolutional neural networks and deep belief networks have represented the two primary research directions.

DL models offer versatile applications spanning cybersecurity, finance, medicine, image processing, search engines, and pattern recognition. Inspired by the human brain, deep learning algorithms excel in analytical and logical thinking. Today, automatic car driving stands as a prominent example of deep learning's practicality, with numerous studies showcasing its application to bolster cybersecurity [26].

DL is a sophisticated computational system comprised of a multitude of techniques from the field of machine learning. It employs a set of nonlinear neurons arranged in different layers of processing to extract and transform feature variable values from the input vector, thereby creating various levels of abstraction to represent the data.

- **Operation**

The architecture of a deep learning is structured into layers of neurons: an input layer, multiple hidden layers, and an output layer. Each pair of adjacent layers is connected. The relationships between them are known as edges. The "neurons" within the same layer are often referred to as "nodes" and are not connected. The standard architecture of a deep neural network model is illustrated in Figure 3-9

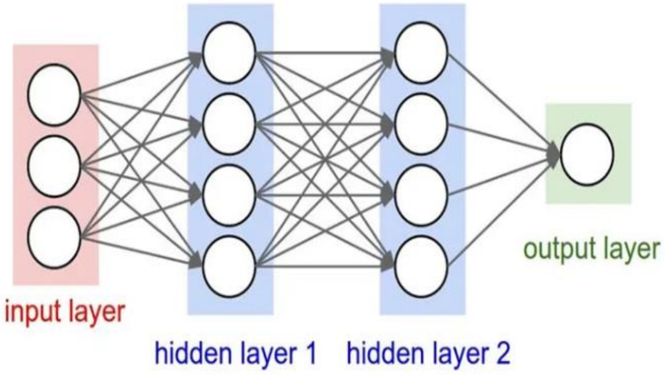


Figure 3-9: The architecture of a Deep Learning model

3.4.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are composed of various nodal layers, including an input layer, one or more hidden layers, and an output layer. Each node, also referred to as an artificial neuron, is connected to another and has specific weights and thresholds. When the output of a node exceeds the defined threshold, this node is activated and sends information to the next layer of the network. Otherwise, no information is transmitted to the next layer of the network. These details are illustrated in Figure 3-10.

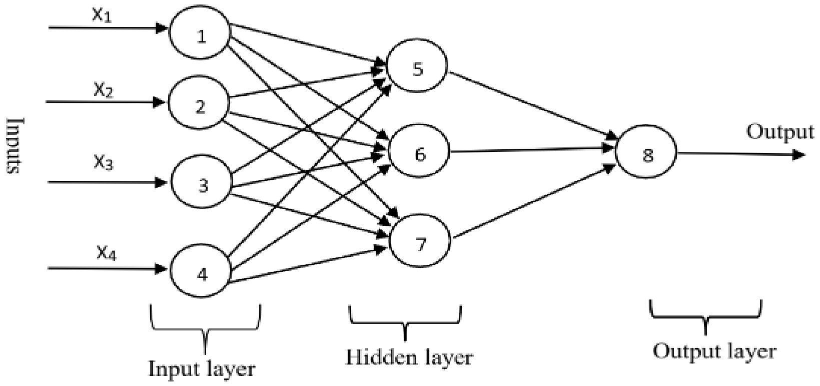


Figure 3-10: An Artificial Neural Network Architecture

Neural networks utilize training data to acquire knowledge and enhance their accuracy over time. However, when these learning algorithms become sufficiently sophisticated, they evolve into powerful tools for computing and artificial intelligence, providing the capability to classify and group data rapidly. Tasks such as speech or image recognition can be completed in just a few minutes, whereas manual identification by human experts takes several hours. The most renowned neural network is Google's search algorithm [34].

- **MLP (Multi-Layer Perceptron)**

MLPs are a specific type of feed forward artificial neural network (ANN) known as multilayer perceptron's. The name "MLP" is imprecise; it could refer to any feed-forward artificial neural network (ANN) or to networks composed of multiple layers of perceptron's that are activated by thresholds. Neural networks with multiple hidden layers, especially multilayer perceptron's, are commonly referred to as "Vanilla" neural networks.

An MLP consists of at least three tiers of nodes, which are composed of an input layer, a hidden layer, and an output layer. All nodes, with the exception of the input nodes, are neurons with nonlinear activation functions. During training, MLP uses the supervised learning technique known as back propagation. The numerous layers and non-linear activation of MLP set it apart from a linear perceptron. It can distinguish between data that isn't linear and data that is [35].

3.4.2 Deep Neural Network

A deep neural network (DNN) is a type of artificial neural network comprised of numerous hidden layers between the input and output layers as shown in Figure 3-11.

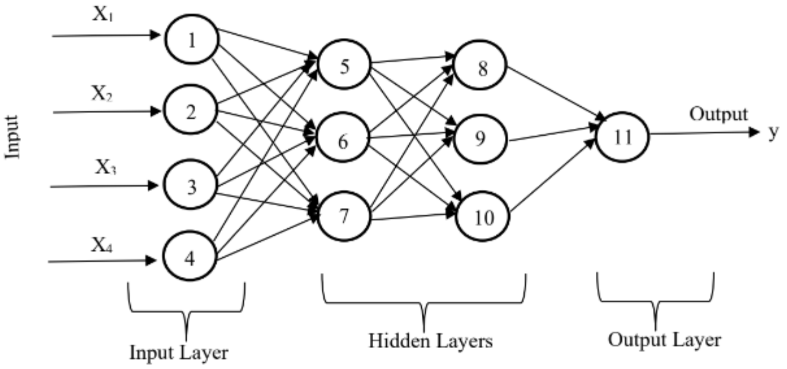


Figure 3-11: A Deep Neural Network Architecture

The operation of a DNN can be likened to the operation of the human brain to facilitate better understanding. Just as the human brain learns to recognize faces based on a reference model, a DNN learns from input data and their relationship with pre-established patterns. When a face is perceived, the brain identifies differences compared to its reference face model, such as the eyes, ears, and eyebrows, in an instant. These differences are quantified by varying electrical signals, and all deviations are combined to produce an outcome.

The nodes in a DNN are similar to neurons in the human brain, and each layer is composed of multiple nodes that react to stimuli. The neural network then interprets the data collected by sensors or directly provided by a programmer, such as images, texts, or sounds, by converting them into numerical values.

Data is progressively processed between the input layer and the output layer to solve a task or make a prediction. The first layer of the network receives the data and processes it using activation functions to produce a result, such as a probability prediction. This result is then transmitted to the next layer of neurons via connections associated with weights, which determine the influence of the data on the output produced by the next layer and eventually on the final output. In summary, deep neural networks enable the automatic learning of complex data representations through their multi-layer architecture, rendering them highly effective for numerous AI tasks [26].

- **Comparison between ANN and DNN**

Understanding the differences between ANNs and DNNs is crucial for grasping their applications and performance characteristics. The table 3-1 bellow presents a detailed comparison between ANN and DNN across various features.

Feature	ANN (Artificial Neural Network)	DNN (Deep Neural Network)
Type	Feed-forward neural network	Multi-layered neural network
Structure	Typically 3-5 layers	Multiple layers, often 10+
Input Processing	Processes inputs in a forward-facing direction	Processes inputs in a forward-facing direction
Feature Learning	Can learn features but not as effectively as DNNs	Can learn complex features through multiple layers
Accuracy	Less powerful than DNNs for image classification	More powerful than ANN for image classification
Training Data	Can be trained with limited data	Requires large amounts of data for effective training
Applications	Suitable for tabular data, text data, and simple image classification	Suitable for complex image classification, computer vision, and natural language processing

Advantages	Fault tolerance, ability to work with incomplete knowledge	High accuracy, ability to learn complex features
Disadvantages	Limited feature learning capabilities, hardware dependence	Requires large amounts of data and computational resources, can be difficult to train

Table 3-1: Comparison between ANN and DNN [36]

3.5 Conclusion

In contemporary cybersecurity, AI plays a critical role in mitigating zero-day vulnerabilities. Zero-day vulnerabilities pose a serious threat to conventional security measures because they are unpatched software defects that are not yet known about. By learning from patterns and anomalies in real-time, even in the absence of established signs, AI improves our capacity to proactively detect and counter these new threats. Because zero-day attacks are so fast and sophisticated, it is imperative that real-time analysis and reaction take place. Compared to conventional techniques, AI can identify potentially harmful behaviors and zero-day assaults early and with higher accuracy performance. Furthermore, by automating the incident response procedure, AI-driven solutions quickly contain and eliminate risks. Integrating AI makes cybersecurity systems more predictive and adaptive, providing a proactive protection against the constantly changing cyber threat scenario.

In conclusion, artificial intelligence emerges as a crucial means to combat cybersecurity threats. Its rapid advancement, particularly through machine learning and deep learning methods, has contributed to enhancing defenses against sophisticated and ever-evolving attacks. The applications of artificial intelligence in detecting anomalies, analyzing suspicious behaviors, and automatically responding to incidents have greatly bolstered the resilience of digital infrastructures. However, it is paramount to remain vigilant about the ethical challenges and potential risks associated with the use of AI in cybersecurity. By continuing to develop robust technologies and adopting a comprehensive approach to security, we can enhance trust in cyberspace and ensure the security of critical data and systems.

CHAPTER 4:

Experimental Materials, Methods and Results

4 Experimental Materials, Methods and Results

4.1 Introduction

After defining the basic theoretical concepts of cybersecurity and different learning algorithms in the third and fourth chapters, we move on to the second part of our approach.

In this chapter, we will present the steps that led to the results of our proposed learning model and experimental workflow, which involve several essential steps. We begin with the collection of the relevant datasets. Next, we proceed with the cleaning and preprocessing of the datasets to ensure data quality and consistency. We then perform feature selection to identify the most significant features. Subsequently, we train and validate several models based on different learning algorithms to determine which are the most suitable for our work objectives in terms of various performance metrics such as accuracy, F1-score, confusion matrix and time consumption.

4.2 Experimental workflow

Our experimental workflow involves a sequence of successive steps, including: datasets cleaning and preprocessing, features selection, models training, models validation and evaluation. An additional intermediate step is parameters optimization that's plays a great role in improving the performance of the models. Figure 4-1 illustrates the complete flowchart of the experiment. Where each experiment step is further explained in next sections.

4.2.1 Datasets Collection

The datasets used in this study are the CIC-IDS2017 and CSE-CIC-IDS2018, and CIC-DDoS2019, which provide comprehensive network traffic data for training and testing intrusion detection models over different times.

- **CIC-IDS2017 dataset**

The CIC-IDS2017 dataset is a labeled open-access dataset designed for intrusion detection and cybersecurity research. It was created by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick and includes records of various types of attacks and normal (benign) network activity collected over several days [37]. The table 4-1 presents dataset specifications:

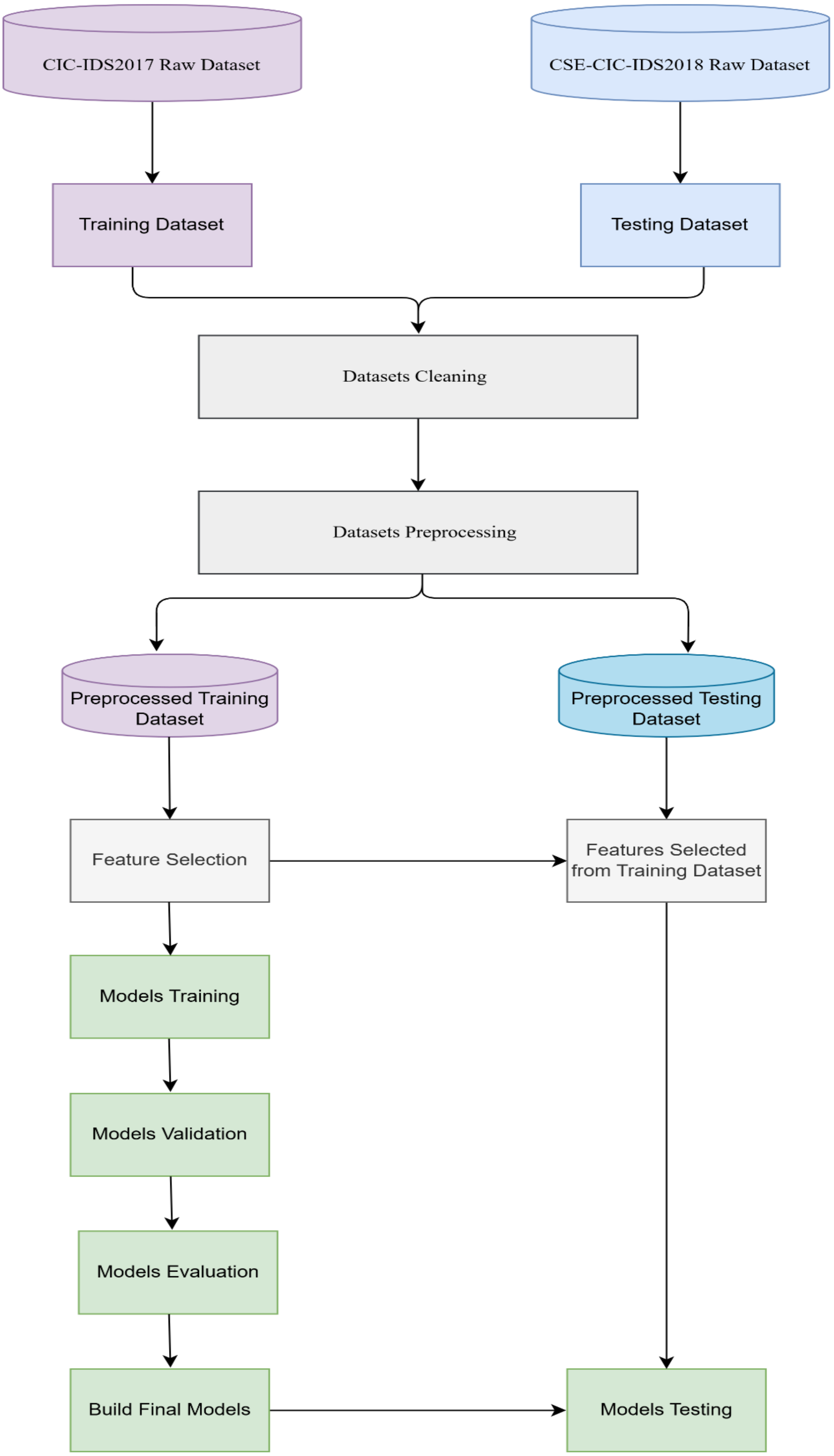


Figure 4-1: The Experimental Workflow: this workflow includes only 2 datasets CIC-IDS2017 and CSE-CIC-IDS2018, the same workflow was applied for CIC-IDS2017 and CIC-DDoS2019

Dataset Name	Number of CSV Files	Number of Features	Total Samples	Duration of Capture
CIC-IDS2017	8	85	3,119,345	5 days

Table 4-1: Specifications of CIC-IDS2017 Dataset

The CICIDS2017 dataset was collected directly from the Canadian Institute for Cyber Security (CIC) via its official website at: https://www.unb.ca/cic/datasets/ids-2017.html

- **CSE-CIC-IDS2018 dataset**

The CSE-CIC-IDS2018 dataset is a labeled open-access dataset that is publicly available for research and analysis purposes, it is the result of a collaborative project between the Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC), This dataset aims to provide a realistic cyber defense scenario for evaluating intrusion detection systems. Expands upon the CIC-IDS2017 dataset by incorporating more sophisticated attack scenarios and a wider variety of network traffic patterns [38]. The table 4-2 presents dataset specifications:

Dataset Name	Number of CSV Files	Number of Features	Total Samples	Duration of Capture
CSE-CIC-IDS2018	10	80	16,232,943	10 days

Table 4-2: Specifications of CSE-CIC-IDS2018 Dataset

The CICIDS2018 dataset was collected by installing the AWS Command Line Interface (AWS CLI) and running the command: [**aws s3 sync --no-sign-request -- "s3://cse-cic-ids2018/**] <destination-folder>" provided on the AWS Open Data website at: https://registry.opendata.aws/cse-cic-ids2018/ in the command prompt. This command downloaded all the 10 files of the CSE-CIC-IDS2018 dataset in the local destination folder.

- **CIC-DDoS2019**

The CIC-DDoS2019 dataset, the latest version published in cybersecurity research. It is a labeled open-access dataset developed by the same community of above datasets. This dataset aims to provide a realistic scenario for evaluating DDoS (Distributed Denial of service) threats detection systems. The table 4-3 presents dataset specifications:

Dataset Name	Number of CSV Files	Number of Features	Total Samples	Duration of Capture
CIC-DDoS2019	18	88	20GB	2 separate days

Table 4-3: Specifications of CSE-CIC-IDS2018 Dataset

To provide more suitable data for building and training models, the 3 datasets are passed through the same series of preprocessing operations. Here is a summary of the key preprocessing operations:

- Combining Dataset Files: We merged the 8 files from the CIC-IDS2017 dataset together, the 10 files from the CSE-CIC-IDS2018 dataset together, the 18 files of the CIC-DDoS2019 dataset together.
- Treating Missing or infinity values and duplicates: We analyzed all the datasets to detect and handle any samples with 'NaN' (Not A Number) values, 'INF' (infinite) values, and duplicate entries. Once these values were identified, we removed them from the datasets to prevent any negative impact on the performance of the final models.
- Handling high imbalance classes: this step is very important, especially in network intrusion detection symmetry to ensure models are not biased towards one type of traffic. Thus, we calculated the classes distributions in each dataset, we sum up the benign samples and all the attack samples, then we down sampled the majority class to find an equal number of classes (ratio of 1:1).
- Datasets Labeling Adjustment: For the 'Label' column, which represents the class of each instance, we simplified the labels into two main categories: 'benign' and 'malicious'. All labels of malicious traffic that were not 'benign' were changed to 'malicious', while all ' benign ' labels were retained to denote normal traffic. We also ensured that all datasets maintained an identical set of columns in the same sequence.

Only 10% of the datasets were used to be further analyzed due to hardware and runtime issues.

4.2.2 Feature Selection

Feature selection is an important task after the data preprocessing step, that not all features are significant to models building, some features may include noise and are not relevant so that the model will be difficult to understand, and as well run slow and not perform well.

There are several techniques available to perform this task. In this experiment, we employed the Random Forest algorithm with Random Forest Classifier to perform feature selection on CIC IDS 2017 only. After training the Random Forest classifier on the CIC IDS 2017 training dataset and compute feature importance score, only the features with the highest score were retained (**20 in our study**).

The top 20 features selected from the features set are shown in Figure4-2: (Init_Win_bytes_forward, Destination Port, Packet Length Std, Packet Length Variance, Average Packet Size, Bwd Packet Length Min, Total Length of Fwd Packets, Packet Length Mean, Init_Win_bytes_backward, Bwd Packet Length Std, Subflow Fwd Bytes, Fwd Packet Length Mean, Bwd Packet Length Mean, Avg Bwd Segment Size, Min Packet Length, Fwd Header Length, Subflow Bwd Bytes, Bwd Header Length, Fwd Packet Length Max, Max Packet Length).

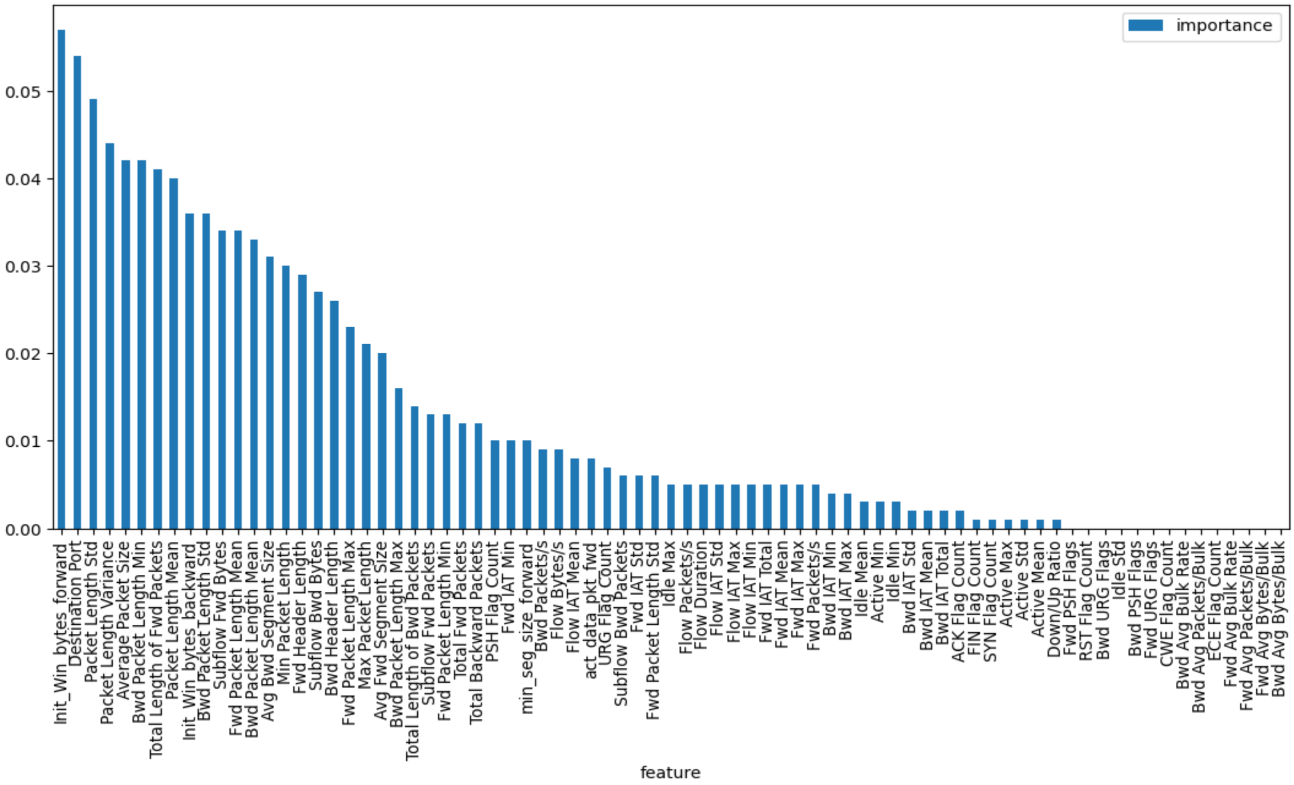


Figure 4-2: features scores

These returned features were again further reduced using brute force, it means train the models by using different number of features. The accuracy of each model with respect to the number of features will be recorded to select the most concise feature set. Figure4-3 shows the accuracy of the models with respect to the number of features, where **number of features=11** produce the best accuracy.

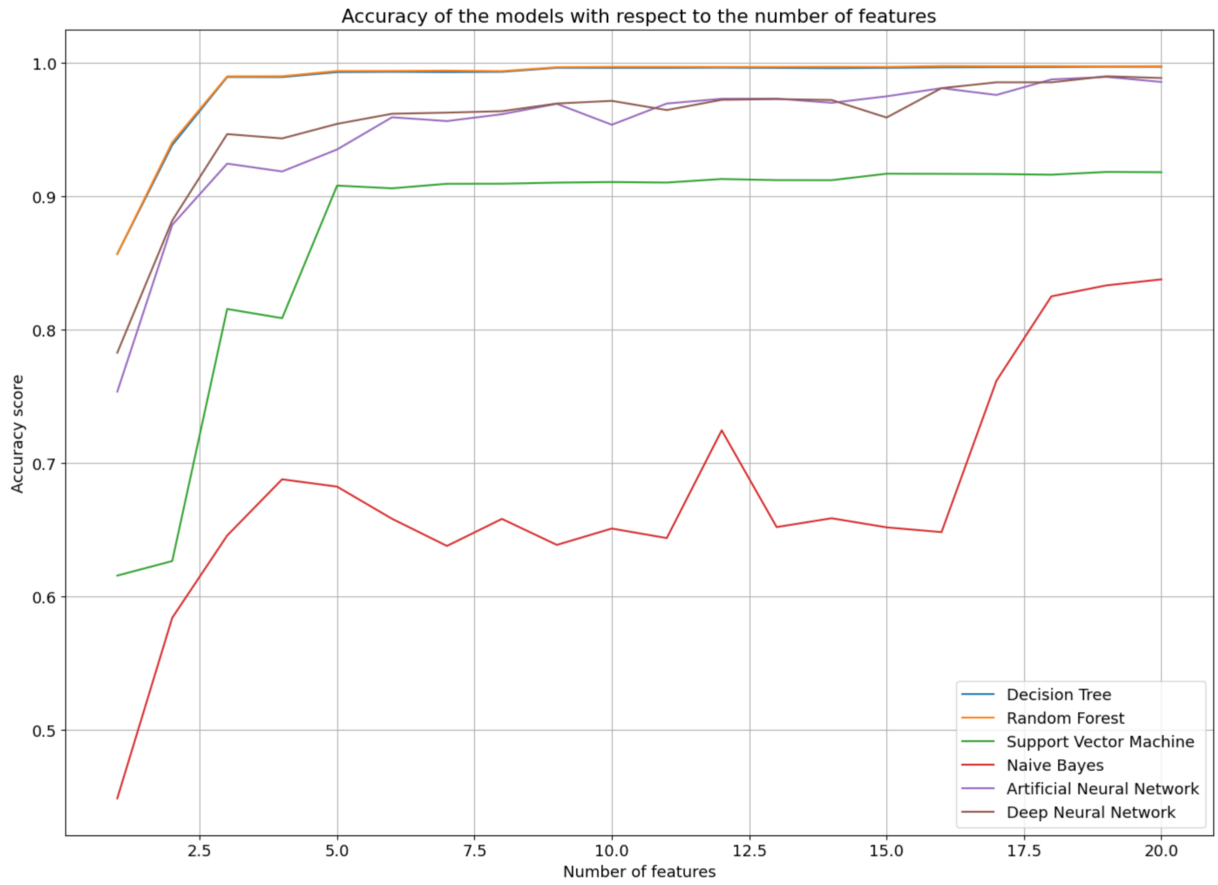


Figure 4-3: Models accuracy score with respect to brute force features

4.2.3 Models Training

After significant features selection, we train the models using the CIC-IDS2017 training dataset only. At this stage, we divide the data into training and testing set and then we perform models parameters tuning or optimization on the training set, then test the best optimized models using the test set.

Parameters optimization involves adjusting the hyperparameters of a model to improve its performance. This process aims to find the most effective combination of parameters that maximize the model's accuracy, efficiency, or other performance metrics. Optimization techniques, such as grid search, random search, or advanced methods like Bayesian optimization, can be employed to systematically explore and identify the best parameter settings.

In this study, we optimized the hyperparameters of the models using grid search (GridSearchCV function in scikit-learn). The models include: Decision tree (DT), Random forest (RF), Support vector machine (SVM), Naïve Bayes (NB), Artificial neural network (ANN) and Deep neural network (DNN), where each model was instantiated with specific hyperparameters optimized for the task. The hyperparameters used for each model are summarized in table 4-4:

Model	**Hyper parameters**
DT	Criterion : entropy , ccp_alpha: 2.0963433176353222e-07
RF	Max depth: 30, Min samples leaf: 1e-05, Min samples split: 1e-05, Number of estimators: 350, Criterion: gini
SVM	C: 100, Gamma: 1, Kernel: rbf
NB	Var smoothing: 0.012328467394420659
ANN	Hidden layer sizes: (40,), Activation: relu, Alpha: 1e-05, Solver: adam, Max iterations: 1000
DNN	Hidden layer sizes: (15, 15, 15), Activation: tanh, Alpha: 1e-05, Solver: adam, Max iterations: 1000

Table 4-4: Models hyperparameters

4.2.4 Models validation and evaluation

For an effective verification of models accuracy on the CIC-IDS2017 training dataset, we decided to apply k fold cross-validation. Cross-validation (CV) is a technique used to evaluate the performance and generalizability of a machine learning model. It involves dividing the dataset into multiple subsets or folds, training the model on some of these folds, and validating it on the remaining folds. For example, in k=5 cross-validation, the dataset is split into 5 folds. The model is trained on 4 of these folds and validated on the remaining fold. This process is repeated 5 times, with each fold serving as the validation set once. The results are then averaged to provide a more reliable estimate of the model's performance. An example of cross validation is described in Figure 4-4.

Next, we trained the final models using 70% of the training dataset. Afterward, we evaluated the models' accuracy using the remaining portion of the training dataset. Finally, we tested the models using the testing datasets (CSE-CIC-IDS2018 and CIC-DDoS2019).

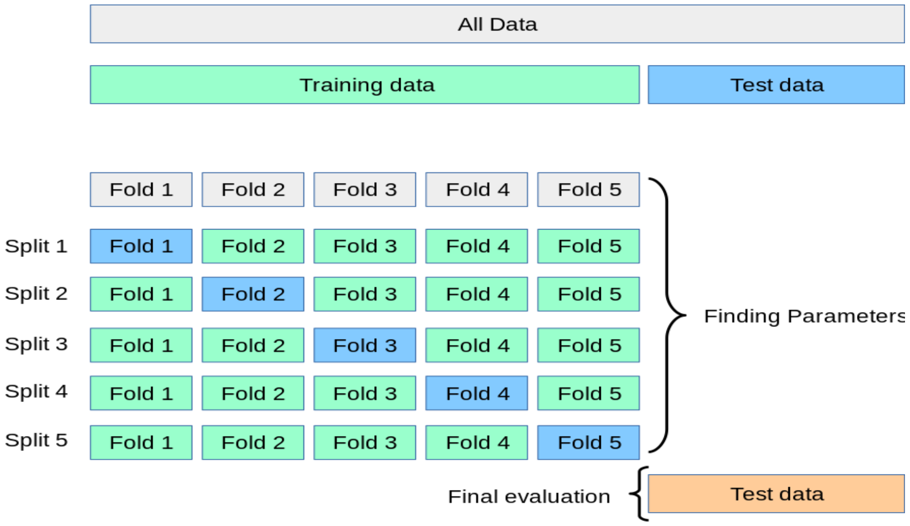


Figure 4-4: 5-fold cross validation

The objective of any classification algorithm is to rely on labeled data to build a model capable of recognizing the class membership of unlabeled data with the highest possible accuracy. Several performance measures can be used. we are particularly focused on the following metrics [31]:

- **Classification Report**

- **Accuracy:** It is a ratio of correctly classified samples/applications to all samples in a dataset. The higher value of accuracy shows the correctness of the classifier. A higher value of accuracy is desirable [31]. Mathematically, it can be expressed as:

$$Accuracy = \frac{TP+FN}{TP+FP+TN+FN} \qquad 4\text{-}1$$

- **Precision:** It is a ratio of correctly classified benign/positive samples to all classified benign in the dataset [31]. Mathematically, it can be expressed as:

$$Precision = \frac{TP}{TP+FP} \qquad 4\text{-}2$$

- **Recall:** also known as Sensitivity. It is a percentage of benign samples correctly classified to the total benign samples in the dataset [39]. Mathematically, it can be expressed as:

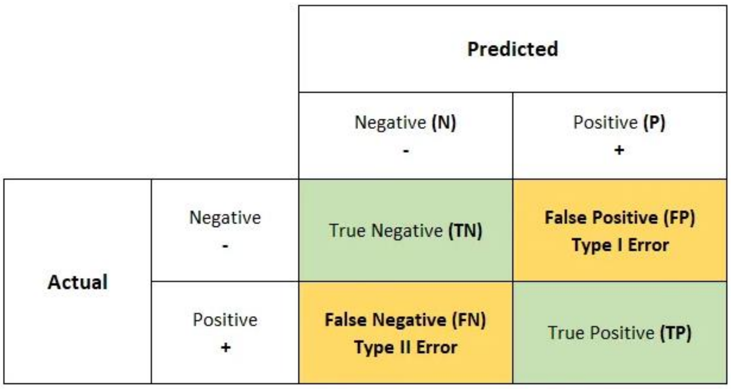
$$Recall = \frac{TP}{TP+FN} \qquad 4\text{-}3$$

- **F1-Score:** It is a measure of calculating the accuracy of the model using the values of precision and recall This measure will be helpful if the user seeks a balance between recall and precision, and sample distribution is an uneven class distribution. Mathematically [39], it can be expressed as:

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \text{4-4}$$

- True Positive (TP): the count of normal traffic that are correctly classified by the model.
- True Negative (TN): The count of malicious that are correctly classified by the model.
- False Positive (FP): the count of malicious that are misclassified as normal/positive by the model. Also known as the Type 1 error.
- False Negative (FN): The count of normal traffic that are misclassified as abnormal/negative by the model. Also known as the Type 2 error.
- **Confusion Matrix**

A confusion matrix is regarded as one of the formal ways to present the details of the learning model. A confusion matrix, also termed as an error matrix, is a tabular summary of the number of correct and incorrect predictions made by a classifier. The four main components of a confusion matrix are mentioned in the figure 4-5 below.



		Predicted	
		Negative (N) -	Positive (P) +
Actual	Negative -	True Negative (TN)	False Positive (FP) Type I Error
	Positive +	False Negative (FN) Type II Error	True Positive (TP)

Figure 4-5: Confusion matrix components

4.3 Experimental materials

4.3.1 Software Tools

Machine learning and Deep learning are fields that require substantial hardware resources, particularly GPUs and CPUs capable of performing intensive computations. For our experiment, we utilized the following resources tools:

- ✓ Python programming language;
- ✓ Google Colaboratory.

- **Python**

Python is a high-level interpreted, object-oriented programming language that is becoming increasingly popular for research in many fields due to its simplicity, readability, and adaptability. It is particularly suitable for machine learning, deep learning and many other artificial intelligence tasks, because of its ease of learning and its wide availability of libraries designed for these specific applications [39].

- **Google Colaboratory**

While there are several Python Integrated Development Environments (IDEs) available (IDLE, PyCharm, Anacond, Jupyter, PyDev, Spyder, Atom, Visual Studio Code…) each with its unique features and capabilities, we opted for Google Colaboratory due to its access to free CPUs and GPUs, which is crucial for handling our large datasets. Developed by Google Research, Google Colaboratory is a free Jupyter Notebook platform, providing 12 GB of RAM, over 100 GB of storage space, and pre-installed libraries necessary for our experiments, including NumPy, Pandas, Matplotlib, and Scikit-learn [40].

4.4 Experimental Results and Discussion

4.4.1 Hyperparameters tuning models accuracy

As clearly observed from table 4-5, expect SVM and NB there were a slight difference between the models' accuracy before and after hyperparameters optimization. Therefore, hyperparameters optimization may not be so important for these models and they can simply work well with their default parameters.

Models	Models accuracy with brute force default parameters	Models accuracy after Hyperparameters optimization
DT	1	0.99607
RF	1	0.99642
SVM	0,93	0.97590
NB	0,82	0.64745
ANN	0,98	0.94503
DNN	0,98	0.96933

Table 4-5 : Default parameters and hyperparameters optimization accuracy results

4.4.2 Accuracy validation

- **Model's 5-fold CV accuracy results:**

Except NB, it is clearly observed that all models achieved a high accuracy exceeds 96%, with a very minimal standard deviation.

Models	Accuracy (5-fold CV)	Mean Accuracy	Standard Deviation
DT	[99.70%,99.70%,99.72%, 99.70%, 99.71%]	99.70%	0.01%
RF	[99.74%,99.73%,99.72%, 99.74%, 99.72%]	99.73%	0.01%
SVM	[97.48%, 97.34%, 97.42%, 97.49%, 97.51%]	97.45%	0.06%
NB	[65.99%, 62.92%, 72.23%, 62.79%, 62.67%]	65.32%	3.67%
ANN	[96.83%,97.20%,96.81%, 96.95%,97.18%]	96.99%	0.16%
DNN	[97.54%,97.48%,97.75%, 97.54%,97.49%]	97.56%	0.10%

Table 4-6: 5-fold CV accuracy results

- **CIC-IDS2017 Training dataset models evaluation:**

After confirming that the models had achieved satisfactory accuracy, they were retrained with the optimal hyperparameters and then tested using the CIC-IDS2017. However, they were tested using CSE-CIC-IDS2018 and CIC-DDoS2019. The results in Table8 and the confusion matrix of each model in figure 4-6 indicate that DT and RF perform the best, closely followed by DNN, SVM and ANN. NB exhibit inferior performance biased towards the benign class.

Model	Precision		Recall		F1-Score		Accuracy
	Benign	Malicious	Benign	Malicious	Benign	Malicious	
DT	0.9988	0.9948	0.9948	0.9988	0.9968	0.9968	**0.9968**
RF	0.9984	0.9963	0.9963	0.9984	0.9974	0.9973	**0.9973**
SVM	0.9960	0.9515	0.9499	0.9961	0.9724	0.9733	0.9729
NB	0.6958	0.7553	0.7928	0.6486	0.7411	0.6979	0.7212
ANN	0.9912	0.9465	0.9447	0.9915	0.9674	0.9685	0.9680
DNN	0.9960	0.9516	0.9500	0.9961	0.9725	0.9733	0.9729

Table 4-7: CIC-IDS2017 models performance

- **CSE-CIC-IDS2018 Testing dataset models evaluation:**

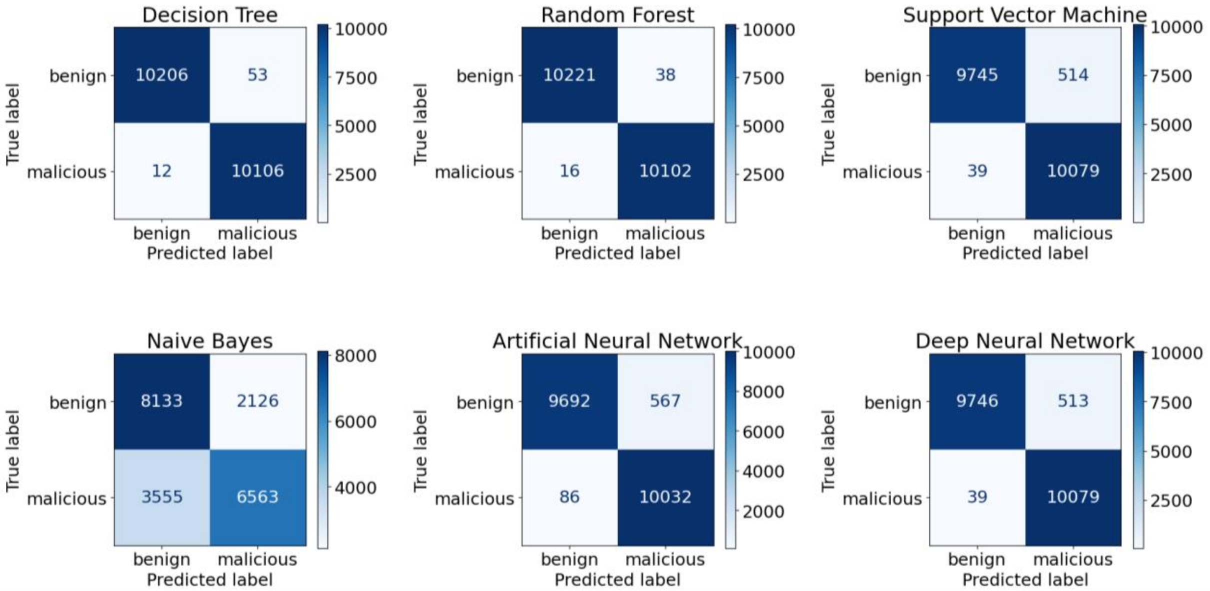


Figure 4-6: CIC-IDS2017 confusion matrix

Table 4-8 and figure 4-7 shows the results obtained using CSE-CIC-IDS2018 testing dataset. All the models exhibit notably poor performance, except SVM with an accuracy of 71% followed by DT with 67% accuracy. Furthermore, the confusion matrix also show that all the models suffer from overfitting (bias towards malicious samples), where SVM and DT are less sensitive.

Model	**Precision**		**Recall**		**F1-Score**		**Accuracy**
	Benign	Malicious	Benign	Malicious	Benign	Malicious	
DT	0.6149	0.9030	0.9575	0.3979	0.7489	0.5524	**0.6783**
RF	0.5735	0.8737	0.9592	0.2835	0.7178	0.4281	0.6221
SVM	0.6492	0.8619	0.9201	0.5007	0.7613	0.6334	**0.7109**
NB	0.5868	0.6557	0.7570	0.4648	0.6611	0.5440	0.6112
ANN	0.5862	0.8460	0.9394	0.3341	0.7219	0.4791	0.6374
DNN	0.6073	0.8643	0.9390	0.3903	0.7376	0.5378	0.6652

Table 4-8: CSE-CIC-IDS2018 models performance

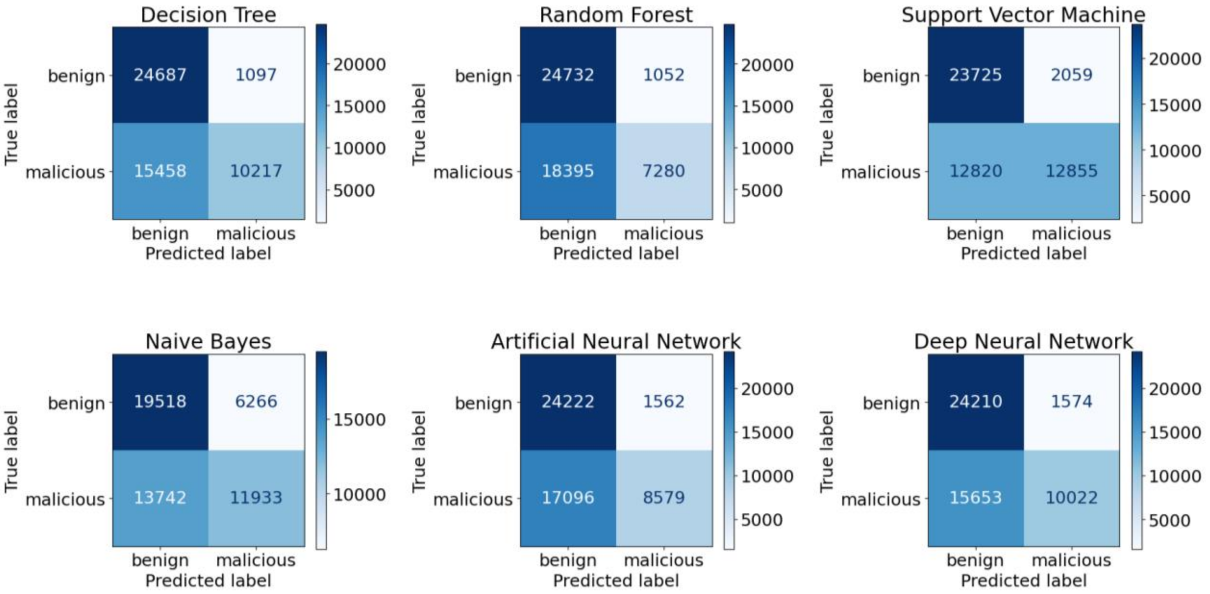


Figure 4-7: CSE-CIC-IDS2018 confusion matrix

- **CIC-DDoS2019 Testing dataset models evaluation:**

While results conducted using CIC-DDoS2019 testing dataset shows the worst performance, except NB with 45% accuracy. All the models here suffer the most from overfitting. These details are outlined in table 4-9 and figure 4-8.

Model	**Precision**		**Recall**		**F1-Score**		**Accuracy**
	Benign	Malicious	Benign	Malicious	Benign	Malicious	
DT	0.0582	0.4583	0.9774	0.0012	0.1098	0.0024	0.0592
RF	0.0590	0.2500	0.9922	0.0002	0.1113	0.0003	0.0591
SVM	0.0582	0.9145	0.9357	0.0434	0.1095	0.0829	0.0946
NB	0.0943	0.9920	0.9452	0.4272	0.1716	0.5972	**0.4579**
ANN	0.0572	0.8520	0.9391	0.0221	0.1078	0.0431	0.0766
DNN	0.0575	0.8958	0.9304	0.0378	0.1084	0.0725	0.0908

Table 4-9: CIC-DDoS2019 models performance

4.4.3 Comparison between the performance of CIC's Datasets

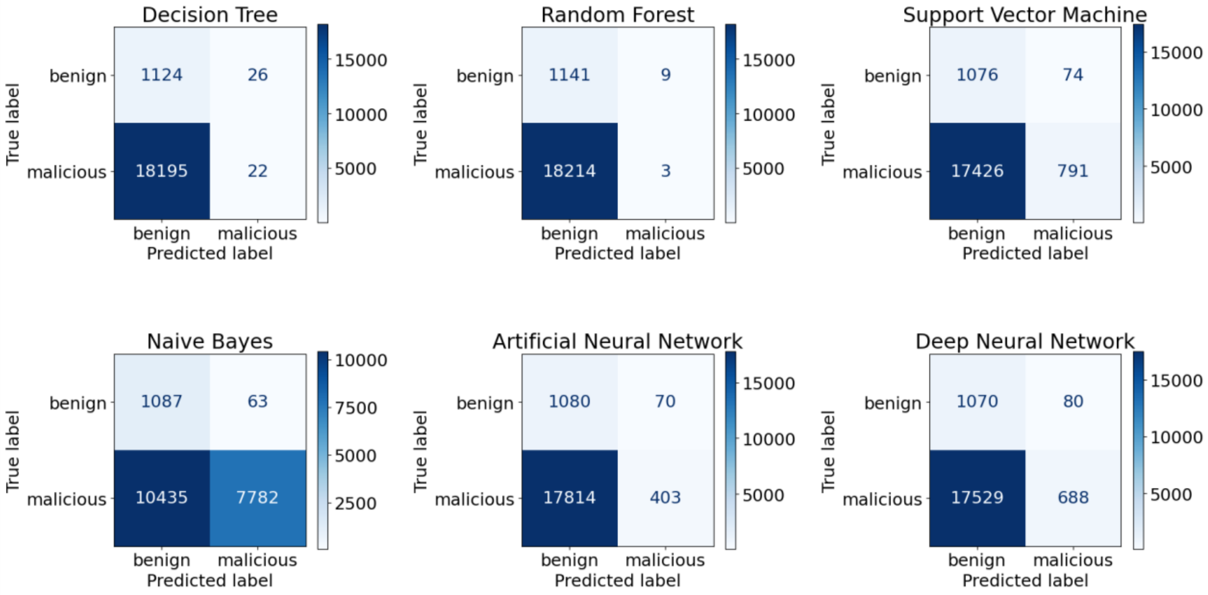


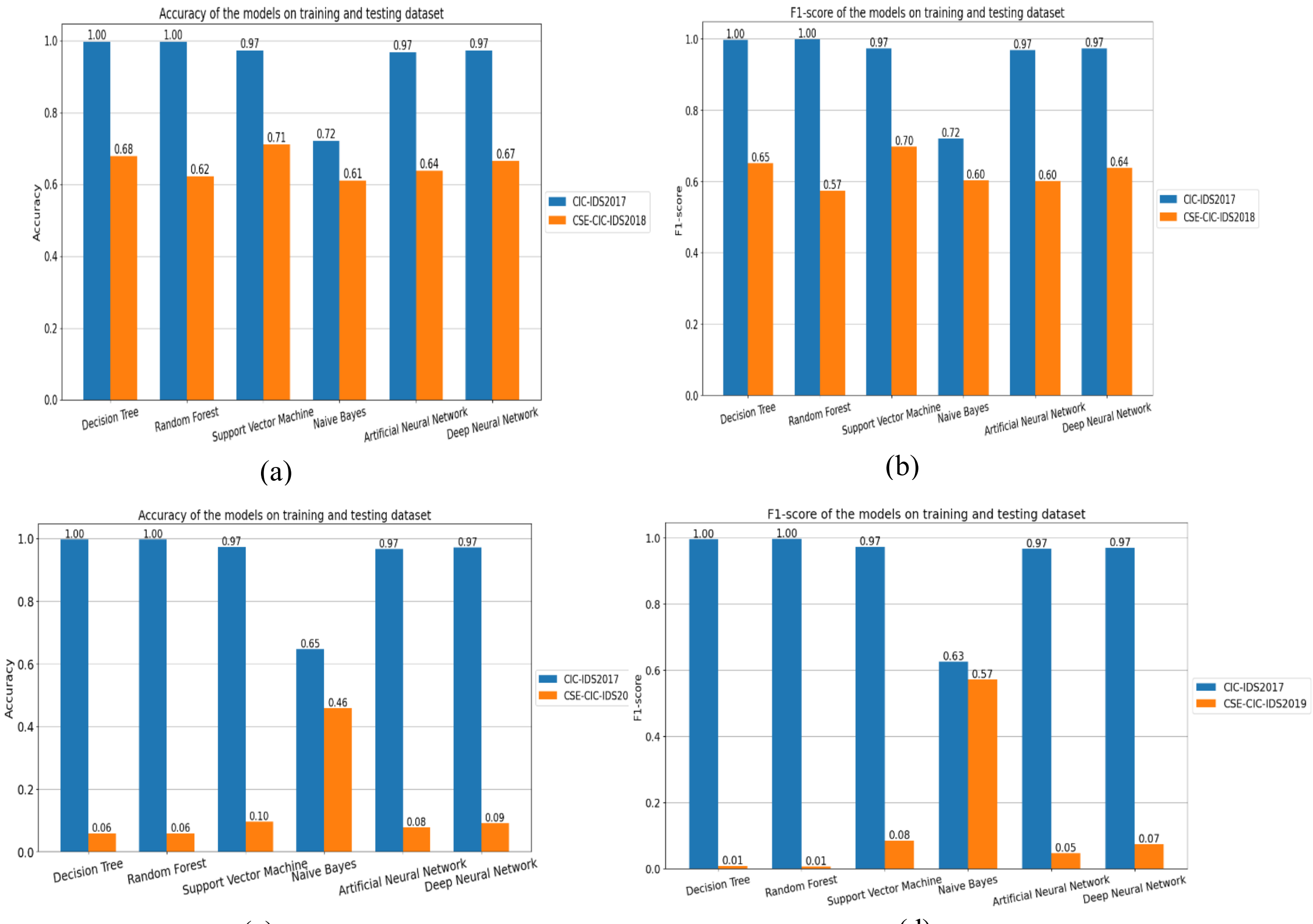
Figure 4-8 : DDoS2019 confusion matrix

The comparison between the training dataset, CIC-IDS2017, and the testing datasets: CSE-CIC-IDS2018 and CIC-DDoS2019, was a critical aspect of this study, as it aimed to evaluate the performance of the selected models over time. As CSE-CIC-IDS2018 and CIC-DDoS2019 were created later than the latter are created later than the Specifically, the assessment focused on the accuracy and F1-score of each model across the 3 datasets. This analysis is presented in detail in Figures 4-9.

In fact, this analysis can better determine which model is most affected by overfitting over time. According to figure 4-9 (a), we clearly observe that NB has the smallest drop in terms of both accuracy and F1-score, followed by SVM and DNN. For DT and RF, both accuracy and

F1-score drop significantly on the testing dataset. As a result, NB and SVM are less prone to overfitting compared to DT, RF, ANN.

On the other hand, from figure 4-9 (d), we can see that NB has the smallest drop in term of f1-score and accuracy as well. In contrast, the other models show a dramatic drop in these metrics.



(a)

(b)

(c)

(d)

Figure 4-9: The accuracy and F1-score of the 3 CIC's datasets (1 training set and 2 testing set)

In terms of time consumption, SVM is most expensive to train and test. DT and NB are the most efficient models for both training and prediction. Whine ANN and DNN have significantly longer training times compared to the other models, their testing times are considerably reduced. Additionally, the time required for RF consistently decreases. These details are visualized in figure 4-10.

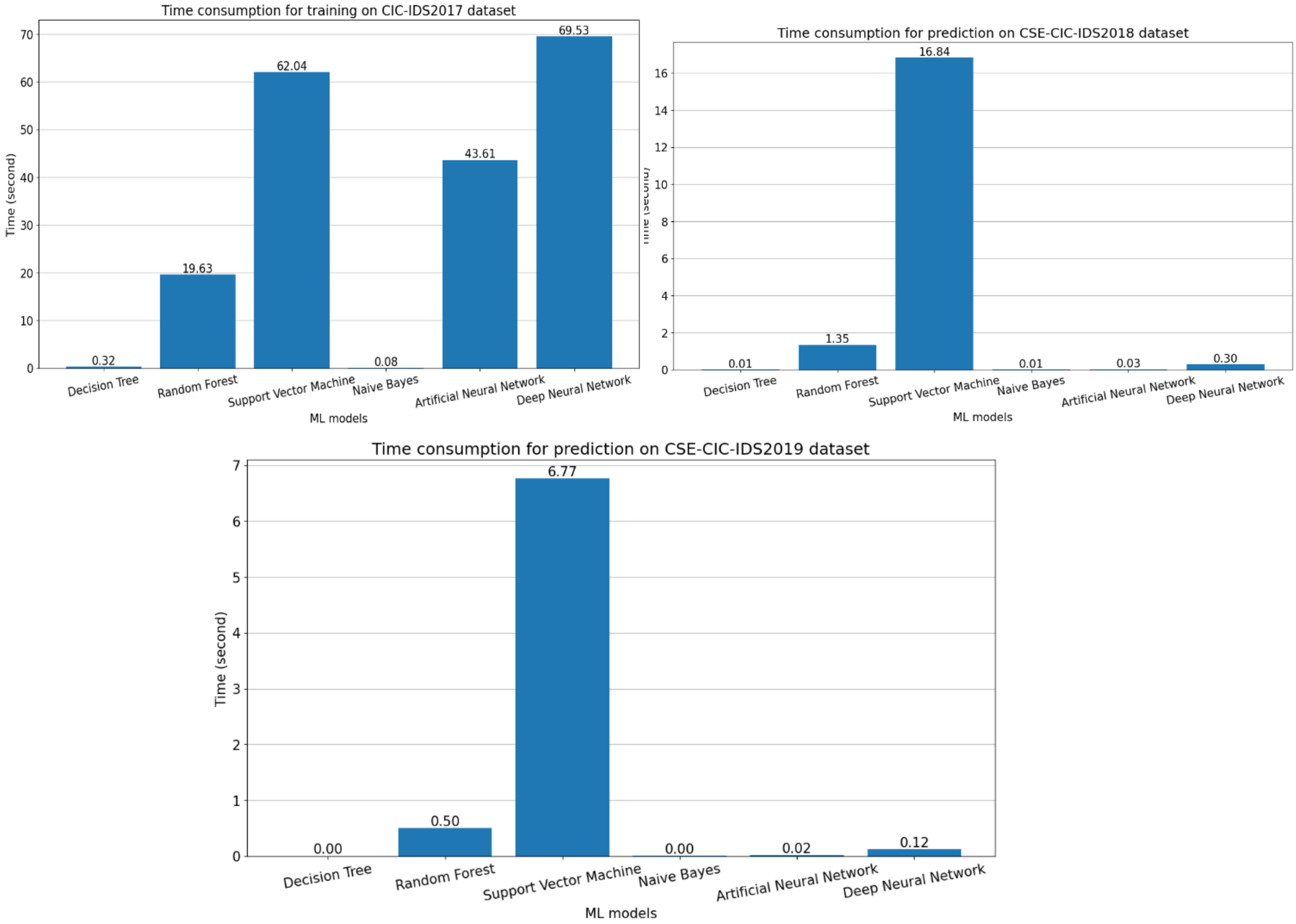


Figure 4-10: The training and prediction time of the 3 CIC's datasets (1 training set and 2 testing set)

4.5 Comparison with Previous Methods

Models	**Our work**	**[41]**	**[42]**
DT	**1.00**	0.95	0.94
RF	**1.00**	0.94	0.94
SVM	**0.93**	-	0.80
NB	**0.82**	**0.87**	0.31
ANN	**0.98**	0.97	0.96
DNN	**0.98**	-	0.94

Table 4-10: Comparison with Previous Methods

4.6 Conclusion

In conclusion, this chapter discussed a step-by-step implementation of our proposed workflow using 6 models and 3 distinctive datasets.

CHAPTER 5:

Conclusion and Future Direction

5 Conclusion and Future Direction

5.1 Conclusion

The work summarized in this dissertation presents a brief overview and a comprehensive analysis of multiple machine and deep learning algorithm-based IDS for attack detection. This research provides new insights into understanding the evolution of network traffics, network attackers and developers and deciphering the precise dynamic evolution of IDS systems.

The study of the zero-day attack detection problem from different training and testing datasets has allowed us to propose an effective learning model. We followed a framework that was highly accurate in elucidating the IDS's ability to detect unknown attacks by evaluating the long-term performance of various sophisticated ML and DL models.

Concerning the literature review, we provided briefs information on cybersecurity fundamentals, various types of cyberattacks, and intrusion detection systems, which are among the promising countermeasures due to their high ability to detect abnormal activities in the target network. We also discussed different machine learning and deep learning methods that are widely used in previous literatures. These methods were successfully applied in this study but with different training and testing datasets, two of which were created later than the first one.

For practical experimentation, we followed a framework with a successive sequence of analysis steps to achieve our objectives.

5.2 Future Direction

Along with this research, some directions for further inquiry are listed below:

1. Detecting zero-day attacks and other malicious activities by fusing multiple network traffic datasets.
2. Evaluating the long-term performance of intrusion detection systems by adopting unsupervised ML models and other modern artificial intelligence approaches.
3. Detecting zero-day attacks and other malicious activities using multi-classification.
4. Enhancing the feature selection method used in this dissertation, as the chosen features significantly influence the models performance. Improving feature selection could potentially reduce the overfitting observed in our initial experiments.

Reference

[1] Verma, A.; Ranga, V. On evaluation of Network Intrusion Detection Systems: Statistical analysis of CIDDS-001 dataset using machine learning techniques. *Pertanika J. Sci. Technol.* **2018**, *26*, 1307–1332.

[2] Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2020**, *50*, 102419.

[3] Kilincer, I.F.; Ertam, F.; Sengur, A. Machine learning methods for cyber security intrusion detection: Datasets and comparative study. *Comput. Netw.* **2021**, *188*, 107840.

[4] Hindy, H.; Atkinson, R.; Tachtatzis, C.; Colin, J.N.; Bayne, E.; Bellekens, X. Utilising deep learning techniques for effective zero-day attack detection. *Electronics* **2020**, *9*, 1684.

[5] D. Denning, 'Cryptography and Data Security', *SERBIULA (sistema Librum 2.0)*, Jan. 1982.

[6] 'What is Cybersecurity? Types, Threats and Cyber Safety Tips', www.kaspersky.com. Available: https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security

[7] R. VON Solms and J. VAN Niekerk, 'From information security to cyber security', *Computers & Security*, vol. 38, pp. 97–102, Oct. 2013, doi: 10.1016/j.cose.2013.04.004.

[8] 'The CIA Triad — Confidentiality, Integrity, and Availability Explained', freeCodeCamp.org. Available: https://www.freecodecamp.org/news/the-cia-triad-confidentiality-integrity-and-availability-explained/

[9] A. P. Namanya, A. Cullen, I. Awan, and J. Pagna Diss, *The World of Malware: An Overview*. 2018. doi: 10.1109/FiCloud.2018.00067.

[10] J.-J. M. Malasi, 'Potentiel de la fouille des données en cybersécurité', masters, Université du Québec en Outaouais, Gatineau, 2023.

[11] G. C. School, 'Types d'attaques cyber et catégorie d'attaque informatique'. Available: https://guardia.school/boite-a-outils/panorama-des-attaques-cyber.html

[12] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, 'Survey of intrusion detection systems: techniques, datasets and challenges', *Cybersecurity*, vol. 2, no. 1, p. 20, Jul. 2019, doi: 10.1186/s42400-019-0038-7.

[13] C. Chapman Chris, *Network Performance and Security: Testing and Analyzing Using Open Source and Low-Cost Tools*. in Elsevier Science. 2016.

[14] 'What is a Zero-Day Exploit? | IBM' Available: https://www.ibm.com/topics/zero-day

[15] Y. Guo, 'A review of Machine Learning-based zero-day attack detection: Challenges and future directions', *Computer Communications*, vol. 198, Nov. 2022, doi: 10.1016/j.comcom.2022.11.001.

[16] S. Ali, S. Rehman, A. Imran, G. Adeem, Z. Iqbal, and K.-I. Kim, 'Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection', *Electronics*, Nov. 2022, doi: 10.3390/electronics11233934.

[17] R. Kalakuntla, A. Vanamala, and R. Kolipyaka, 'Cyber Security', *Holistica*, vol. 10, pp. 115–128, Aug. 2019, doi: 10.2478/hjbpa-2019-0020.

[18] K. Narayanasamy and P. Arumugam, *S2SE:An Encryption Methodology*. 2013.

[19] 'What Is a VPN Tunnel?', Palo Alto Networks. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-a-vpn-tunnel

[20] M. Dahmani and K. Ouardani, 'Proposition d'un outil d'assistance pour la construction des systèmes de détection d'intrusion', Master thesis, niversité du Ibn Khaldoun, TIARET, 2020.

[21] 'Les IDS par la pratique : Snort'. Available: https://www-igm.univ-mlv.fr/~dr/XPOSE2004/IDS/IDSSnort.html

[22] I. F. Kilincer, F. Ertam, and A. Sengur, ‘Machine learning methods for cyber security intrusion detection: Datasets and comparative study’, *Computer Networks*, vol. 188, p. 107840, Apr. 2021, doi: 10.1016/j.comnet.2021.107840.

[23] “The Timeline of Artificial Intelligence - From the 1940s,” Verloop.io Available: https://verloop.io/blog/the-timeline-of-artificial-intelligence-from-the-1940s/

[24] ‘Définition et Histoire - Qu’est-ce que l’intelligence artificielle ?’, Intelligence artificielle & Data Analytics. Available: https://ia-data-analytics.fr/intelligence-artificielle/

[25] S. G a, ‘The Review of Artificial Intelligence in Cyber Security’, *International Journal of Applied Science and Engineering Research*, vol. 10, pp. 1461–1468, Jan. 2022, doi: 10.22214/ijraset.2022.40072.

[26] K. Shaukat, S. Luo, V. Varadharajan, I. Hameed, and M. Xu, ‘A Survey on Machine Learning Techniques for Cyber Security in the Last Decade’, *IEEE Access*, 2020, Available: https://www.semanticscholar.org/paper/A-Survey-on-Machine-Learning-Techniques-for-Cyber-Shaukat-Luo/4c30d041c137948aa75e40912f14558234bf1ce2

[27] A. Sakina, ‘Présenté en vue de l’obtention du diplôme de MASTER Filière : (Mathématiques/Informatique) Option : Réseaux & Sécurité informatique’.

[28]“What is a Decision Tree? | IBM.” Available: https://www.ibm.com/topics/decision-trees

[29] “What Is Random Forest? | IBM.” Available: https://www.ibm.com/topics/random-forest

[30] M. N. Samir, ‘Optimisation des IDS du Cloud Computing par les techniques de machines Learning’.

[31] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, ‘Survey of intrusion detection systems: techniques, datasets and challenges’, *Cybersecurity*, vol. 2, no. 1, p. 20, Jul. 2019, doi: 10.1186/s42400-019-0038-7.

[32] J. Kacprzyk, M. Ezziyyani, and V. E. Balas, Eds., *International Conference on Advanced Intelligent Systems for Sustainable Development: Volume 3 - Advanced Intelligent Systems on Agriculture and Health*, vol. 713. in Lecture Notes in Networks and Systems, vol. 713. Cham: Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-35248-5.

[33] Augusta, ‘3 algorithmes de Machine Learning utiles pour votre business’, Invenis. Available: https://invenis.co/blog/3-algorithmes-de-machine-learning-bien-utiles-business

[34] “Qu’est-ce qu’un réseau de neurones ? | IBM.” Available: https://www.ibm.com/fr-fr/topics/neural-networks

[35] D. Nyale and S. Angolo, ‘A Survey of Artificial Intelligence in Cyber Security’, *International Journal of Computer Applications Technology and Research*, pp. 474–477, Dec. 2022, doi: 10.7753/IJCATR1112.1014.

[36] ‘Difference between a Neural Network and a Deep Learning System’, GeeksforGeeks. Available: https://www.geeksforgeeks.org/difference-between-a-neural-network-and-a-deep-learning-system/

[37] ‘IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB’. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[38] ‘IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB’ Available: https://www.unb.ca/cic/datasets/ids-2018.html

[39] A. Rayhan and D. Gross, *The Rise of Python: A Survey of Recent Research*. 2023. doi: 10.13140/RG.2.2.27388.92809.

[40] ‘colab.google’, colab.google. Available: http://0.0.0.0:8080/

[41] Kostas, K. Anomaly Detection in Networks Using Machine Learning. Master’s Thesis, University of Essex, Colchester, UK, 2018.

[42] Vinayakumar, R.; Alazab, M.; Soman, K.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep learning approach for intelligent intrusion detection system. *IEEE Access* **2019**, *7*, 41525–41550.