People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Larbi Ben M'hidi University of Oum El Bouaghi
Faculty of Exact Sciences and Sciences of Nature and Life
Department of Mathematics and Computer Science
Research Laboratory in Computer Science's Complex Systems ReLa(CS)^2

Master Thesis

Submitted in partial fulfillment of the requirements for the degree of
Master in Computer Science
Option:Artificial vision

A reinforcement learning based intrusion detection system for MANETs.

By:
LAALA Youcef

Presented for defense in a public examination on: June 24 th 2023.

Supervised by:

NASRI Ahlem.
Associate Professor.
University of Oum El Bouaghi.

Examination Committee:

Dr. Mazziz Asma	University of Oum El Bouaghi	Chairman
Dr. Bazziz Mohammed	University of Oum El Bouaghi	Examiner

2022/2023


بسم الله الرحمن الرحيم

Acknowledgements

Firstly, I thank ALLAH for giving me the courage, the health and the motivation to finish this dissertation in the best conditions.

I would like to thank very warmly my supervisor DR. Ahlem Nasri, for having proposed this project to me, and for guiding me to the end of this work thanks to these orientations and precious advice. I also thank her for her efforts and her patience.

Sincere thanks to DR. mazziz asma , Doctor at Oum El Bouaghi University, for agreeing to be president of the jury of my project.

Sincere thanks to DR. mohammed bazziz, Doctor at Oum El Bouaghi University, for for agreeing to be an examiner of the jury of my project.

Last but not least, I would like to thank my family father and mother. and my brothers.

Youcef Laala

Abstract

The attention given to mobile ad hoc networks (MANET) is currently significant owing to their potential to significantly influence various real-world applications, including banking, medicine, and even the military sector. With the increasing use of MANETs in various applications, securing these networks against malicious intrusions has become a major concern. Intrusion detection systems (IDS) are among the best solutions to address intrusions and malicious behaviors. However, traditional intrusion detection systems often struggle to cope with the dynamic and decentralized nature of MANETs. In this thesis, we propose a new approach to intrusion detection in MANETs using reinforcement learning (RL). We leverage RL capabilities to enable the system to learn from its interactions with the environment and improve its detection accuracy over time. The obtained results demonstrate its effectiveness in detecting several categories of intrusions while minimizing false positives. Moreover, the system exhibits adaptability and robustness to changes in network conditions and attack strategies.

Keywords: Intrusion Detection Systems (IDS), Mobile Ad hoc Networks, Deep Learning approaches, Reinforcement Learning.

Résumé

L'attention accordée aux réseaux mobiles ad hoc (MANETs) est actuellement importante en raison de leur potentiel à influencer considérablement diverses applications réelles, notamment la banque, la médecine et même le secteur militaire. Avec cette utilisation croissante des MANETs dans diverses applications, la sécurisation de ces réseaux contre les intrusions malveillantes est devenue une préoccupation majeure. Les systèmes de détection d'intrusions (IDS) sont parmi les meilleures solutions pour faire face aux intrusions et aux comportements malicieux. Cependant, les systèmes traditionnels de détection d'intrusion ont souvent du mal à faire face à la nature dynamique et décentralisée des MANET. Dans cette thèse, nous proposons une nouvelle approche pour la détection d'intrusions dans les MANETs en utilisant l'apprentissage par renforcement (RL). Nous tirons parti des capacités RL pour permettre au système d'apprendre de ses interactions avec l'environnement et d'améliorer sa précision de détection au fil du temps. Les résultats obtenus démontrent son efficacité à détecter divers types d'intrusions tout en maintenant un faible taux de faux positifs. De plus, le système présente une adaptabilité et une robustesse aux changements des conditions du réseau et des stratégies d'attaque.

Mots clés: Les réseaux Ad-hoc , Les système de détection d'intrusions, Les approches de l'apprentissage approfondie, L'apprentissage par renforcement.

الملخـص

يعد الاهتمام الموجه إلى شبكات المتحركة (MANETs) كبيرًا حاليًا بسبب قدرتها على التأثير بشكل كبير على تطبيقات المختلفة في الواقع ، بما في ذلك الخدمات المصرفية والطب وحتى القطاع العسكري. مع هذا الاستخدام المتزايد لـ MANETs في مختلف التطبيقات ، أصبح تأمين هذه الشبكات ضد الاختراقات الخطيرة مصدر قلق كبير. تعد أنظمة كشف التسلل (IDS) من بين أفضل الحلول للتعامل مع عمليات التطفل والسلوك الضار. ومع ذلك ، غالبًا ما تكافح أنظمة الكشف عن التسلل التقليدية للتعامل مع الطبيعة الديناميكية واللامركزية لـ MANETs. في هذه الأطروحة ، نقترح نهجًا جديدًا للكشف عن عمليات الاقتحام في MANETs باستخدام التعلم المعزز (RL). نحن نستفيد من قدرات RL للسماح للنظام بالتعلم من تفاعلاته مع البيئة وتحسين دقة الكشف بمرور الوقت. تظهر النتائج التي تم الحصول عليها فعاليتها في الكشف عن أنواع مختلفة من التدخلات مع الحفاظ على معدل منخفض من الإنذارات الكاذبة. بالإضافة إلى ذلك ، يُظهر النظام القدرة على التكيف والمتانة للتغيرات في ظروف الشبكة واستراتيجيات الهجوم.

.

الكلمات الرئيسية: أنظمة الكشف عن التسلل ، شبكات المتحركة المخصصة (MANETs) ، مناهج التعلم العميق، التعلم المعزز.

Contents

List of Figures

List of Tables

General Introduction

The rapid growth and extensive utilization of Mobile Ad hoc Networks (MANETs) have brought about numerous advantages and opportunities for communication and collaboration in dynamic and decentralized environments. MANETs have found widespread applications in various domains, including banking, medicine, and even the military sector. These networks effectively facilitate information exchange among participants in a meeting, actors involved in rescue operations, workers on a construction site, or elements engaged in a battlefield scenario. However, the inherent characteristics of MANETs, such as their open nature, limited resources, and lack of a centralized infrastructure, expose them to various security vulnerabilities and attacks. The reliance on radio interfaces for communication makes MANETs particularly susceptible to different forms of attacks. Additionally, the implementation of security mechanisms designed for wired networks poses challenges, and in some cases, proves impossible for MANETs. Quencequantly, ensuring the security and integrity of data transmission in MANETs has become a critical challenge.

However, intrusion Detection Systems (IDS) is the most important detection tool and defense against sophisticated and growing network attacks. It is a mechanism used to analyze network traffic and supervise network activities in order to identify all abnormal or suspicious activities. Thus, it is affirmed that IDS play a crucial role in identifying and moderating security risks in MANETs. Traditional IDS approaches, such as rule-based and anomaly-based systems, have shown limitations in effectively detecting and adapting to the ever-evolving attack strategies in dynamic network environments. Therefore, there is a need for innovative and adaptive approaches that can enhance the detection capabilities of IDS in MANETs.

Reinforcement learning (RL) is one from a particularly effective technique in this field. Automated learning algorithms can be used to categorize network traffic as either normal or malicious based on known patterns. It offers a dynamic and adaptive tool that enables intrusion detection systems to continuously learn and improve their detection capabilities. Therefore, the integration of reinforcement learning techniques into intrusion detection systems offers great potential to enhance the security posture of this type of network and can reduce the risks posed by emerging attacks.

This thesis aims to address the limitations of existing IDS solutions by leveraging the power of reinforcement learning techniques. By utilizing reinforcement learning algorithms, this thesis aims to develop an intelligent and adaptive IDS that can learn from interactions with the network environment, make informed decisions, and effectively detect and respond to security

attacks in real-time.

This thesis is organized as follows: chapter one aiming to present the theoretical foundations and key aspects relevant to the field of mobile ad hoc networks. The chapter looks at the importance and challenges associated with mobile ad hoc networks, more specifically their security, by exploring topics such as their characteristics, network architectures and vulnirabilities.
Chapter Two of the thesis is dedicated to exploring the most effective solutions for ensuring the security of mobile ad hoc networks (MANETs). This chapter delves into the concept of intrusion detection systems (IDS) and provides a detailed description of their functionalities and various types. By examining different IDS approaches, including anomaly-based and signature-based detection, chapter two aims to offer a comprehensive understanding of the diverse strategies available for detecting and preventing security attacks in MANETs.
Chapter Three of the thesis is dedicated to exploring the application of deep learning approaches in the field of intrusion detection systems, with a particular focus on reinforcement learning techniques. This chapter provides an in-depth analysis of the existing literature and related works in this area. By examining various studies, research papers, and case studies, chapter three aims to provide a comprehensive overview of the advancements and contributions made in the field of intrusion detection using deep learning and reinforcement learning approaches. The chapter explores the benefits, challenges, and practical considerations associated with implementing these techniques in the context of intrusion detection systems for enhanced security in MANETs network.
Chapter Four of the thesis presents the proposed solution for addressing the security challenges in mobile ad hoc networks. This chapter focuses on the design and implementation of the proposed solution, providing a detailed discussion of the methodology and techniques employed. The chapter also presents the obtained results, including performance evaluations and analyses of the solution's effectiveness.

Chapter 1

Mobile Ad-hoc networks

Contents

1.1 Introduction:

Ad hoc networks are decentralized wireless networks that do not rely on centralized infrastructure. Instead, the devices within the network communicate directly with each other, creating a fast and easy-to-set-up self-organizing network. One specific type of ad hoc network that is the focus of our work is the mobile ad hoc network (MANETs). MANETs consist of mobile devices that can dynamically form a network without the need for a pre-existing infrastructure. These networks are highly adaptable and can be quickly deployed in various environments, making them suitable for scenarios such as military operations, emergency response, and temporary communication setups.[1]
In this chapter, we will examine the challenges and opportunities associated with ad hoc networks in general and especialy mobile ad hoc networks (MANETs). We will explore the specific characteristics of MANETs, with a focus on their dynamism, resource constraints, and associated security issues.

1.2 Definition of Ad hoc Network

Ad hoc networks are a new paradigm of wireless communication for mobile hosts (called nodes). Ad-hoc networks do not have fixed infrastructure such as base stations or mobile switching centers. Mobile nodes within radio range of each other communicate directly over the wireless link, while mobile nodes that further away rely on other nodes to relay messages. Node mobility in ad-hoc networks causes frequent changes in network topology. Figure 1.1 shows an example. Initially, nodes A and D are directly connected. When D moves out of the radio range from A, the connection is broken. However, network A is still connected because D is reachable through C, E, and F.
Military tactical operations are still the main application of ad-hoc networks today. B. Radio-equipped military units (soldiers, tanks, planes, etc.) can form ad-hoc units. Connect as you move across the battlefield. Ad hoc networks can also be used forever further away rely on, and rescue operations. Ad-hoc networks are relatively inexpensive and quick to deploy, making them an attractive option for commercial applications such as sensor networks and virtual claThe connection is broken when D moves out of the radio range from A . [1]

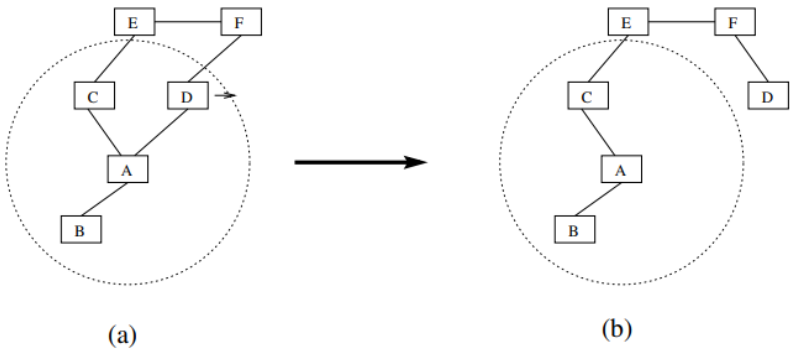

Figure 1.1: Topology Changed in Ad hoc Networks.[1]

1.3 History of Ad hoc Network

Ad hoc networks are not a new technology and have been in development for over 30 years. In the past, years of R&D activity have been primarily for the US government, and more specifically for the Defense Advanced Research Projects Agency (DARPA). This chapter describes the history of ad-hoc networks. Introducing the most important projects in the field of ad-hoc networking. This story is summarized in Figure 1.2.[2]

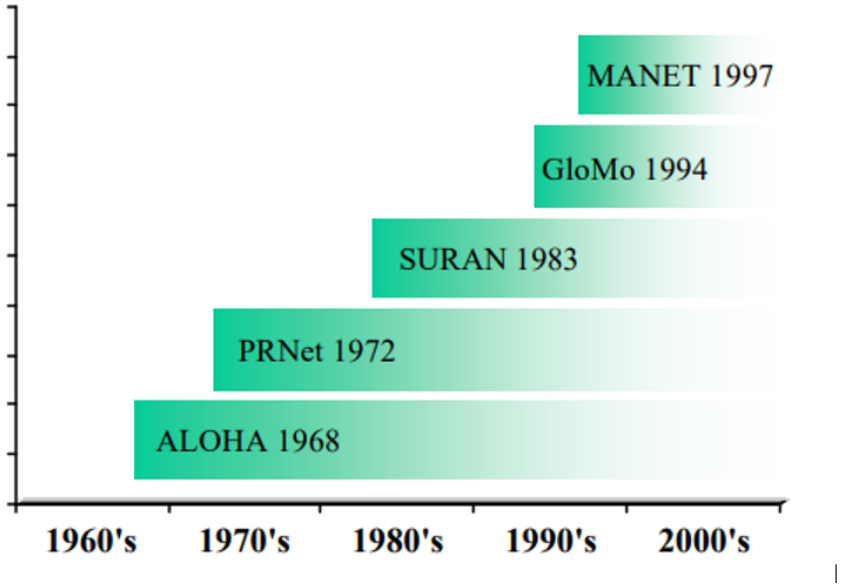

Figure 1.2: The history of ad hoc networks.[2]

1.4 Types of Ad hoc Networks

1.4.1 Wireless Sensor Network(WSN)

A wi-fi sensor community is a set of nodes prepared right into a cooperative community. Each node includes processing capability (one or extra micro controllers, CPUs, or DSP chips), can also additionally include more than one variety of memory (program, data, and flash memories), has an RF transceiver (commonly with an unmarried omnidirectional antenna), have an energy source (e.g., batteries and solar cells), and accommodate numerous sensors and actuators (as figure 1.3 illustrates). The nodes talk wirelessly and frequently self-arrange after being deployed in an advert hoc fashion. Systems of 1000's or maybe 10,000 nodes are anticipated. Such structures can revolutionize the manner we stay and work.
Currently, wi-fi sensor networks are starting to be deployed at an increased pace. It is not unreasonable to anticipate that during the 10-15, year processing arena may be protected with wi-fi sensor networks with getting admission to them thru the Internet. This may be taken into consideration because the Internet turning into a bodily community. This new era is interesting with limitless capability for several utility regions along with environmental, medical, military, transportation, entertainment, disaster management, fatherland defense, and clever spaces.[23]

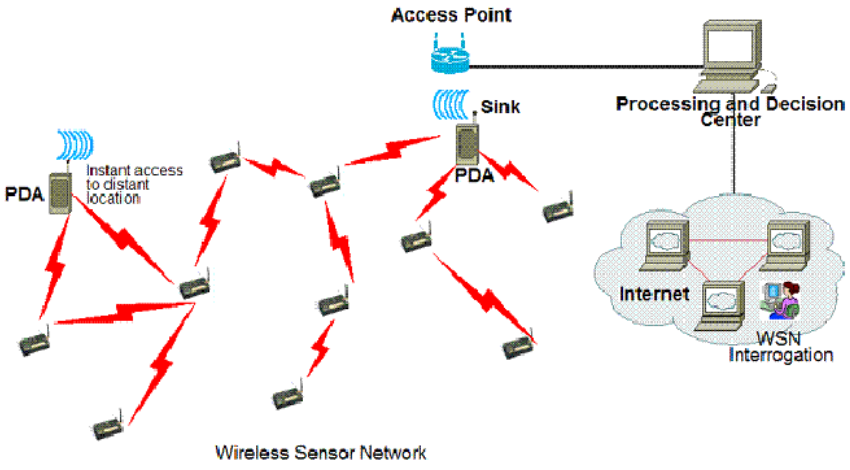

Figure 1.3: Architecture of wireless sensor networks[3]

1.4.2 Wireless Mesh Network(WMN)

A communication network of wireless nodes organized in a mesh topology. A wireless mesh network often consists of mesh clients, mesh rout, and gateways. Mesh clients are often laptops, cell phones, and other wireless devices, and mesh routers route traffic to and from gateways. The gateway can connect to the Internet, but it is not required. A coverage area of wireless nodes operating as a single network is sometimes referred to as a mesh cloud. Access to this mesh cloud relies on wireless nodes working together to create a wireless network.[24]

1.4.3 Mobile Ad hoc Networks (MANET)

Mobile Ad Hoc Networking (a.kla.Mobile Packet Radio Networking) is a call presently being given to a generation below improvement for the beyond 20 or so years, thru studies investment backed with the aid of using the U.S. Government. Its preliminary sponsors blanketed the Defense Advanced Research Projects Agency (DARPA), the U.S. Army, and the Office of Naval Research (ONR). Significant early packet radio packages blanketed the Survivable, Adaptive Networks (SURAN) Program, the Low-fee Packet Radio (LCR) Program, and the Survivable Communication Networks (SCN) Program. Today, government-backed paintings continue to be underway in networking packages inclusive of the Tactical Internet and the Near-Term Digital Radio (NTDR).[25]

1.4.4 Fly Ad-Hoc Network (FANET)

FANET is a similar subclass to MANET.In other words, FANET also has mobility commonality. FANET is a special class of MANET,VANET. That is, we have something in common Mobility of FANETs as well as MANETs and VANETs. A FANET is a type of network made up of mobile agents. It's called a very small aircraft (MAV). those flight agents Form a group of

small MAVs linked on an ad-hoc basis To communicate for necessary purposes. The presence of As with MANETs and VANETs, these flying agents typically lead to frequent changes in the FANET's network topology NET. FANET is a type of network composed of mobile agents called Micro Aerial Vehicles (MAV). These flying agents form a group of smFANETs and are a special class of MANETs, vanette. In other words, FANET has mobility in common with his MANET and VANET. FANET is a type of network composed of mobile agents called Micro Aerial Vehicles (MAV). These flying agents form groups of small MAVs linked on an ad-hoc basis to communicate for any desired purpose. The presence of these flying agents typically causes frequent changes in the FANET network topology. All MAVs are connected ad-hoc to communicate for any purpose. The presence of these flying agents typically leads to frequent changes in the FANET's network topology. [4]

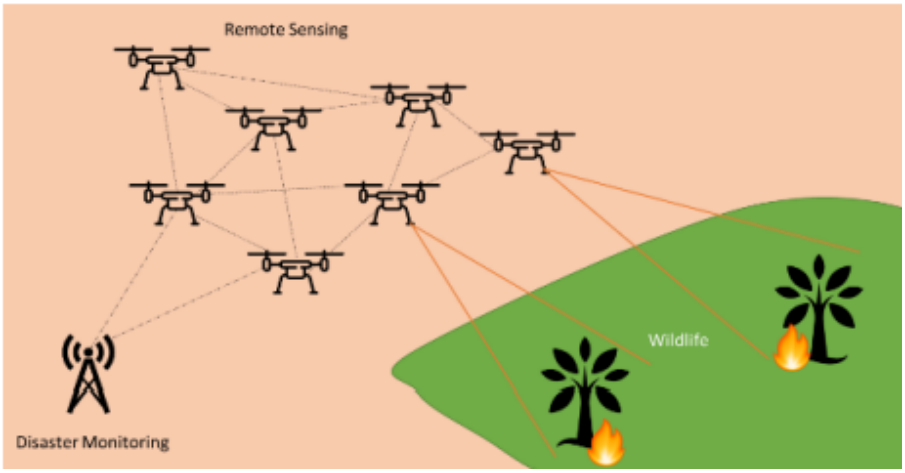

Figure 1.4: architecteur of FANET [4]

1.4.5 Vehicular Ad-Hoc Network (VANET)

VANET is a mobile radio technology The car forwards messages from one node to another through the nodes. Nodes communicate single-hop and multi-hop and also provide an extensive network for capturing signals and sending messages. Nevertheless, VANET technology improves safety and traffic. Vehicle communication is connected with nearby vehicles and the proper design of VANET enhances driving safety. VANET communications include vehicle-to-vehicle or vehicle-to-vehicle (VV) communications, vehicle-to-vehicle or vehicle-infrastructure (VI) communications, and application-to-road communications. communication.[5]

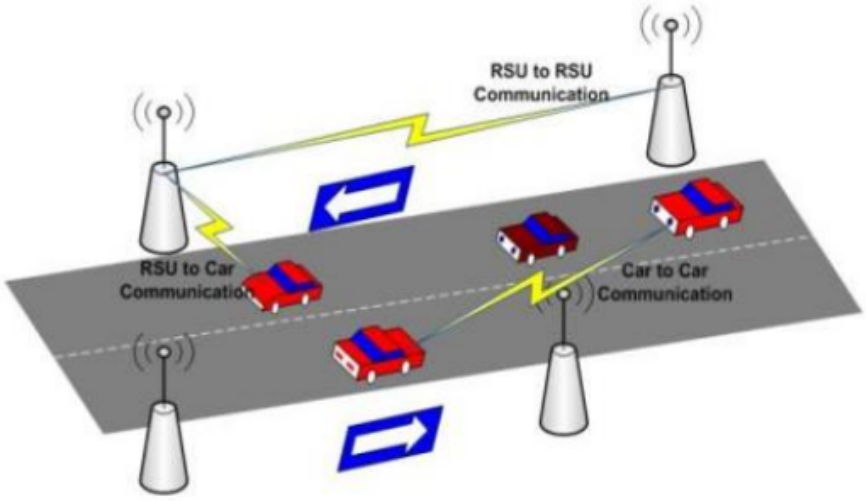

Figure 1.5: Architecture of Vehicle Ad-Hoc Network[5]

1.4.6 SANET

SANET is designed for boats, ships, submarines, Interconnected Vessels or Unmanned Surface Vehicles (USV) Form a comprehensive network. centered on the network Improved underwater connectivity. Usually,y the node is Determined by node density within a given area. Since the nodes are spread across the ocean, Knot density is moderate. In situations of high UPS mobility in SANET, the topology can change frequently compared to MANET. however, VANET and FANET, significant changes in topology Slower[4]

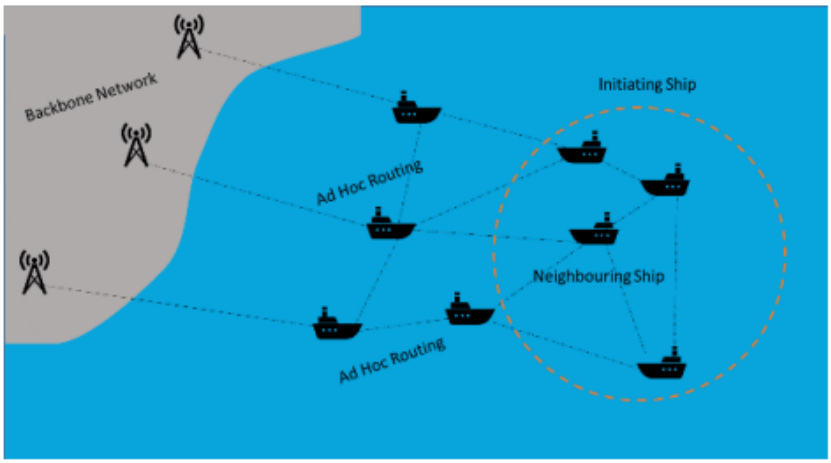

Figure 1.6: Architecture of SANET [4]

1.5 Differences Between the MANET VANET FANET SANET

This table shows MANET, VANET, FANET, and SANET.[4]

Characteristics	MANET	VANET	FANET	SANET
Node Type	Laptop, Tablet, Smartphone	Vehicle, Bus, Truck, Motorcycle, Traffic Light	Aircraft, Helicopter, Drone	Boat, Ship, Vessel
Node Setup	Positioning	Road or Infrastructure	Airfields	Water
Topology Change	Slow	Fast	Fast	Slow
Node Density	Low	High	Very High	Medium
Node Speed	Low to Medium	Medium to High	Medium to High	Medium to High
Routing Protocol	Proactive, Reactive, and Hybrid	Proactive, Reactive, and Hybrid	Proactive, Reactive, and Hybrid	Proactive, Reactive, and Hybrid
Node Mobility	Low	High	Very High	Medium
Mobility Model	Random	Regular	Regular or Random	Random
Computational Power	Low	High	High	High
Frequency Band	2.4 GHz / 5.0 GHz	5.9 GHz	2.4 GHz / 5.0 GHz	5.0 GHz

Table 1.1: Characteristics of MANET, VANET, FANET, and SANET[4]

1.6 Application of Ad Hoc networks

Self-organization, low cost, rapid deployment, and resilience of sensor networks It is an attractive solution in various industries.

- Military: It makes it possible, for example, to monitor all motions (allies or enemies)

or study the terrain before sending troops there (detection of chemical, biological, or radiological agents).

- Medical and Veterinary: Microsensors ingested or inserted under the skin may assist in the monitoring of a live organism's critical activities. For about 24 hours (battery life), it is possible to receive real-time images of a part of the body without any surgery. It's also possible to use biomedical technology in more ambitious ways. For example, blood sugar monitoring could be used to keep track of blood sugar levels, monitor vital organs, detect cancers before they spread, and keep an eye on patients.

- Environmental: Thermo-sensors may be disseminated from aircraft, balloons, and ships to monitor and report on environmental issues in the field (fires, pollution, epidemics, and meteorological dangers), allowing for increased environmental awareness and control strategy efficacy. Farmers might sow sensors with the seeds to detect plant water stress or low nutrient levels in the soil water and optimize water and fertilizer inputs, drainage, and irrigation. Sensors may be deployed in a network on industrial sites, nuclear power plants, and oil tankers to detect dangerous product leaks (gas, chemicals, radioactive materials, oil, etc.) and inform users and emergency services more rapidly, allowing for an effective response. Many microsensors could be placed all over the forest or in specific protected areas to get information about the health and behavior of wildlife, plants, and fungi (movements, activity, and so on).

- Emergency services: Search and rescue missions for people, fires, flooding, and earthquakes, with the specific aim of replacing the wired infrastructure.

- Collaborative work and communications in companies: In the context of a meeting or a conference, for example.

- Home Network: Application sharing and mobile equipment communications.

- Commercial applications: For remote electronic payment (taxi) or for mobile Internet access.

- Networks in motion: On-board computing and communicating vehicles.

- Building: During earthquakes, to help rescuers find the victims (sensors trapped in the concrete at the construction that detect the noise level).

- Industry: For inventory management.

- Transport: For traffic management.

1.7 Challenges of Ad Hoc Networks

The salient functions of ad hoc networks pose each demanding situation and possibilities in accomplishing those safety goals.

- First, the use of Wi-Fi hyperlinks renders an advert hoc community liable to hyperlink assaults starting from passive eavesdropping to energetic impersonation, message replay, and message distortion. Eavesdropping may supply an adversary get admission to mystery information, violating confidentiality. Active assaults may permit the adversary to delete messages, inject inaccurate messages, alter messages, and impersonate a node, for this reason violating availability, integrity, authentication, and non-repudiation.

- Secondly, nodes, roaming in an opposed environment (e.g., a battlefield) with surprisingly bad physical protection, have the non-negligible opportunity of being compromised. Therefore, we have to now no longer simplest consider malicious assaults from out of doors a community, however, additionally recollect the assaults released from inside the community through compromised nodes. Therefore, to gain excessive survivability, ad hoc networks have to have an allotted structure without vital entities. Introducing any vital entity into our safety answer ought to cause considerable vulnerability; that is, if this centralized entity is compromised, then the entire community is subverted.

- Thirdly, an advert hoc community is dynamic due to common modifications in each its topology and its membership (i.e., nodes regularly be part of and depart the community). Trust dating amongst nodes additionally modifications, for example, while sure nodes are detected as being compromised. Unlike different wi-fi cell networks, which includes cell IP, nodes in an advert hoc community might also additionally dynamically emerge as affiliated with administrative domains. Any safety answer with a static configuration might now no longer suffice. It is perfect for our safety mechanisms to conform on-the-fly to those modifications.

- Finally, an advert hoc community might also additionally encompass loads or maybe lots of nodes. Security mechanisms have to be scalable to deal with this sort of big community. [26]

1.8 Mobile ad hoc networking(MANET)

1.8.1 Definition of Mobile Ad hoc Network (MANETs)

A MANET is essentially a chaotic network of portable devices with wireless communication capabilities that can be dynamically connected anytime, anywhere. In this type of network, mobile hosts that simultaneously act as routers are connected by wireless links and move slightly randomly to dynamically change the network topology and create an autonomous system of mobile nodes without base stations. . In MANET, each node has a finite transmission range, so multiple wishes are used to forward packets from each initiating node to each endpoint node in the network.[27]

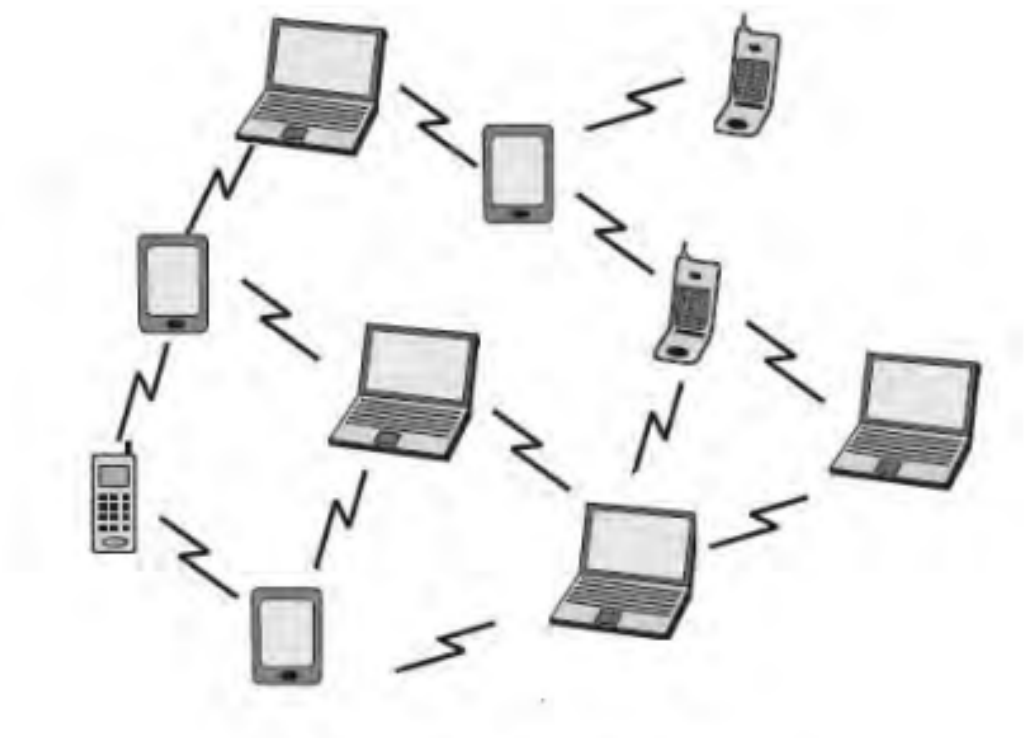

Figure 1.7: architecteur of MANET)[6].

1.8.2 Application of Mobile Ad hoc Network

Distinctive MANET applications include:[27]

- Military area: Ad-hoc networking allows the Army to leverage traditional networking expertise to maintain information networks between vehicles, units, and intelligence headquarters.
- group work: To facilitat e commercial settings, the office atmosphere and the need for coordinated data processing outside the environment are very important compared to the internal environment. People want to be outside of meetings, share information and collabo,rate on assigned tasks.
- Restriction level: Ad-hoc networks can use laptop computers to freely connect to instant hypermedia networks in addition to instant hypermedia networks and share information with all participants. classrooms and meetings. There may be additional valid and restricted applications in the home network where these devices are used. You can connect directly while exchanging information.
- Pan and Bluetooth:A PAN is a localized short-range network whose devices usually belong to one network. specific person. Her MANET, which has a limited range like Bluetooth, can simplify exchanges between multiple portable devices such as laptops and mobile phones.
- Business area: Ad-hoc networks can be used for rescue and emergency processes in emergency relief operations such as floods, fires, and earthquakes. If the transmission

structure is damaged and non-existent, and if rapid preparation of the transmission network is required, emergency rescue procedures should be carried out.

- Sensor network: Management of home appliances with his MANET both near and far. Tracking objects such as creatures. Weather-related activities.
- Backup service: Rescue operations, post-tragedy rescue, hospital diagnosis or status or file transfer, fixed infrastructure replacement.
- Education Department: Telecommunications equipment setup for a computer-generated conference room, classroom, or laboratory

1.8.3 MANET Characteristics

- Autonomous and infrastructure-free: MANET does now no longer depend on hooked up infrastructure or valuable control. Each day, nodes perform in dispensed peer-to-peer mode, performing as unbiased routers and producing unbiased data. Communication community control needs to be dispensed to numerous nodes, making failure detection and control difficult.[28]
- Multi-hop routing: no default router is available. Each node acts as a router, forwarding packets to each other and allowing information exchange between mobile hosts.
- Dynamic topology: In mobile ad-hoc networks, network topologies, typically multi-hop, can change frequently and unpredictably as nodes are free to move, resulting in route changes, frequent network partitions, and packet loss may occur.
- Different link and node capabilities: Each node can be equipped with one or more radio interfaces with different transmit and receive capabilities and operating in different frequency bands. Due to the heterogeneity of the radio capabilities of this node, the connection can become asymmetric. Also, each mobile node may have different software/hardware. Configurations result in variability in processing power. Designing network protocols and algorithms for this heterogeneous network can be complex, requiring dynamic adaptation to the changing conditions (power and channel conditions, traffic load/distribution variations, congestion, etc.).Energy-constrained operation. Because batteries carried by each mobile node have a limited power supply, processing power is limited, which in turn limits services and applications that can be supported by each node. This becomes a bigger issue in mobile ad hoc networks because, as each node is acting as both an end system and a router at the same time, additional energy is required to forward packets from other nodes.
- Network scalability: Currently, popular network management algorithms were mostly designed to work on fixed or relatively small wireless networks. Many mobile ad hoc network applications involve large networks with tens of thousands of nodes, as found for example, in sensor networks and tactical networks. Scalability is critical to the successful deployment of these networks. The step to a large network of nodes with limited resources is not trivial and exists there are still many unsolved challenges in areas such as addressing, routing, location management, configuration management, interoperability, security, and high-capacity wireless technology.

1.8.4 MANET Advantages

These are The advantages of MANET [28]:

- Provide access to information and services regardless of geographic location.
- these networks can be set up anytime, anywhere. Independence from central network management. Self-configuring network, nodes also act as routers. less expensive
- compared to wired networks.
- Scalable - more nodes can be added.
- Greater flexibility.
- Robust due to decentralized administration.

1.8.5 MANET Challenges

Regardless of the attractive applications, the characteristics of MANETs lead to several challenges that need careful consideration before we can expect broad commercial applications. These include [29]

- Routing|: The constantly changing topology of networks makes routing packets between pairs of nodes a daunting task. Most protocols should be based on reactive routing rather than proactive routing. Multicast routing is Another challenge arises because the random movement of nodes within the network makes the multicast tree no longer static. Routes between nodes can involve multiple hops, which is more complex than single-hop communication.
- Safety and Reliability: In addition to normal functions Wireless connectivity vulnerabilities, ad-hoc networks have certain security issues due to nasty neighbors forwarding packets, etc. Distributed operation capabilities require different authentication and key management schemes. wireless too Link characteristics also lead to limitations in radio transmission range, the broadcast nature of the wireless medium (such as the hidden terminal problem), packet loss due to mobility, and reliability issues due to data transmission errors.
- Quality of Service (QoS): Providing varying levels of quality of service in a constantly changing environment is difficult. Due to the inherent probabilistic nature of MANET communication quality, it is difficult to reliably guarantee the service provided to a device. To support this, you need to implement adaptive QoS over traditional resource reservation. multimedia services.
- Networks: In addition to communication within ad-hoc networks, networks between MANETs and fixed networks (mainly IP-based) are often expected. The coexistence of routing protocols on such mobile devices is a challenge for harmonized mobility management.

- Power Consumption: Most lightweight mobile devices require communication-related functions to be optimized for low power consumption. Energy saving and energy-conscious routing should be considered.

- Multicast: Multicast is desirable to support wireless communication with multiple participants. Because multicast trees are no longer static, multicast routing protocols must be able to handle mobility, including multicast membership dynamics (leaves and joins).

- Location-Aided Routing: Location-Aided Routing uses location information to define regions of interest so that routing can be spatially directed and constrained. This is analogous to ABR's associative, constrained broadcast.

1.8.6 Security in MANET

Security in mobile ad hoc networks (MANETs) is a crucial aspect that needs to be addressed due to the unique characteristics of these networks. The lack of centralized authority and the open nature of communications make MANETs vulnerable to various security threats, including eavesdropping, data tampering, malicious node attacks, and network disruptions. In this section we will discuss the security aspects related to MANETs networks.

1.8.6.1 The Security Requirement in MANET

According to [30] security goals, it is to control access to resources. This is done according to assurance criteria that, if met, guarantee that the resulting MANET is secure. Here are the safety standards:

- Availability is an Attributes that are the result of processing resources, processes, and systems that are appropriate, accessible, and timely to use when needed, but no more and no more than necessary. This is an important property that creates survivability. Survivability is the ability of a system to perform its mission in a timely manner in the event of an attack, failure, or accident.

- Confidentialityauthorized users only, An entity or process can access and use the system Resources and Disclose ure/Send or Disclosure/Receive System processing result, method and timing Authorized. In other words, this is true in ad-hoc networks. The property ensures that no data exists between nodes is accessible by unauthorized third parties; intermediate and untrusted nodes do not understand this Contents of the packet and to be sent.

- Integrityis the property that assures that system inputs are accurate particular and carry the right authorization and system processing is complete and accurate, that is accuracy and completeness are preserved or system resources,

- Authorization determines whether aparticular user or entity has the right to carry out an activity. Authorisation establishes rules that define what each network node is or is not allowed to do.

- Authenticationproof of who a user or entity is he/it claims to be authentication authorize node communicating parties o source. This will protect your opponent from Masking nodes to ensure the security of Confidential Information and the Manipulation of another node

- Non-repudiation ensures that a node cannot refuse to send Sending and receiving data. This helps detect and Quarantine the compromised node. all knots You can be blamed for this if you receive incorrect information Evidenced sender convinces other nodes on the compromised node.

1.8.6.2 Vulnerability in MANET

A vulnerability is a weakness in a security system. Constant The system may be vulnerable to unauthorized data manipulation Because the system does not verify the user's identity in advance Allow data access. MANET is more vulnerable than wired Communication networks. Some of the vulnerabilities are:[29]

- Lack of central administration: MANET has nothing Centralized monitoring server. Lack of management Attacks are difficult to detect because it is not the east to monitor Data traffic in highly dynamic and large ad-hoc networks. Lack of centralized control hinders trust management for knots.

- Resource Availability: Resource availability is a key issue at MANET. Provide secure communication of such changes; environment and protection against specific threats Attacks have led to the development of various security schemes, architecture. A collaborative ad-hoc environment also allows this Implementation of self-organizing security mechanisms.

- Scalability: Ad Hoc Scaling with Node Mobility Networks are always changing. So scalability is a big issue for security issues. A security mechanism should be able to: It supports not only small networks but also large networks.

- Cooperativeness: General MANET Routing Algorithms It assumes that the node is cooperative and non-malicious. As This can easily turn a malicious attacker into a primary attacker Ignore the routing agent and interrupt network operations. protocol specification.

- Dynamic Topology: Dynamic Topology and Variable Nodes Membership can disrupt trust relationships between nodes. of some nodes compromised. This dynamic behavior could be better protected Use distributed and adaptive security mechanisms.

- Limited Power Supply: Nodes in Mobile Ad Hoc Networks Limited power supply must be considered. Some problems. A node in a mobile ad-hoc network can operate as follows: In a selfish way when you find yourself limited power supply.

- Bandwidth Constraints: Low-capacity variable links compared to more vulnerable wireless networks Effects of external noise, interference, and signal attenuation.

- In-Network Attackers: In-Network Mobile Nodes MANETs are free to join and leave the network. knot in it Networks can also behave maliciously. it's hard to see Node behavior

is malicious. So this attack is more It is more dangerous than an attack from the outside. These nodes are compromised node.

- No Defined Boundaries: In mobile ad-hoc networks, A network's physical boundaries cannot be precisely defined. of Nodes operate in a nomadic environment where movement is permitted. Joining and leaving wireless networks. once an opponent Once in radio range of a node, you can communicate with this knot. Attacks include Identity theft; tempering, replay, and denial of service (DoS) attacks

1.8.7 Types of Intrusion in MANETs Network

1.8.7.1 Intrusion Classification

Securing ad-hoc wireless networks is a very difficult task. Understanding possible forms of attack is always the first step To developing better security solutions. the safety of Communication over MANET is important for secure transmission Lack of central coordination of information Mechanisms and Shared Wireless Media Make MANET More Useful They are more vulnerable to digital/cyber attacks than wired networks. The number of attacks impacting MANET. These attacks It can be divided into two types:[29]

- External attacks: External attacks are performed by nodes that do not belong to the network. cause overload Either the routing information is incorrect or the service becomes unavailable.
- Internal attacks: Internal attacks originate from compromised nodes. they are part of a network. For internal attacks, malicious A node from the network is illegally accessed, Pretend to be a real node. we can analyze the traffic between Other nodes and may participate in other network activities.

1.8.7.2 Types of attacks in MANET

The main types of attacks that occur in wireless ad-hoc networks are as follows[29]

- Denial of Service Attack: The purpose of this attack is to: Availability of nodes or the entire network. when attacking The service cannot be used normally. attacker
- Black Hole Attack: In this attack, the attacker Zero metric for all targets and zero for all surrounding nodes Route packets there. Malicious nodes send bogus routes Information claiming that there is an optimal route and cause Other good nodes routing data packets through bad nodes one. Malicious nodes drop packets received instead Forward these packets normally. Attacker hears this Requests with flooding-based protocols
- Wormhole attack: In a wormhole attack, the attacker receive the packet at some point in the network and "tunnel" it there Play to another point in the network, Network here. Routing can be interrupted during routing Control messages are tunneled. This tunnel between us Covert attacks are called wormholes.

- Man-in-the-Middle Attack: The attacker stands between the man-in-the-middle. Eavesdrop on all information sent by senders and recipients Between two nodes. In some cases, attackers can impersonate. Sender communicates with recipient or impersonates recipient Reply to sender.

- Gray hole attack: This attack is also known as routing. Cheating attacks that lead to dropped messages. gray A hole attack has two phases. In the first phase, nodes advertise. Second, with itself as a valid route to the destination During the phase, nodes drop intercepted packets with a certain probability

1.8.8 Solutions for Secure MANET:

According to [31] there are many solution used for security of MANET network. Among theses solutions we can cite:

1.8.8.1 Secure Routing

MANET is a Self-organized WLAN The network has vulnerable attacks, It can easily damage your entire network. that's why Even some of their work should provide some solutions A mobile node in your network has been compromised. one of The main challenge of secure routing is the deployment Authentication (trustworthiness) of users within the network.

1.8.8.2 Key Management

Certificate Authority (CA) is one of them. A mechanism that provides key management. if it is Then the entire network could easily be compromised Damaged. One of the main functions of the key Management, sales, and provision of MANET A solution to mobility problems.

1.8.8.3 Intrusion Detection System

It is a complete security solution that we provide. Information about malicious activity on your network. Also used to detect and report malicious activity. MANET is also the design of the route traffic mechanism. Network congestion, some nodes are faulty Dynamic behavior changes topology.(and we will know it in the next chapter)

1.8.8.4 Cryptographic techniques

Encryption algorithms and cryptographic protocols can be employed to protect the confidentiality and integrity of data transmitted in MANETs. Techniques such as symmetric key encryption, public key encryption, and digital signatures can be utilized to secure the communication channels and prevent unauthorized access.

1.8.8.5 Trust-based mechanisms

Trust-based mechanisms establish trust relationships among nodes in MANETs. Nodes can evaluate the reputation and trustworthiness of their neighbors based on their past behavior and interactions. Trust-based mechanisms can help identify and isolate malicious nodes, enhancing the overall security of the network.

1.8.8.6 Genitic Algorithm

Genetic algorithms are considered as one of the solutions for enhancing the security of Mobile Ad hoc Networks (MANETs). Genetic algorithms are optimization algorithms inspired by natural selection and evolution. They can be applied to various security aspects in MANETs, including key management, intrusion detection, and routing optimization.

1.9 Conclusion

In conclusion, this chapter has provided an overview of Mobile Ad hoc Networks (MANETs) and their security challenges. MANETs offer numerous advantages in terms of flexibility, scalability, and decentralized communication, making them suitable for various applications. However, their inherent characteristics such as open nature, limited resources, and absence of a centralized infrastructure pose significant security risks.
In this field, security is a rapidly evolving domain that requires ongoing research and innovation. By addressing the unique security challenges of MANETs and exploring new techniques and technologies, we can strive towards creating more secure and resilient mobile ad hoc networks using Intruction detection systems.

Chapter 2

Intrusion Detection System

Contents

2.1 Introduction

Over the last few decades, the Internet and computer systems have created many security problems due to the explosive use of networks. CERT Statistics (Computer Emergency Response Team) reports that the number of robberies has increased excessively over the years. Malicious intrusions or attacks on network vulnerabilities, computers, or information systems can lead to catastrophic consequences and violate computer security policies. Confidentiality, Integrity, and Availability (CIA). To date, threats to network and information security remain an important research topic.[32].
IDS is one of many solutions used to secure MANET. in this chapter we present IDS

2.2 Definition of Intrusion Detection System

Intrusion detection systems (IDS) are typically deployed as a second line of defense to protect information systems and other proactive security mechanisms such as access control and authentication. There are several reasons why intrusion detection is a necessary part of an overall defense system. First, many traditional systems and applications were designed with wrist security in mind. In other cases, a system or application was designed to work in another environment and becomes vulnerable when deployed in the current environment. (For example, a system is perfectly safe but vulnerable when isolated. Intrusion detection provides a way to identify attacks on these systems and enable responses to them. Second, due to limitations in information security and software engineering practices, computer systems and applications may have design flaws or flaws that can be used by intruders to attack the system or application. As a result, certain protection mechanisms (such as firewalls) may not be as effective as expected. Intrusion detection complements these protection mechanisms to improve system security. Furthermore, even if proactive security mechanisms successfully protect information systems, it is important to understand what kind of intrusions have occurred or is being attempted so that security threats and risk detection management and future attacks can be prepared. [33]

2.3 History of an IDS

IDS stands for Intrusion Detection System, a security technology that monitors network traffic and computer systems for signs of malicious activity or policy violations. The history of IDS dates back to the late 1980s and early 1990s. Back then, the Internet was still in its infancy, and computer security was still a relatively new field.
The first of his IDSs was developed in 1987 by computer science researcher James Anderson at the University of California, Davis. Anderson's system, called the Computer Misuse Detection System, is designed to detect unauthorized access to computer systems by analyzing system audit logs. In the early 1990s, a team of researchers at Carnegie Mellon University's Software Engineering Institute (SEI) developed the first network-based he IDS named Network Security Monitor (NSM). NSM is designed to detect network-based attacks such as port scans and network probes by analyzing network traffic. [34]

2.4 Classification of Intrusion Detection System

Intrusion Detection Systems (IDS) can be classified into several categories based on different criteria. According to [35] intrusion system detection are classified into 3 types:

2.4.1 HIDS

This type is placed on devices such as servers.A workplace where data is analyzed locally We collect this data from various sources.HIDS can use both anomaly detection and abuse detection systems.

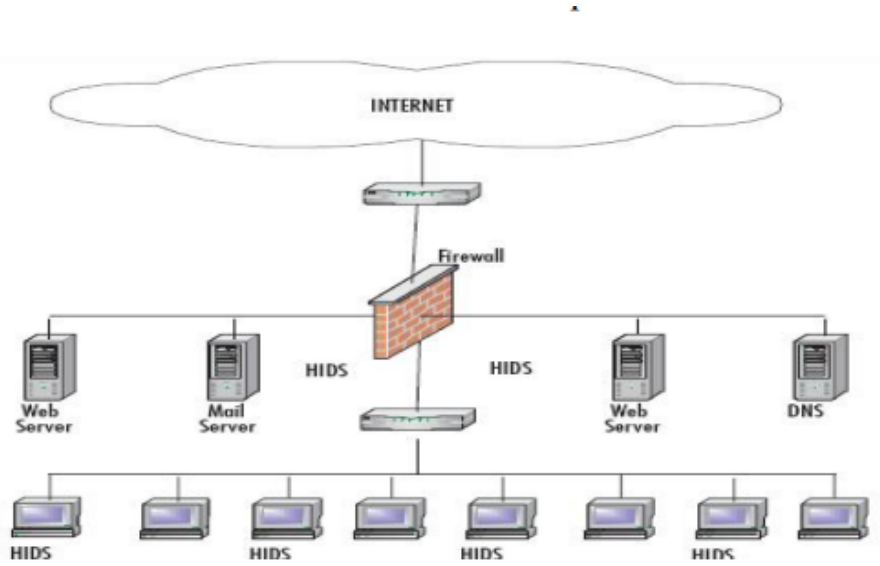

Figure 2.1: Architecture of HIDS[7]

2.4.2 NIDS

NIDS are placed at strategic points in the network infrastructure. NIDS can collect and analyze data and Detect known attacks by comparing patterns and signatures Illegal activity detection by database or scan traffic with unusual activity. Also known as NIDS. Because ”packet-sniffers” capture packets passing through thorough communication media..

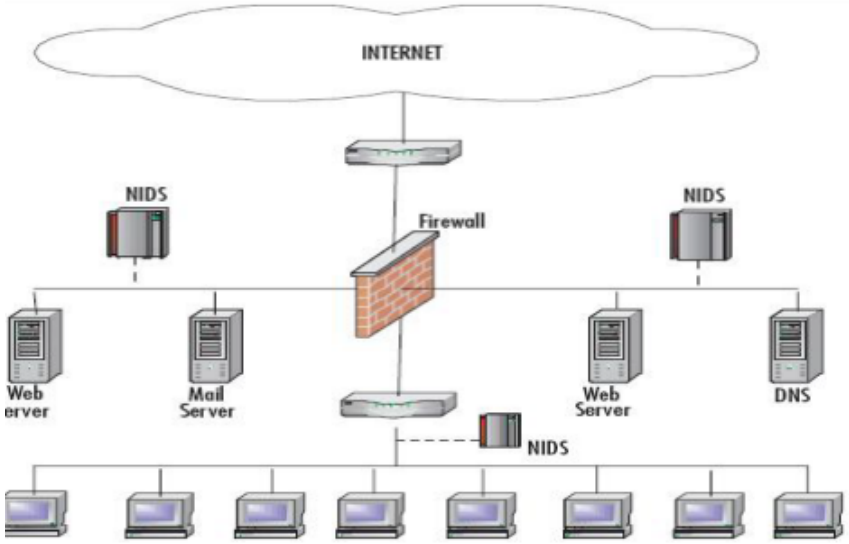

Figure 2.2: Architecture of NIDS[7]

2.4.3 Hybrid Based IDS

Management and alerting of both network and host-based intrusion detection devices, and logical Complementary to NID and HID - Central Intrusion detection management

according to the [20] Table 2 shows a summary of comparisons between HIDS and NIDS.

		Advantages	Disadvantages	Data source
Technology	HIDS	• HIDS can check end-to-end encrypted communications behaviour. • No extra hardware required. • Detects intrusions by checking hosts file system, system calls or network events. • Every packet is reassembled • Looks at the entire item, not streams only	• Delays in reporting attacks • Consumes host resources • Needs to be installed on each host. • It can monitor attacks only on the machine where it is installed.	• Audits records, log files, Application Program Interface (API), rule patterns, system calls.
	NIDS	•Detects attacks by checking network packets. •Not required to install on each host. •Can check various hosts at the same period. •Capable of detecting the broadest ranges of network protocols	•Challenge is to identify attacks from encrypted traffic. •Dedicated hardware is required. •It supports only identification of network attacks. •Difficult to analysis high-speed network. •The most serious threat is the insider attack.	•Simple Network Management Protocol (SNMP) •Network packets (TCP/UDP/ICMP), •Management Information Base (MIB) •Router NetFlow records

Table 2.1: Comparison of IDS technology types based on their positioning within the computer system [20]

2.5 IDS Detection Technique

Intrusion Detection Systems (IDS) employ various detection techniques to identify and respond to potential security threats and attacks. Here are some commonly used IDS detection techniques

[36]:

2.5.1 Signature-Based Detection

Use well-defined attack patterns that exploit vulnerabilities in Systems and application software for identifying intruders. These systems detect intrusions based on patterns. malicious activity. Very useful for recognizing acquaintances Attack patterns, known system vulnerabilities. of System compares network/system activity to known Signatures or other indicators of abuse to generate alerts. of High report dropout rate. regular updates from Signature required. These systems can also be detected Intrusion attempt; partial signature tried to break in. Examples include Haystack, Bro, IDES, or discoveries. An advantage of this system is it has more accuracy and standard alarms understood by the user [8].

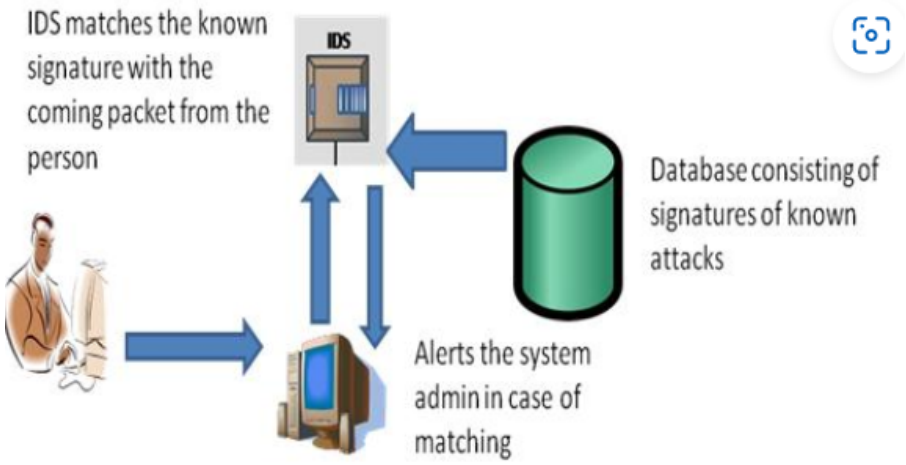

Figure 2.3: Signature-based Intrusion Detection System [8]

2.5.2 Anomaly-Based Detection

It uses normal usage patterns to identify intrusions and is trained using normal traffic patterns. An attack is malicious activity based on deviations from normal behavior. You can detect unknown intruders. Report dropout rate is low. These systems build a model on regular data and see if the data fits the model. This system can find unknown attacks. However, it has lower false alarm rate and accuracy when compared to signature-based approaches. Anomaly detection systems can use supervised or unsupervised learning techniques. Examples include IDES, NIDES, and EMERALD. The domain and nature of anomalies change over time, and intruders adapt network attacks to bypass existing intrusion detection solutions. Figure 2.4 shows the Anomaly-based detection system architecture.
In case of any anomaly or discrepancy, the administrator is alerted. An advantage of this system is it can detect new and unique attacks [8].

2.5.3 Machine Learning-Based Detection

Machine learning techniques, such as supervised learning, unsupervised learning, or deep learning, can be used for intrusion detection. These techniques involve training models on labeled or unlabeled data to recognize patterns and anomalies associated with attacks. Machine learning-based detection can adapt to evolving attack patterns and provide effective detection, but it requires a significant amount of training data and ongoing model maintenance.

It is important to highlight that IDS solutions typically utilize a combination of these detection techniques to enhance their detection capabilities and achieve better accuracy. The choice of the appropriate detection technique depends on specific requirements, the characteristics of the network or system being safeguarded, and the types of threats and attacks that need to be identified.

Table 2.2 represents a comparison between the two techniques signature-based and anomaly-based. It involves evaluating their advantages and disadvantages.

2.6 IDS Architecture

according to [9]Existing IDS architectures for MANET fall into three basic categories: Panos, Xenakis, and Stavrakakis, 2010) (a) standalone, (b) collaborative, (c) hierarchical.

- Standalone: In a standalone architecture, each node runs IDS locally without collaboration. and respond locally. Network attacks. There are limits to the detection accuracy and types of attacks it detects.

- Collaborative: In this architecture, every node of the MANET has its own local IDS system. node Collaborate in a decentralized way to make decisions. When detecting an intruding knot Share this information, the level of attack risk, and take necessary steps to eliminate intrusions Use of active or passive precautions. At the same time, all nodes participate in global discovery decision-making. this is more appropriate For a flat MANETs.

- hierarchical: A tiered architecture is Cluster the network. A specific node is chosen (based on certain criteria) to act as the cluster head. Assume different responsibilities and roles in intrusion detection. These are usually different from those of simple cluster members. Main benefits This architecture makes efficient use of constrained resources, but suffers from high mobility MANET for setting zones and discovering responsible nodes in a cluster.

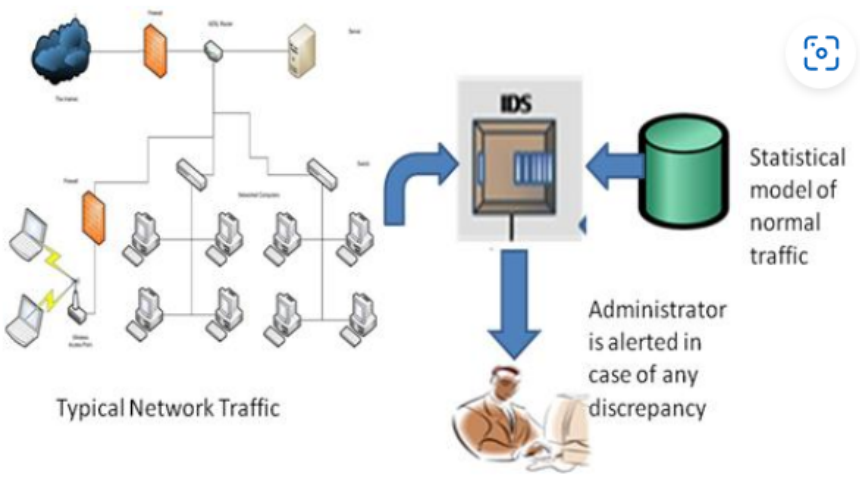

Figure 2.4: Anomaly-based Intrusion Detection System [9]

Detection Methods	Advantages	disAdvantages
Signature Based Detection	1. Is able to detect accurately 2. Generate much fewer false alarms	1. Cannot detect novel or unknown attacks
Anomaly Detection	1. Is able to detect new/unknown attacks based on audit 2. Less dependent on Operation system specific mechanisms 3. Can detect abuse of privileges 4. High False alarm rate	1. High false alarm and limited by training data. 2. The entire scope of behavior is not usually covered during learning phase. 3. Behavior change over time causing the system to perform poor. 4. During learning the system may be undergoing attack which will result in poor results.

Table 2.2: Advantages and Disadvantages of Intrusionn Detection Methods [9]

2.7 IDS Components

according to [37] The components of an IDS work in a structured manner to alert the administrator of an intrusion.

1. sensorFirst, there are two interfaces for the acquisition network.interface, the second is the management network interface.Its main function is detection and reporting. How the sensor listens Sniff the network and capture network traffic The interface transfers all collected data to a buffer. after that The recognition engine examines the contents of the buffer, Perform network protocol analysis. signature-based and Intrusion detection based on anomalies also occurs here.

2. backendThe backend is also called the main function of IDS. Its main functions are collecting and alerting. the event Events detected by the sensor are recorded in the event log. database system. Then decide how your backend reacts to emails, ads and blocks for each event Used to respond to important events.

3. frontendConfigurable command and control IDS Configured and updated by the user from the front end. all Events collected from the backend are displayed on the frontend. So the frontend provides a comfortable interface Users can now manage these logged events. receive To get the most out of your IDS, you need to fine-tune your reports. Important events only. Therefore, the user can make fine adjustments to IDS detection and response through this console. when it's done with the accuracy that the IDS reasonably provides to the user Early warning before robbery

2.8 Difference between IDS and IPS

according to the [38] IDS and IPS were originally developed to address requirements not present in most firewalls. IDS is basic and Used to detect threats or intruders within network segments. However, IPS does not protect against these threats or intervention to block or stop their activity

2.8.1 IDS

An IDS is a software or an appliance that detects threats, unauthorized or malicious network traffic. identification It has its own set of pre-defined rules. The purposes of intrusion detection are monitoring, auditing, and forensics.Report malicious network activity.

- Preventing network attacks
- Identifying the intruders
- Preserving logs in case the incident leads to criminal prosecution

2.8.2 IPS

IPS doesn't just detect bad packets caused by attack However, you can also take steps to prevent these network activities from harming your network.

- IPS stops the attack itself
- IPS changes the security environment

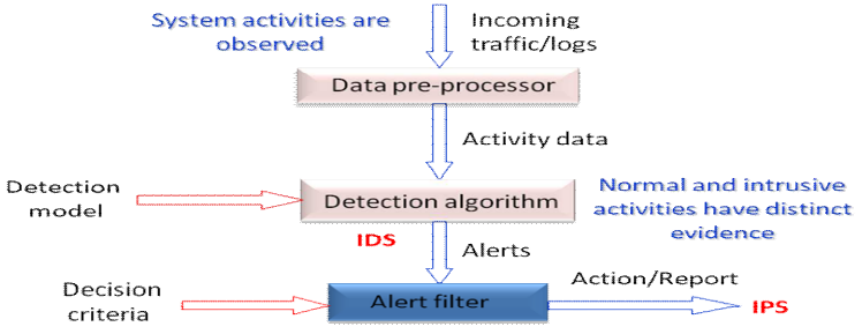

Figure 2.5: Role of IDS and IPS technology in network security

2.9 Challenges Faced IDS

New systems are being researched to automatically detect abnormal system usage. Furthermore, Denning reported on the development of an intrusion detection model proposed as a framework for general-purpose IDS . Since then, experts have developed and applied several algorithms to automate the network ID process. We have also continuously sought more accurate, faster, and more scalable methods to this end. With the advent of the "IoT" and big data era, the number of connected devices is expected to exceed 26 billion by 2020. Along with this trend, the types and number of cybersecurity issues are also expected to increase. Figure 2 summarizes the challenges of IDS. These challenges are false alarm rates, low detection rates, imbalanced datasets, and response times.[39]

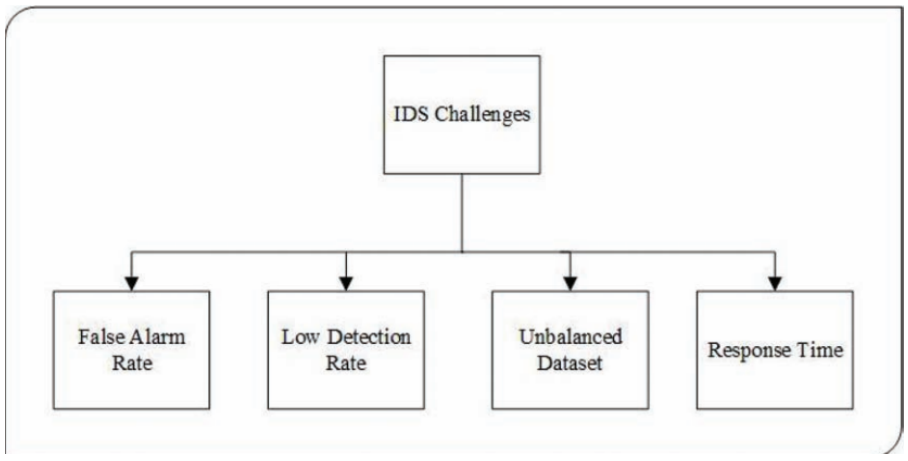

Figure 2.6: Intrusion detection system (IDS) challenges [10]

Some researchers have recently proposed more categories of IDS. Liao et al. , for example, called for a further division of IDS into five subcategories. These subcategories may belong to any of the above classes. Suggested subclasses are his IDS, which are pattern-based, rule-based, statistical-based, state-based, and heuristic-based. In the meantime, such a classification can be confusing because there are many similarities between the strengths of each technique, and there are no clear criteria to distinguish one technique from another. . Signature or rule-based IDSs are typically associated with a certain percentage of false positive (FP) alarms and are unable to detect new forms of attacks . IDS should show a high level of his FP detection. Identity systems that rely on stateful protocol analysis have varying recognition performance depending on the degree of profile definition . A major challenge with this approach is maintaining an up-to-date profile as new protocols evolve over time.[39]

2.10 IDS and its Benefits

according to [40] the IDS can be placed in 2 ways:

1. placed strategically on the network as a NIDS (network-based intrusion detection) which uses hardware sensors deployed at strategic points on the organization's network

2. installed on each individual system as a HIDS (host-based intrusion detection)

HIDS has the advantage of being able to detect attempts to modify or rewrite system files, or suspicious activity within your organization. Use anomaly or signature-based detection methods to identify threats.

- It monitors the working of routers, firewall, key servers and files. It uses its extensive attack signature database, raises an alarm and sends appropriate notifications on detecting a breach.

- By using the signature database, IDS ensures quick and effective detection of known anomalies with a low risk of raising false alarms.

- It analyzes different types of attacks, identifies patterns of malicious content and help the administrators to tune, organize and implement effective controls.

- It helps the company maintain regulatory compliance and meet security regulations, providing greater visibility across the entire network.

IDSs are typically passive systems, but in addition to detecting and generating alerts, some active IDSs block IP addresses or prevent access to restricted resources when anomalies are detected.

2.11 Conclusion

We presented an overview of intrusion detection systems in this chapter containing types and detection techniques and their components and the differences between IDS and IPS

In the upcoming chapter, we will delve into the fundamental concept of deep learning techniques and explore the principles of reinforcement learning.

Chapter 3

Deep Learning Approach for IDS

Contents

3.1 introduction

Both deep learning and reinforcement learning are machine learning functions, which in turn are part of a wider set of artificial intelligence tools. What makes deep learning and reinforcement learning functions interesting is they enable a computer to develop rules on its own to solve problems.
this chapter has been divided in two parts, in the first part we present the necessary basic concepts of deep learning and reinforcement learning, in the second we present state of art connected to the usage of deep learning the reinforcement learning

3.2 Basic Concepts

3.2.1 Deep Learning

3.2.1.1 Definition

Deep learning is a subset of machine learning, essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain and allow it to ”learn” from large amounts of data, even if it is well below the human brain’s capabilities. A single-layer neural network can still make approximate predictions, but additional hidden layers help optimize and improve accuracy. [11]

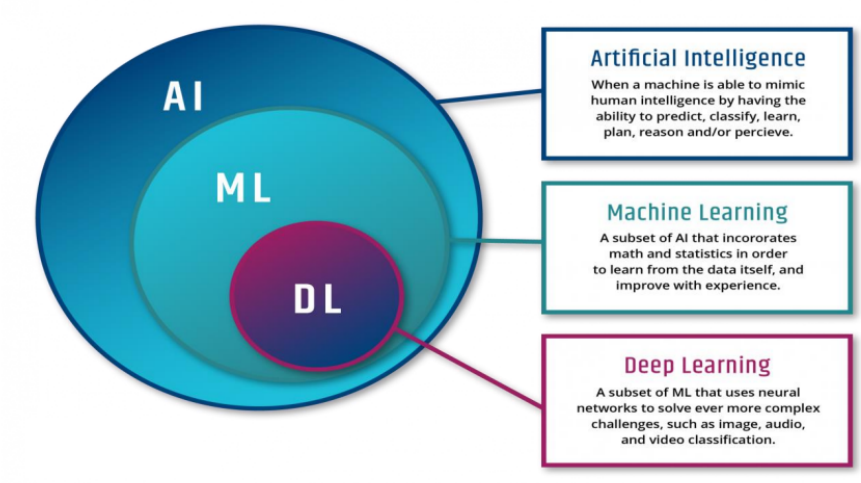

Figure 3.1: Relation between IA and Machine learning, Deep learning [11]

3.2.1.2 History of Deep Learning

The world is now seeing a global AI revolution across all industries. And one of the driving factor of this AI revolution is Deep Learning, Deep Learning now has become a popular term

and people might think that it is a recent discovery. But you might be surprise to know that history of deep learning dates back to 1940s [41]

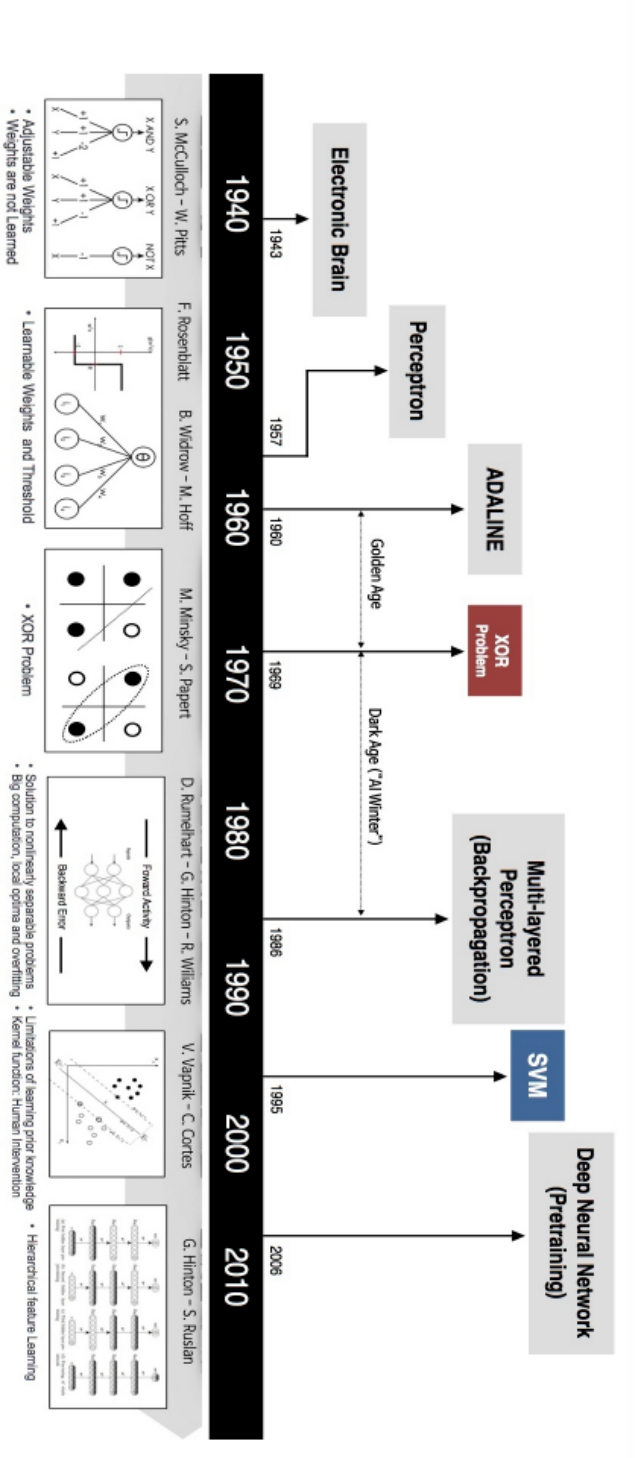

Figure 3.2: Deep learning history [12]

3.2.1.3 Principles of Deep Learning

Deep learning is a nonlinear neural network structure with multiple hidden layers. deep nerve A network consists of an input layer, some hidden layers, and an output layer. Each layer has several It is a neuron and has connection weights between neurons. Each neuron mimics a human neuron Connections between nodes mimic connections between neurons [42]
First layer (input layer): You can get the raw data directly, not just the features.
the middle layer(hidden layer): Contains multiple neurons, each neuron containing two functions. One is a weighted function.The other is the overlaid value and The transformed value is passed to the next layer.
Output Layer: Make decisions based on the final result of the received multi-layer operation.

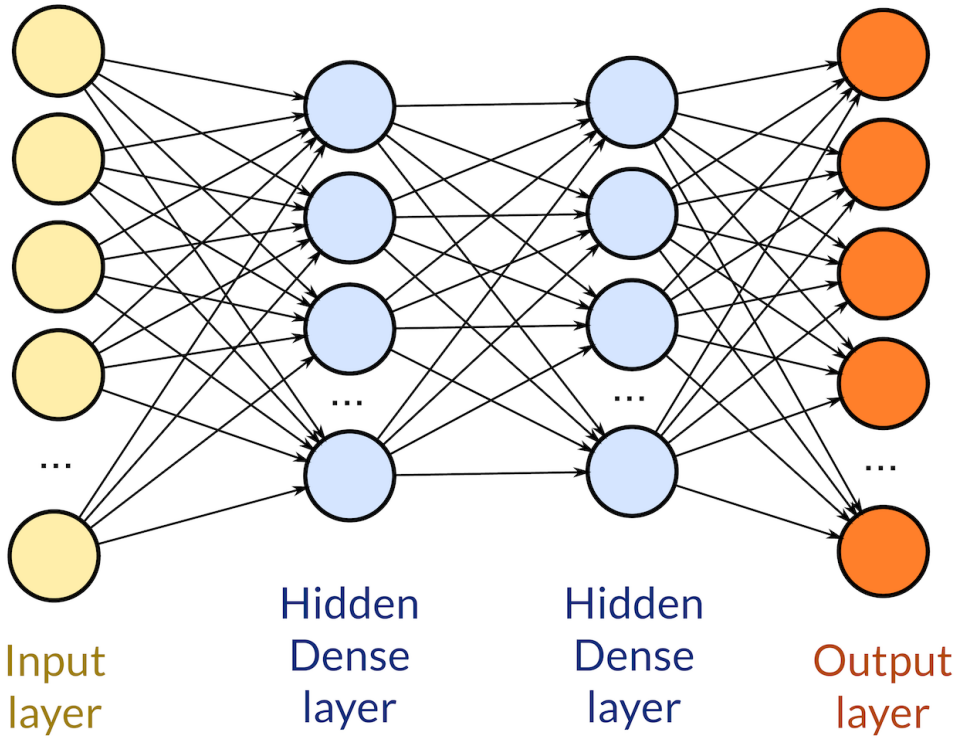

Figure 3.3: Deep learning network [13]

3.2.1.4 application

According to the [42] these are the main applications of deep learning in the field of security

1. Information Security Deep learning is not a panacea for all information security problems as it requires extensive labeling.record. Unfortunately there are no such flagged records. However, there is an important point. Improving information security cases with deep learning networks running existing solutions.Malware detection and network attack detection are just two of those areas, and deep learning has them.It showed clear improvements over rule-based and traditional machine learning solutions.

2. Smart Phone Intrusion DetectionWith the rapid development of science and technology, the number of mobile smart devices and users is increasing Traffic on the Andriod Platform Exponentially Increases and Damages Equally Attacks by malicious intrusion programs for mobile phone users are increasing. Intrusion detection is turned on A preventative security mechanism that monitors the status of smartphones and network behavior, and determines whether unauthorized user behavior or intrusion has occurred. Malicious Android An application recognition system based on deep learning that breaks down the barriers of conventional low technology Efficiency has been obtained on the Internet, not just in proof-of-concept theory, and in practice Validation also yielded better detection results.

3. Detection principleThe principle of deep learning intrusion detection is as follows: First use static code analysis technology to extract multiple types of behavioral feature data for Android applications, then convert the feature data into sample feature matrices, and then use the convolutional neural network algorithm file to train the sample feature matrices . The last batch download did not participate in training the Andriod application of the deep neural network, and then perform system steps on its APK to get the relevant prediction report of the unknown sample APK

4. detection modalsmartphones intrusion detection models based on deep belief networks are mainly composed of a restricted Boltzmann machine model (RBM) and a BP neural network. First process the input data, and then use RBM for unsupervised training to make the output of the feature by each layer more significant. Ensure that when the features are mapped inward to different feature spaces, retain as much feature information as possible to form a training model. To the more obvious feature information; the last layer uses the BP neural network for classification. The BP network layer receives the feature vector output by the RBM layer as its input data, and the training process of this layer is supervised training. After training with a fixed number of layers, set the classification parameter to 2 to get the final error value and calculate the corresponding accuracy rate

3.2.1.5 How Deep Learning Works

Deep learning, neural networks, or artificial neural networks try to mimic this. The human brain with a combination of data inputs, weights, and biases, as shown in the figure. These elements work together to accurately detect, classify, and Describe the objects in your data [14]

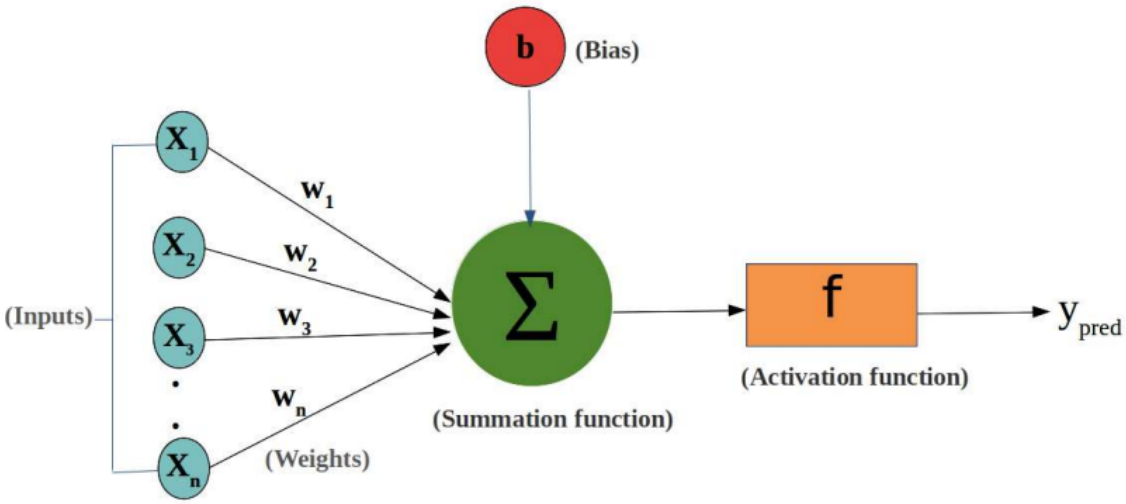

Figure 3.4: input, weight, bais in neural network [14]

Each deep neural network consists of multiple interconnected layers of nodes fine-tune and optimize the forecast based on the previous level, or classification. The progress of this computation through the network is called the forward direction propagation. The input and output layers of a deep neural network are called hidden layers layer. The input layer is where the deep learning model ingests the data process and the output layer does the final prediction or classification

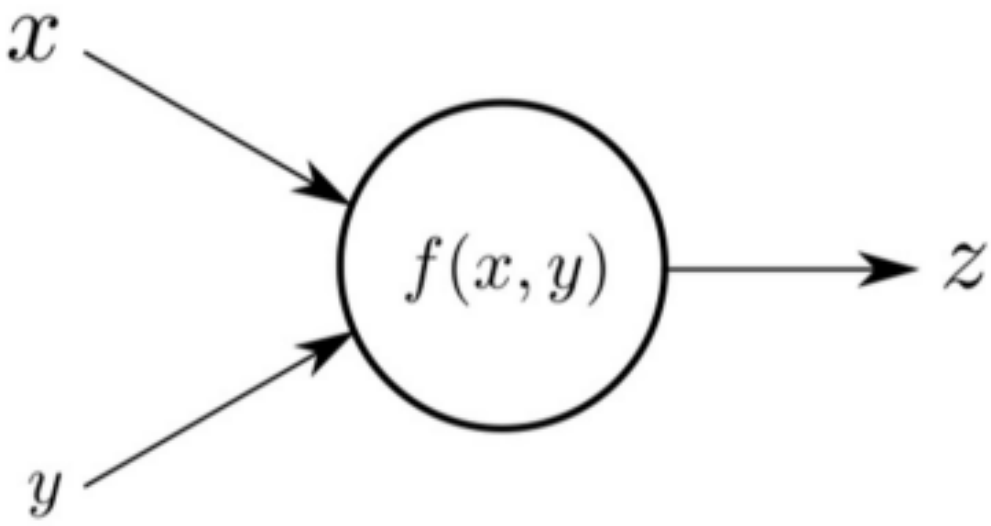

Figure 3.5: forward propagation in neural network[14]

Another process called backpropagation uses algorithms such as gradient descent to Calculate the error of the prediction and adjust the weights and biases of the function by going backward through the layers and training the model as shown below. in Figure. Forward

and backpropagation together enable neural function A network for making predictions and correcting errors accordingly. Algorithms over time gradually become more accurate

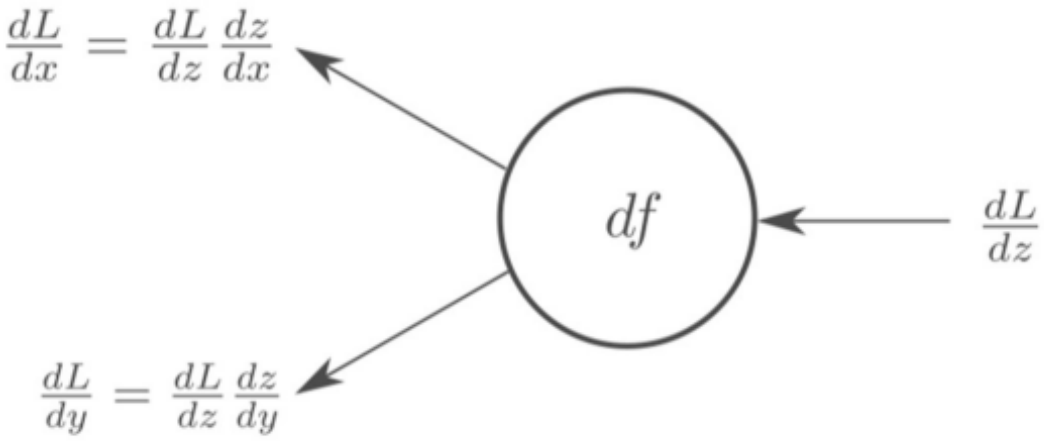

Figure 3.6: backpropagation in neural network[14]

3.2.1.6 Type of Neural Network in Deep Learning

3.2.1.6.1 ANN Artificial Neural Networks (ANN) are considered synonymous, Neural networks, but are not the same. In fact, ANN is By the way, in neural networks, which are essentially feed-forward networks Information moves from one layer to another without touching nodes Twice. This type of nervous system is built in the same way as neurons Inside the human brain to recognize patterns in raw data and help solve complex problems process. Another similarity to the human brain is that Ann's New input received. That is, ANN improves on its own, or continuously improving

3.2.1.6.2 CNN Convolutional neural networks are primarily known for their roles in images and image processing. video recognition, recommendation system, image analysis, and classification. Previously, object identification required manual work pictures. However, CNNs have played an important role in scaling processes using linearity. Algebraic principles for recognizing patterns in images. CNN is based on The following main layers:

- Conventional layer
- Polling layer
- Fully connected layer

Each layer increases the CNN's complexity in understanding the image. of The first level focuses on interpreting simple image features. B. its edge and color. Images are processed through layers, so the network can Recognize complex features such as the shape of objects. the last is the deepest Objects that can be identified

3.2.1.6.3 RNN A recurrent neural network (RNN) is a type of neural network that The output of the processing node from the previous step feeds the input of the stream step. This is how the model learns to predict the outcome of shifts. each node of The RNN model acts as a memory cell that continues the computation, operating. If the network prediction is wrong, the system will Self-learning while continuing to work on correct predictions

3.2.1.7 Learning Technique in Deep Learning

according to [43]Reinforcement learning is considered its own branch of machine learning, though it does have some similarities to other types of machine learning, which break down into the following four domains:

1. Supervised learningIn supervised learning, algorithms are trained on a set of labeled data. A supervised learning algorithm can only learn the attributes specified in the data set. A common application of supervised learning is image recognition models. These models are given a set of labeled images and learn to distinguish common attributes of predefined shapes

2. Unsupervised learningIn unsupervised learning, developers apply algorithms to fully unlabeled data. Algorithms learn by cataloging their own observations about features in the data without being told what to look for.

3. Reinforcement learningIt's a completely different approach. The agent places the agent in an environment with well-defined parameters that define useful and unhelpful activities, and an overarching endgame that must be achieved. It is similar to supervised learning in that the developer must give the algorithm a clear goal and define the reward and punishment. This means that the level of explicit programming is higher than unsupervised learning. But once these parameters are set, the algorithm works by itself and is much more self-directed than supervised learning algorithms. For this reason, reinforcement learning is sometimes referred to as a branch of semi-supervised learning, but it is most often recognized as a separate type of machine learning. this techniques we learn it in the next section

3.2.2 Reinforcement Learning

3.2.2.1 Definition of Reinforcement Learning

RL is learning what to do, i.e. how to match situations with actions. Maximize numerical reward signals. Learners are not informed of actions to take Instead of taking, you have to find through trial and error which actions bring the greatest reward. of In the most interesting and difficult cases, action cannot be immediately limited. It includes not only rewards, but also the next situation and therefore all subsequent rewards. these two Features - trial-and-error search and delayed reward [44]

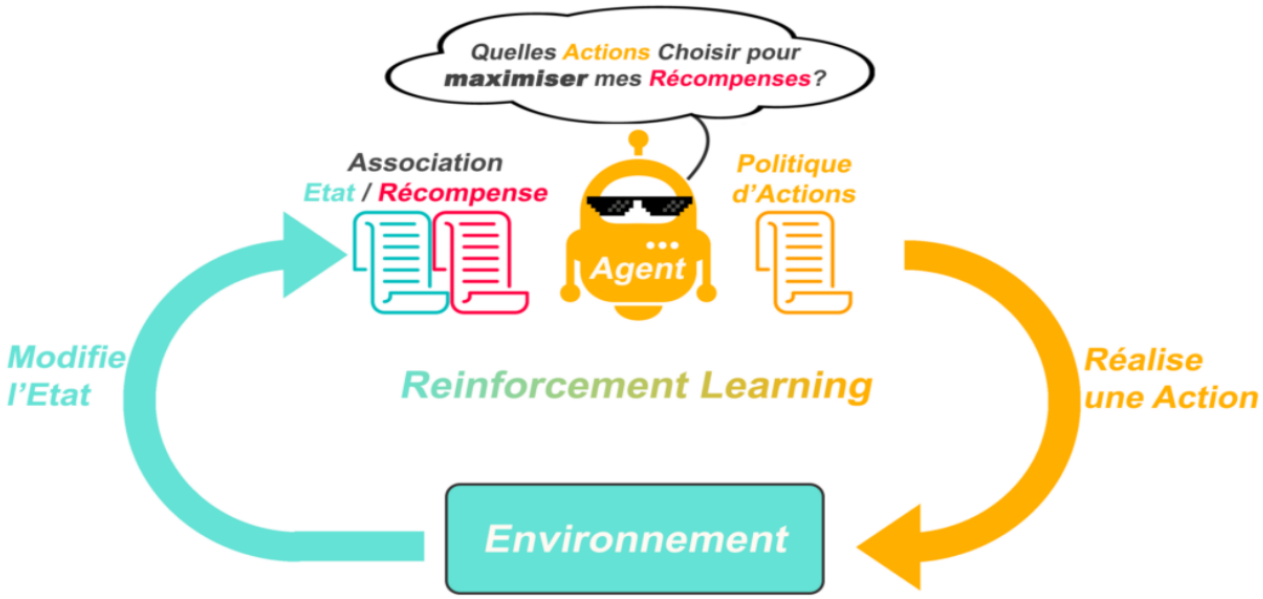

Figure 3.7: Shema of Reinforcement Learning [15]

3.2.2.2 Reinforcement Learning Elements

according to [17]Beyond agents and environments, we can identify four major sub-elements of a. Reinforcement learning systems: policies, rewards, value functions, and optionally models environment.

- An agent: physical or software entity;
- An environment: the area which the agent runs in and reacts with;
- A policy defines the learning agent's way of behaving at a given time;
- A reward signal defines the goal in a reinforcement learning problem;
- A value function specifies what is good in the long run;
- A model of the environment which is something that mimics the behavior of the environment, or more generally, that allows inferences to be made about how the environment will behave.

3.2.2.3 Reinforcement Learning Aims

- Improve the strategies used to solve any problem continuously by relying on the feedback received
- Maximize the rewards while taking steps to solve the problem
- Achieve optimal steps that maximize the rewards to solve the problem at hand.

3.2.2.4 Building Unit for RL

- Policy Function $\pi(a|s)$: Probabilistic function for actions depending on states, indicating how to act in a certain situation.
- Return G: Cumulative sum of future rewards in time, scaled by discount factor γ It is defined as:

$$G_t \sum_{i=t=1}^{\infty} \gamma^{i-t-1} r_i = r_{i+1} + \gamma G_{i+1} \tag{3.1}$$

- Value Function $V(s|\pi)$: Expected return when policy π is followed at state s, defined as:

$$V^{\pi}(S) = E[G_t|S - t = s, \pi] \tag{3.2}$$

- Action-Value Function $Q(s, a|\pi)$: Expected return when policy π is followed, except action a at first step at state s, defined as:

$$Q^{\pi}(s, a) = E[G_t|s_t = s, a_t = a, \pi] \tag{3.3}$$

3.2.2.5 Markov Decision Process

MDP is the mathematical foundation of RL and if you want to fully understand RL we should always start the algorithm with MDP.
MDP is essentially a framework for making decisions under uncertainty. it can provide a way Compute the optimal decision-making policy
MDP consists of :

- The state space S as the set of all possible states
- Action field A as a set of all possible actions
- Model function $T(s'|s, a)$ as the state transition probability.
- reward function R(s) as a reward mapping of state, action, next state tuples reward.
- Discount factor $\gamma \in [0, 1]$, a real number that determines the importance of future rewards controlled object.

Work is done in three phases:

- Write down where you are.
- Act in accordance with policies and receive rewards.
- Write down the reward you received for performing this action in this state.

3.2.2.6 Reinforcement Learning Algorithms

according to[16]The main used algorithms are:

- Q-learning is an out-of-policy RL algorithm used for time-difference learning. Staggered learning is the possibility to compare predictions that follow each other in time. Learn the value function Q(S, a). This means the ability to act "a" in a given state "s". The following flowchart explains how Q-Learning works.

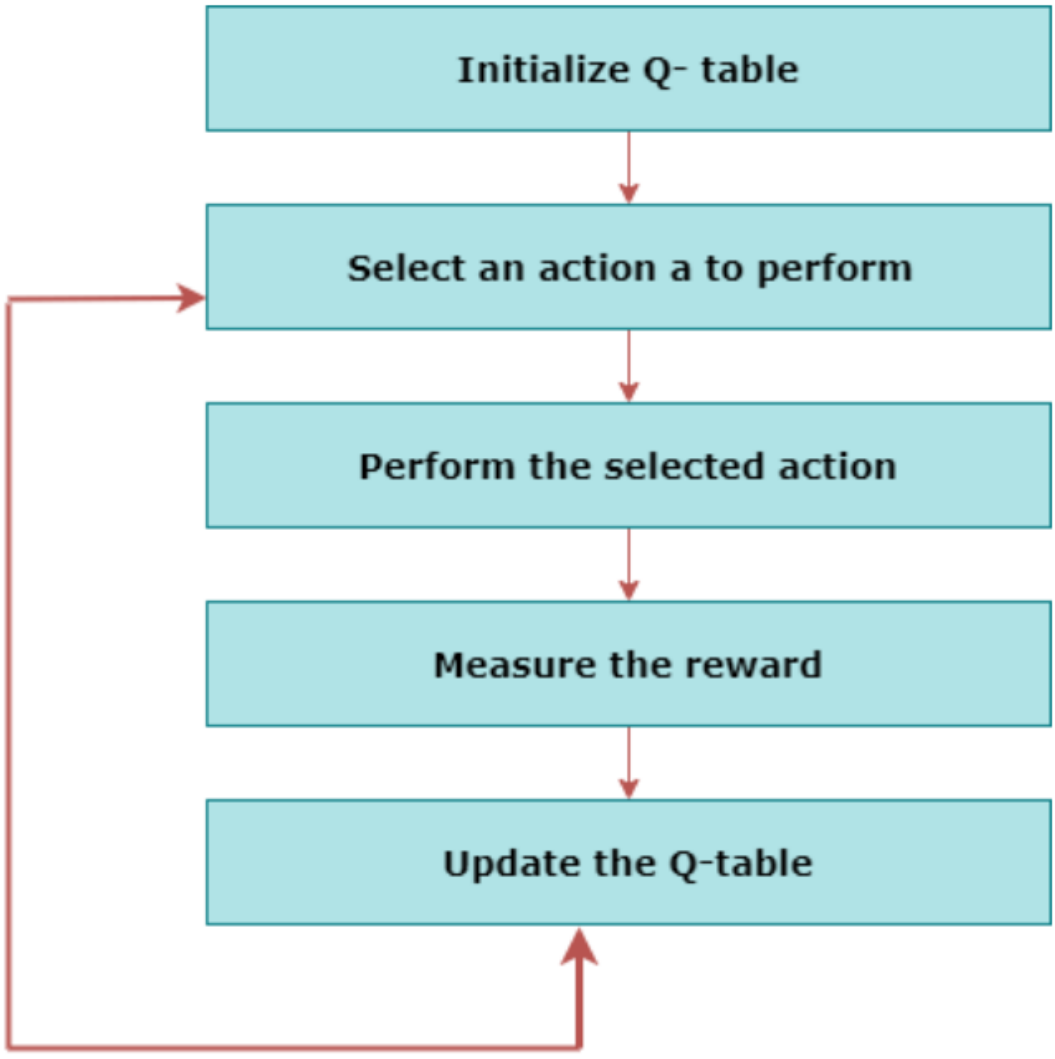

Figure 3.8: Q-Learning algorithm [16]

- SARSA stands for State Action Reward State Action and is a policy-compliant method for learning time differences. On-policy control schemes select actions for each state while learning using a specific policy.
  The goal of SARSA is to compute her $Q\pi(s, a)$ for all pairs of the current policyπ and $(s - a)$ chosen.
  The main difference between Q-learning and SARSA algorithms is that, unlike Q-learning,

no maximal state reward is required to update the Q-values
in the table. SARSA selects new actions and rewards using the same policy that determined the original actions. SARSA is so named because it uses the quintuple $Q(s, a, r, s`, a')$.
where: s: original condition
a: original action
r: Reward for tracing the state
s' and a': new condition, action pair.

- DQN as the name suggests, is Q-learning using neural networks. In a large state-space environment, defining and updating Q-tables becomes a difficult and complex task. The DQN algorithm can be used to solve such problems. Instead of defining Q-tables, neural networks approximate Q-values for each action and state.

3.2.2.7 Reinforcement Learning Differs from Supervised and Unsupervised Learning

according to the [17] The main difference is about the inputs and outputs of each one of the three algorithms previously discussed, the figure below shows this difference.

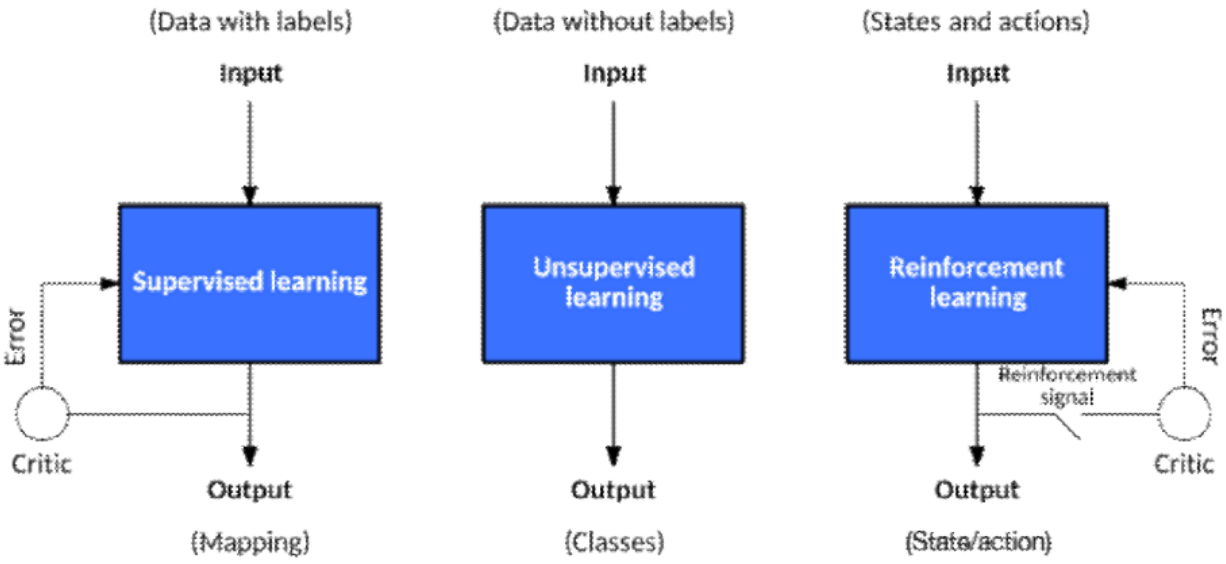

Figure 3.9: the difference between DL algo [17]

3.2.3 Usage of DRL

3.2.3.1 DRL Model

Reinforcement learning is a machine learning technique based on the Markov decision process (MDP), which is a function consisting of S, A, T, and R, where S is a set of states, A is a set of actions, T is a mapping function for each state-action pair to transition to a new state, and R is

the reward function obtained from this process. In the MDP, the transition from the current state-action pair to the next state is entirely determined by T, which possesses the Markov property. Therefore, once the MDP is defined, its policy is a one-to-one mapping of each state to action, and the MDP enables learning the optimal policy corresponding to each state and the best action it should take to maximize the total expected reward R.
The optimality criterion is frequently linked through the value function V, which is an estimate of the value of each state, and the strategy, according to the valuation of the action in the current state Q can be obtained, with V represents the valuation of each state and Q represents the valuation of each state-action pair.
To obtain the best model policy, observe the state-action space as much as possible and use the -greedy algorithm to explore the actions to be executed in the present state. The agent also selects the current status With probability p, choose a random action with probability $1-p$. ongoing dialogue with The agent surrounds and adjusts its own V and Q functions to approximate the actions predicted by the Q function and chosen by the model to best suit itself in the current state. Considerable total expected reward.
The main goal of the DQN algorithm is to fit a Q function that reflects the maximum expected value. The rewards that the environment can provide in a given state and activity. Q function is determined by state and system activity. Once you have $Q(s, a)$ you can get the strategy function. $Guideline = argamax(Q(s, a))$ is a state-dependent policy function that chooses the action that maximizes the value of Q.

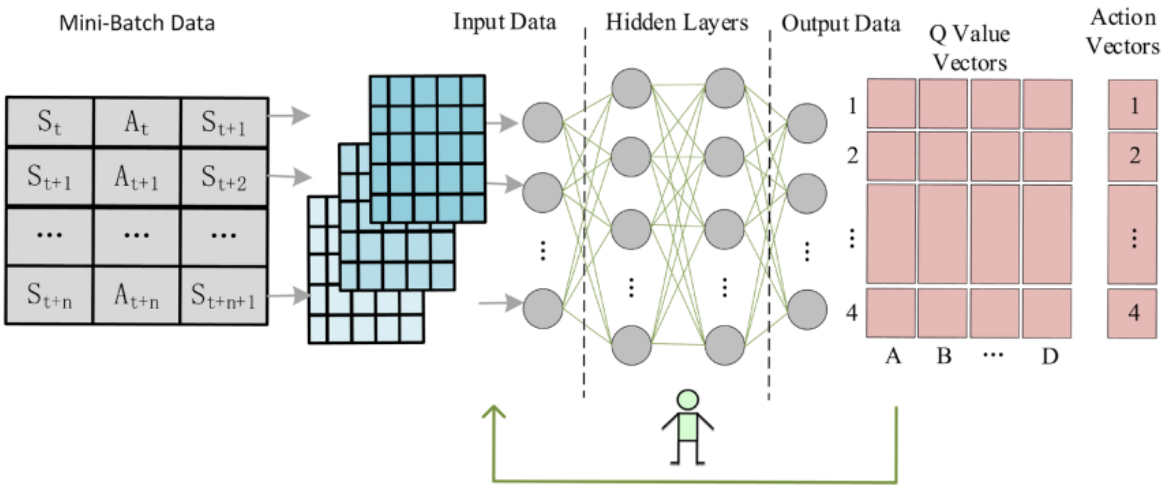

Figure 3.10: shematic of DQL model structure [18]

the DRL model was generated using the DQN method, The Mini-Batch samples selected by features are used as the model's input, and the feature values are extracted after convolutional layers. The feature values are then Flattened as the input data into the 3-layer fully connected layer, and the activation function of each layer in the fully connected network is then evaluated. Each layer's activation function in a fully linked network is the ReLU function, which ensures that all Q values calculated are positive.
the DQN algorithm is primarily applied to the fully connected layer of the DRL model, and the model will calculate the prediction a_t and a_{t-1} corresponding to s_t and s_{t-1} states respectively, then the predicted action a hat t and the state st correct action at continue to compare, if they are the same then the reward is 1; otherwise it is 0, and the reward value is obtained as r_t
notably, the reward discount factor of the model is set to 0.01 in order to get the most excellent

Algorithm 2: Recursive Feature Elimination with DQN (DQN + RFE)

Input: the all features set F in the dataset;
Output: the selected features subset F_i;
Step 1. Train the DQN model using all features
Step 2. Determine the model's accuracy
Step 3. define F to denote the importance of each feature to the model
Step 4. for each subset of F_i, $i = 1 \ldots N$ do
define the F_i as the important features
train the DQN model using F_i features
$Policy(s) = arg_a\, max(Q(s,a))$, get Q function
get reward R,
$Policy(s_{t+1}) = arg_a\, max\big(Q(s_{t+1},a)\big)$, get Q_{t+1} function
$q_{ref} = r_t + \lambda * q_{t+1}$, get the q_{ref}
find the loss of the DQN model
evaluated the accuracy of the model and correction model
find the F_i as the most important features
end for
Step 5. Calculate the accuracy of the model and find the optimal feature subset F_i
Step 6. use the model corresponding to the optimal Fi feature subset and rank the features by importance, $F_i = (F_1 > F_2 > F_3 > \cdots)$

Figure 3.11: algorithme DQL [18]

performance and encourage the model to focus on the present learning reward, given that the dataset is labeled and the labels are uncorrelated.[18]

3.2.3.2 Performance Metric

Accurate identification of attack traffic is more important to IDS than regular verification. traffic jam. Besides accuracy, one of the metrics to consider when evaluating model performance is Also, analyze model performance using F1 score, precision, recall, and ROC metrics.
These metrics are generated using a lattice-structured confusion matrix consisting of TP, TN, FP, and FN. This allows you to visualize your model's performance. TP stands for True Positives and is displayed correctly Anticipated attack traffic. TN stands for True Negative and indicates correctly predicted normal traffic. FP stand False positives indicating normal traffic expected to be attack traffic. FN stands for False Negatives, which indicate attack traffic expected as normal traffic. FN is an integral part. the lower An IDS is less likely to misjudge attack traffic, and our methodology aims to minimize that value.
Basic notation for the above metrics is described below.
Accuracy The number of correct predictions made by the model, expressed as a percentage of the total number of predictions. predict.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.4}$$

Precision this metric measures the percentage of attack traffic correctly predicted as attack traffic and is mathematically defned as follow

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

F1-Scores This metric is a combined form of model accuracy and sensitivity, and is a reconciled average of model accuracy and sensitivity. In an unbalanced dataset, better F1-Scores indicate fewer misclassifed fows, and this metric is the focus of our study.

$$F1\text{-}score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.6}$$

Receiving operating characteristics curve (ROC): ROC is a combination of response sensitivity and continuous specifcity variables that may indicate the link between sensitivity and specifcity; the greater the area of the curve, the better the model's performance.[18]

3.3 State of the Art

Several works have been proposed in the field of an intrusion detection system based on deep learning for MANET. In this section, we present a state of the art of workers who have a relationship with our field of interest.

3.3.1 DQL Based on RL

Authors in [21] proposed an approach called deep Q-learning-based reinforcement learning method for network intrusion detection. It combines Q-learning with a deep feed-forward neural network to detect different types of network intrusions. The approach has the potential to improve network security by continuously enhancing its detection capabilities.

3.3.2 Recurrent neural network

Authors in [45] proposed another approach deep learning approach for intrusion detection using recurrent neural networks (RNN-IDS). It can potentially extract better representations from the data to create better models. The authors study its performance in binary and multiclass classification and compare it with other machine learning methods on a benchmark dataset.

3.3.3 Deep Auto-encoder Model

Another approach proposed by the auteur in [46] deep learning method for intrusion detection systems that uses a Deep Auto-Encoder (DAE) model. The DAE model is trained greedy layer-wise to avoid overfitting and local optima. The approach aims to address the challenge of identifying network attacks due to the extensive number of vulnerabilities in computer systems and the creativity of attackers.

3.3.4 Deep Reinforcement Learning-based Adaptive Cloud IDS

Author in [47] presents another approach called "a deep reinforcement learning-based adaptive cloud IDS architecture" that performs accurate detection and fine-grained classification of new and complex attacks. The proposed architecture consists of three components: host network, agent network, and administrator network. The agent network uses deep reinforcement learning techniques to detect and classify attacks, while the administrator network manages the IDS and makes decisions based on the output of the agent network. Extensive experimentation using the benchmark UNSW-NB15 dataset shows better accuracy and less false positive rate

3.3.5 Industrial Internet of Things

A new approach proposed by the author in [48] is a new deep hardening-based algorithm (02) for the Industrial Internet of Things environment to detect multiple types of cyber threats. The system uses an I-source-based feature selection algorithm to extract the most effective special collection, effectively reducing the computational complexity of the model. It builds deep enhanced learning based on a multi-layer aware with 3 layers of hidden layers as a deep neural network structure shared by value networks and policy networks in this intrusion detection system. Finally, it uses 5, the function as a categorical output in the case of reduced over-fitting. According to experiments carried out on real data sets, this intrusion detection system performs well in detecting multiple types of cyber attacks against the Industrial IoT and has better precision, call-back rate, mouth score, and significantly reduced training time.

3.3.6 Discussion

These studies collectively highlight the potential of reinforcement learning-based IDS in improving the accuracy and adaptability of intrusion detection systems. However, it is important to note that further research is still needed to address challenges such as training data availability, computational complexity, and interpretability of the learned models. Future work should focus on enhancing the scalability, robustness, and real-time capabilities of these systems to ensure their practical applicability in diverse network environments.

3.4 Conclusion

This chapter represents two parts, the first part includes the basic concept of deep learning and the reinforcement learning necessary for this field
in the second part of this chapter,r we present a state-of-the-art
in the next chapter, we will discuss two things, first on discuss of the conception of the architecture proposed and the second steps of implementing this system

Chapter 4

Design and implementation of the Proposed Approach

Contents

4.1 Introduction

The objective of this chapter is to offer a thorough understanding of the methodology employed in developing the proposed solution, emphasizing important design choices and implementation considerations. The chapter is divided into two parts for clarity. The first part focuses on the design of the proposed approach, outlining the key decisions made during the development process. It delves into the architectural design and critical components of the solution. The second part of the chapter is dedicated to presenting and explaining the tools utilized in the implementation phase. It provides a comprehensive overview of the main implementation details, offering a detailed description of the software utilized.

4.2 Design

Ad hoc networks are particularly vulnerable to attacks due to their decentralized and dynamic nature, which makes them difficult to protect using traditional security mechanisms.

An IDS can help to monitor network traffic, detect suspicious behavior, and alert network administrators in real-time. This can help to prevent attacks such as denial-of-service (DoS), distributed denial-of-service (DDoS), and other types of cyber attacks that could compromise the network.

By deploying an IDS in an ad hoc network, network administrators can gain greater visibility and control over network traffic, identify potential security vulnerabilities, and take proactive measures to secure the network. This can help to ensure the confidentiality, integrity, and availability of network resources, and safeguard against data breaches and other security incidents.

The system inputs are the trafic of a simulated mobile ad hoc network , and the outputs are alerts of existing intrusions intended for the network administrator. The system architecture is presented in Figure 4.1. These requirements are implemented as independent modules as follows:

- The ad-hoc network to evaluate the performance of the intrusion detection system we need to apply them to a network model. So we have to define the network first. Furthermore, this model will be simulated to obtain the expected results.

- Intrusion detection tool: after we define the ad-hoc network we apply an intrusion detection system this can be used in two ways the first way is used to secure all the network and the second way is to install it in the every node in the mobile ad hoc network

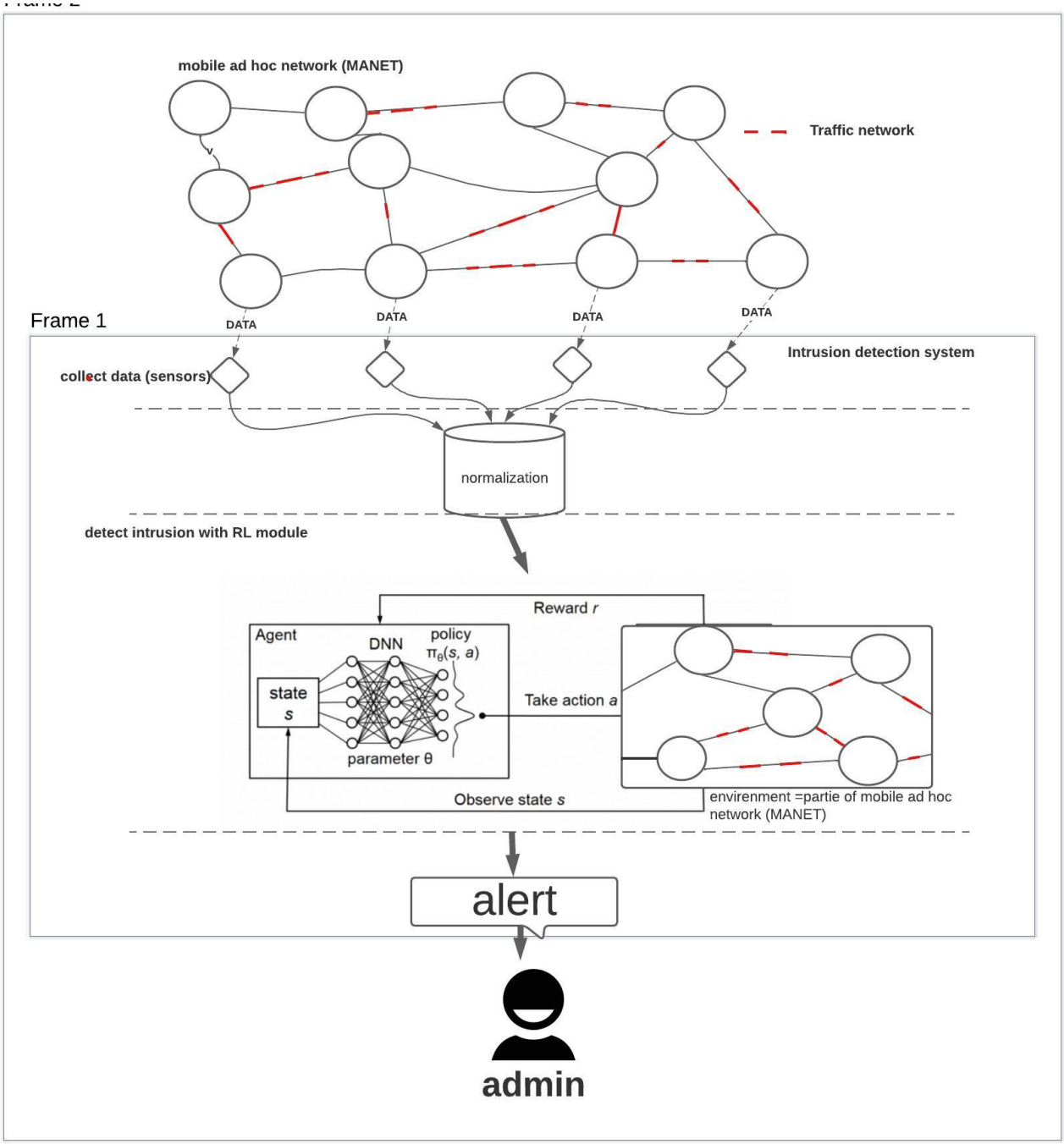

Figure 4.1: Architecture of the proposed solution.

this intrusion detection system contains four module

- Data collect In order to build a robust intrusion detection system, it is important to collect data related to normal network traffic and potential intrusions. This can be achieved by setting up a network sensor that captures network traffic data and stores it in a dataset.

- Normalize Data: Once the data has been collected, it is important to normalize it in order to prepare it for processing. This involves cleaning and organizing the data and ensuring that it is in a consistent format.

- Detect intrusion module with RLThe development of intrusion detection modules utilizing reinforcement learning (RL) techniques. By training RL agents on records that contain normal and malicious activity, agents can learn how to recognize and respond to abnormal behavior. RL agents interact with the environment, observe its state, perform actions, receive rewards, and update policies or value functions accordingly.

- Generate Alerts module: Alert generation module plays a crucial role in intrusion detection systems (IDS) by converting detected anomalies into actionable alerts. when the intrusions are detected this module can generate an alert to notify the system administrator. The alerts can be in the form of emails or messages to a centralized system. once the system administrator notify by the alerts., That's the last one can take actions against this intrusion

4.2.1 Detect Intrusion in Reinforcement Learning

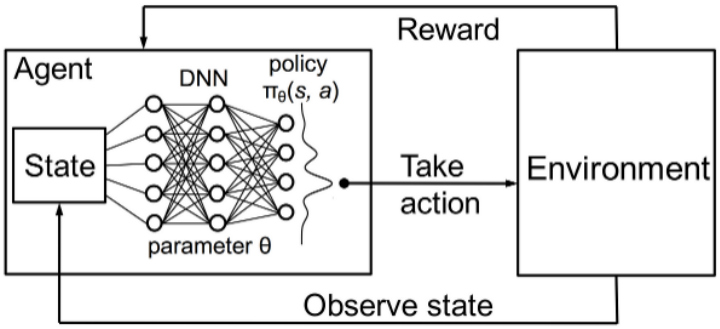

Figure 4.2: deep Q learning [19]

- Observation State: The agent observes the current state of the network, which includes information such as network topology, network traffic, and other relevant network parameters. This information is typically represented as a vector or a matrix.

- Action State: The agent selects actions that correspond to different defensive mechanisms or countermeasures to protect the network from attacks. These actions could include routing, encryption, data filtering, and other security measures.

- Reward Function: The agent receives a reward signal based on its actions and the network's state. The reward function evaluates the agent's performance and assigns a score that indicates how well it has accomplished the task of detecting and responding to attacks.

- Training Algorithm: The agent's behavior is learned using a reinforcement learning algorithm, such as Q-learning, policy gradient, or actor-critic

4.2.2 Training Algorithm:Q-Learning

Q-Learning is a reinforcement learning algorithm, considered one of the most fundamental. In its most simplified form, it uses an array called array-Q to store all the Q-values of all possible action-state pairs. It updates this table using the Bellman equation, while the Stock selection is usually done with a ϵ-greedy policy.

4.2.2.1 Q-value Refresh

- To update the Q-value of any action taken from the previous state, we use the Bellman optimal equation.

- Where is the next state and has any action that can be performed from this point forward

- The formula for calculating the new Q-value for the equity status pair (s,a) in step t is as follows:

$$Q^{new}(s,a) = (1-\alpha) \underbrace{Q(s,a)}_{\substack{ancienne \\ valeur}} + \alpha \overbrace{\left(R_{t+1} + \gamma \max_{a'} Q_*(s',a')\right)}^{\substack{nouvelle \\ valeur}}$$

Figure 4.3: update Q-value

4.3 Implementation

4.3.1 Software Used

4.3.1.1 Anaconda (Python Distribution)

Anaconda is a big collection of scientific Python packages and tools and resources and IDEs. This package contains many essential tools that data scientists can use to harness the incredible power of Python. Anaconda Individual Edition is free and open source. This makes working with Anaconda accessible and easy. Just visit the website and download the distribution[49]

4.3.1.2 VS code

is a text editor developed by Microsoft and it is free and open-source. VS Code is available for many OS such as Windows, Linux, and macOS. Although this editor is relatively lightweight, it includes some powerful features that make VS Code one of the most popular tools in modern development environments.[50]

4.3.1.3 Python

It is a dynamically interpreted (byte code compiled) language. The source code has no type declarations for variables, parameters, functions, and methods. This makes the code shorter and more flexible and loses compile-time source code type checking. Python keeps track of all value types at run time and flags code that doesn't make sense during execution.[51]

4.3.2 Libraries Used

There are many existing libraries adopted by deep learning that have helped us a lot. In this project

4.3.2.1 Pandas

Pandas is a popular Python package that offers efficient, adaptable, and user-friendly data structures specifically designed for handling labeled or relational data. It serves as a fundamental and powerful tool for conducting practical data analysis tasks in Python, enabling users to work with real-world data-sets in a fast and intuitive manner.[52]

4.3.2.2 Keras

Keras is a Python-based deep learning application programming interface (API) that runs on top of the TensorFlow machine learning platform. It is specifically designed to enable rapid experimentation in deep learning. The main goal of Keras is to streamline the process from idea to meaningful results in research by emphasizing speed and efficiency.[53]

and another library that we used in our project such as :

numPY	NumPy is a basic scientific computing package for Python. It provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and a series of quick operations on arrays such as mathematical, logical, shape manipulation, sorting, selection, and I/O. A Python library that provides routines for Discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulations, and more. [54]
matplotlib	Matplotlib is a versatile Python library that offers extensive capabilities for generating visual representations, including static, animated, and interactive visualizations.[55]
sklearn	is a robust and cost-free solution for machine learning. It offers extensive support for both supervised and unsupervised learning, covering a broad spectrum of tasks including classification, regression, clustering, and dimensionality reduction. Moreover, it incorporates a diverse collection of algorithms to cater to various needs in the field of machine learning. [56]

Table 4.1: Libary python

4.4 Project Explanation

4.4.1 Data-set

The NSL-KDD record is the historically flagged network intrusion detection record. It has been used in many tasks to evaluate various deep learning-based algorithms for the development of Various IDS strategies. The NSL-KDD dataset contains 41 features, Regular or specific attack types (i.e. class). we used the one-hot encoding method for this. Reprocessing the dataset to convert categorical features to their numeric counterparts' Values, because deep learning models work only on numeric or floating point values. We used mix-max to normalize the training and test datasets to values between 0 and 1. Normalization strategy. The 41 functions shown in the NSL-KDD dataset can be grouped. It is divided into four functions. B. Basic, Content-Based, Time-Based, and Host-Based Traffic Capabilities. The values of these features are mainly based on continuous, discrete and symbolic values. NSL-KDD records include normal denial of service (DoS), Probe, Route to Local (R2L), and Unauthorized to Route (U2R). These attack classes are: Each NSL-KDD data is identified by a corresponding characteristic. Table 4 NSL-KDD attack classes considered in the study .[21]

Categories	Notation	Definitions	Samples #
Normal	N	Normal activities based on the features	148,517
DoS	D	Attacker tries to avoid users of a service Denial of Service attack	53,385
Probe	P	Attacker tries to scan the target network to collect information such as vulnerabilities	14,077
U2R	U	Attackers with local access to victim's machine tries to get user privileges	119
R2L	P	Attacker without a local account tries to send packets to the target host to get access	3882

Table 4.2: NSL-KDD data categories attack .[21]

Each record has 41 attributes representing different characteristics of the flow, each assigned an attack type or normal label. the Attribute details, i.e. attribute name and descriptions are shown in Table 4.3. [22]

Num	feature	description
1	duration	Length of time duration of the connection
2	protocol-type	Protocol used in the connection
3	service	Destination network service used
4	flag	Status of the connection – Normal or Error
5	source byte	Number of data bytes transferred from source to destination in single connection
6	destination byte	es Number of data bytes transferred from destination to source in single connection
7	land	if source and destination IP addresses and port numbers are equal then, this variable takes value 1 else 0
8	wrong fragment	Total number of wrong fragments in this connection
9	urgent	Number of urgent packets in this connection. Urgent packets are packets with the urgent bit activated
11	hot	Number of 'hot' indicators in the content such as: entering a system
12	Num failed- logins	Count of failed login attempts
13	Logged-in	Login Status : 1 if successfully logged in; 0 otherwise
14	Num-compromised	Number of "compromised' ' conditions
15	Root-shell	1 if root shell is obtained; 0 otherwise
16	Su-attempt ed	1 if "su root" command attempted or used; 0 otherwise
17	Num-root	Number of "root" accesses or number of operations performed as a root in the connection
18	Num-file-creations	Number of files creation operations in the connection
19	Num-shells	Number of shell prompts
20	Num-access-files	Number of operations on access control files
21	Num-outbound-cmds	Number of outbound commands in an ftp session
22	Is-hot-login	1 if the login belongs to the "hot" list i.e., root or admin; else 0

Table 4.3: List of NSL-KDD features with their descriptions [22]

4.4.2 The RLagent

In this subsection, we present how the agent has been defined with :

1. Actions there are two actions (0 and 1) can the agent makes : 1 for the alers and 0 for ignoration

2. Rewards and two corresponding rewards (0 and 1) : 0 for non-intrusion and 1 for intrusion

The next figure 4.4 shows that :

```python
import numpy as np
class RLAgent:
    def __init__(self, model, num_actions):
        self.model = model
        self.num_actions = num_actions
    def predict(self, states ):
        return self.model.predict(states)

    def train_on_batch(self, states, actions , targets):
        q_values = self.predict(states)
        actions = np.asarray(actions, dtype=np.int32)  # Convert actions to integer type
        q_values[np.arange(len(states)), actions.argmax(axis=1)] = targets.flatten()
        # q_values[np.arange(len(states)), actions] = targets
        return self.model.train_on_batch(states, q_values)
    def get_action(self, states, epsilon):
        if np.random.rand() <= epsilon:
            # Random action (alert or ignore)
            actions = np.random.randint(0, self.num_actions, len(states))
        else:
            # Choose action based on model predictions
            q_values = self.predict(states)
            actions = np.argmax(q_values, axis=1)
        return actions
    def get_reward(self, intrusion_labels, actions):
        rewards = np.zeros(len(intrusion_labels))
        for i in range(len(intrusion_labels)):
            if intrusion_labels[i] == 1 and actions[i] == 1:
                # True positive: intrusion correctly detected
                rewards[i] = 1
            elif intrusion_labels[i] == 0 and actions[i] == 0:
                # True negative: non-intrusion correctly ignored
                rewards[i] = 0
            elif intrusion_labels[i] == 1 and actions[i] == 0:
                # False negative: intrusion missed
                rewards[i] = -1
        return rewards
```

Figure 4.4: the agent definition

The RLagent definition :

1. import in the training process with the aim of training it to detect intrusion
2. import in the testing process to test it

Figure 4.5 shows the structure of our project :

- folder Dataset include NSL-KDD dataset
- folder model include model training

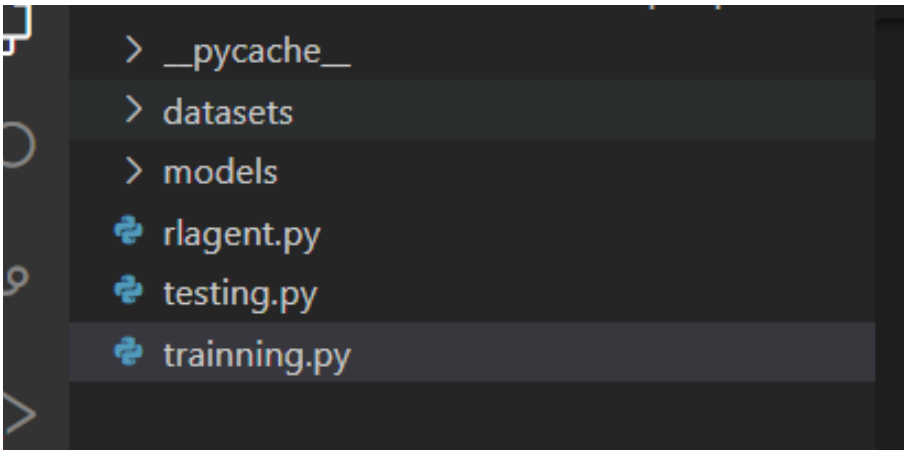

Figure 4.5: structure of project

4.4.3 Training Model

After we define the agent we must train it to detect intrusion using a separate dataset called NSL-KDD with the Q-learning algorithm, the line summarize in figure 4.6is used to load data from the file train in the dataset

```
# Load NSL-KDD train.csv
kdd_path = r'C:\Users\j\Desktop\exemples\datasets\kdd_train.csv'


try:
    df_train = pd.read_csv(kdd_path)
    print(df_train)
```

Figure 4.6: Load data

The next figure 4.7 corresponds to the creation and compilation of a neural network model using the Keras library

- The first line creates a new sequential model object. A sequential model is a linear stack of layers.
- The second line adds a detailed layer to the model. A dense layer is a fully connected layer, meaning that every neuron in this layer is connected to every neuron in the previous layer. Dense layer has 100 units/neurons.
- The third line adds a detailed layer to the model. A dense layer is a fully connected layer, meaning that every neuron in this layer is connected to every neuron in the previous layer. Dense layer has 100 units/neurons.
- The fourth line Add a final dense layer containing num-actions units/neurons.

- The last line assembles the model for training. Specify the optimizer and loss function to use during training. The optimizer used here is SGD (Stochastic Gradient Descent) with a learning rate of 0.2. The loss function used is the mean squared error (MSE), which is often used for regression problems.

```
    model = Sequential()
    model.add(Dense(100, input_shape=(env.state_shape,), activation='relu'))
    model.add(Dense(100, activation='relu'))
    model.add(Dense(num_actions))
    model.compile(SGD(lr=0.2), "mse")

except FileNotFoundError:
```

Figure 4.7: neural network model

After compiling the model, it can be used for training and prediction tasks in a reinforcement learning environment.

We define the different parameters and hyperparameters :

- Num episode : This variable represents the number of episodes in the training loop. In reinforcement learning, an episode is a series of interactions between an agent and its environment. Each episode usually consists of multiple iterations or steps .
- Iteration epoisodes : This variable represents the number of iterations or steps within each episode and determines how long an episode runs before it is considered complete. In the code, each episode will consist of 100 iterations of him.
- EpsilonEpsilon is a hyperparameter used in search strategies that require epsilon. Control the balance between exploration and exploitation during the training process. In your code, epsilon is set to 0.1This means that the agent has a 0.1 probability of exploring (performing random actions) and a 0.9 probability of exploiting (performing the best-known action) during training.
- Decay rate: This variable represents the rate at which epsilon falls over time. Decay is typically used to decrease the search speed gradually as the agent gains experience. In the code, the damping factor is set to 0.99This means that epsilon is reduced by 1% each episode.
- Gamma : Gamma is a discount factor used in reinforcement learning algorithms to determine the importance of future rewards compared to immediate rewards. This influences the agent's decision-making process by weighing potential future benefits. The code sets the gamma to 0.001, suggesting less emphasis on future rewards compared to immediate rewards.

Each episode's loss and total reward are stored in the loss chain and reward chain lists respectively. this was summarized in Figure 4.8

```
87    #
88    num_episodes = 10
89    iterations_episode = 100
90    epsilon = 0.1
91    decay_rate = 0.99
92    gamma = 0.001
93    loss_chain = []
94    reward_chain = []
95
```

Figure 4.8: initialization of hyperparameeter

Now let's go to the main loop where all the process will happen, and that summarized in figure 4.9

```
# Training loop
for episode in range(num_episodes):
    loss = 0
    total_reward_by_episode = 0
    states = env.reset()
    exploration = epsilon * decay_rate ** (episode * iterations_episode)

    for iteration in range(iterations_episode):
        if np.random.rand() <= exploration:
            actions = np.random.randint(0, num_actions, env.batch_size)
        else:
            q_values = model.predict(states)
            actions = np.argmax(q_values, axis=1)

        next_states, reward, done = env.act(actions)
        q_prime = model.predict(next_states)
        max_q_prime = np.max(q_prime, axis=1)
        targets = reward + gamma * max_q_prime
        q_values = model.predict(states)
        q_values[np.arange(env.batch_size), actions] = targets
        loss += model.train_on_batch(states, q_values)

        states = next_states
        total_reward_by_episode += np.sum(reward)

    loss_chain.append(loss)
    reward_chain.append(total_reward_by_episode)

    print(f"Episode: {episode + 1} | Loss: {loss:.4f} | Total Reward: {total_reward_by_episode}")
```

Figure 4.9: trainning loop

During the training process, agents interact with the environment and generate their own

data based on the actions they take, and plotting the total reward and the loss by number of episodes .that summarize in figure 4.10

```
# Save the trained model
model.save_weights(r"C:\Users\j\Desktop\exemples\models\model_weights.h5")
with open(r"C:\Users\j\Desktop\exemples\models\model_architecture.json", "w") as f:
    f.write(model.to_json())

# Plotting the total reward and loss by episode
plt.plot(range(num_episodes), reward_chain)
plt.xlabel("Episode")
plt.ylabel("Total Reward")
plt.show()

plt.plot(range(num_episodes), loss_chain)
plt.xlabel("Episode")
plt.ylabel("Loss")
plt.show()
```

Figure 4.10: save model training

The figure below 4.11 shows the result obtained for the train process :

- Epoch represent the number total of episodes
- Total reward x episodes This value indicates the cumulative reward received by the model during the training process. Rewards are often used in reinforcement learning to guide a model to desired behavior.
- Loss A metric representing the difference between the predicted and actual values during training. Small loss indicates good model performance
- Ones /Zeros represent the number of (1) intrusion and (0) non intrusion
- Total reward for each intrusion type this may display the total reward for each intrusion: 'DoS','Probe','R2L','U2R'

```
Epoch 9.000000/10.000000 | Loss 1.7340 | Tot reward x episode 3177.000000 | ones/zeros 3177.000000\23.000000
Total reward for each intrusion type:
-------------------------------------------------
DoS:  0.0
-------------------------------------------------
Probe:  0.0
-------------------------------------------------
R2L:  0.0
-------------------------------------------------
U2R:  0.0
-------------------------------------------------
```

Figure 4.11: result of trainning

The figure below 4.12 represent the total reward by episode

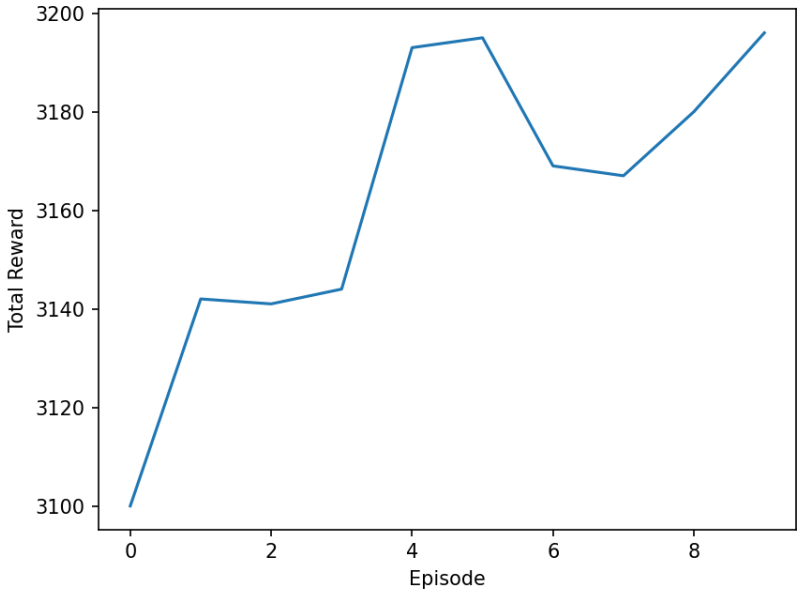

Figure 4.12: total rewards by episodes

The figure below 4.13 represent the loss by episode

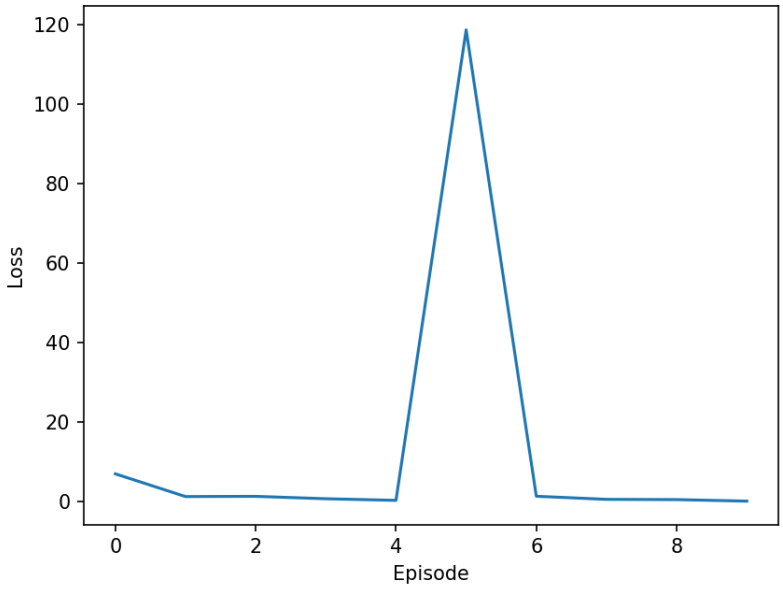

Figure 4.13: Loss by episodes

4.4.4 Testing Model

Once the agent has been trained and the model has been saved, and evaluate its performance. the model's ability to detect intrusion should be assessed based on metrics such as precision and recall and F1-score

- Accuracy: Accuracy is the ratio of correct predictions to the total number of predictions. This is one of the simplest measurements of the model. You should aim for high model accuracy. If the model is highly accurate, we can conclude that the model is making correct predictions in most cases. [57]
- F1-score:It is a harmonious medium of precision and memory. Both false positive and false negative results are considered. So performance is good for imbalanced datasets [58]
- Recall: The ratio (dip) of true positives to the total number of actual positives. Measures the ability of an intrusion detection system to accurately identify positive incidents.
- Precision : is the ratio of true positives to the total number of instances (dips) predicted positive. It measures the accuracy of positive predictions
- Confusion matrix is a matrix to represent the number of True Positives, False Positives, True Negatives, and False Negatives [57]

This has been summarized in the figure 4.14and the figure 4.15

```
# Calculate evaluation metrics
accuracy = accuracy_score(ground_truth_labels, predicted_actions)
f1 = f1_score(ground_truth_labels, predicted_actions, average='weighted')
precision = precision_score(ground_truth_labels, predicted_actions, average='weighted')
recall = recall_score(ground_truth_labels, predicted_actions, average='weighted')

# Print the evaluation metrics
print('Accuracy:', accuracy)
print('F1 Score:', f1)
print('Precision:', precision)
print('Recall:', recall)
```

Figure 4.14: plot evaluation metric

```
15
16      # Calculate the confusion matrix
17      cm = confusion_matrix(ground_truth_labels, predicted_actions)
18
19      # Plot the confusion matrix
20      plt.figure(figsize=(8, 6))
21      plot_confusion_matrix(cm, classes=['Normal', 'Dos', 'Prob', 'R2L'], normalize=True)
22      plt.show()
```

Figure 4.15: plot matrix confusion

4.5 Performance

In this section, we present the obtained result from the approach proposed
after the agent trained to detect intrusion the figure below 4.16 represent the performance metric

```
Epoch 2253/2254 | Ones/Zeros: 12548/8266  Tot Rew --> 20814
Total reward: 20814 | Number of samples: 22540 | Acuracy = 92.34250221827861%
Accuracy: 0.9234250221827861
F1 Score: 0.9226469940342361
Precision: 0.9270446293179364
Recall: 0.9234250221827861
Normalized confusion matrix
[[0.85128733 0.14871267]
 [0.02197973 0.97802027]]
```

Figure 4.16: perfermonce metric

The figure below 4.17 represent the confusion matrix

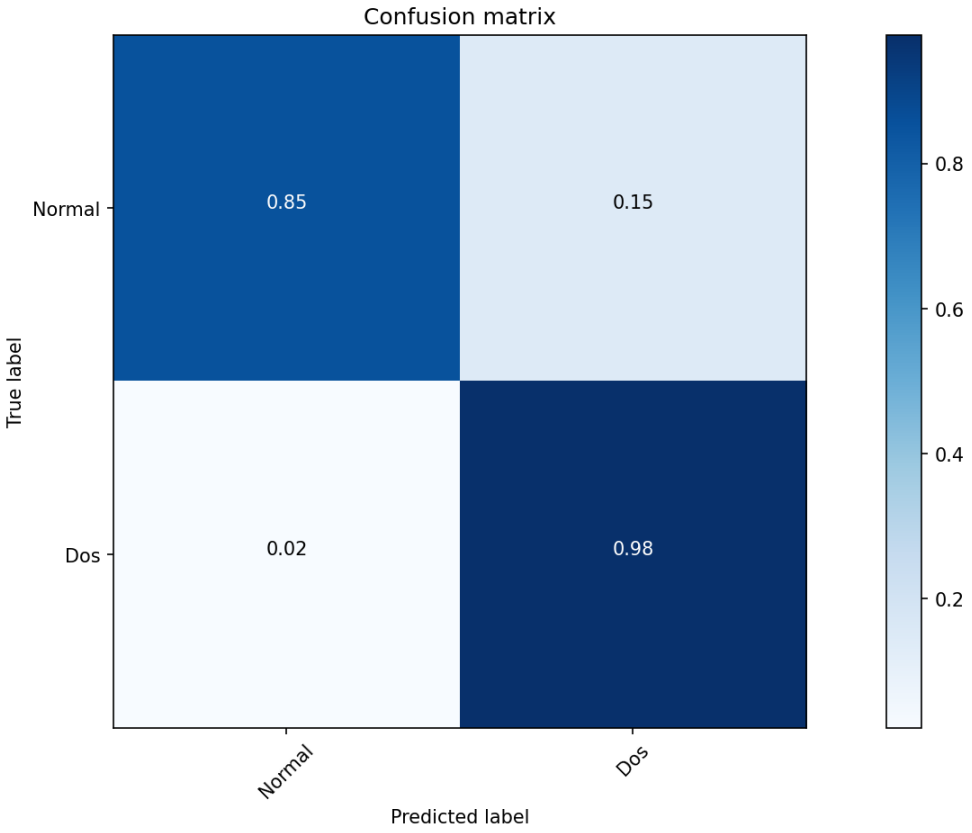

Figure 4.17: matrix confusion

4.6 Conclusion

In this chapter, we presented and explained the developed tools used in our proposed architecture, and provided a thorough description of the main implementation details. The evaluation results are presented and discussed.

General Conclusion

In this thesis, the focus was on the development of an intrusion detection system for Mobile Ad Hoc Networks. Throughout this work, We explored the difficulties posed by the dynamic and decentralized characteristics of MANETs and proposed an approach that utilizes reinforcement learning to improve intrusion detection accuracy.

We have succeeded to overcome the drawbacks of current IDS solutions by harnessing the capabilities of reinforcement learning techniques. By employing reinforcement learning algorithms, we create an intelligent and adaptable IDS capable of learning from network environment interactions, making informed decisions, and efficiently detecting and responding to security attacks in real-time.

The obtained results demonstrated the effectiveness of our intrusion detection system, which was able to autonomously learn from interactions with the environment. By leveraging the capabilities of reinforcement learning, our system adapted to behavioral changes and malicious attacks in MANETs, thereby reinforcing communication security.

While our reinforcement-based IDS has demonstrated promising results, it is crucial to acknowledge that there is still scope for further investigation and improvement. Subsequent research endeavors can concentrate on optimizing the learning algorithms, enhancing the system's scalability, and integrating supplementary security mechanisms. These efforts aim to deliver a comprehensive security solution tailored to the specific requirements of MANETs.
Furthermore, our work can extend to the administrator who receives intrusion alerts. We can emulate the behavior of the administrator in their reactions by employing imitation learning techniques, treating the administrator as an agent. This approach allows the system to learn from the administrator's actions and reactions, ultimately enhancing the overall effectiveness of the intrusion detection and response process.

Bibliography

[1] "Manet: Self-organizing wireless network.." https://chat.openai.com/chat/6d3a2378-e4f5-4066-80a7-72ee89a2f600. (Accessed on 03/15/2023). XI, 5

[2] K. Nieminen, "Introduction to ad hoc networking," Networking Laboratory, Helsinki University of Technology, 2003. XI, 6

[3] L. Yong-Min, W. Shu-Ci, and N. Xiao-Hong, "The architecture and characteristics of wireless sensor network," in 2009 International Conference on Computer Technology and Development, vol. 1, pp. 561–565, IEEE, 2009. XI, 7

[4] S. Yogarayan, "Wireless ad hoc network of manet, vanet, fanet and sanet: A review," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 13, no. 4, pp. 13–18, 2021. XI, XIII, 8, 9, 10

[5] I. A. Sumra, P. Sellappan, A. Abdullah, and A. Ali, "Security issues and challenges in manet-vanet-fanet: A survey," EAI Endorsed Transactions on Energy Web, vol. 5, no. 17, pp. e16–e16, 2018. XI, 8, 9

[6] M. Kumar and R. Mishra, "An overview of manet: history, challenges and applications," Indian Journal of Computer Science and Engineering (IJCSE), vol. 3, no. 1, pp. 121–125, 2012. XI, 13

[7] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," Journal of network and computer applications, vol. 36, no. 1, pp. 42–57, 2013. XI, 23, 24

[8] "Basics of intrusion detection system, classifactions and advantages." https://www.elprocus.com/basic-intrusion-detection-system/. (Accessed on 03/30/2023). XI, 25

[9] E. Amiri, H. Keshavarz, H. Heidari, E. Mohamadi, and H. Moradzadeh, "Intrusion detection systems in manet: a review," Procedia-Social and Behavioral Sciences, vol. 129, pp. 453–459, 2014. XI, XIII, 26, 27

[10] M. Aljanabi, M. A. Ismail, and A. H. Ali, "Intrusion detection systems, issues, challenges, and needs," International Journal of Computational Intelligence Systems, vol. 14, no. 1, pp. 560–571, 2021. XI, 29

[11] "What is deep learning? | ibm." https://www.ibm.com/topics/deep-learning. (Accessed on 07/06/2023). XI, 34

[12] "Deep learning essentials | packt." https://www.packtpub.com/product/deep-learning-essentials/9781785880360. (Accessed on 03/24/2023). XI, 35

[13] "Demystifying deep learning." https://ekababisong.org/demystifying-deep-learning/. (Accessed on 07/10/2023). XI, 36

[14] "Université d'oum-el-bouaghi: Deep learning based approach for predicting depression using twitter data." http://bib.univ-oeb.dz:8080/jspui/handle/123456789/14292. (Accessed on 04/09/2023). XI, XII, 37, 38, 39

[15] "Logiciel de présentation : Powerpoint." file:///C:/Users/j/Pictures/Screenshots/Chapitre4.pdf. (Accessed on 03/25/2023). XII, 41

[16] "Reinforcement learning tutorial - javatpoint." https://www.javatpoint.com/reinforcement-learning. (Accessed on 03/26/2023). XII, 43

[17] "Th.m.inf.fr.2021.16.pdf." http://dspace.univ-tiaret.dz/bitstream/123456789/5487/1/TH.M.INF.FR.2021.16.pdf. (Accessed on 04/05/2023). XII, 41, 44

[18] "Id-rdrl: a deep reinforcement learning-based feature selection intrusion detection model | scientific reports." https://www.nature.com/articles/s41598-022-19366-3. (Accessed on 04/16/2023). XII, 45, 46, 47

[19] "Week of 3/7." https://sratcaltechct.blogspot.com/2019/03/week-of-37.html. (Accessed on 04/20/2023). XII, 52

[20] "Survey of intrusion detection systems: techniques, datasets and challenges | cybersecurity | full text." https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0038-7. (Accessed on 04/03/2023). XIII, 24

[21] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep q-learning based reinforcement learning approach for network intrusion detection," Computers, vol. 11, no. 3, p. 41, 2022. XIII, 47, 56

[22] L. Dhanabal and S. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," International journal of advanced research in computer and communication engineering, vol. 4, no. 6, pp. 446–452, 2015. XIII, 56, 57

[23] J. A. Stankovic, "Wireless sensor networks," computer, vol. 41, no. 10, pp. 92–95, 2008. 6

[24] V. Student and R. Dhir, "A study of ad-hoc network: A review," Int. J, vol. 3, no. 3, pp. 135–138, 2013. 7

[25] J. P. Macker and M. S. Corson, "Mobile ad hoc networking and the ietf," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 2, no. 1, pp. 9–14, 1998. 7

[26] L. Zhou and Z. J. Haas, "Securing ad hoc networks," IEEE network, vol. 13, no. 6, pp. 24–30, 1999. 12

[27] N. Raza, M. Umar Aftab, M. Qasim Akbar, O. Ashraf, and M. Irfan, "Mobile ad-hoc networks applications and its challenges," Communications and Network, vol. 8, no. 03, pp. 131–136, 2016. 12, 13

[28] M. Chitkara and M. W. Ahmad, "Review on manet: characteristics, challenges, imperatives and routing protocols," International journal of computer science and mobile computing, vol. 3, no. 2, pp. 432–437, 2014. 14, 15

[29] P. Goyal, V. Parmar, R. Rishi, et al., "Manet: vulnerabilities, challenges, attacks, application," IJCEM International Journal of Computational Engineering & Management, vol. 11, no. 2011, pp. 32–37, 2011. 15, 17, 18

[30] O. O. Obi, "Security issues in mobile ad-hoc networks: a survey," The 17 th White House Papers Graduate Research In Informatics at Sussex, 2004. 16

[31] S. Lalar, "Security in manet: Vulnerabilities, attacks & solutions," International Journal of Multidisciplinary and current research, vol. 2, pp. 62–69, 2014. 19

[32] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013. 22

[33] P. Ning and S. Jajodia, "Intrusion detection techniques," The Internet Encyclopedia, vol. 2, pp. 355–367, 2003. 22

[34] "Projet master pfe laala - online latex editor overleaf." https://fr.overleaf.com/project/63fcd0eb70fc4ba66a8541b0. (Accessed on 04/01/2023). 22

[35] A. S. Ashoor and S. Gore, "Importance of intrusion detection system (ids)," International Journal of Scientific and Engineering Research, vol. 2, no. 1, pp. 1–4, 2011. 23

[36] "(pdf) intrusion detection systems: A review." https://www.researchgate.net/publication/320517757_INTRUSION_DETECTION_SYSTEMS_A_REVIEW. (Accessed on 03/21/2023). 25

[37] "(pdf) intrusion detection system." https://www.researchgate.net/publication/3165599266_INTRUSION_DETECTION_SYSTEM. (Accessed on 03/25/2023). 28

[38] A. S. Ashoor and S. Gore, "Difference between intrusion detection system (ids) and intrusion prevention system (ips)," in Advances in Network Security and Applications: 4th International Conference, CNSA 2011, Chennai, India, July 15-17, 2011 4, pp. 497–501, Springer, 2011. 28

[39] "Intrusion detection systems, issues, challenges, and needs | atlantis press." https://www.atlantis-press.com/journals/ijcis/125951139/view#fig-F2. (Accessed on 03/29/2023). 29, 30

[40] "Benefits of having intrusion prevention/detection system in your enterprise." https://www.seqrite.com/blog/benefits-of-having-intrusion-preventiondetection-system-in-your-enterprise/. (Accessed on 03/30/2023). 30

[41] "Brief history of deep learning from 1943-2019 [timeline] - mlk - machine learning knowledge." https://machinelearningknowledge.ai/brief-history-of-deep-learning/. (Accessed on 03/24/2023). 35

[42] H. Zeng, Z. Liu, and H. Cai, "Research on the application of deep learning in computer network information security," in Journal of Physics: Conference Series, vol. 1650, p. 032117, IOP Publishing, 2020. 36

[43] "What is reinforcement learning? | definition from techtarget." https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning. (Accessed on 04/03/2023). 40

[44] "Reinforcement learning : Définition et application." https://datascientest.com/reinforcement-learning. (Accessed on 03/24/2023). 40

[45] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," Ieee Access, vol. 5, pp. 21954–21961, 2017. 47

[46] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in 2018 20th International Conference on Advanced Communication Technology (ICACT), pp. 178–183, IEEE, 2018. 47

[47] "Sci-hub | deep reinforcement learning based intrusion detection system for cloud infrastructure. 2020 international conference on communication systems & networks (comsnets) | 10.1109/comsnets48256.2020.9027452." https://sci-hub.ru/10.1109/comsnets48256.2020.9027452. (Accessed on 05/06/2023). 48

[48] "Drl-ids:deep reinforcement learning based intrusion detection system for industrial internet of things." https://www.jsjkx.com/EN/10.11896/jsjkx.210400021. (Accessed on 05/07/2023). 48

[49] "An overview of the anaconda distribution | by dan root | towards data science." https://towardsdatascience.com/an-overview-of-the-anaconda-distribution-9479ff1859e6. (Accessed on 05/12/2023). 54

[50] "What is visual studio code?." https://www.educative.io/answers/what-is-visual-studio-code. (Accessed on 05/13/2023). 54

[51] "Python introduction | python education | google for developers." https://developers.google.com/edu/python/introduction. (Accessed on 05/25/2023). 54

[52] "Package overview — pandas 2.0.1 documentation." https://pandas.pydata.org/docs/getting_started/overview.html. (Accessed on 05/15/2023). 55

[53] "About keras." https://keras.io/about/. (Accessed on 05/15/2023). 55

[54] "What is numpy? — numpy v1.24 manual." https://numpy.org/doc/stable/user/whatisnumpy.html. (Accessed on 05/15/2023). 55

[55] "Matplotlib documentation — matplotlib 3.7.1 documentation." https://matplotlib.org/stable/index.html. (Accessed on 05/15/2023). 55

[56] "Introduction to scikit-learn (sklearn) in python • datagy." https://datagy.io/python-scikit-learn-introduction/. (Accessed on 05/15/2023). 55

[57] "Understanding accuracy, recall, precision, f1 scores, and confusion matrices | by rahul banerjee | towards data science." https://towardsdatascience.com/understanding-accuracy-recall-precision-f1-scores-and-confusion-matrices-561e0f5e328c. (Accessed on 05/27/2023). 64

[58] "Performance metrics: Confusion matrix, precision, recall, and f1 score | by vaibhav jayaswal | towards data science." https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-and-f1-score-a8fe076a2262. (Accessed on 05/27/2023). 64