PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

Analysis and detection of routing attacks in the internet of Things using Deep learning

SMAILI Abdelkarim & KACHOUR Imad Eddine

A THESIS

Submitted in partial fulfillment of the
requirements for the degree of:

Master in Networks and Telecommunications

Ibn khaldoun university - Tiaret

Publicly defended in Tiaret by the committee composed of:

Mr. DAHMANI Youcef	P.R	IBN-KHALDOUN university -Tiaret	President
Mr. ALEM Abdelkader	M.A.A	IBN-KHALDOUN university -Tiaret	Supervisor
Mr CHADLI Abdelhafidh	M.C.B	IBN-KHALDOUN university -Tiaret	Reviewer

2020-2021

ACKNOWLEDGMENTS

First and foremost, thank ALLAH who gave us strength and the Patience to accomplish this modest work.

A very big thank to:

Our parents who followed us during our studies.

We would like to thank our supervisor Mr. ***ALEM Abdelkader*** *for his continuous and unlimited support, feedback and adjustment of the research. We would also like to thank the committee consisting of Mr.* ***DAHMANI Youcef*** *and Mr* ***.CHADLI Abdelhafidh*** *. As we don't forget Mr.****BOUAZZA Abdelhamid*** *for his support and assitance.*

Secondly, we would like to thank all teachers and staff at UIK for help and support during our studies. The year

Dedication

I dedicate my master's thesis to:

my mother, my father, my brothers and sisters ,my family ,my dear uncle mustapha and my friends ibrahim and abdelkader for the unlimited encouragement and support during my master's study at UJK.

ABDELKARIM SMAILI

Dedication

To my parents .

The reason of what I become today.

Thanks for your great support and contious care .

To my brothers .

I am really grateful to both of you, you have ben my inspiration, and my soul mates

To my friends

Thank you for being in my life.

EMAD EDDINE KACHOUR

Abstract

Internet of things applications is growing day by day, as they are being used in many areas and systems, and as their uses and modes of employment increase, there are many gaps with them, the most important problem is security.

IOT has a large number of connected devices and therefore mobile data traffic is large and routing protocols are a key element.

IOT has many routing protocols, the most widely used is RPL protocol, which takes into account limited power and the device's capabilities, but it suffers from several weaknesses, the most important one is routing based attacks which targeting this protocol.

In this work, we aim to solve the problem of Internet of Things exposure to RPL-based attacks as routing protocol. We built an anomaly **intrusion detection system** based on deep learning and an IoT attacks dataset (Minerva-IoT) containing the most important attacks built through Cooja simulator and implementation of different scenarios that allowed for the extraction of important features with the addition of new sensitive features such as **nodes power** and their **geographical location, balancing** the dataset by fix minority classes (**rare attacks**) to avoid fake performance using smart algorithms.

The results were very satisfactory after the most important challenges in intrusion detection systems were achieved from a **false alarm rate** (false positive), **accuracy** and **precision**.

Keywords: Internet of things, DODAG, RPL, Security, Attacks, intrusion detection systems

الملخص

تزداد الحاجة يوما بعد يوم إلى تطبيقات إنترنت الأشياء، حيث أصبحت تُستخدم في العديد من المجالات والأنظمة، ومع هذا التزايد الواسع لاستخداماتها وطرق توظيفها، بدأت تظهر معها ثغرات عديدة أهمها مشكل الحماية .

إن شبكات أنترنت الأشياء تحتوي على عدد كبير من الأجهزة المتصلة وبالتالي حركة البيانات المتنقلة فيها تكون كبيرة و بروتوكولات التوجيه هي عنصر أساسي لذلك.

يوجد العديد من بروتوكولات التوجيه في أنترنت الأشياء, أكثرها إستخداما وإعتمادا هو بروتوكول RPL الذي يأخذ بعين الإعتبار محدودية الطاقة و إمكانيات الأجهزة المتصلة لكنه يعاني من عدة نقاط ضعف أهمها الهجمات والتسللات التي تستهدف هذا البروتوكول.

نهدف في هذا العمل إلى حل مشكلة تعرض شبكات أنترنت الأشياء إلى الهجمات التي تعتمد على بروتوكول RPL كبروتوكول توجيه .قمنا ببناء نموذج بإستخدام التعلم العميق ومجموعة بيانات تحتوي على أهم الهجمات تم بنائها من خلال عمل محاكاة بأداة Cooja وتنفيذ سيناريوهات مختلفة سمحت بالتوصل إلى سمات مهمة مع إضافة سمات جديدة مؤثرة كطاقة العقد ومقرها الجغرافي مما جعل مجموعة البيانات شاملة لأغلب هجمات التوجيه الخاصة بأنترنت الأشياء كما تم حل مشكل البيانات القليلة (فئات الهجمات) بالإعتماد على خوارزميات ذكية لجعل عددها كافي من أجل بناء نموذج قوي

كانت النتائج المتوصل إليها مرضية للغاية وذلك بعد تحقيق أهم التحديات في أنظمة كشف التسلل من معدل أدنى من الإنذارات الكاذبة (إيجابية كاذبة) و معدل أقصى لإكتشاف الهجمات .

الكلمات المفتاحية : أنترنت الأشياء , أنظمة كشف التسلل,بروتوكول التوجيه ,الامن,RPL,DODAG....

Table of contents

Abstract 5
الملخص 6
List of Figures I
List of Tables II
List of acronymes : 3
General Introduction: 1
I - Background: 1
II - Research problem: 1
III - Research Objectives: 2
IV - Document Outline: 2
Chapter 1 : Internet of Things 3
Introduction : 3
I - Definition: 4
II - IoT connectivity models: 4
1 - Device to device: 4
2 - Device to cloud: 4
3 - Device to gateway: 5
4 - Back-end Data-Sharing: 6
III - IoT applications: 6
1 - Smart home: 6
2 - Enterprise asset management: 7
3 - Wearables: 7
4 - Health: 7
5 - Traffic monitoring: 7
6 - Water supply: 8
7 - Agriculture: 8
IV - IoT Architecture: 8
1 - Three-and Five-Layer Architectures: 8
V - Communication protocols, standards and regulations: 10
VI - Low-power and lossy network protocols: 11
VII - IoT Protocols: 12
1 - IEEE 802.15.4: 13
2 - 6LOWPAN: 14
3 - MQTT: 14

4 - Coap: 14
5 - RPL: 15
VIII - Vulnerabilities and threats in the Internet of Things: 17
1 - Vulnerabilities of internet of things: 17
IX - RPL Routing attacks in IoT: 19
1 - RPL Routing attacks examples: 20
X - Security requirements in the IoT: 21
1 - Confidentiality: 21
2 - Integrity: 22
3 - Availability: 22
4 - Authentication and authorization: 22
XI - Security Challenges in the IoT: 22
1 - Interoperability: 23
2 - Resource constraints: 23
3 - Resilience to physical attacks and natural disasters: 23
4 - Autonomic control: 23
5 - Scalability: 23
6 - Information volume: 23
XII - Conclusion: 23
Chapter 2: Intrusion Detection systems 24
Introduction: 24
I - Intrusion detection systems (IDSs): 24
II - IDS functionalities: 24
1 - Data Collection: 25
2 - Feature Selection: 25
3 - Analysis: 25
4 - Action: 25
III - IDS Architecture: 25
IV - Taxonomy of Intrusion Detection Systems (IDSs): 26
1 - Information (data) source**:** 26
2 - The analysis strategy**:** 26
3 - Time aspects: 27
4 - IDSs architectures: 27
5 - Detection response: 27
V - IDSs types: 28
1 - A host-based intrusion detection system: 28

2 - A network-based intrusion detection system: 28
VI - Comparison between types of IDS: 29
VII - IDSs detection techniques: 29
1 - Misuse-based intrusion detection 29
2 - Anomaly-based intrusion detection: 30
VIII - IDSs: performance evaluation: 32
IX - IDS Challenges: 33
X - Limitations of Intrusion Detection Systems: 33
XI - Conclusion: 34
Chapter 3: Deep learning 35
Introduction: 35
I - Difference between machine learning and deep learning: 36
II - Deep learning applications: 36
III - Neural Networks: 36
- 1 Neuron: 37
2 - Mathematical equation for neural networks: 37
3 - Cost-function equation: 38
4 - Data path in neural networks: 39
5 - Back propagation: 40
6 - Neural networks Types: 40
7 - Self-organizing map (SOM): 41
8 - Convolutional neural networks: 41
9 - . Deep neural networks (DNN): 42
10 - Recurrent neural networks (RNN): 43
IV - Principles for deep learning IDS in IoT: 44
V - Conclusion: 45
Chapter 4: Contribution in the detection of intrusions in the IOT environment 44
Introduction: 44
I - Related work: 44
II - Proposed framework: 46
1 - Overall Architecture: 47
2 - Training and Optimization of CNN Framework: 49
III - Rpl routing attacks dataset: 50
1 - Dataset generation: 51
2 - Dataset Description: 53
3 - Data balancing: 54

4 - Dataset splitting: 55
IV - Evaluation and Metrics: 55
1 - Comparative study: 55
2 - Comparative study with related works: 56
3 - Test effectiveness of final Model with NSL-KDD dataset (10%): 57
V - Implementation Tools: 59
1 - Programming environment python: 59
2 - Anaconda Integrated development environment (IDE): 60
3 - Used Libraries: 60
VI - Conclusion : 62
General conclusion: 63
Bibliography 63
Web Sources 69

List of Figures

Figure 1:IOT connected devices distribution 3
Figure 2:device to device communication model 4
Figure 3:device to cloud communication model 5
Figure 4:device to gateway communication model 5
Figure 5: Back-end Data-Sharing communication model 6
Figure 6:Conventional IoT architecture 9
Figure 7:The five-layer IoT architecture 10
Figure 8:Comparison of wireless power consumption with data rates 12
Figure 9:LLN networks protocols 13
Figure 10:(a) send DIO, (b) update DIO message, (c) send DAO 17
Figure 11:Taxonomy of attacks against RPL networks 19
Figure 12:Blackhole Attack (data received from legitimate nodes is dropped) 20
Figure 13:Flooding Attack (A means attacker node) 21
Figure 14:IDS Functionality 24
Figure 15:IDS Architecture 26
Figure 16:Taxonomy of intrusion detection systems according to proposed five criteria 27
Figure 17:Comparative architecture between NIDS and HIDS 28
Figure 18:typical misuse detection 30
Figure 19:Typical anomaly detection 31
Figure 20:Confusion Matrix for IDS System 32
Figure 21:Intrusion detection system (IDS) challenges 33
Figure 22:Artificial intelligence areas 35
Figure 23:Neural networks design 36
Figure 24:Neuron shape 37
Figure 25: mathematical calculation for neural networks 37
Figure 26:Mathematical equations simple example 38
Figure 27:Cost function for each sample 39
Figure 28:Forward Propagation (predicted values calculation) 39
Figure 29:Back propagation (weights update) 40
Figure 30:Feed forward neural network 40
Figure 31:self-organizing map 41
Figure 32:CNN architecture 42
Figure 33:DNN architecture 42

Figure 34:Recurrent neural networks 43
Figure 35:Some Deep learning algorithms 44
Figure 36:Conv 1D CNN architecture 47
Figure 37: Dataset before balancing 48
Figure 38:Dataset After balancing 48
Figure 39:CNN framework 49
Figure 40:training and optimization of the proposed CNN framework 49
Figure 41: features extraction algorithm 52
Figure 42: Different steps to build the dataset. 52
Figure 43:Confusion Matrix with an IOT-dataset 56
Figure 44: NSL-kdd dataset statistics 58
Figure 45:confusion matrix with NSL-KDD dataset 59

List of Tables

Table 1:layer-Wise IoT protocols 11
Table 2: OWASP top 10 vulnerabilities 18
Table 3: Comparison between HIDS and NIDS performance 29
Table 4: comparison between DL and ML 36
Table 5:Features description 53
Table 6:dataset statistics 54
Table 7:original dataset and oversampled dataset 55
Table 8:train, validation and test set 55
Table 9:Comparative study between classifiers 56
Table 10:comparison with used dataset in each work 57
Table 11:comparison with related works performance 57
Table 12:train, validation and test set after sampling 58
Table 13:model performance with NSL-KDD 58

List of acronymes :

6LoPAN: IPv6 over Low -Power Wireless Personal Area Networks

ACK: Acknowledgment

DAO: Dodag advertisement Object

DIO: Dodag information Object

DIS: Dodag information sollicitation

DODAG: Destination Oriented Directed Acyclic Graph

IDO: Internet Des Objects

LLN:Low power and lossy networks

IDS: Intrusion Detection Systems

IOT: Internet of Things

LTE-A : Long Term Evolution—Advanced

M2M: Machine to machine

P2P : Peer-to-Peer

RFID: Radio-frequency identification

RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks

WLAN: Wireless Local Area Network

WSN: Wireless sensor network

DL: deep learning

ML: Machine learning

OF : objective function

OWASP : Open Web Application Security Project

HIDS: host-based intrusion detection system

General Introduction:

I - Background:

The Internet of Things (IoT) is a network of everyday physical objects that can connect to Internet to communicate and synthesize data using existing network resources. These objects (nodes) are the interconnected digital devices or sensors that are capable of exchanging this information over the global Internet. These interactions between sensors, connectivity, and people and processes creates new applications and services. These digital devices or sensors are referred to as the "Things" in the Internet of "Things". which are connected to the Internet via 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) is used to improve the quality of life (e.g., smart cities, smart buildings, smart cars). Each node can reach other nodes and exchange routing informations using RPL(ipv6 routing protocol for low-power and lossy networks).

However, due to its ad-hoc and limited resource structure, IoT systems are very vulnerable to attacks. Generally, attacks target the usability and energy consumption of a node connected to a heavy data stream. Attack detection systems are one of the security measures and are crucial in an IoT ecosystem.

RPL is a novel distance vector routing protocol standardized for constrained 6LoWPAN networks enabling nodes to communicate in a mesh topology. Unfortunately, several attacks exist on the RPL protocol that target a node's availability, and increase dramatically its power consumption.

II - Research problem:

Security issues pose the greatest challenge against the routing in internet of things. Unlike the traditional networks, IoT networks suffer from a lack of well-established and standardized design concepts such as the client-server model. This deficiency prevents a large variety of traditional security solutions from being implemented in IoT networks.

With the increasing number of IoT devices, IoT is becoming a lucrative platform for a variety of Internet attacks which can occur in various forms, targeting different resources on a variety of IoT devices. Continuous monitoring and analysis are needed for securing IoT systems. Because of the vast amount of network and sensing data produced by IoT devices

and systems, Big Data and ML (machine learning) methods are highly effective in continuous monitoring and analysis for the security of IoT systems.

Traditional ML methods such as Bayesian networks (BN), support vector machines (SVM) and others have been applied for cyber security, however the large scale of data generated in IoT call for a deep learning-based method which performs better with large data sizes and is adaptable todifferent attack scenarios.

III - Research Objectives:

The goal of this research is to develop a DL (Deep learning) intrusion detection system for detection of routing attacks in IoT.In this study we have focused on specific IoT routing attacks named, decreased rank, version number, Blackhole, Hello flood. Accordingly, it can perform a mitigation action if an intrusion is detected. It is designed to work for a wide variety of IoT networks, ranging from Personal Area Networks (PANs) to Metropolitan Area Networks (MANs), to enterprise class Local Area Networks (LAN) and Wide Area Networks (WANs).

The objectives of this research are:

- Deeply understand of IoT routing mechanisms and attacks.
- Develop a deep learning-based IDS model for IoT networks.
- Evaluate the model based on (False alarm rate, Accuracy, Precision).

IV - Document Outline:

The thesis is organized as follows:

- Chapter 1 presents a study on the IoT, RPL protocol and security problems in IoT.
- The second one presents intrusion detection systems
- The third presents DL and its implementation on the field of cyber security.
- Last chapter represent related works and our contribution.

Chapter 1 : Internet of things

Chapter 1 : Internet of Things

Introduction :

The Internet of Things (IoT) refers to a system of interrelated, internet-connected objects that are able to collect and transfer data over a wireless network without human intervention.

in this chapter we are going to talk about Internet of Things and its important applications in all aspects of our lives, and then we are going to introduce low-power networks with resource-constrained devices as the infrastructure of the Internet of Things, and then study its various protocols. Finally look at an overview of security in the Internet of Things before getting specifically into the topic of IoT routing.

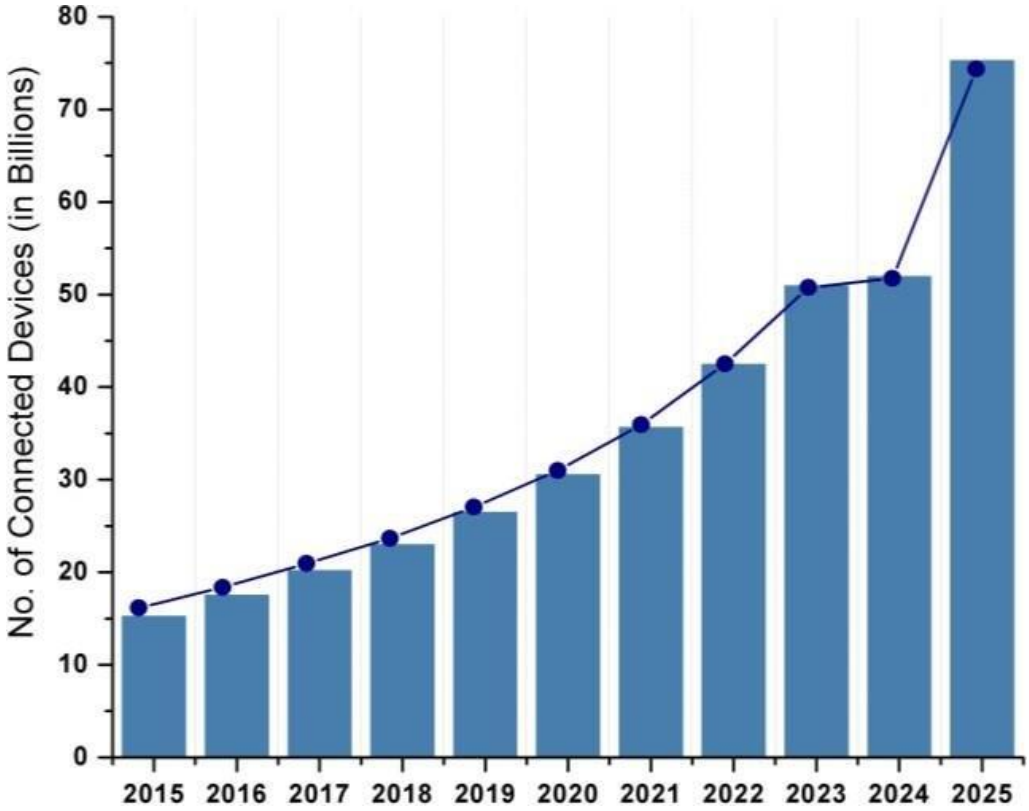

Figure 1:IoT connected devices distribution **[1]**

I - Definition:

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

II - IoT connectivity models:

1 - Device to device:

Device-to-device communication represents two or more devices that directly connect and communicate between one another **[2]** . They can communicate over many types of networks, including IP networks or the Internet, but most often use protocols like Bluetooth, Z-Wave, and ZigBee.

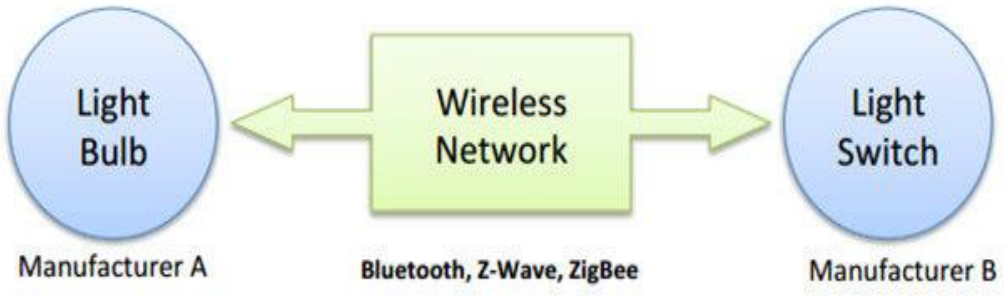

Figure 2:device to device communication model **[3]**

2 - Device to cloud:

Many IoT devices connect to the cloud, often with the use of wired Ethernet or Wi-Fi. Connecting to the cloud allows users and related applications to access the devices, making it possible to course through commands remotely as well as push necessary updates to the device software **[2]** . Through this connection, the devices can also collect user data for the improvement of their service providers.

Figure 3:device to cloud communication model **[3]**

3 - Device to gateway:

Before connecting to the cloud, IoT devices can communicate first with an intermediary gateway device **[2]**. The gateway can translate protocols and add an additional layer of security for the entire IoT system. In the case of a smart home, for example, all smart devices can be connected to a hub (the gateway) that helps the different devices to work together despite having different connection protocols.

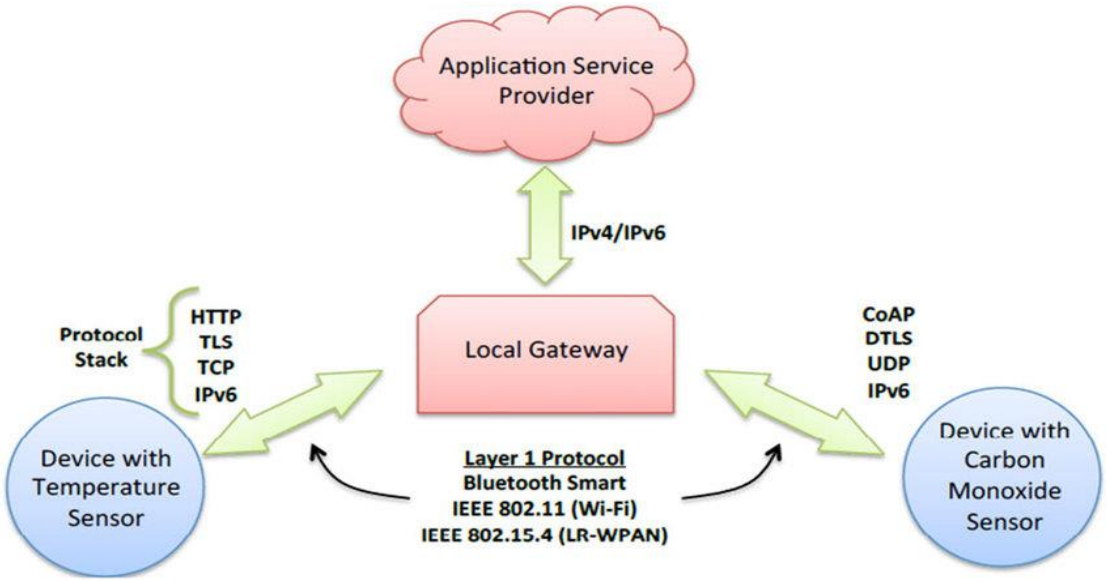

Figure 4:device to gateway communication model **[3]**

4 - Back-end Data-Sharing:

An extension of the device-to-cloud model, this model allows users to gain access to and analyze a collection of data from different smart devices **[2]**. A company, for instance, can use this model to access information from all of the devices working inside the company building as organized together in the cloud.

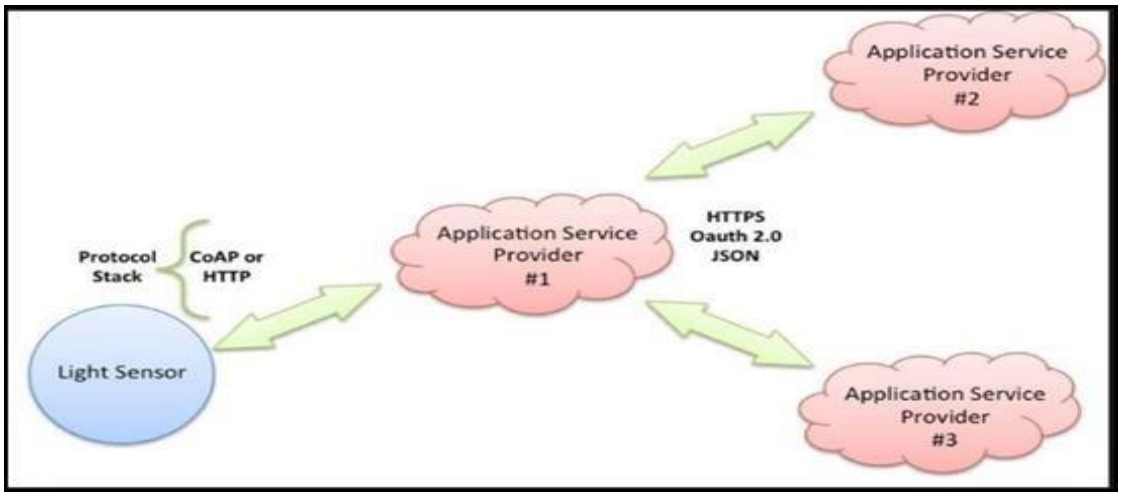

Figure 5: Back-end Data-Sharing communication model **[3]**

III - IoT applications:

The Internet of Things (IoT) promises to bring immense value to every organization. By continuing to connect all our things, people, and environments, we will unlock tremendous organizational value and achieve feats that will truly seem like magic. But because IoT is so broad and far-reaching of a concept, we have found that many are confused about what the potential applications for IoT are exactly. How can my business actually implement IoT solutions? How should my city think about creating value for residents using IoT? Below we will give some Internet of Things examples and applications to clear things up.

1 - Smart home:

Whenever we think of IoT systems, the most important and efficient application that stands out every time is Smart Home ranking as highest IoT application on all channels. The number of people searching for smart homes increases every month with about 60,000 people and increasing **[4]**.

2 - Enterprise asset management:

Enterprise asset management involves measures taken to improve device and machine health to achieve greater output. It is among the prime internet of things examples in the industrial setup. Machines retrofitted with IoT sensors inform users about the machine's current status and whether it needs any maintenance. It allows for more efficient checks for safety and compliance purposes **[4].**

3 - Wearables:

Virtual glasses, fitness bands to monitor for example calorie expenditure and heart beats, or GPS tracking belts, are just some examples of wearable devices that we have been using for some time now. Companies such as Google, Apple, Samsung and others have developed and introduced the Internet of Things and the application thereof into our daily lives.

These are small and energy efficient devices, which are equipped with sensors, with the necessary hardware for measurements and readings, and with software to collect and organize data and information about users **[4].**

4 - Health:

The use of wearables or sensors connected to patients, allows doctors to monitor a patient's condition outside the hospital and in real-time. Through continuously monitoring certain metrics and automatic alerts on their vital signs, the Internet of Things helps to improve the care for patients and the prevention of lethal events in high-risk patients.

Another use is the integration of IoT technology into hospital beds, giving way to smart beds, equipped with special sensors to observe vital signs, blood pressure, oximeter and body temperature, among others **[4].**

5 - Traffic monitoring:

When we use our mobile phones as sensors, **[4]** which collect and share data from our vehicles through applications such as Google Maps, we are using the Internet of Things to inform us and at the same time contribute to traffic monitoring, showing the conditions of the different routes, and feeding and improving the information on the different routes to the same destination, distance, estimated time of arrival.

6 - Water supply:

A sensor, either incorporated or adjusted externally to water meters, connected to the Internet and accompanied by the necessary *software* , helps to collect, process and analyze data, which allows understanding the behavior of consumers, detecting faults in the supply service, report results and offer courses of action to the company that provides the service.

Likewise, it offers final consumers the possibility of tracking their own consumption information, through a web page and in real time, even receiving automatic alerts in case of detecting consumption out of range to their average consumption record, which could indicate the presence of a leak.

7 - Agriculture:

Smart farms are a fact. The quality of soil is crucial to produce good crops, and the Internet of Things offers farmers the possibility to access detailed knowledge and valuable information of their soil condition.

Through the implementation of IoT sensors **[4]**, a significant amount of data can be obtained on the state and stages of the soil. Information such as soil moisture, level of acidity, the presence of certain nutrients, temperature and many other chemical characteristics, helps farmers control irrigation, make water use more efficient, specify the best times to start sowing, and even discover the presence of diseases in plants and soil.

IV - IoT Architecture:

IoT architecture comprises a collection of physical objects, sensors, cloud services, developers, actuators, communication layers, users, business layers, and IoT protocols **[5]**, **[6]**. Because of the wide domain of internet objects, there is no single consensus on IoT architecture, which is universally agreed. Different architectures were proposed by different researchers.

1 - Three-and Five-Layer Architectures:

1-1 - Three-layer architecture:

According to most of the researcher’s opinions, IoT architecture is considered to be three layers.

- Perception layer :

The perception layer is also known as the recognition layer **[6]**. is the physical layer with environmental information sensors. In the environment, some physical parameters are sensed, or other intelligent objects are identified

- Network layer :

The network layer connects to other intelligent devices, network devices, and servers. Its functions are also used for transmitting and processing sensor data.

- Application layer :

The application layer is responsible for providing the user with specific application services. The primary responsibility of this layer is to link the wide gap between users and applications **[7]**. It defines different applications for the IoT, such as smart homes, smart cities, and intelligent health.

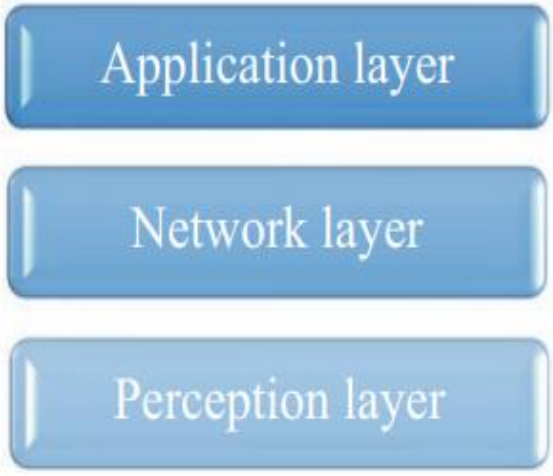

Figure 6:Conventional IoT architecture **[8]**

1-2 - Five-layer architecture:

The three-layer architecture defines the main idea of IoT, but it is not sufficient for IoT research, because research often focuses on the finer aspects of IoT. Hence, the five-layer architecture is defined. The role of the perception and application layers in this architecture is the same as three-layer architecture **[7]**. The functions of the other three layers are as follows:

- Transport layer :

The transport layer transfers sensor data from the perception layer to the processing layer and vice versa via networks such as Bluetooth, wireless, 3G,4G,5G, LAN, NFC (Near Field Communications), and RFID (Radio-frequency identification).

- Processing layer :

The processing layer is also referred to as the middleware layer. It can store, analyze, and process large quantities of transportation data **[9]**. Also, it can manage and provide a variety of lower layers of services. It uses many technologies, such as databases, cloud computing, and big data processing modules.

- Business layer :

The entire IoT system is managed by the business layer **[9]**, including applications, business and business models, and user privacy.

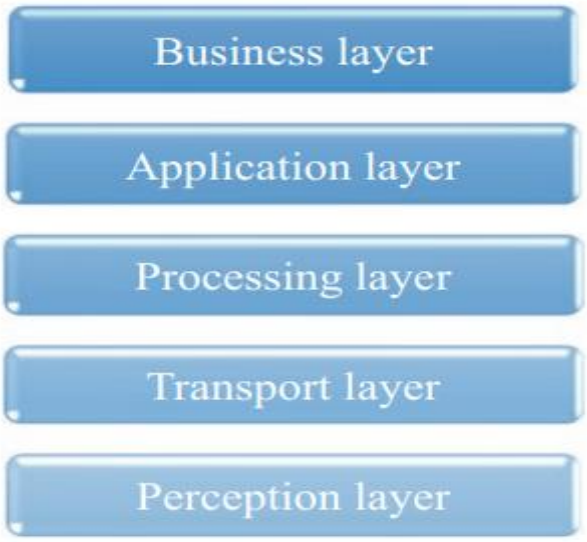

Figure 7:The five-layer IoT architecture **[8]**

V - Communication protocols, standards and regulations:

Communication protocols provide a language that allows devices and users to interact with each other in a more efficient and inter-connected manner. Table 1 enlists some of the popular protocols and their specific features as per the layered architecture of the IoT **[10]**, **[11]**.

Architectural layer	Protocol	Features
Perception and sensor layer (physical and data link layer)	6LOWPAN: Internet Protocol (IP)v6 over low power wireless personal area networks **[12]**	-Allow transmission of IPV6 packets over 802.15.4 links. -can be applied to small and low-power devices having processing constraints.
Network Layer	RPL: routing protocol for low-power and Lossy networks **[13]**	-Routing protocol designed for wireless networks that are susceptible to packet loss and have low power consumption.
Application layer	MQTT: message queuing telemetry protocol **[14]**	-Machine-to-Machine (M2M) communication-based publish subscribe protocol for establishing light -weight connectivity over Transmission. Control protocol (TCP).
	COAP**:**Constrained application protocol **[15]**	-Request response model-based protocol that runs over User Datagram Protocol (UDP) and is developed for resource-constrained environments.

Table 1:layer-Wise IoT protocols

VI - Low-power and lossy network protocols:

The Internet of Things allows billions of physical objects to collect data through surveillance, sensing and control of environments, and to do so distributes them, integrates them into subsets like WSN (wireless sensor network) and connects to the Internet. The primary role of WSN networks is to sensor and collect data through distributed sensors (nodes) and transfer them to network information receiver nodes(gateway). The devices integrated into these networks face some kind of limitations on limited energy, memory and processing resources. This kind of networks called LLN (low-power and lossy networks). The

device size factor is a constraint on these devices. The volume is usually very small and supported by a battery and therefore requires optimal use of resource **[16]**.

In figure 8, we will show a comparison of the technologies used in the Internet of Things in terms of energy consumption for transmitters and receivers, taking into account their data rates **[17]**, **[18]**.

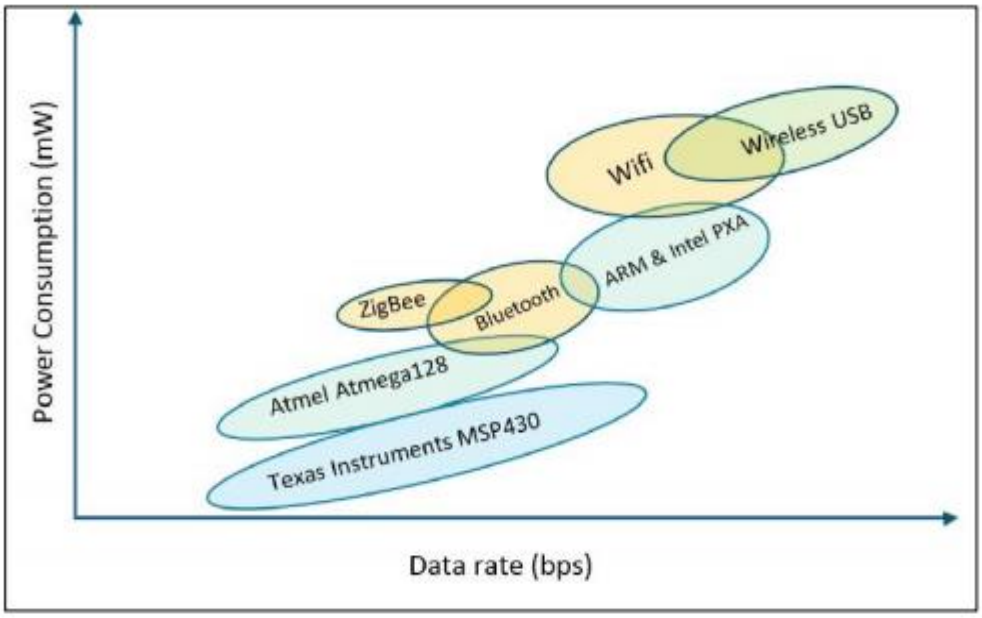

Figure 8:Comparison of wireless power consumption with data rates **[17][18]**

VII - IoT Protocols:

The Standards Authority IETF and IEEE has introduced protocols to meet the requirements of LLN applications as shown in the figure 09.

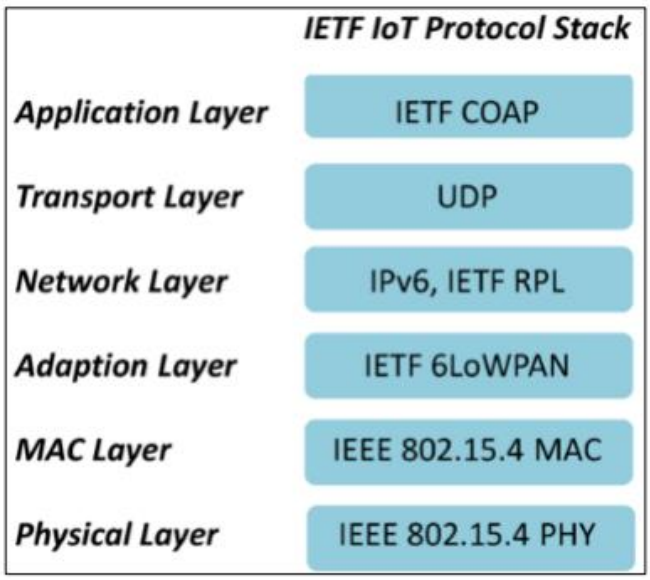

Figure 9:LLN networks protocols**[19]**

1 - IEEE 802.15.4:

The IEEE 802.15 working group defined the physical layer (PHY) and the medium access sub-layer (MAC) for low-complexity, low-power-consumption, low-bit-rate WPAN connectivity. The IEEE 802.15.4 standard, approved in 2003 and amended several times in the following years, contributes to all of these aims, and several compliant products are already available on the market, even if more as development kits than as real end-products.

IEEE 802.15.4 is designed to: 'support three frequency channel bands and use a Direct Sequence Spread Spectrum (DSSS) method. Based on the frequency channels used, the physical layer transmits and receives data on three data rates: 250 kbps at 2.4 GHz, 40 kbps at 915 MHz and 20 kbps at 868 MHz Higher frequencies and wider bands offer high throughput and low latency while lower frequencies offer better sensitivity and cover greater distances. To reduce potential collisions, IEEE 802.15.4 MAC uses CSMA / CA protocol **[20]**.

The devices under this protocol shall be divided into a fully functional device (FFD) and reduced-function device (RFD) .FFD can control and maintain the network, store routing tables and more actions. The main difference between FFD and RFD is when the FFD is a gateway so it must be ON all the time unlike RFD, so it reduces energy consumption very much **[20]**.

2 - 6LOWPAN:

IPv6 over Low power Wireless Personal Area Network (6LoWPAN) is the first and most commonly used standard in this category. It efficiently encapsulates IPv6 long headers in IEEE802.15.4 small packets, which cannot exceed 128 bytes. The specification supports different length addresses, low bandwidth, different topologies including star or mesh, power consumption, low cost, scalable networks, mobility, unreliability and long sleep time. The standard provides header compression to reduce transmission overhead, fragmentation to meet the 128-byte maximum frame length in IEEE802.15.4, and support of multi-hop delivery.

Frames in 6LoWPAN use four types of headers: No 6loWPAN header (00), Dispatch header (01), Mesh header (10) and Fragmentation header (11). In No 6loWPAN header case, any frame that does not follow 6loWPAN specifications is discarded. Dispatch header is used for multicasting and IPv6 header compressions. Mesh headers are used for broadcasting; while Fragmentation headers are used to break long IPv6 header to fit into fragments of maximum 128-byte length **[21].**

3 - MQTT:

Message Queue Telemetry Transport (MQTT) was introduced by IBM in 1999 and standardized by OASIS in 2013 **[22].** It is designed to provide embedded connectivity between applications and middleware's on one side and networks and communications on the other side. It follows a publish/subscribe architecture, where the system consists of three main components: publishers, subscribers, and a broker. From IoT point of view, publishers are basically the lightweight sensors that connect to the broker to send their data and go back to sleep whenever possible. Subscribers are applications that are interested in a certain topic, or sensory data, so they connect to brokers to be informed whenever new data are received. The brokers classify sensory data in topics and send them to subscribers interested in the topics.

4 - Coap:

COAP (constrained application protocol) one to one protocol inspired by HTTP (use request /response paradigm), is based on UDP and used for applications where some constraints we need to fellow (energy, latency). Coap supports four different type of messages:

- Confirmable
- Non – confirmable
- Acknowledgment
- Reset

5 - RPL:

Rpl was created by the roll workgroup and IETF adopted it as a standard guidance program for LLN networks. RPL draws the topology of the network through the DAG scheme, which consists of one or more diagrams called a DODAG, each one is a guidance tree created by root nodes. Unlike known guidance protocols, RPL uses more factors to calculate the best paths, for example measures Guidance

In the following sections We will talk about RPL topology and how control messages work.

5-1 - RPL control messages:

New types of ICMPV6 control messages have been proposed in RPL to build a DODAG.

- DODAG Information Object (DIO):

The DIO message contains network data that allows the node to find an RPL instance, build a DODAG, discover its control factors and select the parent list in the network.

First, the DODAG root broadcasts the DIO message downwards to the neighbors, DIO messages have information about DODAG root ID, RPL Instance ID, rank, objective function, metrics. These messages are sent periodically with a serial number Cumulative to begin the father selection process **[23]**.

- DODAG Information sollicitation (DIS) :

DIS is used by any node to explicitly get DIOs from adjacent nodes, it’s also used for Join the node to the DODAG tree and in case it can’t get DIO messages after a predetermined time **[23]**.

- Destination Advertisement Object (DAO):

The DAO message is used to spread destination information upwards along a tree, a DODAG providing reverse path data to record each node visited across the specified upward path. Each node sends a DAO message except for the DODAG root to spread

prefixes and routing tables for children to parents, after that the full path will be created after passing the DAO messages through the path to the DODAG root **[23].**

- DAO-ACK:

The DAO-ACK message from the DAO receiver is sent as a response to the DAO message received **[23].**

5-2 - DODAG Construction:

The root node begins in the first step with the process of building a DODAG by sending a DIO message to its neighbors Figure 10 (a), which contains many information such as node rank information to allow the node to take its positions in the DODAG and to prevent steering loops. Each node that receives the DIO message must process it and determine whether or not it will join the DODAG according to the use. If the node chooses to join the DODAG, it will have a path up to the root, at which point the node calculates its rank and updates its neighbor table, and selects the better father who will be used to redirect messages to the DODAG root. Each node receiving the DIO message must process it and follow the process until all the nodes are accessed in the network Figure 10 (b).

RPL allows the new nodes to join DODAG at any time, the new node uses the DIS message to request the DIO message from a node located in the DODAG. The new node identifies its best father by receiving the DIO message in accordance with the OF.

The nodes send DIO messages periodically to keep the network stable, when the node is already connected to the DODAG and then receives a new DIO message, which will be processed in three different ways **[24]:**

a) Drop the DIO message according to some rules that define by RPL.
b) Process the DIO message to keep her position in the DODAG
c) Update her position by choosing new parent according to the OF, in this case the node must update parent list to avoid DODAG routing loops

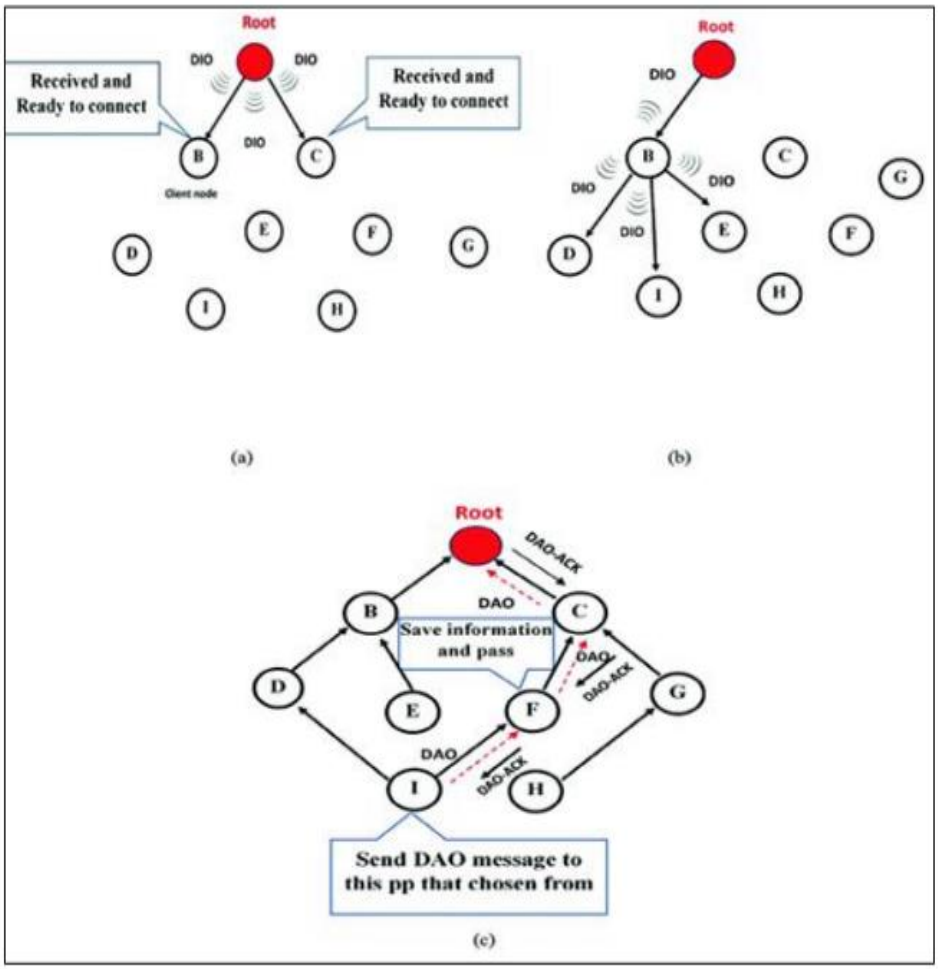

Figure 10:(a) send DIO, (b) update DIO message, (c) send DAO **[24]**

VIII - Vulnerabilities and threats in the Internet of Things:

The level of acceptance of new technologies and services offered by the IoT within the company is strongly linked to the degree of reliability of information and protection of users' private data. Although several projects have been launched with the aim of finding adequate solutions for privacy protection and ensuring rigorous end-user protection, confidentiality, privacy and trust management **[25]**.

1 - Vulnerabilities of internet of things:

The continuous development of the Internet of things (IoT) brings considerable innovations and new use cases for all the people buying connected devices. But at the same time, privacy is more and more put in danger. The revelations about Privacy and Security Threats on the Internet of Things privacy breaches, which are voluntarily done or not, are weekly or almost daily published in the different media. To prevent the privacy leaks, the main threats for the privacy should be analyzed in the context of IoT. This part provides an

overview of the main efforts being carried out to cope with the aforementioned issues. Namely, the section reviews the main security and privacy framework defined in the scope of different initiatives, such as OWASP (Open Web Application Security Project).

The OWASP **[26]** IoT project defines a security framework that gathers information on security issues associated to the IoT development, deployment or technology assessment as mentioned in the Table below:

Name	Description
Insecure web interface	Anyone having access to the web interface if the system is not secured enough could perform attacks such as SQL injection or XSS
Insufficient authentication/authorization	When weak passwords are used or poorly protected. It is prevalent if it is assumed that the interfaces web connections from external networks are not taken into account
Insecure network services	They might be susceptible to buffer overflows or attacks that create denial of service
Lack of transport encryption / integrity verification	Allows data to be viewed as it travels over local networks or the Internet. Often local network is under such risk as it is assumed that it will not be widely visible
Privacy concerns	Lack of proper protection of collected personal data
Insecure cloud interface	Lack of credentials and reset mechanisms or not using SSL for connection to the mobile wireless network
Insufficient security configurability	Lack of granularity in configuration options, especially for user permissions
Insecure software / firmware	They contain hard-coded sensitive data or unprotected network connection for updates of the software/firmware
Poor physical security	USB or other ports can be easily accessed on the device, for instance, to bypass configurations or permissions
Insecure mobile interface	Lack of credentials and reset mechanisms or not using SSL for connection to the mobile wireless network

Table 2: OWASP top 10 vulnerabilities

IX - RPL Routing attacks in IoT:

IoT applications are located in many fields such as smart homes, smart energy monitoring, healthcare systems, smart cities, logistics and etc. Because of this wide range usage, security of IoT is important and routing attacks are a very common threat for IoT **[27]**. RPL is a kind of distance-based protocol. First, each node in the network determines its routing path, then RPL network initializes. In another aspects RPL is a tree-oriented IPv6 routing protocol for 6LoWPAN and it creates Destination Oriented Directed Acyclic Graphs (DODAGs), called as DODAG tree. Each network has one or more DODAG root node as central node and each network has a unique identifier DODAG ID to be identified. Additionally, each node has a rank number and a routing table due to the other nodes' rank numbers. The rank number is used to determine the distance between the node and root **[20].**

RPL attacks can be examined under three categories depending on the vulnerability which they aim to exploit. These categories are **resource-based**, **topology based** and **traffic based**. Resource-based attacks aim to consume energy, power and overload the memory. Topology-based attacks aim to hinder the normal process of the network. This could cause those one or more nodes are broken off from the network. Additionally, these attacks threaten the original topology of the network. Traffic-based attacker nodes aim to join the network as a normal node. Then these attackers use the information of the network traffic to conduct the attack **[28].**

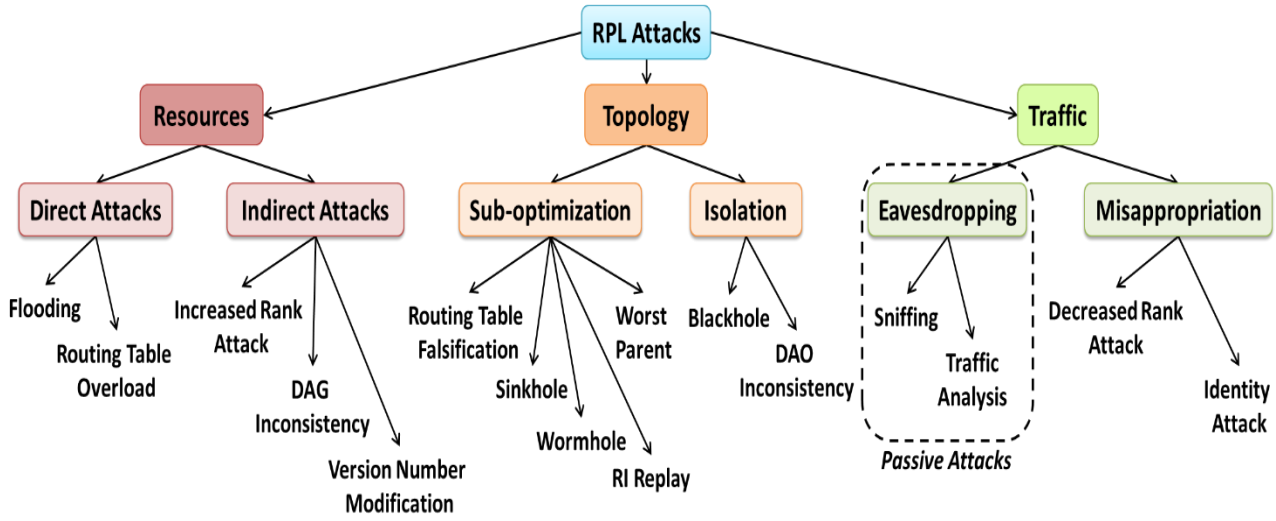

Figure 11:Taxonomy of attacks against RPL networks **[28]**

1 - RPL Routing attacks examples:

1-1 - Sinkhole Attack:

This attack is implemented by creating a malicious node and adding this node to the existing network. The rank of the malicious node is made such that all the network traffic is directed to the malicious node. Usually, it is assigned the next highest value possible after the gateway. The malicious node advertises a different routing path thus attacking many nearby nodes to attract the traffic. So, the malicious node can misuse those packets which were transmitted to them in any way. This attack alone can't harm the system much but when complemented with other attacks, can have adverse effects **[27].**

1-2 - Blackhole Attack:

Blackhole is referred as a place where incoming data is discarded in such a way that source never comes to know that information doesn't reach the destination. In the same way, in RPL protocol a node is made malicious. This malicious node drops the packets which are directed through it. This attack, when combined with the sinkhole, can have more adverse effects **[27]**.

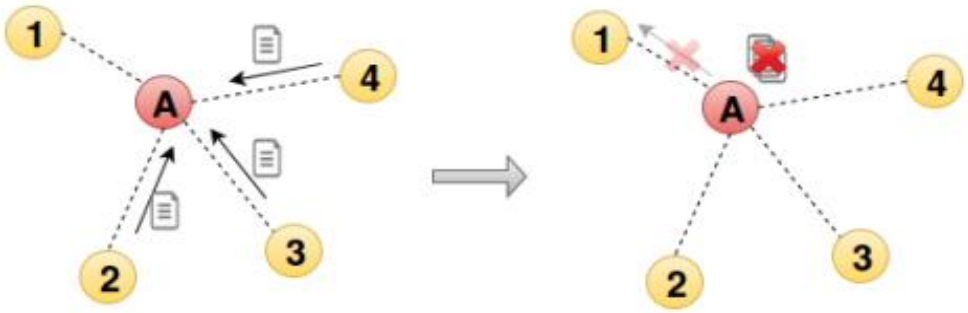

Figure 12:Blackhole Attack (data received from legitimate nodes is dropped)

1-3 - Sybil Attack:

Sybil is also called as the "single node with multiple identities attack. The malicious node can be at multiple places at the same time and it looks like an ordinary node. In other words, the malicious node shows different ID at a different time, due to that other node will take this node as multiple nodes. This attack degrades the performance of the system **[29]**.

1-4 - Hello Flooding Attack:

The HELLO message is the message that a node sends initially before joining the network's DODAG, HELLO message is a DIO message. In Hello Flooding attack, the attacker sends a message with a favorable routing metrics. The malicious node in this attack introduces itself as the neighbor to many nodes. **[27]**.

1-5 - Flooding Attack:

consists of generating a large amount of traffic through DIS messages, causing nodes within range to send DIO messages (used to advertise information about DODAG's to new nodes) and reset their trickle timers (supposed to increase as the network stabilizes). Note that, if secure DIS are used, this attack can still be performed using a compromised node.

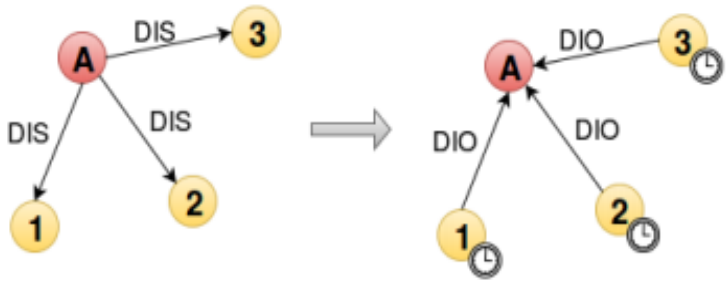

Figure 13:Flooding Attack (A means attacker node)

1-6 - Selective forwarding attack:

The selective forwarding attack is a special case of Blackhole. The malicious node drops some of the packets from the incoming traffic and forwards some of them, according to the hash function applied to it. This attack disrupts the routing paths of the system **[27].**

X - Security requirements in the IoT:

Before evaluating the possible security threats in the IoT paradigm, firstly we should determine the corresponding security requirements. Many studies have investigated and determined the security requirements for IoT **[29]**. Based on them, we define the following security principles.

1 - Confidentiality:

Confidentiality is an important security feature in IoT, but it may not be mandatory in some scenarios where data is presented publicly **[30]**. However, in most situations and

scenarios sensitive data must not be disclosed or read by unauthorized entities. For instance, patient data, private business data, and/or military data as well as security credentials and secret keys, must be hidden from unauthorized entities.

2 - Integrity:

To provide reliable services to IoT users, integrity is a mandatory security property in most cases. Different systems in IoT have various integrity requirements **[31]**. For instance, a remote patient monitoring system will have high integrity checking against random errors due to information sensitivities. Loss or manipulation of data may occur due to communication, potentially causing loss of human lives **[32]**.

3 - Availability:

A user of a device (or the device itself) must be capable of accessing services anytime, whenever needed. Different hardware and software components in IoT devices must be robust so as to provide services even in the presence of malicious entities or adverse situations. Various systems have different availability requirements. For instance, fire monitoring or healthcare monitoring systems would likely have higher availability requirements than roadside pollution sensors.

4 - Authentication and authorization:

Ubiquitous connectivity of the IoT aggravates the problem of authentication because of the nature of IoT environments, where possible communication would take place between device to device (M2M), human to device, and/or human to human. Different authentication requirements necessitate different solutions in different systems. Some solutions must be strong, for example authentication of bank cards or bank systems. On the other hand, most will have to be international, while others have to be local **[32].** The authorization property allows only authorized entities (any authenticated entity) to perform certain operations in the network.

XI - Security Challenges in the IoT:

The security in IoT is characterized by high priority research interest since it is an evolution of the traditional, unsecured Internet model where the communications in the digital world meet the physical world. In particular, the security mechanisms in the IoT have to address the traditional networking attacks and at the same time, they have to offer secure communications for both type of interactions: human-to-machine and machine-to-machine. In

order to fulfill the aforementioned security requirements and specify appropriate countermeasures, the following challenges have to be addressed **[33]**.

1 - Interoperability:

The development and the use of security mechanisms in the IoT should not largely limit the functional capabilities of the IoT devices **[33].**

2 - Resource constraints:

The devices in the IoT are characterized by constrained resources in memory and computation; therefore, they may not support the expensive operations of the conventional security measures, such as the asymmetric encryption **[33]**.

3 - Resilience to physical attacks and natural disasters:

The IoT devices are typically small with limited or no physical protection. For instance, a mobile or a sensor device could be stolen, and the fixed devices could be moved or destroyed by natural disasters.

4 - Autonomic control:

The traditional information systems require the users to configure them. However, the IoT devices have to establish their settings autonomously **[33]**.

5 - Scalability:

The IoT networks usually involve an enormous number of objects. Therefore, the security and privacy protection mechanisms should be able to scale **[33]**.

6 - Information volume:

Many IoT applications such as the smart grid and smart city process a huge volume of sensitive and personal information, which is a potential target of an ever-increasing number of security threats **[33]**.

XII - Conclusion:

The IoT is vulnerable to many attacks and threats, this chapter contains two parts, in the first part we talked about the Internet of things, its applications, and its protocols, especially the routing protocol RPL, and in the second part we talked about some concepts in the security of the Internet of things and their problems. In the next chapter we are going to talk about intrusion detection systems.

Chapter 02: Intrusion detection systems

Chapter 2: Intrusion Detection systems

Introduction:

With the development and widespread use of technology and its reliance on sending and dealing with various types of data, new weaknesses and threats have emerged. Threats and attacks are becoming more frequent and every day we see a new attack and that makes things difficult, so security becomes more and more interested. therefore, there must be a mechanism to monitor and control these activities. in this chapter we present an interested mechanism called IDSs.

I - Intrusion detection systems (IDSs):

Monitoring and analyzing user information, networks, and services through passive traffic collection and analysis are useful tools for managing networks and discovering security vulnerabilities in a timely manner **[34], [35]**. An IDS is a tool for monitoring traffic data to identify and protect against intrusions that threaten the confidentiality, integrity, and availability of an information system **[36]**.

II - IDS functionalities:

The IDS consist of four main functions namely, data collection, feature selection, analysis and action.

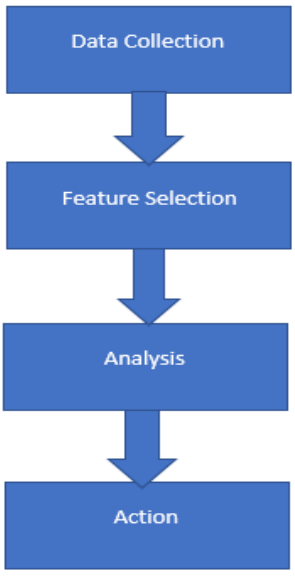

Figure 14:IDS Functionality **[37]**

1 - Data Collection:

This module passes the data as input to IDS. The data is recorded into a file and then analyzed. Network based IDS collects and alters the data packets and in host-based IDS collects details like usage of the disk and processes of system.

2 - Feature Selection:

To select the particular feature large data is available in the network and they are usually evaluated for intrusion **[37]**. For example, the Internet Protocol (IP) address of the source and destination system, protocol type, header length and size could be taken as a key for intrusion selection.

3 - Analysis:

The data is analyzed to find the correctness. Rule based IDS analyses the data where the incoming traffic is checked against predefined signature or pattern **[37]**. Another method is anomaly-based IDS where the system behavior is studied and mathematical models are employed to it.

4 - Action:

It defines about the reaction and attack of the system. It can either inform that the system administrator with all the required data through an email/alarm icons or it can play an active part in the system by dropping packets so that it does not enter the system or close the ports **[37]**.

III - IDS Architecture:

Basic architecture of an intrusion detection system is made up of three modules:

- Sensors: the sensors gather data on the development of the condition of the framework and gives an arrangement of occasions that mirrors this advancement.
- Analyzer: the analyzer figures out which part, fitting to an example of occasions given by the sensor, is attributes of a malevolent action.
- Manager: the manager gathers the beneficiary alarm from the sensor and presents them to the administrator for further activities **[38]**.

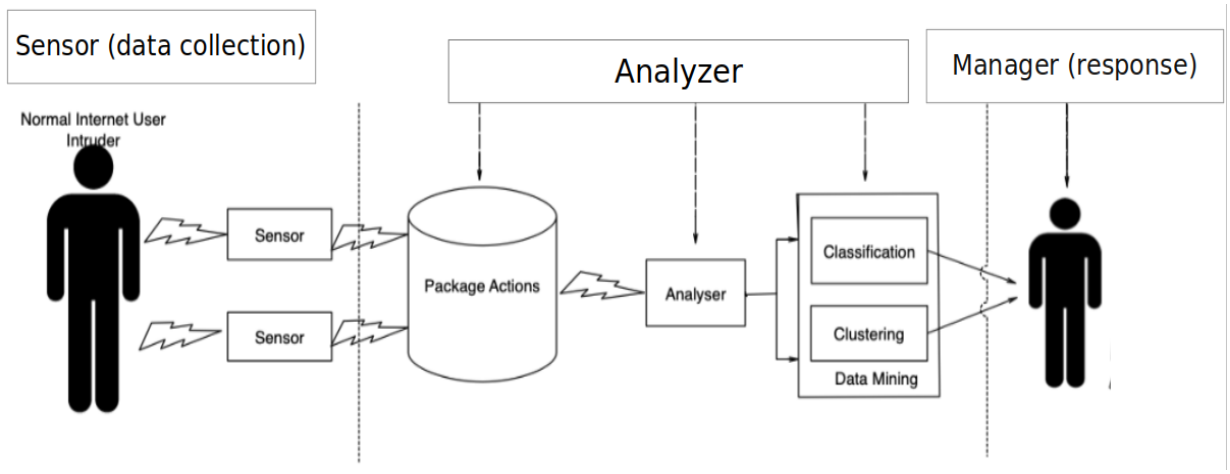

Figure 15:IDS Architecture **[38]**

IV - Taxonomy of Intrusion Detection Systems (IDSs):

There is different Several classifications of intrusion detection, but there is still no universally accepted taxonomy. In this chapter, we present a taxonomy that is based on the synthesis of a number of existing ones **[39]**, **[40]**. We use five criteria to classify IDSs, as summarized in Figure 16.

1 - Information (data) source:

This distinguishes IDSs based on the system that is monitored, i.e., source of input information. The source information can be:

- Audit trails (e.g., system logs) on a host.
- Network connections/packets.
- Application logs.
- Wireless network Traffic
- Intrusion-detection and/or sensor alerts produced by other intrusion-detection systems.

2 - The analysis strategy:

The analysis strategy describes the characteristics of the detector When the IDS looks for events or sets of events that match a predefined pattern of a known attack, this analysis strategy is called misuse detection. When the IDS identifies intrusions as unusual behavior that differs from the normal behavior of the monitored system, this analysis strategy is called anomaly detection.

3 - Time aspects:

Time aspect are used to categorize the IDSs into on-line IDSs that detect intrusions in real time and off-line IDSs that usually first store the monitored data and then analyze it in batch mode for signs of intrusion.

4 - IDSs architectures:

Architecture is used to differentiate between centralized IDSs that analyze the data collected only from a single monitored system and distributed IDSs that collect information from multiple monitored systems in order to investigate global, distributed and coordinated attacks.

5 - Detection response:

Response describes the reaction of the IDS to an attack (intrusion). If the IDS reacts to the attack by taking corrective action (e.g., closing holes) or pro-active action (e.g., logging out possible attackers, closing down services), the response is called active. If the IDS only generates alarms (including paging security analysts) and does not take any actions, the response is called passive **[41]**.

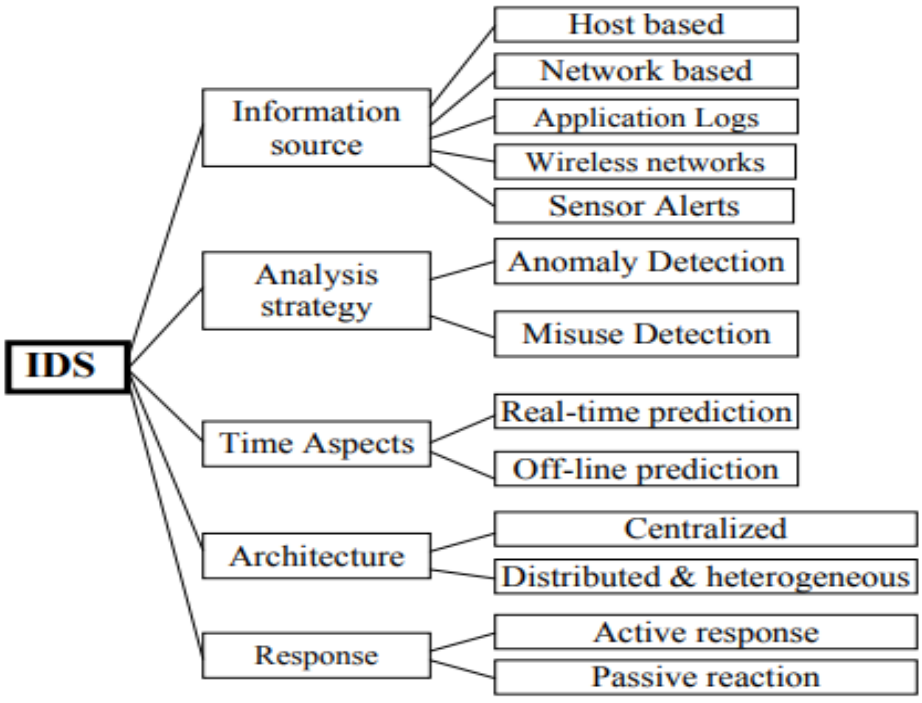

Figure 16:Taxonomy of intrusion detection systems according to proposed five criteria **[41]**

V - IDSs types:

The implementation of an IDS depends on the environment.

1 - A host-based intrusion detection system:

HIDS (host-based intrusion detection system) is designed to be implemented on a single system and to protect that system from intrusions or malicious attacks that will harm its operating system or data **[42]**. A HIDS generally depends on metrics in the host environment, such as the log files in a computer system **[43]**. These metrics or features are used as input to the decision engine of the HIDS. Thus, feature extraction from the host environment serves as the basis for any HIDS.

2 - A network-based intrusion detection system:

NIDS sniffs network traffic packets to detect intrusions and malicious attacks [**43].** A NIDS can be either a software-based system or a hardware-based system. For example, Snort NIDS is a software-based NIDS **[44]**. Network expansion and increasing traffic volumes necessitate the implementation of IDSs as hardware systems, such as a smart sensor architecture **[45]**. For example, field programmable gate arrays (FPGAs) can be used as the basis of a hardware-based NIDS. The special characteristics of FPGAs, such as their ability to support high-speed interfaces, dynamic reprogramming and very high-volume data processing, make FPGAs very suitable for use in NIDSs **[46]**.

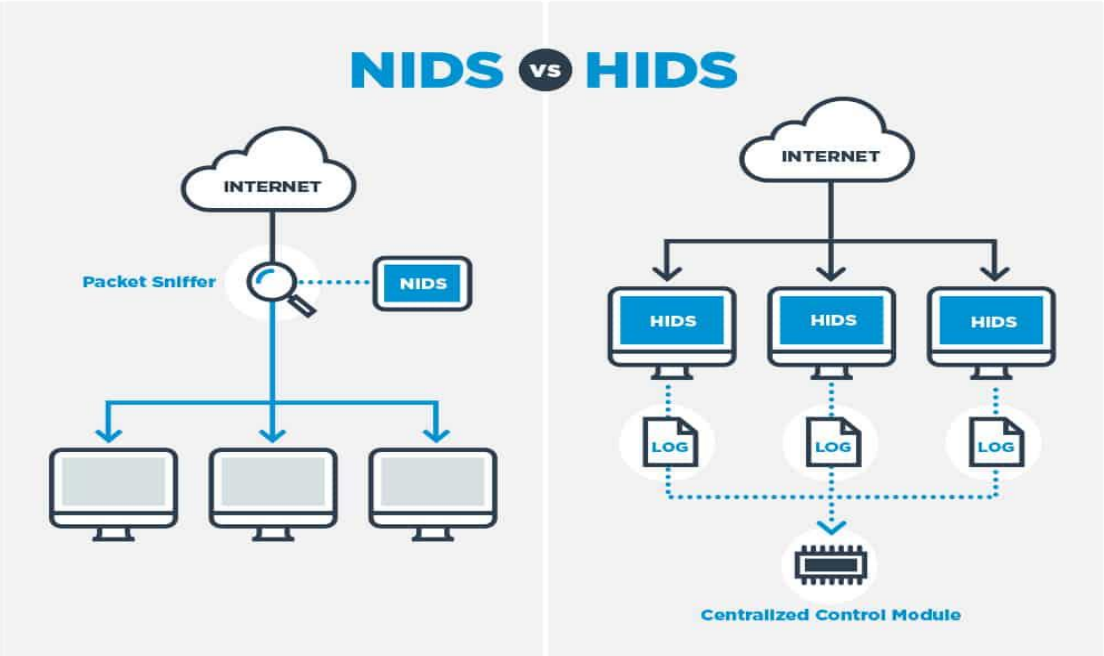

Figure 17:Comparative architecture between NIDS and HIDS

VI - Comparison between types of IDS:

Performance in terms of :	Host Based IDS	Network Based IDS
Intruder deterrence	Strong deterrence for inside intruders	Strong deterrence for outside intruders
Threat response	Week real time response but performs better for a long term attack	Strong response time against outside intruders
Assessing damage	Excellent in determining extent of damage	Very week in determining extent of damage
Intruder prevention	Good at preventing inside intruders	Good at preventing outside intruders
Threat anticipation	Good at trending and detecting suspicious Behavior patterns	Good at trending and detecting suspicious Behavior patterns

Table 3: Comparison between HIDS and NIDS performance

VII - IDSs detection techniques:

Can be misuse detection or anomalies detection.

1 - Misuse-based intrusion detection

A misuse-based intrusion detection technique uses a database of known signatures and patterns of malicious codes and intrusions to detect well-known attacks **[47]**. Network packet overload, the high cost of signature matching, and the large number of false alarms are three disadvantages of misuse-based IDSs **[48]**. In addition, the severe memory constraints in some types of networks, such as WSNs, result in low performance of misuse-based IDSs because of their need to store a large database of attack signatures **[49]**. Moreover, the signature and pattern databases in signature-based IDSs and pattern-matching IDSs need to be continuously updated. Such misuse-based IDSs are designed to detect malicious attacks and intrusions based on previous knowledge.

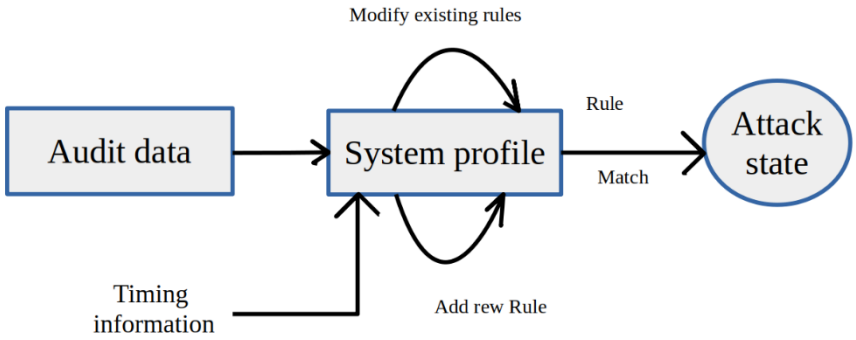

Figure 18:typical misuse detection

1-1 - Advantages:

- Misuse detectors are very effective at detecting attacks without generating an overwhelming number of false alarms
- Misuse detectors can quickly and reliably diagnose the use of a specific attack tool or technique. This can help security managers prioritize corrective measures.
- Misuse detectors can allow system managers, regardless of their level of security expertise, to track security problems on their systems, initiating incident handling procedures **[50]**.

1-2 - Disadvantages:

- Misuse detectors can only detect those attacks they know about, therefore they must be constantly updated with signatures of new attacks.
- Many misuse detectors are designed to use tightly defined signatures that prevent them

prevent them from Detecting variants of common attacks. State-based misuse detectors can overcome this limitation, but are not commonly used in commercial IDSs **[50].**

2 - Anomaly-based intrusion detection:

In an anomaly-based intrusion detection technique, a normal data pattern is created based on data from normal users and then compared against current data patterns in an online manner to detect anomalies **[51]**. Such anomalies arise due to noise or other phenomena that

have some probability of being created by hacking tools. Thus, anomalies are unusual behaviors caused by intruders that leave footprints in the computing environment **[52]**. These footprints are detected in order to identify attacks, particularly unknown attacks. An anomaly-based IDS operates by creating a model of the normal behavior in the computing environment, which is continuously updated, based on data from normal users and using this model to detect any deviation from normal behavior **[53]**.

Anomaly-based IDS algorithms can be used in IoT-based environments depending on the complexity, execution time and detection time requirements

Anomaly IDS techniques (Data mining, Machine learning, Deep learning, Rule model, Payload model, Protocol model, Signal processing model)

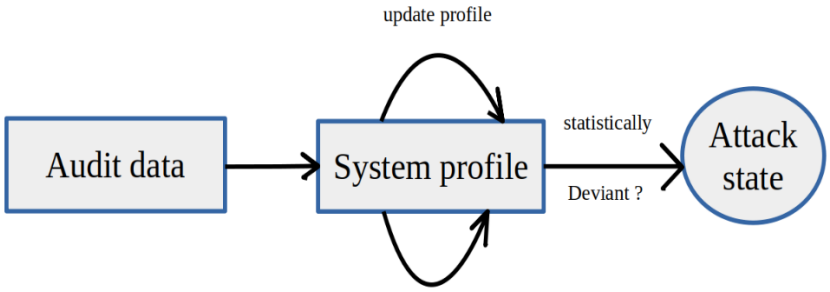

Figure 19:Typical anomaly detection

2-1 - Advantages:

- IDSs based on anomaly detection detect unusual behavior and thus have the ability to detect symptoms of attacks without specific knowledge of details.
- Anomaly detectors can produce information that can in turn be used to define signatures for misuse detectors **[50].**

2-2 - Disadvantages:

- Anomaly detection approaches usually produce a large number of false alarms due to the unpredictable behaviors of users and networks.
- Anomaly detection approaches **[50]** often require extensive "training sets" of system event records in order to characterize normal behavior pattern.

VIII - IDSs: performance evaluation:

There are many classification metrics for IDS, some of which are known by multiple names. Figure 20 shows the confusion matrix for a two-class classifier which can be used for evaluating the performance of an IDS. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. IDS are typically evaluated based on the following standard performance measures.

Actual Class	Predicted Class		
	Class	Normal	Attack
	Normal	True negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True positive (TP)

Figure 20:Confusion Matrix for IDS System

- True Positive Rate (TPR): It is calculated as the ratio between the number of correctly predicted attacks and the total number of attacks. If all intrusions are detected then the TPR is 1 which is extremely rare for an IDS. TPR is also called a Detection Rate (DR) or the Sensitivity. The TPR can be expressed mathematically as

$$TPR = \frac{TP}{TP + FN}$$

- False Positive Rate (FPR): It is calculated as the ratio between the number of normal instances incorrectly classified as an attack and the total number of normal instances:

$$FPR = \frac{FP}{FP + TN}$$

- False Negative Rate (FNR): False negative means when a detector fails to identify an anomaly and classifies it as normal. The FNR can be expressed mathematically as:

$$FNR = \frac{FN}{FN + TP}$$

- Classification rate (CR) or Accuracy: The CR measures how accurate the IDS is in detecting normal or anomalous traffic behavior **[54]**. It is described as the percentage of all those correctly predicted instances to all instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

IX - IDS Challenges:

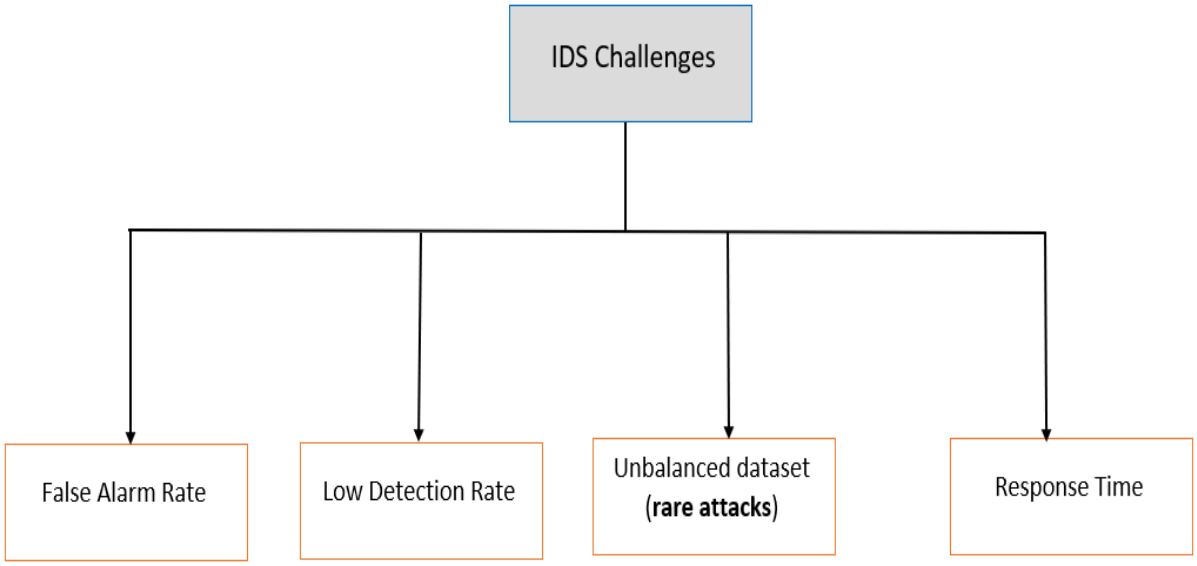

Figure 21:Intrusion detection system (IDS) challenges **[54]**

X - Limitations of Intrusion Detection Systems:

Intrusion detection systems cannot perform the following functions:

- Compensating for weak or missing security mechanisms in the protection infrastructure. Such mechanisms include firewalls, identification and authentication, link encryption, access control mechanisms, and virus detection and eradication.

- Instantaneously detecting, reporting, and responding to an attack, when there is a heavy network or processing load.
- Detecting newly published attacks or variants of existing attacks.
- Effectively responding to attacks launched by sophisticated attackers
- Automatically investigating attacks without human intervention.
- Resisting attacks that are intended to defeat or circumvent them
- Compensating for problems with the fidelity of information sources.
- Dealing effectively with switched networks [50].

XI - Conclusion:

Intrusion detection currently attracts considerable interest in the field of security as we presented, we saw IDS characteristics, architectures, detection type, advantages and disadvantages. Intrusion detection is still an emerging field, and it still exists to improve it due to its importance and necessity.

Chapter 03: Deep learning

Chapter 3: Deep learning

Introduction:

-A type of machine learning, which relies mainly on deep neural networks, to reach the desired value.it is used in all areas of **machine learning (regression, classification etc. ...)**. it needs stronger and faster hardware and massive calculations **[55]**. According to Andrew Ng **Deep learning** is like a rocket. Its fuel is the data it uses. Both the engine and the rocket must be powerful.

-in this **chapter** we are going to take a general cover of deep learning and its algorithms.

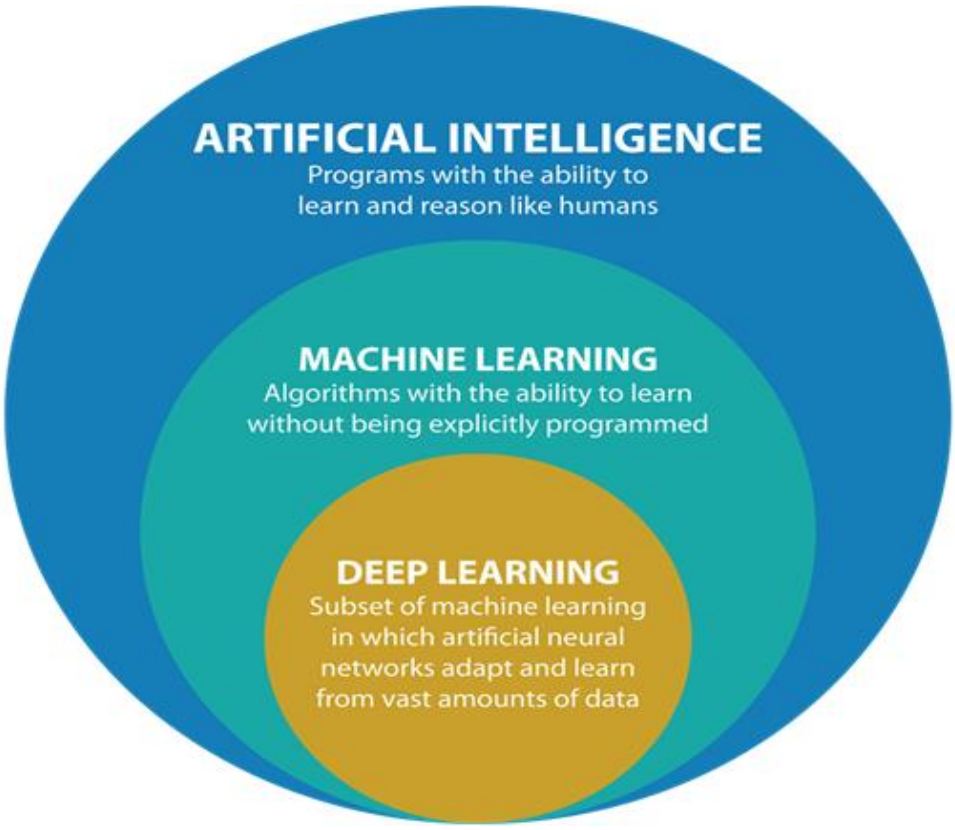

Figure 22:Artificial intelligence areas

I - Difference between machine learning and deep learning:

	Machine learning	Deep learning
operations and time	small	Longer than ML
Type of data set	Structured dataset	Structured and unstructured
Number of samples	A reasonable amount	Huge amount
Dataset size needed	A certain amount of data is sufficient, then it is useless	Deep learning takes advantage of an awful amount of data
Output results	Number or class	Voice, video, Pictures, Texts etc. …

Table 4: comparison between DL and ML

II - Deep learning applications:

- Computer vision
- Natural language processing
- Diagnostics of diseases in healthcare
- Speech recognition
- Robots and self-driving cars
- Intrusion and malware detection systems.

III - Neural Networks:

Using the idea in which brain cells think and analyze data to design an algorithm similar to it. The idea has nothing to do with human thinking, but rather the method of data analysis. Deep learning is based on **NN (neural networks).**

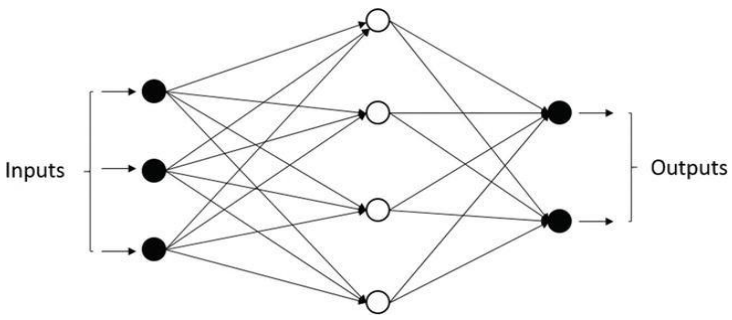

Figure 23:Neural networks design

1 - Neuron:

It was designed to mimic the design of the brain's neuron. The orange circle is the neuron. (X1, X2, X3) are the **inputs**, $\boldsymbol{\theta}$ are the weights and **h(x)** is calculated using sigmoid function or any other activation function.

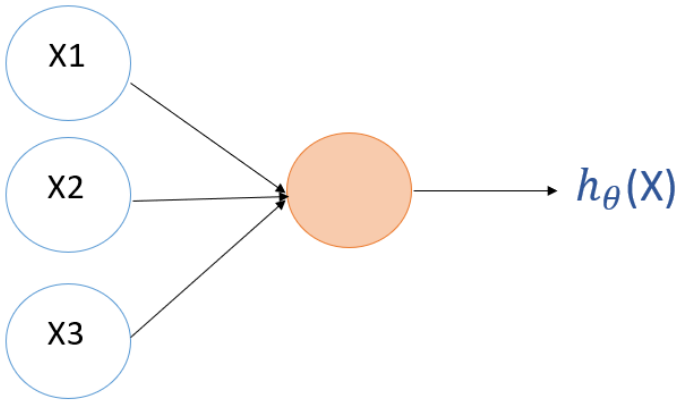

Figure 24:Neuron shape

2 - Mathematical equation for neural networks:

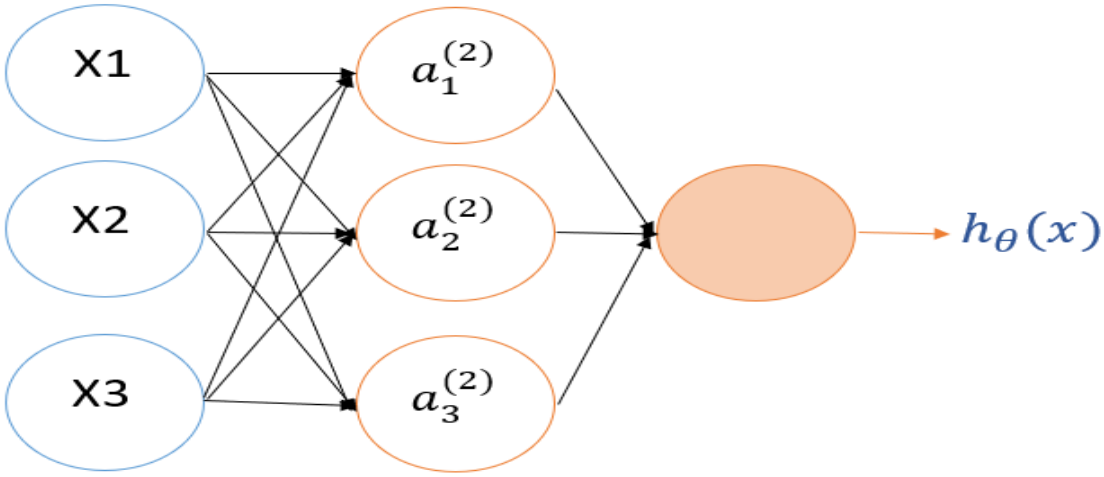

Figure 25: mathematical calculation for neural networks

X1, X2, X3: The inputs

a1, a2, a3: the output values based on the previous layer

Superscript: represent the number of the current hidden layer

Under script: represent the neuron number

θ : the weights

H(x): the final result (predicted value)

$$a_1^2 = \theta_{1\,1}X_1 + \theta_{2\,1}X_2 + \theta_{3\,1}X_3$$

$$a_2^2 = \theta_{1\,2}X_1 + \theta_{2\,2}X_2 + \theta_{3\,2}X_3$$

$$a_1^3 = \theta_{1\,3}X_1 + \theta_{2\,3}X_2 + \theta_{3\,3}X_3$$

Figure 26:Mathematical equations simple example

These equations are repeated with the remaining neurons of the hidden layers until the expected value is reached.

3 - Cost-function equation:

It measures how close the expected value is to the real value **[55].**

$$\boldsymbol{j(\theta) = -\frac{1}{m}\sum_{1}^{m}\ \sum_{k=1}^{k}[y_k^{(i)}\,\log(\,(h_\theta(x^{(i)}))_k) + (1 - y_k^{(i)})\,\log\,(1 - (h_\theta\,(x^{(i)}))_k)] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{sl}\sum_{j=1}^{sl+1}\left(\theta_{j,i}^{(l)}\right)^2}$$

M: represent number of samples

K: is the number of outputs

L: is the number of layers

- We can summarize it as follows:

$$\boldsymbol{cos(t) = y^t\ \log\left(h_\theta\left(x^{(t)}\right)\right) + (1 - y^t)\,\log\,(1 - h_\theta(x^{(t)}))}$$

In a general concept:

$$\cos(i) \approx (h_\theta(x^{(i)}) - y^{(i)})^2)$$

Y: is the actual value

H(X): is the predicted value

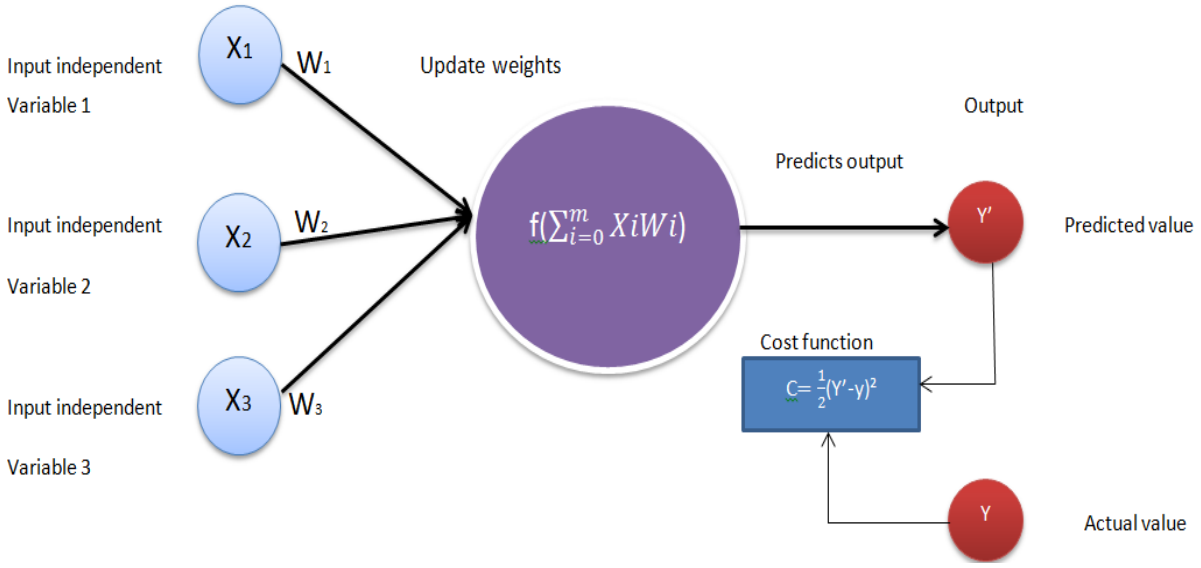

Figure 27:Cost function for each sample

4 - Data path in neural networks:

4-1 - Forward propagation:

It is done according to the values of the outputs through the inputs. This process is carried out through one of the activation equations (Sigmoid, Relu, Tanh, Softmax etc..). Certain values of theta are imposed before the beginning.

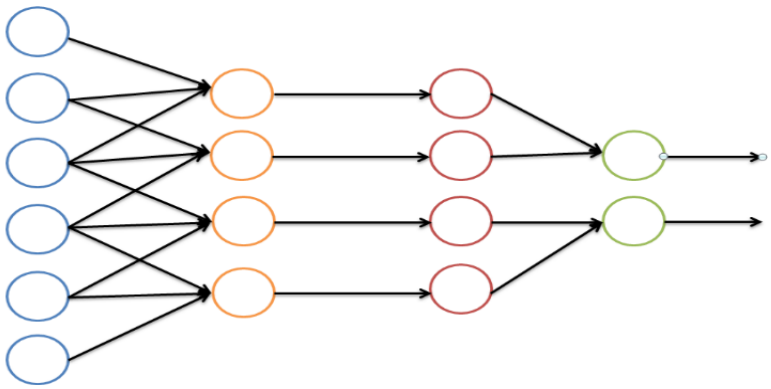

Figure 28:Forward Propagation (predicted values calculation)

5 - Back propagation:

The **θ** values are inversely calculated. This is done by finding the difference between the expected value and the true value, followed by partial differentiation. It is used to modify the assumed **θ** values.

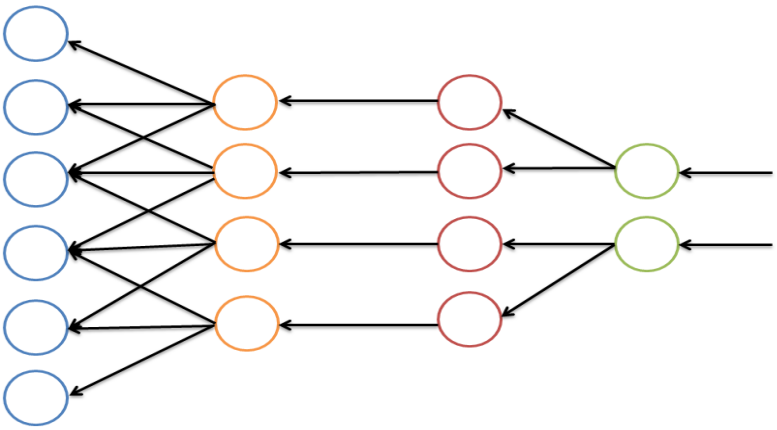

Figure 29:Back propagation (weights update)

6 - Neural networks Types:

6-1 - Feed forward neural networks (FNN):

The sigmoid function is often used inside it. there is a development for it called multi-layer feed forward neural networks (MLFNN) .Often used in voice recognition, computer vision, and self-driving cars.

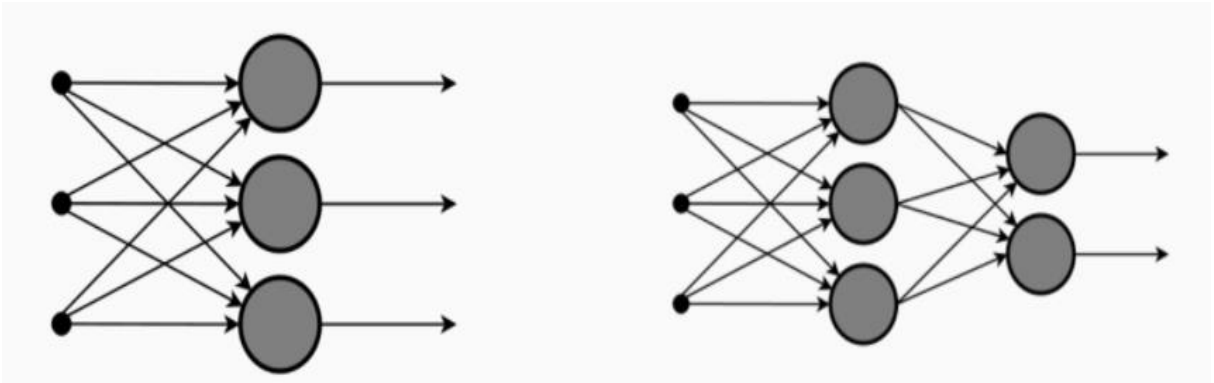

Figure 30:Feed forward neural network

7 - Self-organizing map (SOM):

It mainly deals with unsupervised learning, based on the idea of getting a large amount of data and arranging it specifically during training. It's called also **Kohonen NN**.

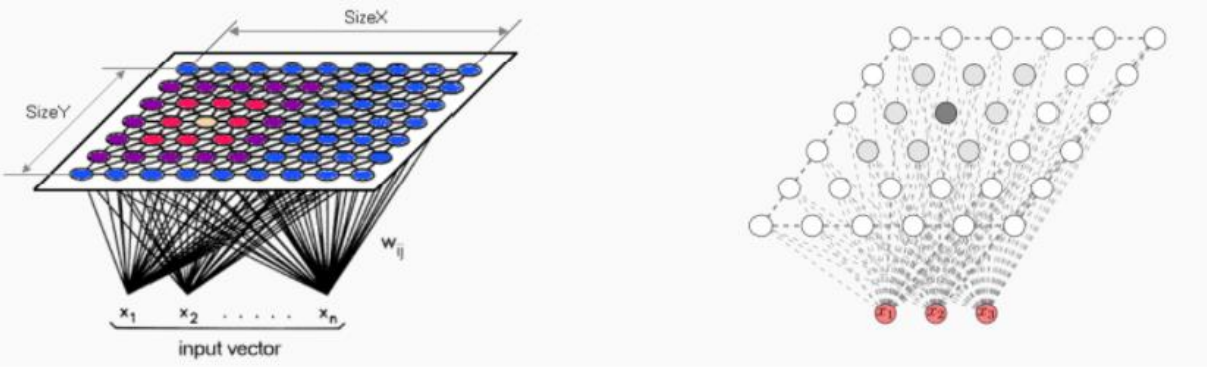

Figure 31:self-organizing map

8 - Convolutional neural networks:

Convolutional neural network (CNN), as shown in Figure 31, is a deep neural network that is composed of multiple layers. The three main types of layers are the following

- Convolutional layer: it applies a set of filters also known as a convolutional kernel, on the input data. Each filter slides over the data to produce a feature map. By stacking all the produced feature maps together, we get the final output of the convolution layer.
- Pooling layer: it operates over the feature maps to perform subsampling, which reduces the dimensionality of the feature maps. Average pooling and max pooling are the most common pooling methods.
- Fully connected layer: It takes the output of the previous layers, and turns them into a single vector that can be an input for the next layer.

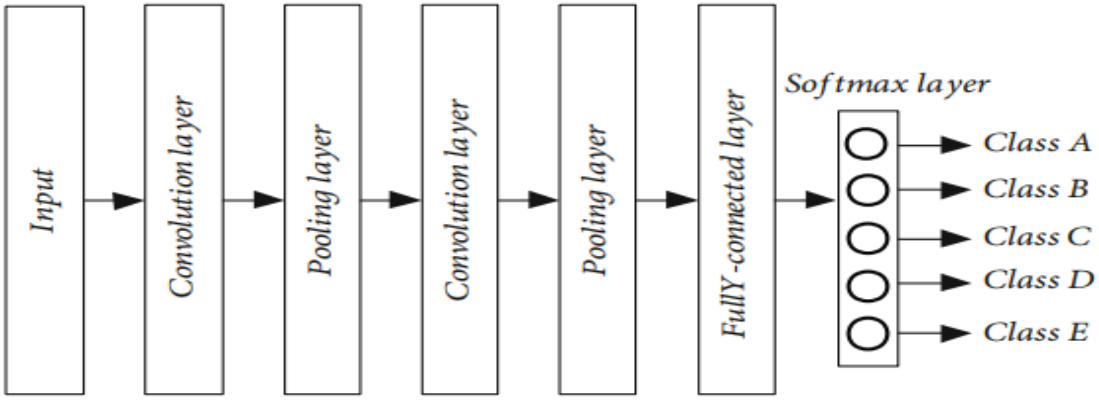

Figure 32:CNN architecture

-CNN used widely in computer vision.

9 - . Deep neural networks (DNN):

A deep neural network is a neural network with a certain level of complexity, a neural network with more than two layers. Deep neural networks use sophisticated mathematical modeling to process data in complex ways.

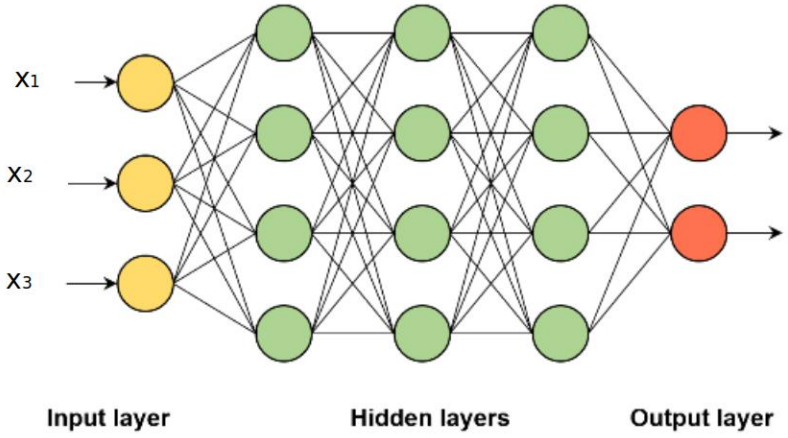

Figure 33:DNN architecture

DNN use the basic concept of neural networks such as forward propagation and back propagation.

$$A = g(Z)$$

$$[Z] = [W]*[X] + b$$

W it is the matrix of weights, **X** is the features matrix, b is the bias and **G** is the activation function (Sigmoid , Relu ..). Finally, **A** is the output and **Z** is the result of weights and features multiplication **[55].**

10 - Recurrent neural networks (RNN):

Its idea depends on feeding the input or hidden layers with the output. Used to complete writing and translation and convert audio to texts.

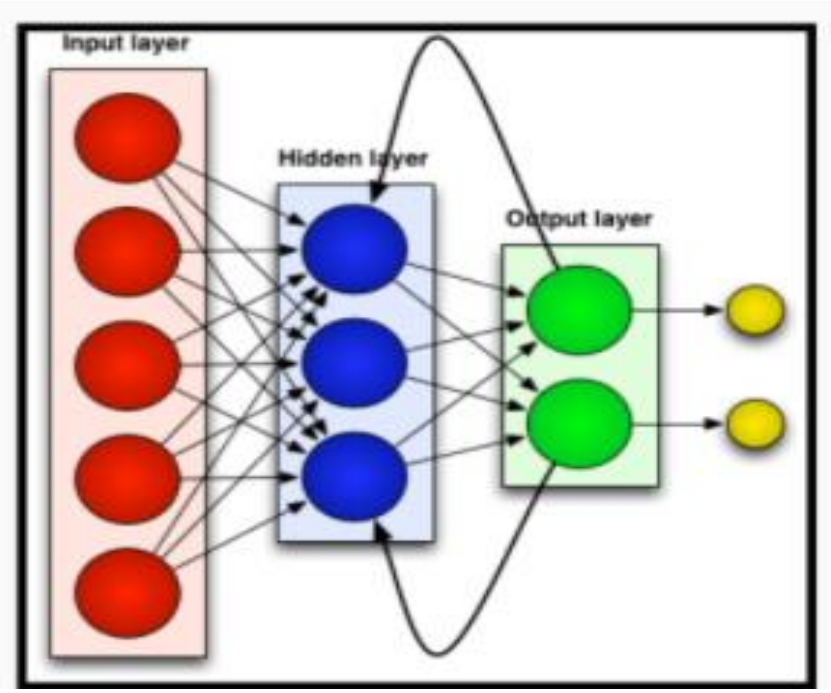

Figure 34:Recurrent neural networks

-some other deep learning algorithms are represented in figure 35:

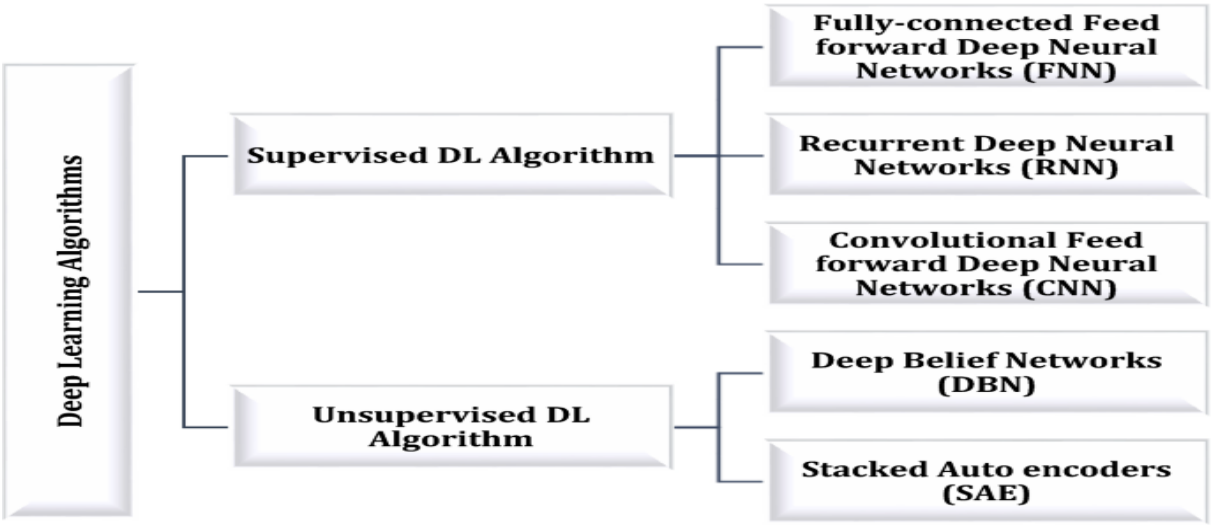

Figure 35:Some Deep learning algorithms

IV - Principles for deep learning IDS in IoT:

The objective of deep learning-based IDS solutions for IoT is to generate models that perform well in terms of effectiveness and efficiency. However, each model adopts some design choices that might limit its ability in achieving this objective. For example, some deep learning IDSs in IoT do not consider the overfitting problem, or apply their model on an unbalanced dataset, or neglect employing feature engineering, which negatively affects their performance in terms of accuracy, memory consumption, and computational time. Also, some IDSs do not try to optimize their learning model, and some are evaluated on outdated or irrelevant datasets, which do not reflect real-world IoT network traffic. Motivated by the above observations, the deep learning-based IDS solution for IoT should advocate the following key design principles:

- Handling overfitting: overfitting happens when the model achieves a good fit on the training data, but it does not generalize well on unseen data. In deep learning, overfitting could be avoided by regularization and dropout layers **[56]**.
- Balancing dataset: data imbalance refers to a disproportion distribution of classes within a dataset. If a model is trained under an imbalanced dataset, it will become biased and rare attacks is the bad problem. By balancing the dataset, the effectiveness of the model will be improved.

- Feature engineering: it allows reducing the cost of the deep learning workflow in terms of memory consumption and time. It also allows improving the accuracy of the model by discarding irrelevant features and applying feature transformation to improve the accuracy of the learning model **[56]**.
- Model optimization: the objective of model optimization is to minimize a loss function, which computes the difference between the predicted output and the actual output. This is achieved by iteratively adjusting the weights of the model. By applying an optimization algorithm such as SGD (Stochastic gradient descent) and Adam **[57]**, the effectiveness of the model will be improved
- Testing on IoT dataset: a deep learning-based IDS for IoT should be tested under an IoT dataset to get results that reflect real-world IoT traffic

V - Conclusion:

Deep learning is a powerful tool that perform well in terms of effectiveness and efficiency

In this chapter we discussed about Deep learning algorithms and most neural networks architecture, basic concepts also key design principles for DL (deep learning) IDS in IoT, in the next chapter we are going to discuss about our proposed framework based on DL and its results.

Chapter 04: Contribution in the detection of intrusions in the IOT environment

Chapter 4: Contribution in the detection of intrusions in the IOT environment

Introduction:

In this chapter, we will talk about our model, how it's built, and the results obtained.

The chapter is divided into four main parts, first we will mention some related works, second, we will discuss about our framework and all steps we have passed to build it and then we will describe the dataset used in our contribution and how it is generated. the last part present performance evaluation and comparative study with some related works.

I - Related work:

In this section we give an overview about some researches about detection of routing attacks in the internet of things using IDS.

1. Yavuz, F. Y., Devrim, & Ensar, G. Ü. L **[58] .**This study is proof of concept for the application of deep learning for security in internet of things

    - They proposed a method for detecting routing attacks for IoT based on deep learning.
    - The main problem in this area is the lack of datasets and the quality of data available. their attack datasets are produced by simulation, using the code of a real sensor and the implementation of the Contiki-RPL RPL protocol.
    - They propose a clearly scalable attack detection methodology based on in-depth learning for the detection of IoT routing attacks that are restricted category, hello-flood type and version number modification attacks, with great precision and accuracy.
    - In addition, they built a deep neural network of models formed using IRAD datasets with assessment information: accuracy, precision and recall rates.

2. Mridula Sharma, Haytham Elmiligi, Fayez Gebali et Abhishek Verma **[59]**

    - They introduced a new Framework to simulate RPL attacks using Contiki-Cooja and had simulated four different attacks using this Framework
    - For the implementation of the experiment, they choose to use four different attacks: the "hello flood" attack, the "DODAG Information Solicitation" (DIS) attack, the "increased version" attack and the "reduced rank" attack.
    - They analyse the characteristics extracted from the network traffic packets and propose a new machine learning model. Using several feature reduction techniques, the number of features required for the classification of attacks is reduced from 58 to 21, a 63.7% reduction in processing and communication energy savings.
    - The selected set of features shows increased efficiency in detecting various attacks using three different classifiers, namely Naive Bayes, RandomForest and the and C4.5 .

Their experimental results show that they could achieve 99.33% classification accuracy using the Random-forest classifier

3. Abd Elmalek Said, Aymen Yahyaoui, Faicel Yaakoubi, et Takoua Abdellatif **[60]**

    - In this article, they proposed an "IDS" intrusion detection system for smart hospitals
    - They offer an RPL attack detection system based on anomalies against an IoT network and especially the RPL using support vector machines.
    - The hospitals they are interested in face many challenges such as resiliency of services, interoperability of assets and protection of sensitive information.
    - They run four simulation scenarios:

- First scenario: IoT network without any malicious mote.
- Second scenario: IoT network with 1 malicious mote randomly placed.

- Third scenario: IoT network with two malicious motes randomly placed.
- Fourth scenario: IoT network with 4 malicious motes randomly placed.
    - Selected IDS is centralized and based on anomalies using an SVM machine learning algorithm.
    - To evaluate the accuracy of the proposed IDS, they use energy consumption as a parameter, and collect data for monitoring power per mote in terms of radio energy, radio transmission energy, receive RX radio energy and interfered INT radio energy.

The results obtained show that the efficiency of the approach through detection accuracy will be higher and more accurate when the number of malicious nodes increases.

II - Proposed framework:

- Because of effectiveness of deep learning and its good results in various areas such as cybersecurity, especially when dealing with large datasets, so we applied it to build an intrusion detection system in the Internet of Things using multiple algorithms and we have chosen CNN like a based architecture of our framework.

The CNN 1D (one dimension) deep learning is a special architecture of regular CNN that used input shape as a vector. Each sample in the dataset is represented as a vector which is the number of rows equal to number of features. Figure 36 shown present CNN 1D.

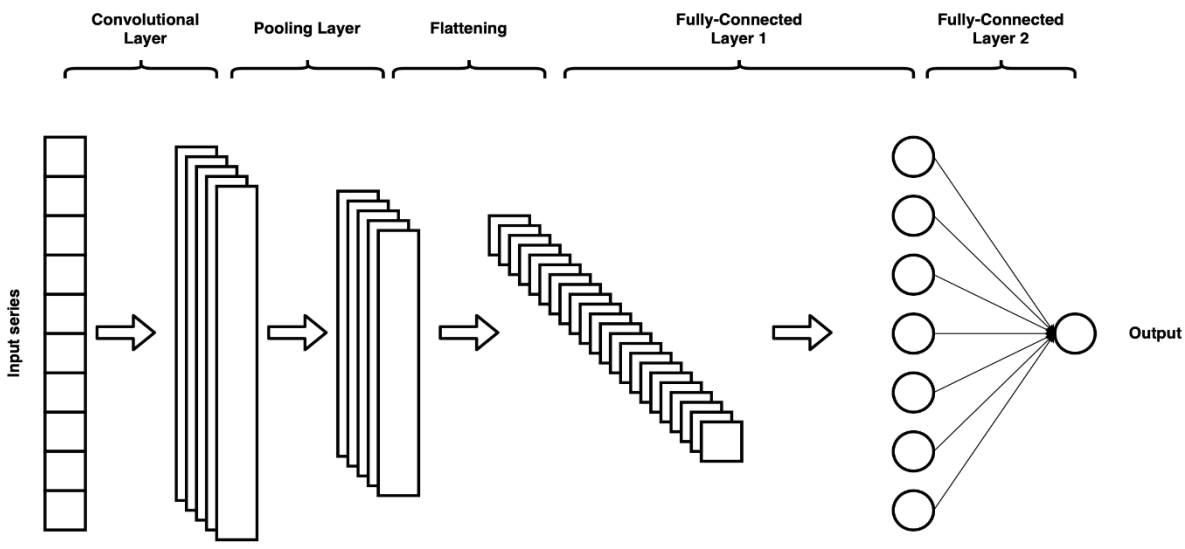

Figure 36:Conv 1D CNN architecture

1 - Overall Architecture:

Figure 39 shows the overall architecture of the proposed CNN framework. The proposed architecture is composed of the following phases:

- Data Balancing: as mentioned above (chapter 3), an imbalanced dataset can produce misleading results. To handle this problem, we use in this phase the oversampling method, which creates synthetic samples of minority classes and is capable of handling mixed dataset of categorical and continuous features.
- Feature engineering: in this phase, we apply feature transformation on the training subset. Min max scaler is applied on the continuous numerical features. In addition, label encoding is applied to categorical features, which simply replaces each categorical column with a specific number. This transformation process is later applied on the validation and the testing subsets.
- Dataset splitting: in this phase, the dataset is split into: training, validation, and testing subsets in order to counter overfitting.
- Training and optimization: in this phase, the CNN model is built. It is trained using the training subset, and its parameters are optimized using Adam optimizer and the validation subset.

- Classification: the generated CNN model is applied on the testing subset to attribute each testing record to its actual class: normal or a specific category of attack

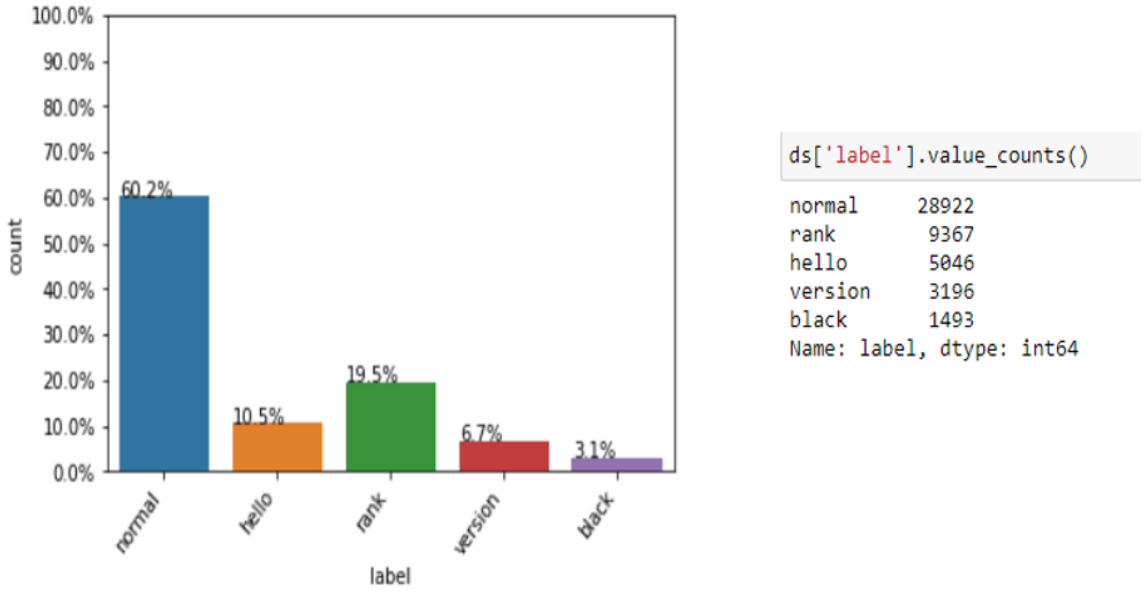

Figure 37: Dataset before balancing

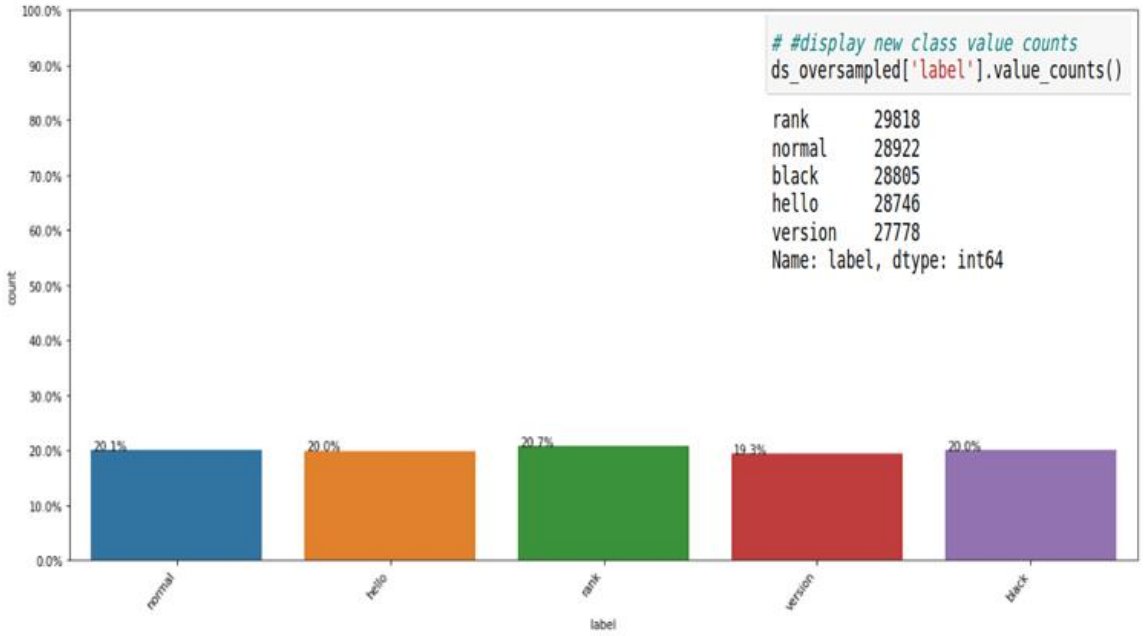

Figure 38:Dataset After balancing

2 - Training and Optimization of CNN Framework:

The training and optimization phase of the proposed CNN is composed of four 1D convolution layers, two dense layers, and a sigmoid layer, which applies sigmoid functions for multiclass classification task. To overcome overfitting, we use tow global maximum pooling, dropout layers and early stopping. We choose Adam optimizer to update weights and optimize sparse categorical crossentropy loss function. Adam optimizer combines the advantages of two stochastic gradient descent algorithms, namely Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp).

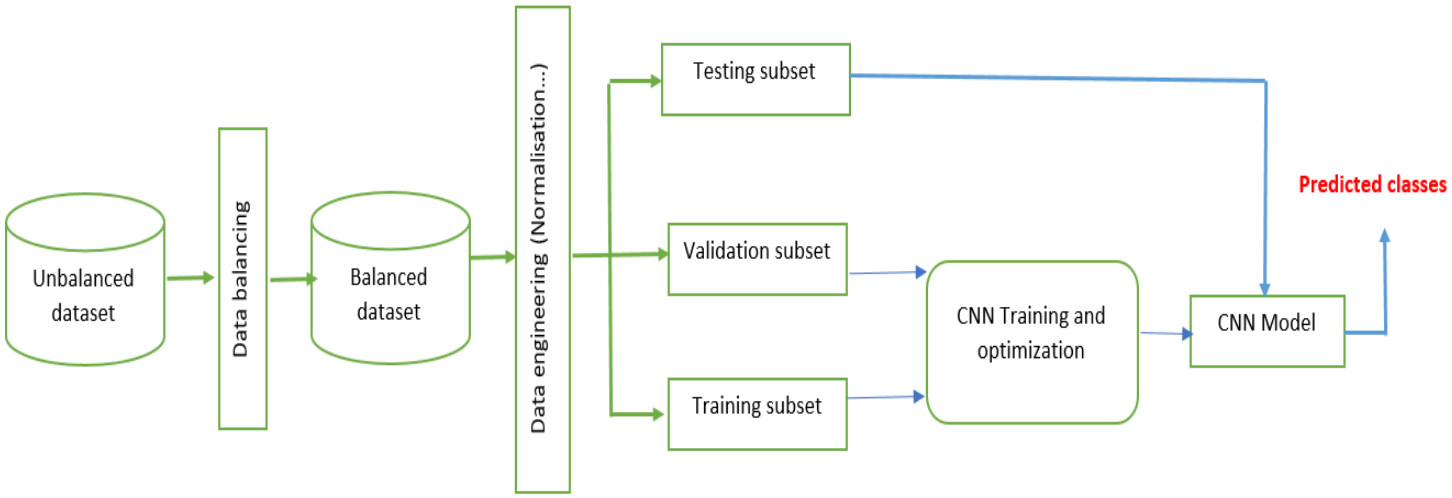

Figure 39:CNN framework

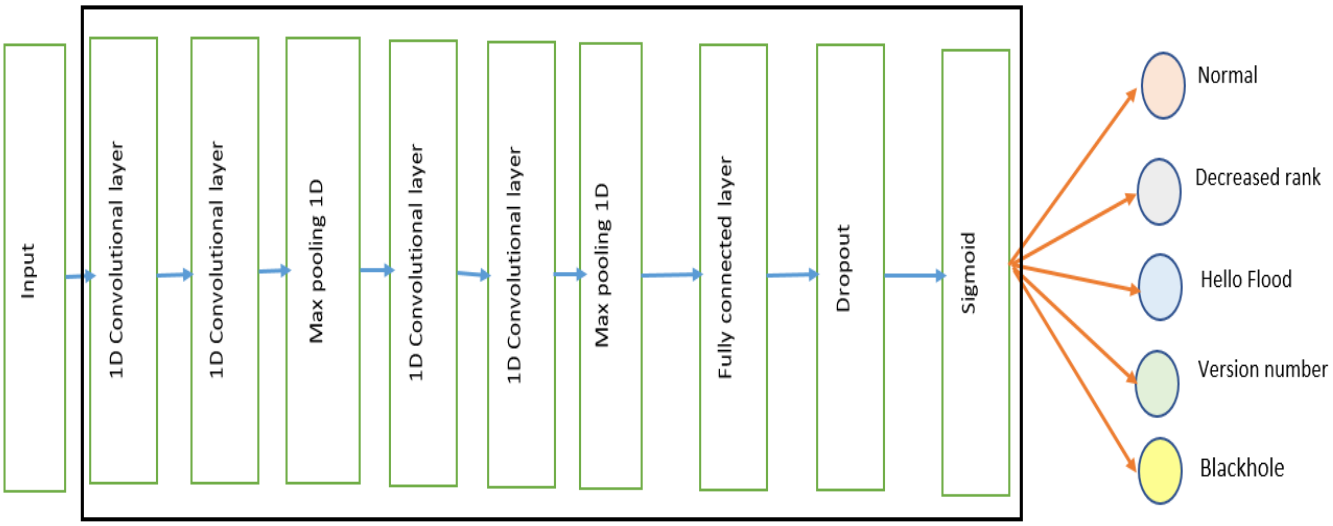

Figure 40:training and optimization of the proposed CNN framework

Specifically, the training and optimization phase of the proposed CNN architecture, as shown in Figure 40, is composed of the following layers:

2-1 - First Layer:

- First 1D convolution layer: it convolves across the input vectors with 64 filters and filter size of 3 with using padding.
- Second 1D convolution layer: it uses 64 filters and a filter size of 3. This second layer before pooling allows the model to learn more complex features
- Max pooling: using pooling size of 2.

2-2 - Second Layer:

- First 1D convolution layer: it convolves across the input vectors with 128 filters and filter size of 3 with using padding.
- Second 1D convolution layer: it uses 128 filters and a filter size of 3. This second layer before pooling allows the model to learn more complex features
- Max pooling: using pooling size of 2.

2-3 - Third Layer:

- Fully connected dense layer: it employs 128 hidden units and a dropout ratio of 50%

2-4 - Fourth Layer:

- Fully connected dense layer with sigmoid activation function: it produces five units that correspond to the five categories of traffic for multiclass classification

III - Rpl routing attacks dataset:

We used an IoT dataset **[61]**, an IoT dataset that was released in 2020 by BOUAZZA Abdelhamid and CHAABI Aissa in the University of Ibn khaldoun – Tiaret. The main problem in this area is the lack of datasets and the quality of data

available. their attack datasets are produced by simulation, using real scenarios by using code of a real sensor and the implementation of the Contiki-RPL protocol.

1 - Dataset generation:

All steps to create the dataset are summarized as follows:

1-1 - Traffic capture:

- They captured all the traffic that went through the IoT network with **different scenarios** as a pcap file by Wireshark with the help of a ready tool in cooja simulator named radio messages. pcap file is converted to csv file.
- All the simulation is divided into a window time of 1000ms, it means in each second, they have captured a number of packets.
- Raw data sets include types of data that cannot be processed by the machine learning algorithm, such as IP addresses make model misleading (overfitting). To avoid this problem source and destination addresses are converted from IPv6 to node ID. For example: IPv6 address 2001:0db8:3c4d:0015:0000:d234::3eee:0011 can be shortened to 11 and broadcast ip address ff02::1a is converted to 99.

1-2 - Generate new features :

All the previous steps generated a total of **13 features** from **6 features** at the beginning

- The transmission and reception time of each packet is calculated. The total time of the duration of each transmission and reception packet in 1000 ms. Then they had calculated the average transmission and reception time for each node, The number of control packets transmitted from each node (concern the control packets: DAO, DIO and DIS) is calculated in windowing size, 1000 ms. Those values had an impact for attack detection like **Hello Flooding** because in this attack **transmission rate** should be higher. The pseudo code of the functionality extraction algorithm is provided below:

Algorithm 1

function

array ← *Dataset.csv*
Sorted array ▷ Sorting by time
Feature conversion
Feature Extraction:
Window Size ← 1000ms
Calculate Feature values within window size
Label the dataset
End of *the Feature Extraction*
End the function.

Figure 41: features extraction algorithm

1-3 - Energy tracking:

They tracked the power of the nodes without attacks, and they found that the attacks consumed the energy of the nodes greatly. Using the simulator, four properties were derived: energy (ON), emission mode (radio TX), reception mode (radio RX) and finally INT (interfered radio).

1-4 - Position and rank tracking:

With changing of position (X, Y) and rang (rank) of nodes They saw that malicious nodes always take on an **important geographical position** and are **close to the root node** to cover and influence as many nodes as possible.

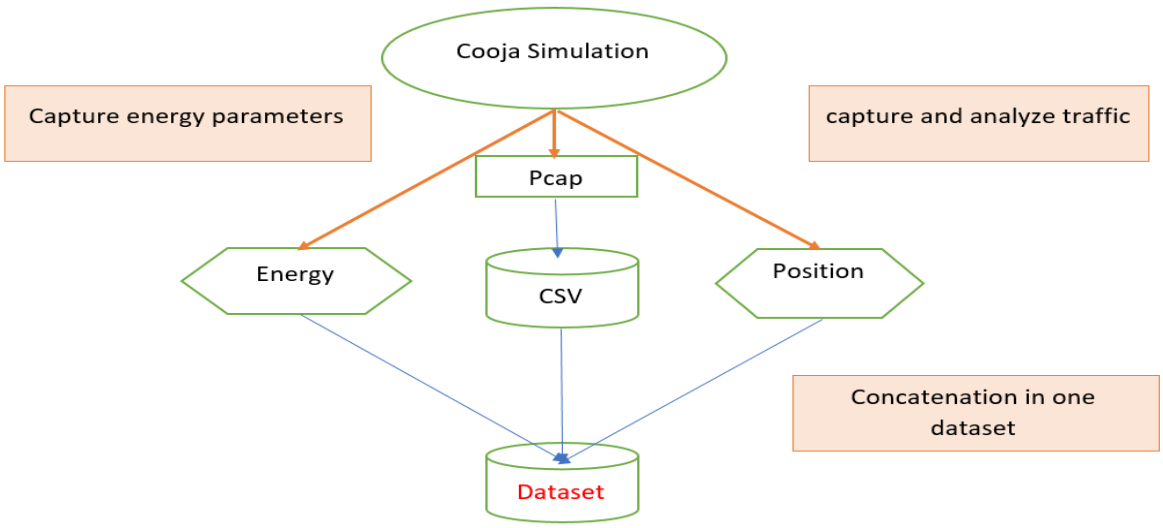

Figure 42: Different steps to build the dataset.

2 - Dataset Description:

This RPL attacks dataset contain **24 features** and **48024 sample**, in the tables below we will describe all details:

N°	Feature name	Description
1	T	Time
2	Src	Source
3	Dst	Destination
4	Protocol	The upper layer protocol decoded
5	Dure_tr	Transmission time during a time window
6	Moy_tr	Transmission media
7	Length_tr	Transsmited Packet size
8	DIS_tr	Trasnssmited DIS number
9	DIO_tr	Trasnssmited DIO number
10	DAO_tr	Trasnssmited DAO number
11	Dure_rec	Reception time during a time window (1s)
12	Moy_rec	Reception media
13	Length_rec	received Packet size
14	DIS_rec	Received DIS number
15	DIO_rec	Received DIO number
16	DAO_rec	Received DAO number
17	ON	Energy
18	TX	Emission energy
19	RX	Reception energy
20	INT	Interfered radio
21	Pos_x	X geographical Position in x axis
22	Pos_y	Y geographical Position in y axis
23	Rang	Node rank in DODAG
24	Class	Attack class

Table 5:Features description

Normal/Attack	Category	Records Numbers
Attack	Decreased Rank	9367
	Version number	3196
	Black Hole	1493
	Hello Flooding	5046
Normal		28922

Table 6:dataset statistics

3 - Data balancing:

In the dataset, there are 28922 normal and 19102 attack samples. We can notice that more than 60% of the samples belong to normal categories, as shown in Table 6. In this way, the learning model will predict the majority classes and fail to spot the minority classes, which means the model is biased. To deal with this problem, different resampling methods have been proposed **[62]** like (1) random oversampling, which randomly replicates the exact samples of the minority classes, and (2) oversampling by creating synthetic samples of minority classes using techniques such as synthetic minority oversampling technique (SMOTE), synthetic minority oversampling technique for nominal and continuous (SMOTE-NC), and adaptive synthetic (ADASYN). In this work, we used the ADASYN technique as it is capable of handling mixed dataset of categorical and continuous features and make us avoided advantages of random oversampling and SMOTE sampling techniques.

3-1 - Adaptive Synthetic Sampling Method for Imbalanced Data (ADASYN):

Adaptive Synthetic (ADASYN) is based on the idea of adaptively generating minority data samples according to their distributions using K nearest neighbor **[63]**. The algorithm adaptively updates the distribution and there are no assumptions made for the underlying distribution of the data. The algorithm uses Euclidean distance for KNN Algorithm. The key difference between ADASYN and SMOTE is that the former uses a density distribution, as a criterion to automatically decide the number of synthetic samples that must be generated for each minority sample by adaptively changing the weights of the different minority samples to

compensate for the skewed distributions. The latter generates the same number of synthetic samples for each original minority sample.

The minority classes such as black hole and hello flood are increased to 28,805 and 28746 samples in the dataset, as shown in Table 7

Before Balancing dataset		**After Balancing dataset**
All Categories	48024	**144069**
Rank	9367	**29818**
Hello	5046	**28746**
Version	3196	**27778**
Black	1493	**28805**
Normal	28922	**28922**

Table 7:original dataset and oversampled dataset

4 - Dataset splitting:

	Training	**Validation**	**Test**
All samples	110213	19449	14407

Table 8:train, validation and test set

IV - Evaluation and Metrics:

In this section we have evaluated the performance of the IDS classifiers, we have focused on Three metrics **Accuracy, precision** and **False alarm rate**.

1 - Comparative study:

We did a comparison of the performance of the algorithms we used. RNN (Recurrent neural network), DNN (Deep neural network), CNN (Convolutional neural network) and by experiments **CNN** was the best algorithm because it gives us best results.

Classifier	Accuracy	Precision	False alarme rate	F1Score	Recall
RNN	0.98112	0.98112	0.0047	0.98112	0.98112
DNN	0,9977	0,99776	0.0005	0.99774	0.99773
CNN	0.9995	0,99951	0,0001	0,99951	0.99951

Table 9:Comparative study between classifiers

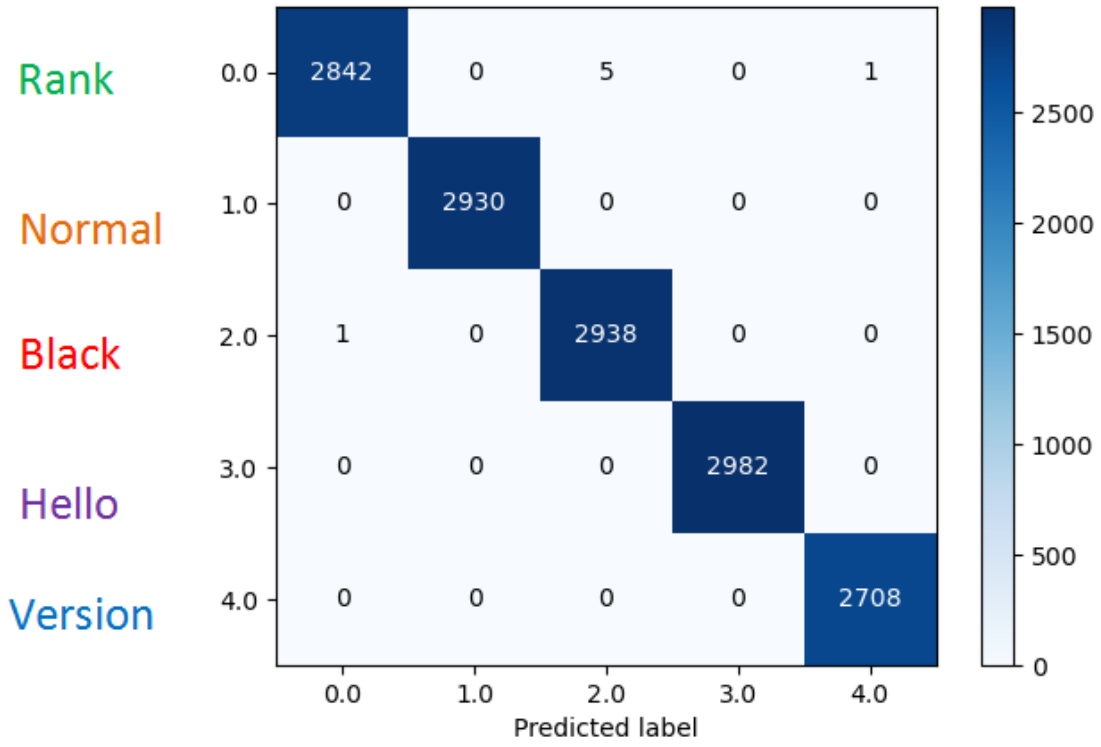

Figure 43:Confusion Matrix with an IOT-dataset

2 - Comparative study with related works:

To Evaluate the performance of our model, we compared its performance with related work **[58]**and **[59].** The result of this comparative study is summarized in the table 10.

	Our model	(Yavuz et al.,2018)	(Sharma et al.,2019)
Attack Types	4	3	4
IoT-dataset	Pcap files, Energy, position	Pcap files	Pcap files
ML/DL	Deep Learning	Deep Learning	Machine Learning
Features number	23	18	21

Table 10:comparison with used dataset in each work

Classifier	Precision	Recall	F1-Score	Accuracy	FAR
(Yavuz et al.,2018)	0.957	0.957	0.957	/	/
(Sharma et al.,2019)	0.994	0.993	/	0.9933	/
Our Model	0.999	0.997	0.997	0.999	0.000

Table 11:comparison with related works performance

3 - Test effectiveness of final Model with NSL-KDD dataset (10%):

We trained our model on the Internet of Things dataset (Minerva), and the result was very satisfactory, and then we wanted to check the efficiency of the model and its hyperparameter with NSL-KDD. More details and results are mentioned in Tables 12 and 13:

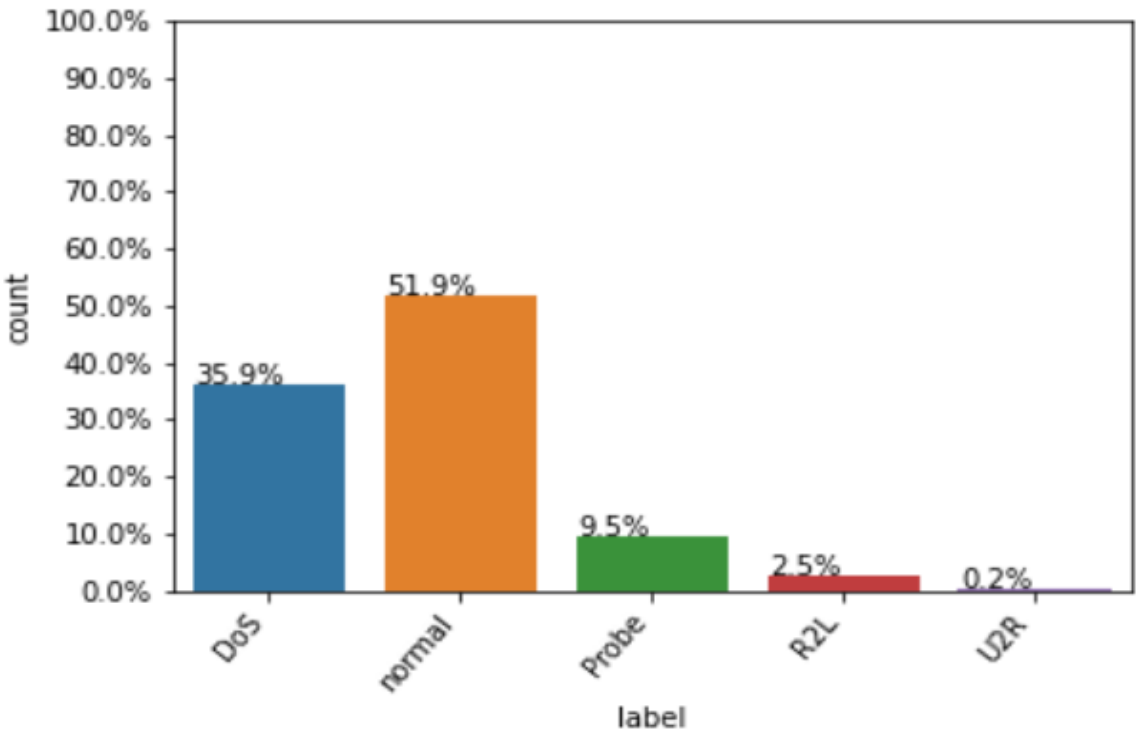

Figure 44: NSL-kdd dataset statistics

We applied the same previous processes as the first dataset **(Normalization, label encoding, data balancing …).**

	Training	**Validation**	**Test**
All samples	294674	52001	38519

Table 12:train, validation and test set after sampling

Classifier	**Precision**	**Accuracy**	**FAR**
Our model	0.9951	0.9951	0.001

Table 13:model performance with NSL-KDD

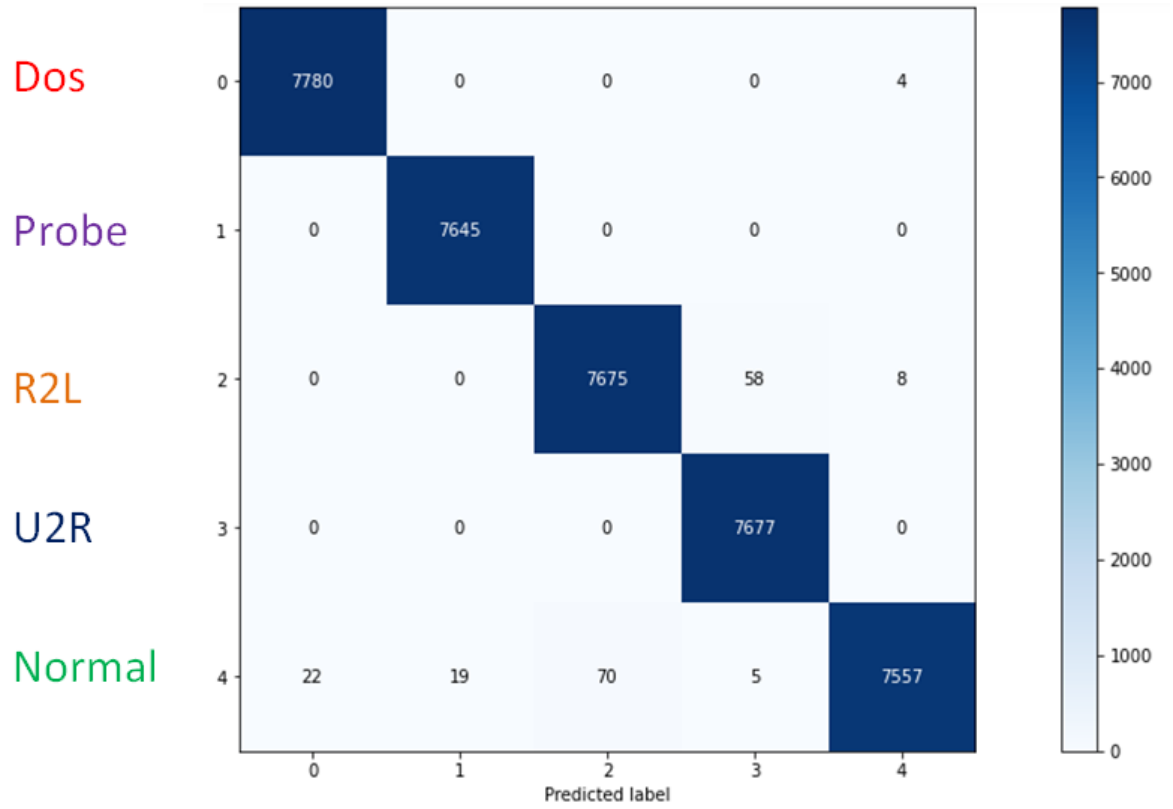

Figure 45:confusion matrix with NSL-KDD dataset

V - Implementation Tools:

In this part we will describe all the tools used in our contribution. To implement the detection learning models, we use Intel core i5-4200M CPU @2.5Ghz*4 processor with 8 GB RAM and 500 GB Hard drive. As for software, we use Python 3.6 programming language, and TensorFlow to build deep learning models.

1 - Programming environment python:

What is python ?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code

reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed **[64].**

2 - Anaconda Integrated development environment (IDE):

What is Anaconda ?

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux**[65]**.

Why Anaconda ?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator **[65].**

3 - Used Libraries:

3-1 - Pandas:

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language **[66].**

3-2 - NumPy :

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more **[67].**

3-3 - Matplotlib:

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

3-4 - Scikit learn :

Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. It's built upon some of the technology you might already be familiar with, like NumPy, pandas, and Matplotlib.

The functionality that scikit-learn provides include:

- Regression, including Linear and Logistic Regression
- Classification**,** including K-Nearest Neighbors
- Clustering**,** including K-Means and K-Means++
- Model selection
- Preprocessing, including Min-Max Normalization

3-5 - Tensorflow :

TensorFlow is an open-source library of software for dataflow and differential programing for various tasks. Similarly, TensorFlow is used in machine learning by neural networks. Developed by Google in 2011 under the name DistBelief, TensorFlow was officially released in 2017 for free. The library is able to run on multiple CPUs and GPUs and is available across multiple platforms, including mobile. The name comes from multidimensional arrays known as tensors, which are commonly used in neural networks**[68]**.

3-6 - Keras :

Keras is an open-source library of neural network components written in Python. Keras is capable of running atop TensorFlow, Theano, PlaidML and others. The library was developed to be modular and user-friendly, however it initially began as part of a research project for the Open-ended Neuro-Electronic Intelligent Operating System or ONEIROS. The principal author of Keras is Francois Chollet, a Google engineer who also wrote XCeption, a deep neural network model. While Keras officially launched, it was not integrated into Google's TensorFlow core library until 2017. Additional support has also been added for Keras integration with Microsoft Cognitive Toolkit.

3-7 - Pycaret :

PyCaret is an open-source, low-code machine learning library in Python that automates machine learning workflows **[69].** It is an end-to-end machine learning and model management tool that speeds up the experiment cycle exponentially and makes you more productive. In comparison with the other open-source machine learning libraries, PyCaret is an alternate low-code library that can be used to replace hundreds of lines of code with few words only. This makes experiments exponentially fast and efficient. PyCaret is essentially a Python wrapper around several machine learning libraries and frameworks such as scikit-learn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt, Ray, and many more.

VI - Conclusion :

In this chapter, we talked about all the steps that we have taken to get to the final model, which is an intrusion detection system for routing attacks in the Internet of Things. The results were excellent with **high accuracy** of 0.9995 and **false alarm rate** with 0.0001.the test experiment was on minerva-iot dataset which is based on recent articles.

General Conclusion

General conclusion:

As the number of devices connected to the Internet of things increases, their security becomes the first obstacle. When we talk about the Internet of Things, it means data everywhere, and it's more dangerous. There's a lot of research in the area of securing these networks, but few of them fit the real environment of the Internet of things and the real scenarios they're exposed to.

In this work, we studied the most important attacks against the routing protocol in the Internet of Things and how it works. The security in IoT is more interested than any other environment because when we talked about IoT we talked about sensitive components and data.

The purpose of this work is to build an intrusion detection system against routing attacks in the Internet of Things based on deep learning algorithms. To train our model, we used a dataset of routing attacks named Minerva **[62]** .This dataset was built with a cooja simulator and it is based **on recent articles** , it contains four main attacks (blackhole , decreased rank , modification version number, hello flood) .it also contain an important features such as node **position** and **energy**.

An effective and efficient Multi -classifier model was then built based on CNN and integrated with ADASYN balancing after going through the most important steps of processing the dataset and using carefully selected **parameters** and **hyperparameters** to achieve a good result. Particularly dependent on **accuracy**, **precision** and **false alarm rate**. The final model has been evaluated and compared with recurrent neural networks (RNN) and deep neural networks (DNN) and other recent related works.

To verify the **strength** of the model and its parameters and hyperparameters we experimented it with **NSL-KDD (10%)** which contains attacks against wired networks, and through data preprocessing and training steps, which also produced an excellent result as shown above that proved our model to be effective.

As part of **future work**, it would be interesting to add another recent attack into the dataset, also **combine** some deep learning algorithms each other to get better results and effective IDSs against most attacks.

Bibliography

[2] "The Four Internet of Things Connectivity Models Explained | InetServicesCloud." http://www.inetservicescloud.com/the-four-internet-of-things-connectivity-models-explained/ (accessed April. 25, 2021).

[3] M. Husamuddin and M. Qayyum, "Internet of Things: A study on security and privacy threats," *2017 2nd Int. Conf. Anti-Cyber Crimes, ICACC 2017*, pp. 93–97, 2017, doi: 10.1109/Anti-Cybercrime.2017.7905270.

[5] D. Serpanos and M. Wolf, "The IoT Landscape," *Internet-of-Things Syst.*, pp. 1–6, 2018, doi: 10.1007/978-3-319-69715-4_1.

[6] K. Rose, S. Eldridge, and L. Chapin, "THE INTERNET OF THINGS: AN OVERVIEW. Understanding the Issues and Challenges of a More Connected World.," *Internet Soc.*, vol. 2, no. October, p. 80, 2015, [Online]. Available: http://electronicdesign.com/communications/internet-things-needs-firewalls-too.

[7] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT Middleware: A Survey on Issues and Enabling Technologies," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–20, 2017, doi: 10.1109/JIOT.2016.2615180.

[8] M. A. Jabraeil Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, *IoT Architecture BT - Towards the Internet of Things: Architectures, Security, and Applications*. 2020.

[9] M. Mukherjee, I. Adhikary, S. Mondal, A. K. Mondal, M. Pundir, and V. Chowdary, "A vision of IoT: Applications, challenges, and opportunities with dehradun perspective," *Adv. Intell. Syst. Comput.*, vol. 479, pp. 553–559, 2017, doi: 10.1007/978-981-10-1708-7_63.

[10] D. P. A. B. B. Gupta, *Handbook of research on cloud computing and big data applications in IoT*. 2019.

[11] A. P. Plageras *et al.*, "Efficient Large-scale Medical Data (eHealth Big Data) Analytics in Internet of Things," 2017, doi: 10.1109/CBI.2017.3.

[12] G. Mtei, J. Mulligan, N. Palmer, P. Kamuzora, M. Ally, and A. Mills, “An Assessment of the Health Financing System in Tanzania: Implications for Equity and Social Health Insurance: Report on Shield Work Package 1,” *Oral Present. Int. Heal. Econ. Assoc. Conf.*, 2007.

[13] N. Accettura, L. A. Grieco, G. Boggia, and P. Camarda, “Performance analysis of the RPL Routing Protocol,” *2011 IEEE Int. Conf. Mechatronics, ICM 2011 - Proc.*, pp. 767–772, 2011, doi: 10.1109/ICMECH.2011.5971218.

[14] U. Hunkeler, H. L. Truong, and A. Stanford-clark, “MQTT-S – A Publish / Subscribe Protocol For Wireless Sensor Networks.”

[15] C. Bormann, A. P. Castellani, and Z. Shelby, “CoAP: An application protocol for billions of tiny internet nodes,” *IEEE Internet Comput.*, vol. 16, no. 2, pp. 62–67, 2012, doi: 10.1109/MIC.2012.29.

[16] Z. Shelby and C. Bormann, *6LoWPAN : The Wireless Embedded Internet Companion Lecture Slides The Book 6LoWPAN : The Wireless Embedded Internet*, no. c. 2009.

[17] H. Fotouhi, “Reliable Mobility Support in Low-Power Wireless Networks,” *Dissertation*, 2015.

[18] L.-O. Varga, “Réseaux de capteurs sans fils multi-sauts à récupération d’énergie : routage et couche liaison de bas rapport cyclique,” *Http://Www.Theses.Fr*, 2015, [Online]. Available: http://www.theses.fr/2015GREAM064.

[19] H. Lin and N. W. Bergmann, “IoT privacy and security challenges for smart home environments,” *Inf.*, vol. 7, no. 3, 2016, doi: 10.3390/info7030044.

[20] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.

[21] T. Salman and R. Jain, “Networking protocols and standards for internet of things,” *Internet Things Data Anal. Handb.*, no. September, pp. 215–238, 2017, doi: 10.1002/9781119173601.ch13.

[22] V. Karagiannis, P. Chatzimisios, F. Vazquez-gallego, and J. Alonso-zarate, “47-94-2-

Pb," vol. 3, no. 1, pp. 9–18, 2015.

[23] E. Aljarrah, M. B. Yassein, and S. Aljawarneh, "Survey and Open Issues," *2016 Int. Conf. Eng. MIS*, pp. 1–6, 2016.

[24] J. V. V. Sobral, J. J. P. C. Rodrigues, R. A. L. Rabêlo, J. Al-Muhtadi, and V. Korotaev, "Routing protocols for low power and lossy networks in internet of things applications," *Sensors (Switzerland)*, vol. 19, no. 9, pp. 1–40, 2019, doi: 10.3390/s19092144.

[25] O. mawloud Ait Mouhoub younes, Bouchebbah fatah, "Proposistion d'un modèle de confiance dans l'iot," Abderrahmanme mira - bejaia, 2015.

[27] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based internet of things," *Int. J. Distrib. Sens. Networks*, vol. 2013, 2013, doi: 10.1155/2013/794326.

[28] A. Mayzaud, R. Badonnel, and I. Chrisment, "A taxonomy of attacks in RPL-based internet of things," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 459–473, 2016.

[29] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," *2015 10th Int. Conf. Internet Technol. Secur. Trans. ICITST 2015*, pp. 336–341, 2016, doi: 10.1109/ICITST.2015.7412116.

[30] J. Lopez, R. Roman, and C. Alcaraz, "Analysis of security threats, requirements, technologies and standards in ireless sensor networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5705 LNCS, pp. 289–338, 2009, doi: 10.1007/978-3-642-03829-7_10.

[31] B. Jung, I. Han, and S. Lee, "Security threats to Internet: A Korean multi-industry investigation," *Inf. Manag.*, vol. 38, no. 8, pp. 487–498, 2001, doi: 10.1016/S0378-7206(01)00071-4.

[32] I. Mashal, O. Alsaryrah, T. Y. Chung, C. Z. Yang, W. H. Kuo, and D. P. Agrawal, "Choices for interaction with things on Internet and underlying issues," *Ad Hoc Networks*, vol. 28, no. January, pp. 68–90, 2015, doi: 10.1016/j.adhoc.2014.12.006.

[33] P. I. Radoglou Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the

Internet of Things: Challenges, threats and solutions," *Internet of Things*, vol. 5, pp. 41–70, 2019, doi: 10.1016/j.iot.2018.11.003.

[34] A. Ismail, H. Mahmud, and H. F., "Reliable Network Traffic Collection for Network Characterization and User Behavior," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 2, 2013, doi: 10.14569/ijacsa.2013.040241.

[35] J. Rubio-Loyola, D. Sala, and I. Ali, "Maximizing packet loss monitoring accuracy for reliable trace collections," *Proc. 2008 16th IEEE Work. Local Metrop. Area Networks, LANMAN 2008*, pp. 61–66, 2008, doi: 10.1109/LANMAN.2008.4675845.

[36] M. T. Ali A. Ghorbani, Wei Lu, *Network Intrusion Detection and Prevention Advances in Information Security*. 2010.

[37] T. Mohit, K. Raj, B. Akash, and K. Jai, "Intrusion Detection System," *Int. J. Technol. Res. Appl.*, no. 2, pp. 38–44, 2017.

[38] N. M. Shanono, N. A. Abu, and W. Mohamed, "Intrusion Detection System Architecture : Issues and Challenges," *Glob. J. Comput. Sci. Technol.*, vol. 62, no. 7, 2020.

[39] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Citeseer, 2000.

[40] R. Bace and P. Mell, "NIST special publication on intrusion detection systems," BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, 2001.

[41] A. Lazarevic, V. Kumar, and J. Srivastava, "Intrusion detection: A survey," in *Managing Cyber Threats*, Springer, 2005, pp. 19–78.

[42] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, 2013.

[43] S. K. Gautam and H. Om, "Computational neural network regression model for Host based Intrusion Detection System," *Perspect. Sci.*, vol. 8, pp. 93–95, 2016.

[45] F. Maciá-Pérez, F. J. Mora-Gimeno, D. Marcos-Jorquera, J. A. Gil-Martínez-Abarca, H. Ramos-Morillo, and I. Lorenzo-Fonseca, "Network intrusion detection system embedded on a smart sensor," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 722–732,

2010.

[46] S. Pontarelli, G. Bianchi, and S. Teofili, “Traffic-aware design of a high-speed FPGA network intrusion detection system,” *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2322–2334, 2012.

[47] W. Bul’ajoul, A. James, and M. Pannu, “Improving network intrusion detection system performance through quality of service configuration and parallel technology,” *J. Comput. Syst. Sci.*, vol. 81, no. 6, pp. 981–999, 2015.

[48] W. Meng, W. Li, and L.-F. Kwok, “EFM: enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism,” *Comput. Secur.*, vol. 43, pp. 189–204, 2014.

[49] A. Abduvaliyev, A.-S. K. Pathan, J. Zhou, R. Roman, and W.-C. Wong, “On the vital areas of intrusion detection systems in wireless sensor networks,” *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.

[50] R. G. Bace and P. Mell, “Intrusion detection systems.” US Department of Commerce, Technology Administration, National Institute of …, 2001.

[51] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network anomaly detection: methods, systems and tools,” *Ieee Commun. Surv. tutorials*, vol. 16, no. 1, pp. 303–336, 2013.

[52] J. Hong, C.-C. Liu, and M. Govindarasu, “Integrated anomaly detection for cyber security of the substations,” *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1643–1653, 2014.

[53] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, “Intrusion detection techniques in cloud environment: A survey,” *J. Netw. Comput. Appl.*, vol. 77, pp. 18–47, 2017.

[54] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[56] A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, “Intrusion Detection System for Internet of Things Based on Temporal Convolution Neural Network and Efficient

Feature Engineering," *Wirel. Commun. Mob. Comput.*, vol. 2020, no. April, 2020, doi: 10.1155/2020/6689134.

[57] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016, [Online]. Available: http://arxiv.org/abs/1609.04747.

[58] F. Y. Yavuz, D. Ünal, and E. Gül, "Deep learning for detection of routing attacks in the internet of things," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 1, pp. 39–58, 2018, doi: 10.2991/ijcis.2018.25905181.

[59] M. Sharma, H. Elmiligi, F. Gebali, and A. Verma, "Simulating Attacks for RPL and Generating Multi-class Dataset for Supervised Machine Learning," *2019 IEEE 10th Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON 2019*, pp. 20–26, 2019, doi: 10.1109/IEMCON.2019.8936142.

[60] A. Yahyaoui, F. Yaakoubi, and T. Abdellatif, "Machine Learning Based Rank Attack Detection for Smart Hospital Infrastructure," in *International Conference on Smart Homes and Health Telematics*, 2020, pp. 28–40.

[61] C. A. BOUAZZA abdelhamid, "Détection des attaques de rout age dans l ' Internet," ibn khaldoun - Tiaret, 2019.

[62] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.

Web Sources

[1] "IoT Connected devices installed base worldwide. | Download Scientific Diagram." https://www.researchgate.net/figure/IoT-Connected-devices-installed-base-worldwide_fig1_342262969 (accessed April.21, 2021).

[4] "The 9 most important applications of the Internet of Things (IoT)." https://www.fracttal.com/en/blog/the-9-most-important-applications-of-the-internet-of-things (accessed April. 25, 2021).

[26] "OWASP Internet of Things Project - OWASP." https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10 (accessed Sep. 12, 2021)

[44] "Snort - Network Intrusion Detection & Prevention System." https://www.snort.org/ (accessed July. 15, 2021).

[55] "Deep Learning | Coursera." https://www.coursera.org/specializations/deep-learning (accessed .june 20, 2021).

[64] "What is Python? Executive Summary | Python.org." https://www.python.org/doc/essays/blurb/ (accessed Sep. 1, 2021).

[65] "Anaconda Navigator — Anaconda documentation." https://docs.anaconda.com/anaconda/navigator/index.html (accessed Sep. 20, 2021).

[66] "pandas - Python Data Analysis Library." https://pandas.pydata.org/ (accessed Sep. 10, 2021).

[67] "What is NumPy? — NumPy v1.21 Manual." https://numpy.org/doc/stable/user/whatisnumpy.html (accessed Sep. 1, 2021).

[68] "TensorFlow." https://www.tensorflow.org/ (accessed Sep. 1, 2021).

[69] "PyCaret — pycaret 2.2.0 documentation." https://pycaret.readthedocs.io/en/latest/ (accessed .Sept 1, 2021).

[63] R. Walimbe, “Handling imbalanced dataset in supervised learning using family of SMOTE algorithm. - Data Science Central,” 2017. https://www.datasciencecentral.com/profiles/blogs/handling-imbalanced-data-sets-in-supervised-learning-using-family (accessed July. 31, 2021).