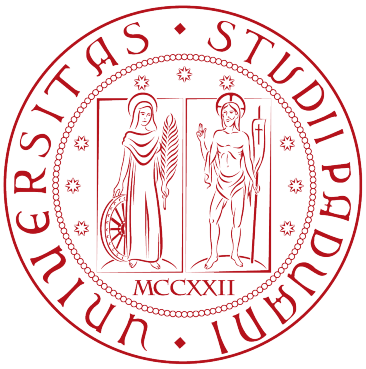

Università degli Studi di Padova

Dipartimento di Diritto pubblico, internazionale e comunitario

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Diritto e Tecnologia

Anomalous traffic detection in IoT systems

Supervisor:
Dott.ssa Sara Baldoni

Candidate:
Liam Ganci
Matricola: 2010308

Academic Year 2023-2024

Sommario

Questa tesi fornisce una panoramica sulla sicurezza dell'Internet of Things (IoT) con un focus principale sulla rilevazione e l'analisi del traffico anomalo e dannoso nei sistemi IoT. Il documento inizia con una revisione completa della tecnologia IoT, discutendo le sue definizioni, caratteristiche e applicazioni. Successivamente, vengono descritti i principi di sicurezza chiave per l'IoT. Questa sezione fornisce anche una panoramica della vasta gamma di attacchi informatici che colpiscono l'IoT. Successivamente vengono esplorate le metodologie di rilevamento del traffico anomalo, analizzando gli attacchi di rete e le misure di difesa.

Abstract

This thesis provides an overview on the cybersecurity of the Internet of Things (IoT) technology with major focus on the detection and analysis of anomalous and malicious traffic in IoT systems. The document begins with a comprehensive review of the IoT technology, discussing its definitions, features and applications. Next, the key security principles for IoT are described. This section also provides a catalogue of the wide range of cyberattacks striking IoT. In the following, anomalous and malicious traffic detection methods are explored. IoT network attacks and defensive measures will be addressed, providing insights into threat landscapes and proactive security strategies. The concluding segment focuses on analysing detection methods and implemented tools to prevent attacks, ensuring IoT security.

Acknowledgements

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of the *Internet of Things* (IoT) has revolutionized numerous sectors, integrating interconnected devices that generate extensive data streams for enhanced automation and real-time decision-making capabilities. This new paradigm has the potential to transform industries such as healthcare, agriculture, manufacturing, and urban management by providing unprecedented levels of efficiency, cost savings, and innovative services. The ability of IoT devices to communicate and interact autonomously opens up a myriad of possibilities for smarter homes, intelligent transportation systems, and more responsive healthcare solutions. However, this rapid expansion and integration of IoT systems also introduces significant security challenges. As the number of connected devices increases, so does the potential for security breaches, data theft, and other cyber threats. These vulnerabilities can compromise sensitive information and disrupt essential services, highlighting the critical need for robust security measures.

First, this Thesis explores the evolution and impact of IoT systems, outlining its significance and transformative influence across various sectors. More specifically, Chapter 2 details the characteristics, structure, applications, and challenges of this innovative technology.

In the following Chapter 3, the Thesis delves into the broader landscape of IoT security challenges, discussing the vulnerabilities found at different layers of IoT architecture, from the perception layer where sensors collect data, to the network layer where data transmission occurs, and finally to the application layer where data is processed and utilized. Understanding these challenges is crucial for developing comprehensive security strategies that address potential weaknesses at every level of the IoT ecosystem.

Chapter 4 delves into the critical aspects of *Intrusion Detection Systems* (IDSs) and anomaly-based detection techniques within IoT networks, which are intrinsically linked to the broader theme of IoT security discussed in Chapter 3. As Chapter 3 outlines the various security chal-

lenges faced by IoT systems at different architectural layers, Chapter 4 builds on this by exploring specific solutions to mitigate these threats. IDSs play a vital role in identifying and responding to potential security breaches, thereby addressing the vulnerabilities discussed earlier. The Chapter categorizes IDSs based on their network positions and detection methodologies, providing a framework for understanding how these systems can be effectively deployed within the IoT ecosystem. It further delves into anomaly-based detection techniques, crucial for detecting irregular patterns that signal potential cyber threats, thereby offering practical solutions to the theoretical challenges highlighted in Chapter 3. By examining *Machine Learning* (ML) and *Deep Learning* (DL) approaches for anomaly detection, the Chapter highlights advanced methods for enhancing IDSs capabilities, directly contributing to the development of robust security measures essential for safeguarding interconnected IoT devices.

The Thesis finally includes a detailed examination of a case study, the Passban IDS [11] in Chapter 5. Passban IDS leverages DL and ML algorithms to monitor network traffic and identify suspicious activities in real-time, thereby providing an additional layer of security to IoT infrastructures. The system's ability to learn from data and improve its detection capabilities makes it a promising solution for protecting IoT devices against evolving cyber-threats. By focusing on this case study, the Thesis provides practical insights into the application of IoT technologies in real-world scenarios, highlighting both their strengths and limitations.

Chapter 2

Introduction to IoT

2.1 The IoT Revolution

A crucial invention that is reshaping the way we live and work is the Internet of Things (IoT), a transformative technology that enables the connection and communication of physical devices over the internet. The IoT concept has been around academic discussions for roughly twenty years now and its definition is still not univocal between researchers.

The term 'Internet of Things' was first introduced in 1999, when Kevin Ashton, computer scientist working at Procter and Gamble, hosted a presentation in which he proposed putting *Radio-Frequency IDentification* (RFID) chips on products to keep track of them through a supply chain [5]. This concept was revolutionary because it marked the beginning of a paradigm shift in how we perceive and interact with technology and the physical world. Kevin Ashton's vision for the Internet of Things aimed to merge the physical world of 'things' with the digital realm of information systems. He saw that communicating objects would pave the way to a new era of efficiency in supply chain management, driving automation processes in industries and reducing costs. The widespread adoption of smart devices is central to this vision.

Predictions for 2024 estimate that there will be at least 17.04 billion connected IoT devices worldwide, with this number expected to nearly double by 2030. This rapid growth of interconnected 'smart' devices, ranging from simple sensors to complex home appliances and smartphones, reflects the far-reaching impact of IoT [16].

The IoT framework allows these diverse devices to communicate and collaborate, unlocking new levels of efficiency and innovation. This seamless connectivity between devices is achieved through *Machine to Machine* (M2M) communication, where devices exchange information with minimal or no human intervention [27].

As technology continues to evolve, IoT promises to redefine how we live and work, connecting the physical and digital worlds in ways that were once unimaginable, demonstrating the transformative effects of this technology on a wide range of industries and everyday life, from smart homes and cities to healthcare, agriculture, manufacturing, and many more innovations that will be further explored in this manuscript.

In summary, the goal of IoT is concrete and ambitious: create a system of things connected anytime, anyplace, with anything and anyone, using any path and any service [24].

2.1.1 Key IoT Characteristics and Features

To fully understand the reasons of IoT being such an extraordinary invention, it is fundamental to explore the key features that define this technology. These features not only set the framework for IoT but also provide insights into the challenges and opportunities it presents [24]. From interconnectivity to flexibility, each of these characteristics plays a critical role in shaping the IoT landscape and driving innovation. The key features of IoT are outlined as follows:" [2, 24]:

- *Interconnectivity:*

  IoT devices can communicate and share information, creating a cohesive ecosystem of interconnected systems. This is the point at which the potential of IoT becomes apparent, with a multitude of devices and tools collaborating to collect, transmit, and process data.

- *Interoperability:*

  The ability of IoT devices and systems to collaborate seamlessly, despite being heterogeneous in terms of hardware platforms and networks. This enables efficient communication and interaction across diverse technologies.

- *Heterogeneity:*

  IoT systems comprise a wide range of devices and platforms, from sensors and actuators to gateways and networks, each with its unique characteristics and requirements.

- *Scalability:*

  IoT infrastructure is designed to scale, allowing for the addition of more devices and sensors without compromising performance. This is crucial for accommodating the growth

of applications and handling increased demands. IoT devices collect, transmit, and process large amounts of data. This requires a robust infrastructure to handle the data load and provide insights.

- *Flexibility:*

  IoT systems must be flexible enough to adapt to changing user needs and evolving environments. This adaptability is crucial for the continuous growth and success of IoT technology.

2.2 The IoT structure

The global community of researchers and practitioners has not reached a consensus on a single architecture for the Internet of Things. Many different architectural models have been proposed, with varying levels of complexity [8].

The most basic architecture proposed has three layers. Each layer has its own role and features in the overall functioning of IoT systems [36]:

- *Perception layer:*

  This layer, also called Device or Physical layer works like people's eyes, ears and nose. It is in charge of identifying and collecting information about the environment. To achieve these objectives, hardware like sensors and actuators are adopted. Many types of sensors are employed and attached to things to gather data, with their selection being dependent on the specific needs and goals of the application. The data collected may cover a range of wide parameters, including location, air quality, environmental conditions, motion, vibration, and more.

- *Network layer:*

  The Network layer, also known as the Connectivity layer, is responsible for enabling communication between devices, serving as a key link between the perception layer and the application layer. To achieve this, various communication protocols are employed, such as *Message Queuing Telemetry Transport* (MQTT), which is a lightweight messaging protocol designed for low-bandwidth, high-latency networks, and *Constrained Application Protocol* (CoAP), a protocol specifically made for devices with limited processing power and memory. The transmission medium can be either wireless or wired, with routers and gateways serving as key components in this layer.

- *Application layer:*

  The final layer in this architecture is the application layer, which delivers user-centric services tailored for specific applications. This layer involves software for device communication and management, enabling the establishment, provisioning, and control of connections with connected devices. Moreover, this layer includes an application platform for the creation and implementation of IoT applications. It outlines a multitude of application scenarios suitable for Internet of Things deployment, exemplified by domains such as smart homes, urban environments, and healthcare.

A graphical representation of the IoT layered architecture is provided in Figure 2.1.

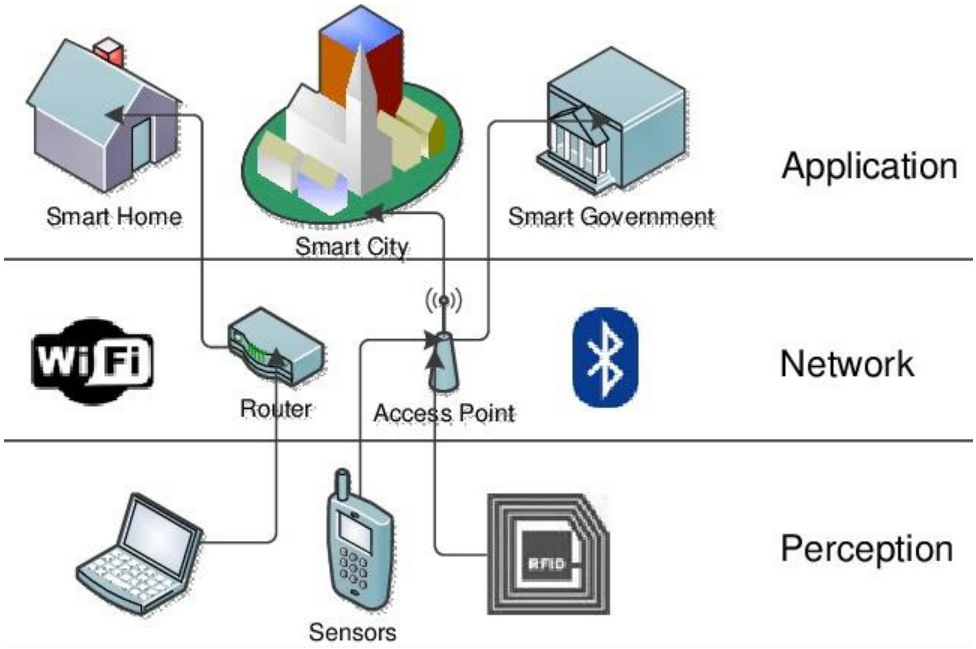

Figure 2.1: The IoT Layered Model [22].

2.3 IoT Applications

Potential IoT applications are extensive and varied, touching virtually every aspect of daily life for individuals, businesses, and society at large.

IoT technology is used to create "smart" environments and spaces across various domains, including transportation, buildings, cities, lifestyle, retail, agriculture, factories, supply chains, emergency response, healthcare, user interaction, culture and tourism, and energy management.

2.3.1 Smart Cities and Smart Homes

The idea behind smart cities is to interconnect facilities together with the use of interactive sensors and digital interfaces; these tools work by collecting information about geographical locations, exposing that information uniformly, offering cities many advantages and solving a wide area of problems.

First of all, in the context of public safety, digital video monitoring and fire control systems integrated with IoT technology allow for more efficient emergency response and public announcements, ensuring a quicker reaction to incidents. Many cities are even equipped with IoT-enabled streetlights which can adjust their brightness based on weather conditions, traffic flow, and time of day, improving energy efficiency and safety. Another attractive use of IoT in cities is smart transportation; smart tools are being implemented to make the urban transportation system interconnected, integrated and automated, making life easier to the population. Automotive and smart mobility aim to improve transportation by enhancing road safety, reducing traffic congestion, and managing traffic flow effectively. In addition, another efficient example is parking monitoring, to ease the reservation and availability of spaces in parking areas, reducing the time and frustration associated with finding parking. Last but not least, using IoT for waste management services is helpful to detect rubbish levels in containers and optimise the collection of waste [24].

Besides the use of IoT in public places, it is not rare to implement an IoT system in private areas like people's houses and rooms. Nowadays, smart homes are increasingly equipped with smart devices, used to collect data, aiming at motion, light and fire detection. These tools can display information about weather conditions (humidity, temperature, pressure, wind speed, and rain levels) and monitor energy and water consumption, to save cost and resources. IoT brings intelligence to everyday appliances. For example, smart refrigerators with *Liquid-Crystal Display* (LCD) screens can track food expiration dates, and suggest ingredients to buy, with this information accessible via a smartphone app. Smart washing machines allow you to monitor and control laundry remotely, while smart kitchen ranges offer app-based remote temperature control and self-cleaning monitoring. Devices interconnected in smart homes can also play a significant role in enhancing the security of an entire private building. An example would be the usage of safety monitoring like cameras and alarms or detection systems to find out violations and break-ins to prevent intruders [24].

2.3.2 Smart Health

In the healthcare industry, it is becoming more frequent to use sensors and devices for health monitoring purposes. This phenomenon is having today a large impact, not only on patients but even on healthcare professionals.

IoT brings an important innovation in the surveillance of infirms conditions inside hospitals and hospices, with tools like fall detection devices to assist elderly or disabled people. Moreover, physical activity monitoring systems use wireless sensors to sense breathing motion and heart rates, providing data that can be available trough apps on smartphones. For instance, dentists and dental surgeons can implement IoT tools to track patients' brushing habits in order to retrieve helpful statistics.

As mentioned before, also healthcare professional can benefit of IoT systems. Medical fridges can be equipped with smart tools to control the conditions inside fridges which often store vaccines and medicines [24].

2.3.3 Smart Energy and Environment

Unlike traditional power grids, which rely on centralized power plants to distribute energy, the smart grid allows for bidirectional flow of energy between producers and consumers. This means that consumers can also generate electricity. A smart grid is the first step when it comes to implementing IoT in the energy field and since its first application it has always been advantageous to monitor and manage energy consumption making the energy resource consumption more efficient. Smart tools can be used to monitor and analyze the flow of energy from wind turbines and power houses, enabling two-way communication with smart meters to analyze consumers' energy consumption patterns.

IoT can additionally be a powerful tool when it comes to environmental challenges. Apart from the already mentioned weather monitoring, smart devices are used to study the quality of water in rivers and sea for eligibility in drinkable use. With IoT, scientists can retrieve insights on air pollution caused by CO_2 emissions in factories and toxic gases in the air.

At the end, it is relevant to be aware of how much the IoT is helping the flora and fauna: tracking collars are used to locate and protect wildlife (especially endangered animals) and smart gadgets are employed to gather information about water in rivers during rainy days, avoiding floods [24].

2.3.4 Smart Agriculture and Smart Industry

Two other fields which were touched by the IoT revolution are the agriculture and the industry sectors.

From centuries farming had been a traditional activity, sometimes with minimal innovation. Today, with the implementation of IoT systems, the way of doing agriculture is mutating to a new era, with modern technologies and tools which are making life easier to all the people whose life is based on the primary sector. For instance, with the use of smart systems in farms, many animals grazing in open pastures can be localized and offspring growth conditions are ensured. Additionally, fields can be monitored to retrieve data about crop status and improve fertilization processes. Even green houses can be implemented with IoT to control micro-climate conditions to maximize the growth of fruit and vegetables plants.

On the other hand, what has introduced a wide area of cutting-edge technology is the secondary sector, the industry field. For factory workers, having a safe job while working in a protected environment is crucial and IoT aims to ensure these conditions are respected. One example is the use of smart detection systems, which have the goal to recognize and read gas levels in industrial environments to prevent leakage and chemical accidents, that can be very dangerous. Eventually, sensors installed in equipment can monitor for signs of malfunction, allowing for early predictions of equipment issues. This enables automatic scheduling of maintenance and repair services before a part actually fails [24].

2.4 IoT Challenges

Although IoT has had a transformative impact, it is crucial to recognize that not everything is as promising as it seems. As this technology continues to evolve and its potential becomes more widely recognized, there are still significant hurdles to overcome.

One of the foremost challenges in the IoT landscape is achieving seamless interoperability among a diverse array of devices, platforms, and applications. As IoT technology continues to evolve, it incorporates a wide range of architectures and communication protocols, leading to significant variability in how devices interact with each other. This heterogeneity creates a complex environment where devices from different manufacturers may not easily communicate or share data. Addressing this issue requires developing and adopting common standards, which can be costly and time-consuming for organizations that rely on custom or proprietary solutions. The lack of interoperability can eventually hinder innovation, as busi-

nesses must invest substantial resources to ensure their systems work together cohesively, potentially slowing the pace of technological advancement [9].

Moreover, scalability is considered as one of the most important issues in the IoT field. Scalability refers to a system's capacity to expand its resources to accommodate a growing number of connected devices, increasing data volumes, and rising network traffic, all while maintaining high performance and reliability [3]. The exponential growth in the number of IoT devices leads to increased demands on infrastructure, processing power, storage, and bandwidth, requiring systems to address issues like network congestion, latency, data storage, and system integration while ensuring security and cost efficiency. To meet these demands, scalable IoT architectures often rely on cloud computing for elastic resource scaling, edge computing for reduced network load, and distributed systems to manage the growing complexity and diversity of IoT application [11]

Power consumption is another significant burden in IoT, as sensor operations typically depend on battery life. Moreover, many modern devices, including smartphones, tablets and laptops, are equipped with sensors that support a range of applications, often consuming a large amount of power. Energy efficiency is particularly crucial in applications like weather prediction, which rely heavily on *Global Positioning System* (GPS) for location tracking. Keeping GPS on throughout the entire sensing process can quickly drain the battery, highlighting the importance of optimizing power consumption to ensure the longevity and reliability of IoT devices [3].

Finally, there is still an enormous obstacle IoT needs to address: security and privacy challenges. A larger number of IoT gadgets is appearing rapidly on our cities and lives everyday; as this number increases, enemies acquire a greater number of ways to compromise the system operations. IoT objects, being linked to a network, can become more vulnerable to malware and botnets, data breaches and *Distributed Denial-of-Service* (DDoS) attacks [15]. In 2019 Avast made a security report highlighting how two out of five IoT devices are exposed to cyberattacks, confirming botnets as the most common type. This scenario is worsened by IoT designers who often neglect security aspects, resulting in companies having completely unsatisfactory security. Due to limited public awareness about IoT devices, a lack of standardization, and the dynamic, constantly evolving nature of IoT, concerns on security and privacy are further arising [6].

Chapter 3

Security in IoT

3.1 Introduction

In the history of IoT technology, managing security has never been an easy task due to the fluid and transient connections among devices, the wide range of actors involved and the eventual constraints on resources.

The increase in cyberattacks is proportionally tied to the exponential growth of IoT systems and devices worldwide, especially in many areas of our lives [21]. In 2022, the global number of cyber attacks targeting IoT devices exceeded 112 million. This marks a considerable increase from the roughly 32 million cases reported in 2018, indicating a significant upward trend in IoT-related cyber threats [25]. Despite the rising threat of IoT-based cyberattacks, only 35% of the organizations surveyed have an IoT security strategy in place, and even fewer, just 28%, have fully implemented one [26]. Additionally, another survey found that while 80% of organizations experienced cyberattacks on theirIoT devices in the past year, 26% still did not employ security protection technologies [17].

These surveys underline the significant security weaknesses present in many IoT devices and highlight the urgent need for organizations to take proactive measures and invest inIoT cybersecurity [21].

3.2 The CIA security model

To avoid security breaches and maintain the integrity of IoT systems, it is fundamental for services and devices to be compliant to what is known as the *Confidentiality, Integrity, Availability* (CIA) triad. Fortinet, chief company in the cybersecurity field, defines the triad as *a*

common model that forms the basis for the development of security systems. It is used for finding vulnerabilities and methods for creating solutions [12]. This model, as mentioned above, is composed of three different security principles, each one essential to establish a secure communication framework for people, software, processes, and devices in an Internet of Things environment [22, 12]. A detailed discussion of each principle is reported in the following:

- *Confidentiality:*

  Data must be secure and only available to authorized users. With regard to IoT, the term user does not only refers to human beings, but also to objects and services, which are the main components acting in the different IoT systems. Human users of IoT should always be informed about the mechanisms used for data management, understand who or what is responsible for overseeing this process, and ensure that data remains secure at every stage. An example of confidentiality issue could be a sensor revealing the collected relevant data to neighboring nodes, with no authorization.

- *Integrity:*

  Integrity ensures that data is reliable, intact, and has not been changed or tampered with, whether intentionally or unintentionally, by someone without authorization. Recall that an IoT system is based on the exchange of data and information between various types devices hence it is crucial to ensure the accuracy of the data being sent with no tampering or interference during the transmission process and received from the right sender. Typically, firewalls and protocols are used to manage data traffic, but these mechanisms may not be suitable to ensure integrity at IoT endpoints due to the low computational power of many IoT nodes, which might struggle to support these security measures.

- *Availability:*

  Availability is the principle that ensures data, systems, and services are accessible and functional when authorized users need them. That means that the IoT users should have all the data available whenever they need and request to access. Data is not the only critical element in IoT: devices and services must also be accessible and operational when required, ensuring that IoT can meet its intended expectations.

The three principles are summarized is Figure 3.1.

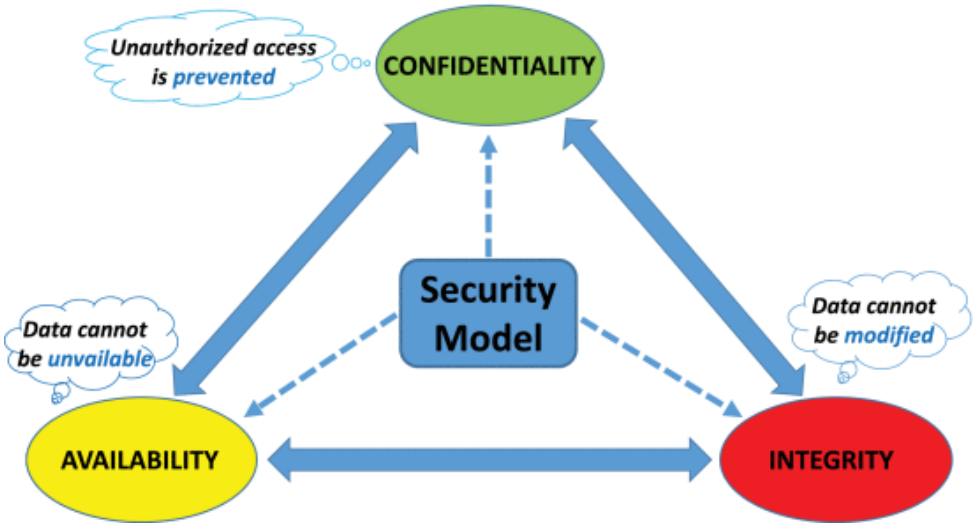

Figure 3.1: The CIA triad [14].

3.2.1 Additional security principles

Beyond the CIA security model, other principles have been explored to enhance the overall security framework of IoT. These are general principles applicable not only to the IoT framework, but also to information security at large. Some of the most shared by the scientific community are described below [22, 32]:

- *Authentication:*

  in the context of security, authenticity ensures that an entity is indeed what it claims to be, and its actions or messages are not forged or manipulated by unauthorized parties. Therefore, every object in the IoT must be able to identify and authenticate other objects clearly. This can be complex due to the involvement of various entities within the IoT ecosystem, including devices, people, services, service providers, and processing units, each with its own characteristics and roles. Additionally, objects may need to interact with unfamiliar objects they have never been encountered before. Given these complexities, a mechanism for mutual authentication of entities in every IoT interaction is essential.

- *Authorization:*

  once an entity has been identified, it is essential to confirm that it owns the appropriate permissions to access the data, resources, or applications within the system.

- *Non-Repudiation:*

this principle is defined as the system's ability to confirm or deny the happening of an action within its framework. It serves to prevent any entity from concealing their actions with hostile intent, ensuring they cannot later deny their involvement. Essentially, non-repudiation ensures that no party can falsely claim that a transaction did not occur when it indeed did, or vice versa.

- *Privacy:*

  When data is being handled, processed, stored or deleted, it is crucial to ensure that individuals' rights concerning the use of personal information are respected. This generally involves compliance with contracts and organizational policies, as well as adherence to applicable regulations or laws. In Europe, for example, the *General Data Protection Regulation* (GDPR), which has been in effect since 2020, sets strict guidelines for protecting personal data.

3.3 Security Challenges in Each layer

As described in Section 2.2, IoT systems are designed following a distinct three-layer architecture, with every layer being different from the other, with its own characteristics and functionalities. This structural diversity also extends to IoT security challenges: each layer has its own specific security problems and threats that must be addressed.

This section offers a comprehensive overview of the various challenges within the realm of IoT security. It emphasizes a layered approach, delving into security concerns for each layer of the IoT architecture.

3.3.1 Perception layer

In the perception layer, three main, security issues are observed. The first concerns the reliability of wireless signals. This challenge arises because the majority of communication links in IoT sensor networks are wireless, which can be disrupted by interfering signals. These disturbances can reduce the efficiency of wireless communication, leading to potential security vulnerabilities and degraded performance in the perception layer. Secondly, the sensor nodes can be an object of tampering, not only by the owner himself but even by attackers: this is due to IoT nodes usually operating in external and outdoor environments. This will eventually result in devices getting physically tampered for malevolent purposes, perhaps by injecting malicious code. Furthermore, since IoT nodes are often moved to different locations,

the network configuration can change frequently. This characteristic creates challenges in maintaining consistent security protocols across the network.

Since sensors can be considered the least secure devices within the entire IoT architecture, threats targeting the sensing layer are particularly significant. This vulnerability makes sensors the simplest entry point for breaching the entire IoT system, enabling attackers to use them as a gateway to launch attacks [22]. Some examples of attack types in the perception layer are reported in the following: [28]:

- *Forged node insertion:*

  the attacker may insert a deceptive or malicious node between network nodes to intercept the data stream without detection, eventually stopping or destroying it, causing irrecoverable damage.

- *Malevolent code insertion:*

  the attacker physically inserts a code that has been previously modified to cause harm to the network service, making it unavailable.

- *RFID unauthorised access:*

  in these types of systems, the tags or chips used to grant access to certain information are often vulnerable because they lack a secure authentication system. It is very common for these tags to be manipulated easily, resulting in individuals gaining access to sensitive information by exploiting this weakness.

- *Slumber denial attack:*

  in IoT networks, nodes can be found in very distant places. These nodes are powered by replaceable batteries: if the device is not in use, the nodes are programmed to slumber to save battery life. The attacker can prevent this process by feeding false input to the node, causing a major waste of energy in the long run and eventually a power failure.

- *Introduction of noise on data:*

  taking in consideration that IoT nodes are located at large distance from one to another, a slight distortion of data caused by the attacker, can be the result of data loss and and potentially compromise the integrity of the entire network.

3.3.2 Network layer

Since the primary function of the network layer in IoT is to transmit aggregated data, the security concerns at this layer revolve around ensuring the availability of network resources, as well as addressing issues related to data integrity, confidentiality, and protection against network attacks. Consequently, this layer is mainly susceptible to *Denial-of-Service* (DoS) and *Man-in-the-middle* (MITM) attacks, eavesdropping , illegal access, destruction and virus infiltrations.

It is important to recall that IoT communication differs from traditional internet communication because it involves M2M interactions. This brings a security challenge of compatibility, as the diversity of IoT devices and networks complicates the use of standard protocols and makes it difficult to ensure effective security measures.

On top of that, attackers can exploit the network's interconnectedness to gather sensitive user data aiming for criminal activities like identity theft or financial fraud. Thus, safeguarding the network infrastructure is crucial to prevent unauthorized access and data breaches.

A thorough examination of common threats in the network layer is provided in the following [19, 1]:

- *Traffic analysis attack:*

  the hackers use packet sniffing tools or port scanning applications on wireless technologies to obtain confidential information (*enumeration*) before proceeding with malevolent attacks.

- *DoS attack:*

  occurs when an attacker floods a network with excessive traffic, overwhelming the system's resources and rendering it inaccessible to users. This can lead to a serious disruption, causing devices and servers to be unable to communicate with users or provide data. An example of DoS attack is SYN flood attack which aims to make a server unavailable to legitimate traffic by consuming all available server resources. This type of attack is discussed in the case study in Chapter 5.

- *Sinkhole attack:*

  this attack compromises data security by intercepting all signals from wireless sensor network nodes and redirecting them to a different location controlled by the attacker. As a result, instead of delivering packets to their intended destination, all data packets are dropped.

- *MITM attack:*

  the attacker interferes between two nodes, exploiting vulnerabilities in the IoT communication protocol to gain unauthorized access to restricted information and violate the privacy of the nodes involved. This type of attack implies managing the communication between the nodes without physically interfering with them.

- *Spoofing:*

  this type of attack involves a malicious device manipulating an authentic device's IP address, MAC address, or other identifying information to masquerade as a legitimate one. This allows the attacker to gain unauthorized access toIoTsystems and potentially manipulate data or disrupt operations.

- *Gateway attack:*

  the connection between sensors and the internet infrastructure is compromised.

3.3.3 Application layer

The IoT, being a relatively new technology, encounters significant security challenges due to inconsistent or incomplete policies and standards that guide application development and interaction. Although some frameworks exist, they are not universally adopted, leading to potential gaps in the application layer security. This absence of standardization contributes to inconsistencies in authentication mechanisms across various software and applications, complicating the integration efforts and increasing the risks to data privacy and identity authentication. Additionally, the enormous volume of connected devices sharing data can limit the applications performance, potentially affecting service availability. Lastly, but equally significant, when developers design applications, it is crucial for them to ensure how users will interact with these systems, the amount and sensitivity of data that will be disclosed, along with clear accountability for data management. Users, on the other hand, should have tools to control the information they share and need to be aware of the risks involved in revealing their data [22].

Therefore, a wide range of threats pose significant risks to the security and reliability of the application layer. These threats can impact everything from data privacy to the integrity and availability of services. A list of common threats that IoT systems must guard against is reported below [19, 1]:

- *Malevolent code attacks and scripts:*

  These attacks target devices such as home routers and security cameras, often using harmful computer worms to compromise and disrupt their functions.

- *Software defencelessness:*

  when programmers write codes for IoT systems they need to follow some coding standards and best practices. If this is not respected, it increases the likelihood of encountering issues such as security vulnerabilities, interoperability problems, and reliability issues within the IoT system.

- *Phishing attacks:*

  adversaries may exploit phishing techniques to deceive users into revealing sensitive information, such as device credentials or access codes, through compromised emails or fake IoT management portals. This can result in the compromise of IoT systems, leading to data breaches, device manipulation, or unauthorized control over connected devices.

- *Virus, spyware and worms:*

  these types of malware are the most common for attackers to infect systems resulting in information theft, DoS and data corruption.

Table 3.1 summarizes the various security challenges of each layer.

Table 3.1: Security Challenges in IoT Layers [28, 19, 1]

Layer	Security Challenges
Perception Layer	**Wireless signal reliability**: Interference can disrupt communication.
	Sensor tampering: Physical tampering can inject malicious code.
	Frequent network reconfiguration: Moving nodes affect security consistency.
	Least secure devices: Sensors are vulnerable entry points.
	Forged node insertion: Malicious nodes intercept data streams.
	Malevolent code insertion: Harmful code disrupts network service.
	RFID unauthorized access: Insecure tags allow access to sensitive information.
	Slumber denial attack: False input prevents energy-saving sleep mode.
	Noise on data: Distorted data compromises network integrity.
Network Layer	**Data integrity and confidentiality**: Ensuring secure data transmission.
	DoS attacks: Excessive traffic overwhelms the network. Discussed in Chapter 5.
	MITM attacks: Unauthorized interception of communication.
	Spoofing: Malicious devices impersonate legitimate ones.
	Traffic analysis attack: Packet sniffing to obtain confidential information.
	Sinkhole attack: Intercepting and redirecting data packets.
	Gateway attack: Compromising connections between sensors and the internet.
Application Layer	**Inconsistent policies and standards**: Lack of uniformity in security practices.
	Authentication mechanism inconsistencies: Varying security measures across applications.
	Data volume and performance: Large data sharing affects availability.
	User interaction and data control: Ensuring user awareness and control over shared data.
	Malevolent code attacks: Harmful scripts target devices like routers.
	Software vulnerabilities: Poor coding practices lead to security issues.
	Phishing attacks: Deceptive techniques to obtain sensitive information.
	Malware: Viruses, spyware, and worms cause data theft and corruption.

Chapter 4

Intrusion Detection Systems and Anomaly-based Detection Techniques in IoT Networks

4.1 IDS Taxonomy

There are many tools which can be implemented in the IoT realm in order to protect and safeguard its vast and interconnected network from malicious attacks and anomalous traffic. IDSs are one of them and will be the main focus of this Chapter, whose goal is to explore how IDSs play a crucial role in securing IoT networks from potential threats.

An IDSs can be both a device or a software application that monitors and protects a network from malicious activities or privacy violations. These tools are therefore specialised in collecting and analyzing network traffic, with detecting intrusions as the biggest goal [33]. A standard IDSs comprises sensors, an analysis engine, and a reporting system. Sensors are strategically placed throughout the network or on specific hosts, aiming at gathering data such as traffic statistics, packet headers, service requests, operating system calls, and file-system modifications. This collected data is then sent to the analysis engine, which examines the information to identify any ongoing intrusions. If an intrusion is detected, the reporting system alerts the network administrator [40].

There are many reasons why IDSs are used and suited for IoT systems [40]. First of all, for some IoT devices, implementing tools such as anti-virus software is difficult and challenging. Anti-virus can be computationally heavy for the device which can result in a performance decrease. An additional drawback of equipping IoT devices with such technologies would also

increase the apparatus cost [33].

IDSs for the IoT are classified into two categories:

- *IDS types based on their positions in the network*
- *IDS types based on their detection techniques*

The first category is based on where the IDSs is located in the IoT network. The second category of classification is based on the techniques used for implementing the IDSs. The classification of IDSs is illustrated using Figure 4.1.

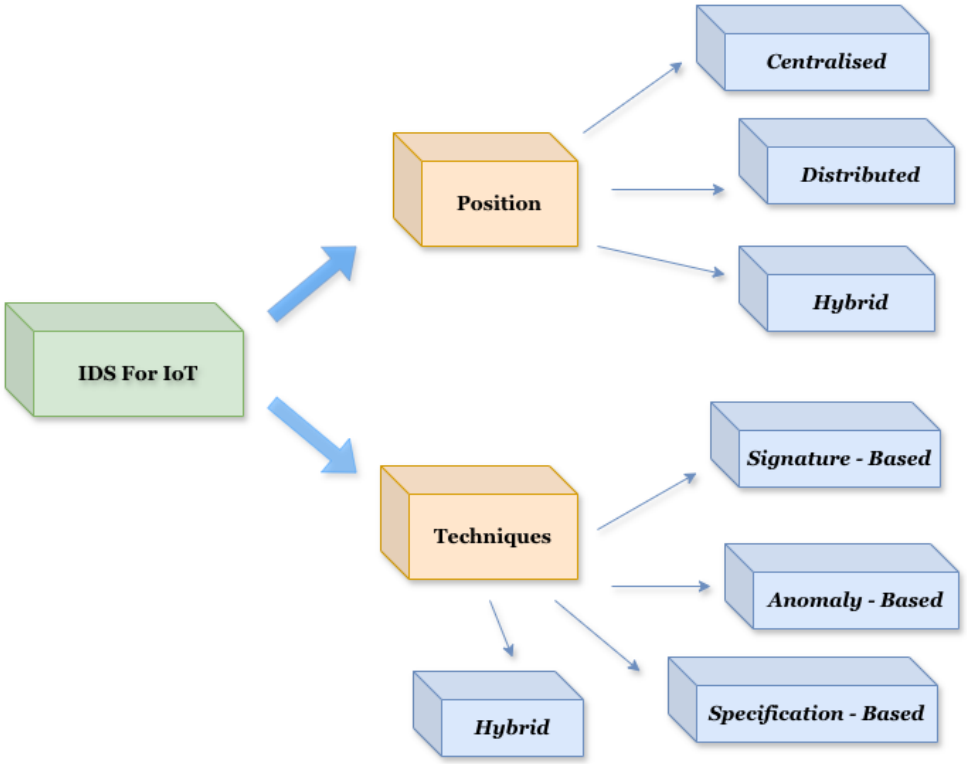

Figure 4.1: Types of IDSs for IoT redrawn from [4].

4.1.1 IDS Types Based on its Position in the Network

While developing an IDSs for IoT networks, the IDSs can be deployed either in the router connecting various devices or in dedicated smart devices and hosts; additionally, a third method involves combining both approaches, utilizing the strengths of router-based and device-based deployments.

In the following the various IDSs placement strategies in IoT networks along with their advantages and disadvantages are described [40, 33, 4].

- *Distributed IDS placement:*

in a distributed IDSs placement strategy, IDSs agents are deployed in every physical device of the IoT network. In this type of strategy, each node in the IoT network is responsible for monitoring and detecting attacks. Given the resource-constrained nature of IoT devices, it is crucial to examine and optimize these properties since the IDSs is installed on each node.

- *Centralized IDS placement:*

  in this strategy, IDSs are installed on a centralized router or a dedicated server. Implementing centralized IDSs in IoT is a straightforward process due to the presence of a centralized edge node, the border router, which connects the IoT network to the Internet. Because data packets from outside enter the IoT environment through the border router, a centralized IDSs can quickly recognize external attackers. Therefore, when the IDSs is deployed in the border router, it can easily monitor, analyze, and drop malicious data packets upon detecting any attacks. However, detecting internal attacks is challenging with this approach, as it requires comprehensive monitoring and analysis of all internal nodes connected to the border router.

- *Hybrid IDS placement:*

  the hybrid placement strategy for IDSs combines the advantages of both centralized and distributed IDSs strategies. In a hybrid IDSs placement strategy, the network is organized into clusters, with a central IDSs node assigned to each. This central node manages and monitors the activities of other nodes within its cluster. While hybrid IDSs generally entails higher resource consumption compared to distributed IDSs, robust nodes are chosen as cluster heads to efficiently manage each cluster.

4.1.2 IDS Types Based on its Detection Techniques

IDSs approaches when it comes to detection can be classified as signature-based, anomaly based and specification-based [40, 33, 4]:

- *Signature-based detection:*

  this method operates by identifying malicious patterns through matching attack signatures stored within the IoT internal database. So, when an attack signature matches with an existing one, an alert is triggered. This approach is highly effective in pinpointing known threats, but struggles to detect new attacks (also known as zero-day attacks) or

variations of exiting ones. In signature-based methods, the costs associated with storing signatures in databases and the computational workload required for running learning algorithms to validate each signature are significant.

- *Anomaly-based detection:*

  it detects unknown attacks by first creating a model of the normal behavior of the system. Often relying on ML algorithms, it then compares anomalous network activities against this model. This method usually has high false positive rates. The ML techniques and statistical methods used for matching algorithms can be resource-intensive, posing a challenge for deployment on low-capacity IoT nodes.

- *Specification-based detection:*

  this approach establishes guidelines for the expected behaviour of network components. Similar to anomaly-detection, deviations from these specifications are flagged as intrusions. Specifications-based detection relies on security experts, whose job is to define specification (sets of predefined guidelines) for each element, which typically results in lower false positive rates. Unlike anomaly detection, this method does not rely on learning algorithms, but it poses the challenge of needing different specifications for different platforms or environments.

- *Hybrid detection:*

  hybrid IDSs combine the concepts and advantages of signature-based, anomaly-based, and specification-based IDSs to achieve high classification accuracy and detection rates.

4.2 Anomaly-based Detection Techniques

Signature-based detection techniques are generally not effective against evolving threats, as these threats frequently mutate, causing their signatures to change. These techniques are therefore not advisable in real-world scenarios when the goal is to detect new variants of attacks. In contrast, anomaly-based detection techniques are widely favored in IoT environments because they operate on the assumption that malicious traffic will exhibit behavioral differences from normal traffic. Consequently, anomaly-based IDSs are employed to monitor normal network behavior and define thresholds to detect deviations, effectively identifying potential security threats [31].

Before delving into the various types and methods of anomaly detection, it is crucial to establish a definition of anomaly. An anomaly refers to a data sample that exhibits a notable deviation from the anticipated behavior within a modeled system. Generally, anomalies are regarded as uncommon occurrences or observations that significantly stray from established patterns of behavior. These deviations may manifest in a single data point, within a particular context or temporal segment, or across the entirety of a dataset [38]. Anomalies are frequently linked to external factors such as sensor malfunctions or malicious attacks.

Figure 4.2 illustrates a visual representation of examples of anomalies.

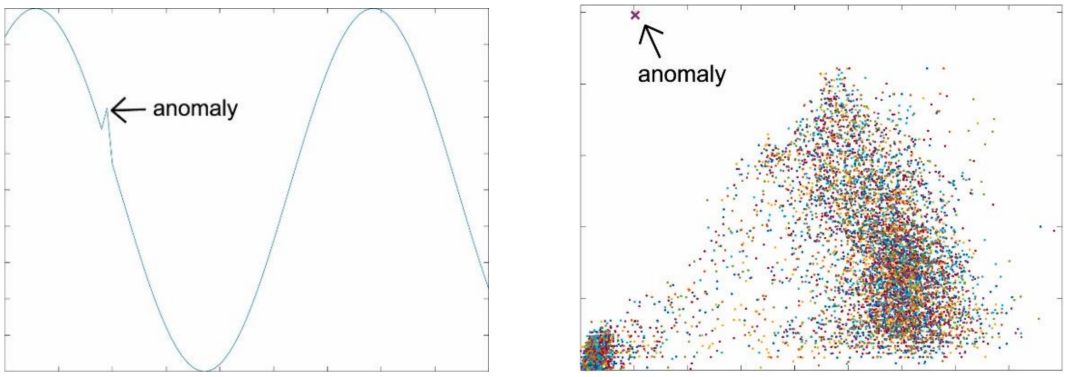

Figure 4.2: Visual representations of anomalous occurrences [38].

In the realm of IoT, anomalies can be classified into three primary types [38]:

- *Point anomalies:*

  single data points that diverge significantly from expected behavior.

- *Contextual anomalies:*

  anomalies that depend on the context in which the data is viewed, requiring both contextual and behavioral attributes for detection.

- *Collective anomalies:*

  involve the entire dataset and are used to identify irregularities over a broader scope.

Furthermore, detection algorithms can be classified based on their latency and scalability characteristics [38]:

- *Online algorithms:*

  process data serially, one data point at a time or within a time window, allowing for real-time detection without needing the entire dataset.

- *Offline algorithms:*

  handle complete datasets for more complex problem-solving, often requiring more computational resources and time but providing comprehensive analysis.

Finally, the methods for finding anomalies in IoT systems can be classified in three different categories [38]:

- *Geometrical methods:*

  use distance-based or density-based strategies to distinguish between normal and anomalous data points. The underlying assumption is that anomalies tend to appear in sparse regions of the dataset. Anomalies are classified by applying a threshold (denoted as t) to the calculated distance (d) between data points, which can be set either statically or dynamically. This threshold-driven classification is represented by the following equation:

$$d = \begin{cases} \leq t & \textit{Normal (under threshold)} \\ > t & \textit{Anomaly (above threshold)} \end{cases} \tag{4.1}$$

- *Statistical methods:*

  these involve modeling normal data patterns using mathematical models and distributions. Anomalies are detected as data points that significantly deviate from these established models. Examples include using probability distributions like Gaussian or Poisson.

- *Machine learning methods:*

  these models learn patterns from the data and use them to classify normal and anomalous events. Supervised learning techniques like *Support vector machine* (SVM) and random forests are used when labeled data is available. Unsupervised learning techniques, such as clustering or outlier detection algorithms, identify anomalies without labeled training data.

In Figure 4.3 an illustration of the different anomaly detection strategies is provided.

4.3 IDS techniques and algorithms

This section examines existing anomaly-based IDSs techniques tailored to safeguard IoT environments, providing an analysis of the various detection methods and approaches.

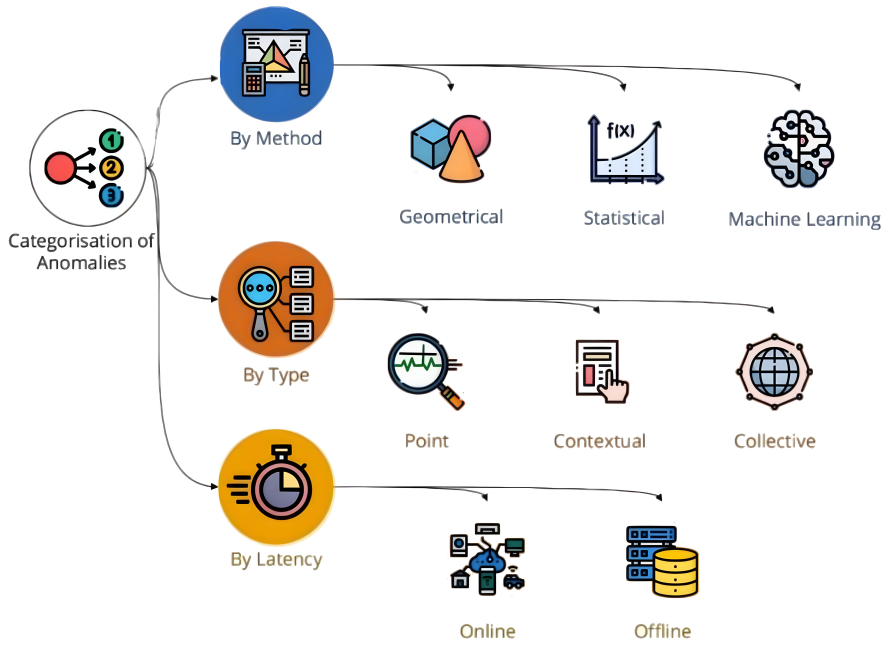

Figure 4.3: An overview of anomaly detection strategies, edited from [38].

4.3.1 Machine Learning approaches

ML, a branch of *Artificial Intelligence* (AI), is a field where machines learn from data, build models, and process new data using the trained models. This technique is widely applied in various areas of anomaly-based IDS to train the system and enabling it to detect new data anomalies [10].

4.3.1.1 Supervised Machine Learning algorithms in anomaly-based IDS

IDS using supervised learning techniques operate by utilizing labeled data for intrusion detection. This method comprises two stages: training and testing. During the training phase, the algorithm is trained using a training dataset consisting of input features and corresponding predefined labels (typically “normal” and “attack”). In the testing phase, separate testing data (excluding training data) is employed to evaluate the accuracy of the trained algorithm. More specifically, the model is presented with unknown data (the test set) for which it generates predicted outputs without prior knowledge of the actual outputs. These predicted outputs are then compared with the actual outputs to assess the accuracy of the model. Various classification algorithms, such as decision trees, Naïve Bayes, and support vector machines, can be utilized [10].

- *Decision Tree:*

  it is a popular tool in supervised ML used for classification and regression tasks. It has

a hierarchical structure, resembling the one of a tree and it is composed by nodes and branches.

The root node, at the top, represents the entire dataset and starts the decision-making process. Internal nodes evaluate specific features to form homogenous subsets, while branches connect these nodes, representing decision outcomes. Leaf nodes at the ends of branches provide the final class label or prediction.

The tree is built by repeatedly splitting the dataset into smaller parts until each part is uniform or meets a stopping condition. To make a prediction, the algorithm starts at the root node and moves through the internal nodes, following the branches based on decisions, until it reaches a leaf node that provides the final result.

Nancy et al. [23] utilized a fuzzy decision tree classifier for intrusion detection in wireless sensor networks, demonstrating its effectiveness in reducing false alarm rates and energy consumption.

- *Naïve Bayes:*

  it derives results by applying conditional probability formulas and it is typically applied in high-dimensional datasets. This algorithm is highly efficient due to its streamlined calculation process and the assumption of conditional independence among features. However, its accuracy may decrease if the assumption of individual independence is incorrect.

  Yang et al. [37] introduced an artificial bee colony-based Naïve Bayes method for intrusion detection, demonstrating its effectiveness compared to other state-of-the-art approaches.

- *Support Vector Machine:*

  SVM constructs a boundary, known as a hyperplane, in an N-dimensional space to distinguish between different classes. The dimensionality of the hyperplane corresponds to the total number of features in the dataset. When the number of features exceeds three, visualizing the hyperplane becomes challenging. To handle data separation, SVM employs a kernel function. Different types of kernel functions, such as linear, Gaussian, hyperbolic, and polynomial, are used for this purpose.

  Safaldin et al. [30] proposed an intrusion detection system for wireless sensor networks based on a variant of Gray Wolf Optimization and SVM, claiming increased accuracy and reduced false alarm rates.

4.3.1.2 Unsupervised Machine Learning algorithms in anomaly-based IDS

Unsupervised ML techniques operate on datasets lacking predefined labels. In contrast to supervised models, which rely on labeled datasets for training, unsupervised methods aim to discern hidden patterns within the data itself.

In an IDS scenario, for instance, an unsupervised technique would initially seek similarities among the dataset's records. Once the data is clustered based on these similarities, smaller clusters might be labeled as intrusion instances, while larger clusters could be labeled as normal occurrences. This approach assumes that normal data would be more prevalent than intrusions, resulting in distinct clusters for each category due to their dissimilarities.

Below follows an overview of the most used in the subject [10].

- *K-means Clustering*

  this technique aims to organize n data points into k clusters, with each data object grouped based on its nearest mean. This iterative process continues until a specified number of iterations is completed, resulting in the final clusters. The total number of clusters is determined by the user.

  Tahir et al. [34] proposed a hybrid technique combining k-means clustering and the Naïve Bayes algorithm for intrusion detection. Their approach yielded relevant accuracy and detection rates.

  Figure 4.4 explains the working principle K-means clustering algorithm.

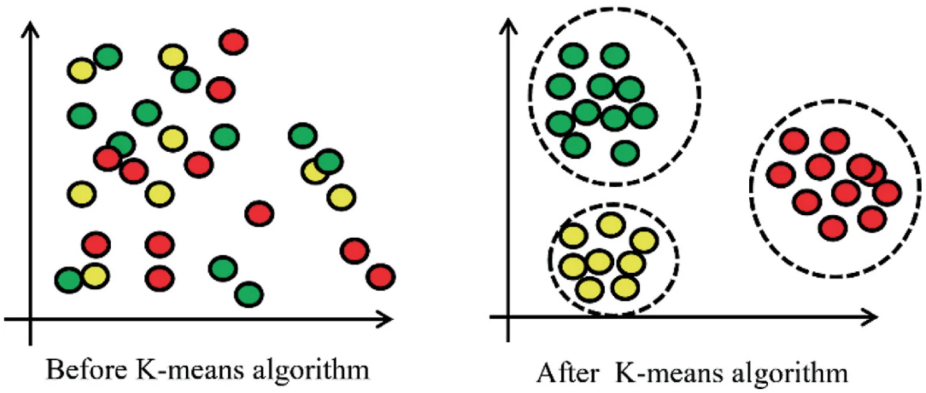

Figure 4.4: K-Means clustering [10].

- *Principal Component Analysis:*

  is an unsupervised learning technique utilized for dimension reduction. The algorithm transforms correlated features into a set of linearly uncorrelated features, known as

principal components. Widely used in data analysis and image compression, *Principal Component Analysis* (PCA) addresses the common problem in IDS of dealing with a large volume of irrelevant data.

Bhattacharya et al. [7] introduced a PCA firefly algorithm for intrusion detection, employing the XGBoost algorithm for classification. They reported that their model exhibited superior performance compared to other ML algorithms.

4.3.1.3 The advantages of Machine Learning Algorithms for Anomaly Detection

ML algorithms enable the automated and efficient detection of abnormal events within the vast amounts of data generated by IoT devices. As IoT systems become more complex and interconnected, traditional rule-based or threshold-based approaches may not suffice to capture the diverse and evolving patterns of anomalies. ML algorithms, by learning from historical data, can identify hidden patterns and adapt to changing conditions, making them well-suited for IoT anomaly detection. A key advantage of ML in IoT anomaly detection is its ability to handle large-scale and heterogeneous data . IoT environments generate various data types, including sensor readings, network traffic data, and system logs. ML algorithms can process and analyze this data to identify abnormal patterns indicative of security breaches, system failures, or other abnormal behaviors. They can handle the high volume, velocity, and variety of IoT data, making them scalable and applicable for real-time monitoring and analysis. ML algorithms also excel at detecting anomalies that are not explicitly recognized or anticipated in advance. Unlike rule-based approaches, these algorithms can learn from historical data to detect anomalies that may not be apparent to humans, allowing for proactive anomaly detection and early warnings of potential issues. This capability helps reduce the risk of system downtime or security breaches.

In summary, automated anomaly detection in IoT, enabled by ML algorithms, significantly reduces the need for manual inspections and analysis. Manual analysis of IoT data is time-consuming, error-prone, and inefficient, especially in large-scale deployments. ML algorithms allow for continuous and efficient monitoring of data streams from IoT devices in real time, freeing human operators to focus on more critical tasks, such as investigating anomalies and taking appropriate actions. By leveraging ML, IoT systems become more secure, reliable, and efficient by proactively detecting and mitigating abnormal events [38].

4.3.2 Deep Learning approaches in anomaly-based IDS

DL is a branch of AI that enables machines to learn from experience, emulating human intelligence. It relies on *Artificial Neural Network*s (ANNs), inspired by human brain neurons, where neurons are interconnected across different layers: the input layer, hidden layer, and output layer. The hidden layer, situated between the input and output layers, performs complex mathematical functions.

Here is a review of few DL algorithms most implemented for anomaly-based IDS [10].

- *Convolutional Neural Networks:*

  are a type of ANNs primarily employed for image classification tasks. *Convolutional Neural Network*s (CNNs) operate as supervised DL algorithms, comprising four key layers: convolution layer, *Rectified Linear Unit* (ReLU) layer, pooling layer, and fully connected layer. In the convolution layer, images pass through filters known as kernels, generating feature maps. Subsequently, the pooling layer reduces the size or volume of the feature map to enhance computational efficiency, a process referred to as subsampling or downsampling. Finally, the fully connected layer processes inputs from the preceding layers, producing output as a one-dimensional array representing the final classes.

  Researchers like Vinayakumar et al. [35] and Riyaz et al. [29] have explored the application of CNNs in IDSs.

- *Recurrent Neural Networks:*

  is a type of ANNs characterized by its dependency on previous outputs when processing current inputs. Unlike traditional neural networks, where the temporal relationship between inputs is not considered, *Recurrent Neural Network*s (RNNs) excel in tasks such as pattern recognition in text, speech, and handwriting, where sequential data streams are prevalent. Previous outputs are stored in a state vector, which affects the computation of the current output. RNNs leverage both previous and current inputs to compute new outputs.

  Researchers like Yin et al. [39] proposed DL models based on RNN for intrusion detection systems, achieving superior accuracy compared to other classification algorithms such as ANNs or SVM.

4.3.2.1 The advantages of Deep Learning Algorithms for Anomaly Detection

The power of neural networks allows DL algorithms to detect IoT anomalies by analyzing the data generated by IoT devices and learning complex patterns and representations. DL algorithms, have demonstrated impressive performance across various domains, such as computer vision, natural language processing, and speech recognition. Their capability to automatically extract hierarchical features and model intricate relationships makes them highly effective for detecting anomalies in IoT data [38].

One key advantage of DL algorithms in IoT anomaly detection is their ability to handle high-dimensional and unstructured data. IoT environments generate vast amounts of data, often in the form of images, sensor readings, or textual information. DL algorithms can efficiently process and analyze this data, capturing subtle and nuanced patterns that may indicate anomalies [38]. Another crucial aspect of DL algorithms is their ability to learn from data without relying on explicit feature engineering. Traditional ML algorithms often require manual extraction of relevant features, which can be time-consuming and challenging, especially in the context of IoT data. DL algorithms can autonomously learn and extract relevant features directly from raw data, eliminating the need for extensive domain knowledge and feature engineering. This allows them to uncover intricate and non-linear relationships in the data, enhancing the accuracy and robustness of anomaly detection [38].

Wrapping up, DL algorithms, with their ability to handle high-dimensional data, automatically learn relevant features, facilitate transfer learning, and analyze sequential dependencies, are extremely helpful in IoT anomaly detection. By leveraging deep neural networks, IoT systems can effectively detect anomalies in the complex and diverse data generated by IoT devices; however, the training of these algorithms is computationally expensive and thus typically performed on cloud platforms rather than directly on the devices [38].

4.4 Challenges in Data Anomaly Detection for IoT

When it comes to detect anomalies in IoT data, several challenges and difficulties arise due to the unique characteristics of data and the constraints of IoT environments. Some of these challenges are mentioned under [38]:

- *Scalability:*

  IoT systems generate massive amounts of data in real-time. Anomaly detection algorithms must scale to handle high data volume and velocity. This requires efficient algo-

rithms and infrastructure to meet computational and storage demands.

- *Imbalanced data:*

  IoT datasets frequently suffer from imbalanced class distributions, where normal instances far outnumber anomalies. This can result in biased models that favor the majority class, making accurate anomaly detection difficult. Techniques like oversampling or undersampling are needed to improve the detection of rare anomalies.

- *High dimensionality:*

  data in IoT is often high-dimensional, encompassing numerous sensors, devices, and data sources. This increases the complexity of anomaly detection, as algorithms must manage many features and capture complex interrelationships. Dimensionality reduction techniques may be necessary to address this challenge.

- *Concept drift:*

  IoT environments are dynamic, with the statistical properties of data changing over time. Models trained on historical data may become less effective with new data patterns. Continuous model updating and adaptation are necessary to address concept drift and detect evolving anomalies.

- *Privacy:*

  datasets in IoT often contains sensitive information, making privacy crucial. Anomaly detection algorithms must preserve privacy, ensuring sensitive data is not compromised. This requires designing algorithms that balance detection accuracy with privacy protection.

- *Real-time Detection:*

  many IoT applications require real-time anomaly detection for timely response and mitigation. Achieving this is challenging due to the computational complexity of some algorithms and the need for near real-time data processing and analysis. Efficient algorithms and scalable infrastructure are essential for real-time detection.

- *Interpretability:*

  understanding why an instance is flagged as an anomaly is vital for effective management and decision-making. However, ML and DL algorithms, while powerful, may lack

interpretability. Balancing accuracy and interpretability is crucial, especially in applications requiring explainability.

Addressing these challenges in IoT data anomaly detection necessitates the development of innovative algorithms, techniques, and frameworks that can effectively detect and stream high-dimensional data, adjust to dynamic environments, maintain privacy, and enable real-time detection. These solutions must also be energy-efficient and easily scalable to accommodate large-scale networks. Additionally, they should be capable of detecting anomalies caused by malicious activities, natural phenomena, and human errors [38].

Chapter 5

Case study: Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices

5.1 Introduction to Passban IDS

Throughout this Chapter, the main focus will be on Passban, an anomaly-based IDS capable of analyzing data generated by a wide range of IoT sources, proposed in [11]. The main goals of Passban are:

- *Security:*

  Passban is an edge-based IDS designed to guarantee comprehensive data protection near IoT data sources. It enhances security and privacy by minimizing network processing burdens and operates efficiently on resource-constrained IoT gateways.

- *Scalability:*

  Passban is a scalable IDS that that adapts to new threats without hardware upgrades, demonstrating that inexpensive IoT devices can provide effective protection in real-world scenarios.

- *Effectiveness:*

  last goal is to deliver an IDS with a reduced *False Positive* (FP) rate and satisfactory detection accuracy, as these are crucial for the effectiveness of any IDSs.

5.2 Structure of Passban IDS

This IDS is composed of five main components:

- *Packet Flow Discovery Block:*

  monitors network traffic by capturing raw packets and identifying network flows.

- *Feature Extraction Block:*

  computes network flow statistics and constructs a feature set to train a ML model. Passban IDS employs "NetMate", a software tool which analyses network traffic using the LipPCAP library, converts raw packets into network flows, and computes flow metrics.

- *Train/Load Model Block:*

  the training phase of the IDS occurs with normal network flow only, thus realizing and anomaly-based IDS. After training, the model is saved in the gateway internal memory to detect attacks on new traffic.

- Action Manager *(AM) Procedure:*

  this module is responsible for taking actions based on the decisions made by the learned model for attack detection. Examples of these actions include logging attack incidents, blocking incoming or outgoing traffic to a target device, and notifying a specific user about the attack.

- *Web Management Interface:*

  allows users to interact with the IDS, including cleaning and reviewing logs, controlling the IDS activation status, and managing the training process.

Passban's operational phases can be summarized in a training and in a prediction phase. These phases are both illustrated in Figure 5.1 all together with the network flow direction and the interfaces between the above-mentioned components of the IDS.

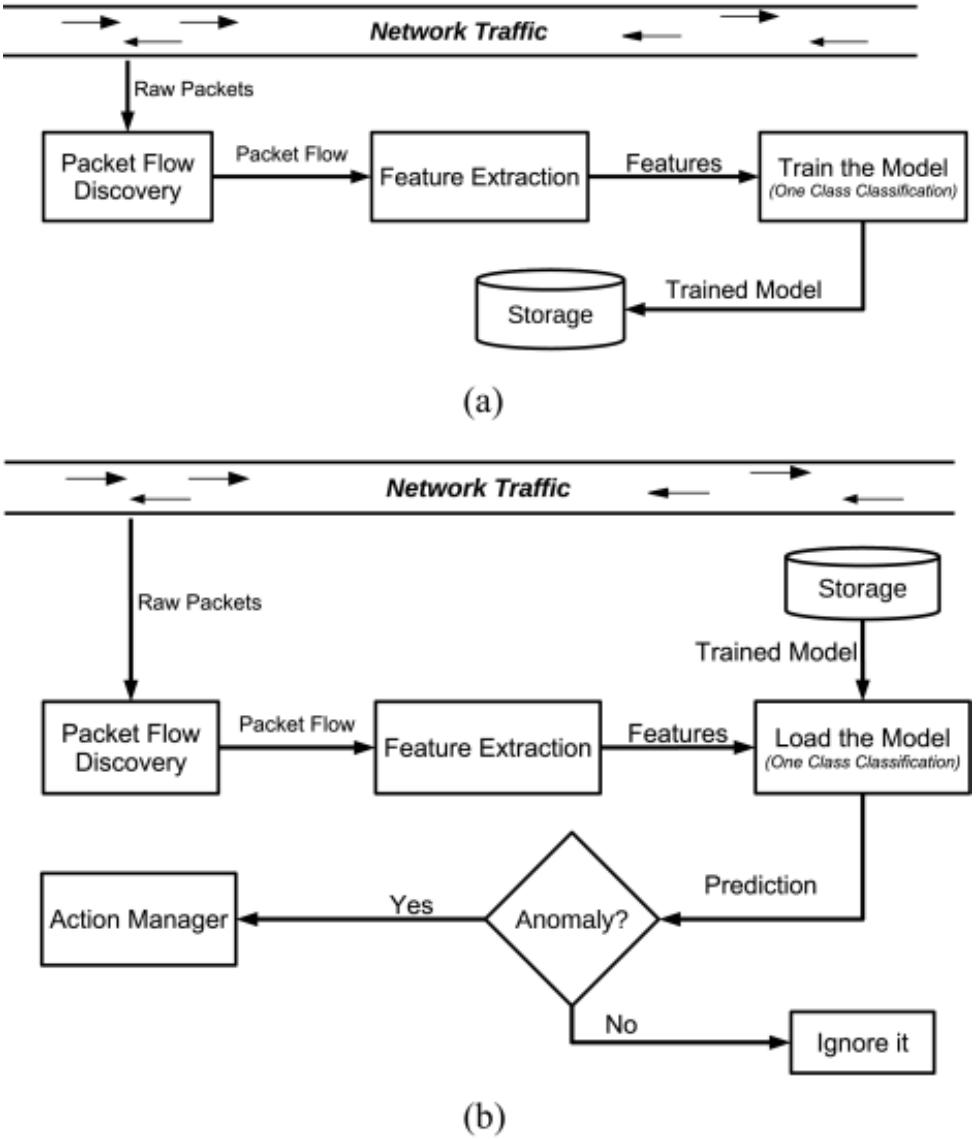

Figure 5.1: Main operational phases of the Passban IDS. (a) Training phase. (b) Prediction phase [11].

5.2.1 Machine Learning algorithms in Passban IDS

Passban utilizes ML, specifically a one-class classification approach, to create a model of the normal system behavior and subsequently detect anomalies in incoming network traffic. Hence, Passban concentrates exclusively on learning the typical patterns of network traffic. By recognizing deviations from this established normal behavior, it identifies potential anomalies and flags them for further investigation; this strategy enables Passban to effectively detect emerging threats in real-time network traffic monitoring, without the need for extensive computational resources.

One-class classification algorithms are based on two types of approaches: profiling and isolation.

- Profiling methods learn key features of normal behavior but may inadvertently increase the false positive rate, while also being computationally intensive.
- Isolation methods, on the other hand, are based on the principle of identifying anomalies as points that are sparse and distinct from the majority of the data. They are more computationally efficient and effective.

In the following, both algorithms used by Passban are explored: iForest, an isolation-based algorithm, and *Lcoal Outlier Factor* (LOF), a profiling-based method:

- *Isolation Forest (iForest):*

  is an ensemble method designed to detect anomalies by leveraging the rarity and distinctiveness of anomalous data points. It constructs a forest of random decision trees, recursively partitioning data until instances are isolated. Outliers are identified by the path lengths in the trees: anomalies result in shorter paths, while normal instances have longer paths. Thus, instances with the shortest paths are flagged as anomalies. Figure 5.2 below depicts the iForest algorithm.

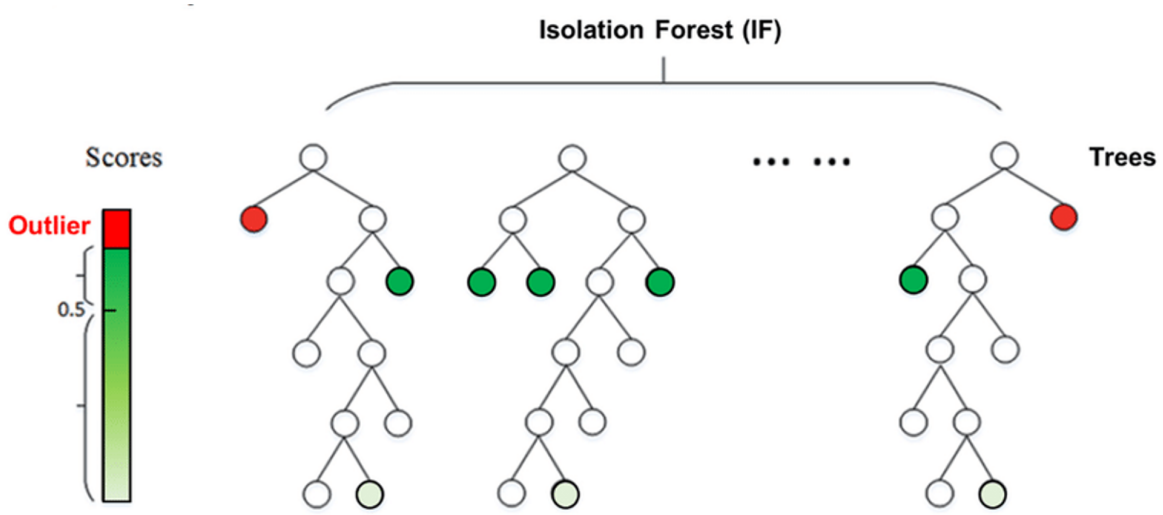

Figure 5.2: Isolation Forest algorithm architecture [18].

- *Local Outlier Factor:*

  is a density-based anomaly detection method that assesses local density by measuring distances between data points and their k-nearest neighbors. Points in dense regions with consistent densities are classified as normal, while those in sparse regions are identified as outliers or anomalies. Due to its reliance on distance measurements, LOF incurs in higher memory usage, processing demands, and computational complexity, making

it less suitable for resource-constrained edge devices. Figure 5.3 illustrates how this algorithm works: point A has a high LOF score because its density is low relative to its neighbors' densities. Dotted circles indicate the distance to each point's third-nearest neighbor whilst the red lines display the k-distance of the object A to the k-th nearest neighbors.

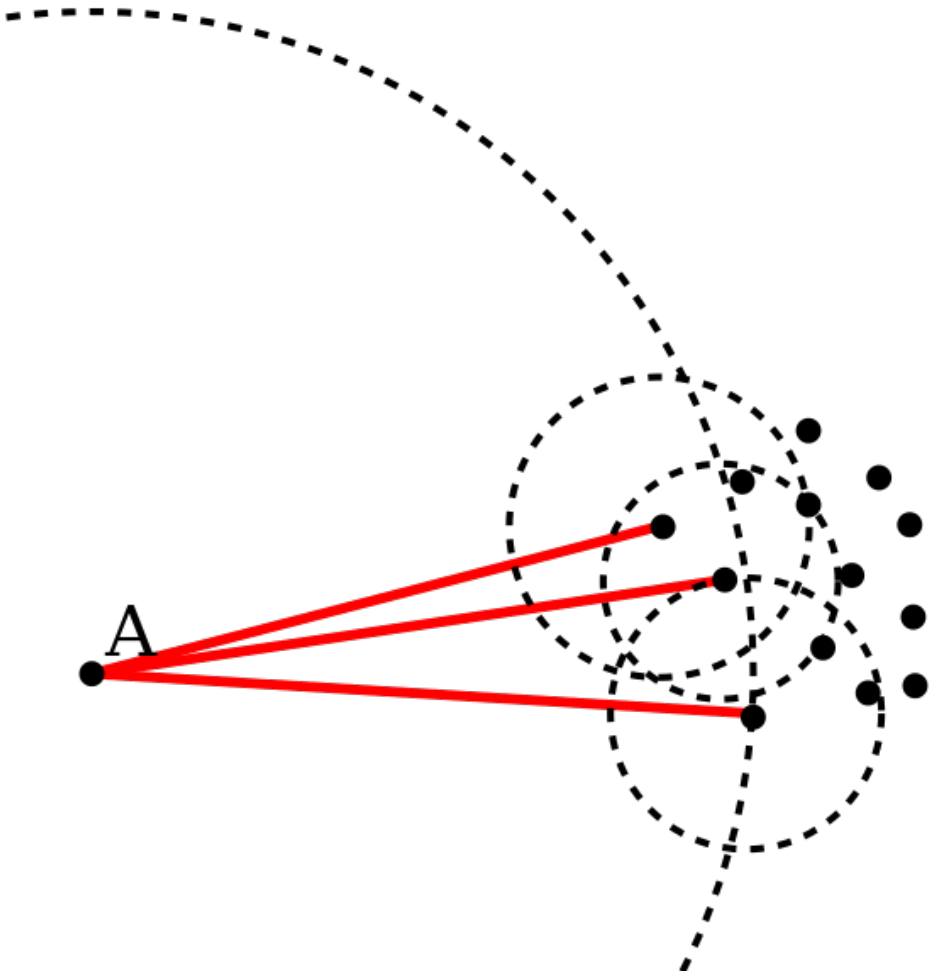

Figure 5.3: Local Outlier Factor basic idea [20].

5.2.2 Training and Prediction Phases

During the training phase (refer to Figure 5.1 (a)), a ML algorithm is trained to learn the normality model hidden in normal data instances. At the end of this phase, the trained model is stored in the local memory of the edge device (e.g., an IoT gateway).

On the other hand, throughout the prediction phase (see 5.1 (b)), the learned model is loaded back from the local memory of the edge device to predict any anomalies in the network traffic. The prediction outcome is a simple "Yes" if an anomaly is spotted, or "No" if everything looks normal. Any detected anomalies are then sent to an AM for further handling.

The training process is outlined in Figure 5.4, described using pseudocode: this algorithm takes the network interface *Ni* as input, collects network flow data, and checks for an existing trained model. If a trained model is found, it is loaded; otherwise, a new training session is initiated. The trained ML model *ML* is then returned as output.

```
Input:  Ni: Network interface;
Output: ML: Trained Machine Learning Object;
1  ML = new MachineLearning();
2  if (Exist trainedModel) then
3  |  ML.loadModel(trainedModel);
4  end
5  else
6  |  ds = new Dataset();
7  |  nf = new NetFlow();
8  |  for (nf in Ni.getNetFlow()) do
9  |  |  nf = Ni.getNetFlow();
10 |  |  ds.add(nf);
11 |  end
12 |  trainedModel = ML.train(ds);
13 |  store(trainedModel);
14 end
15 return ML;
```

Figure 5.4: Passban Initialization Procedure [11].

5.2.3 AM Module

As mentioned in Section 5.2.2, whenever Passban identifies an incoming network flow as an anomaly, the AM module is notified to take proper actions. When this happens, the details of detected attacks are by default logged in the AM. The AM module also provides *Application Programming Interface* (API) callbacks to interact with other applications or devices, enabling actions based on detected anomalies. For example, it can block traffic, notify the network administrator, or turn off critical devices according to a predefined threat alert policy.

5.2.4 Web Manager Interface

The Web manager interface operates on a predefined network port, providing access to the Passban IDS. The interface displays the IDS status and alerts for suspicious network flows. Additionally, it enables users to start or stop the IDS, switch between detection and training modes, and clean detection logs. A screenshot of the Web Manager interface is shown in Figure 5.5.

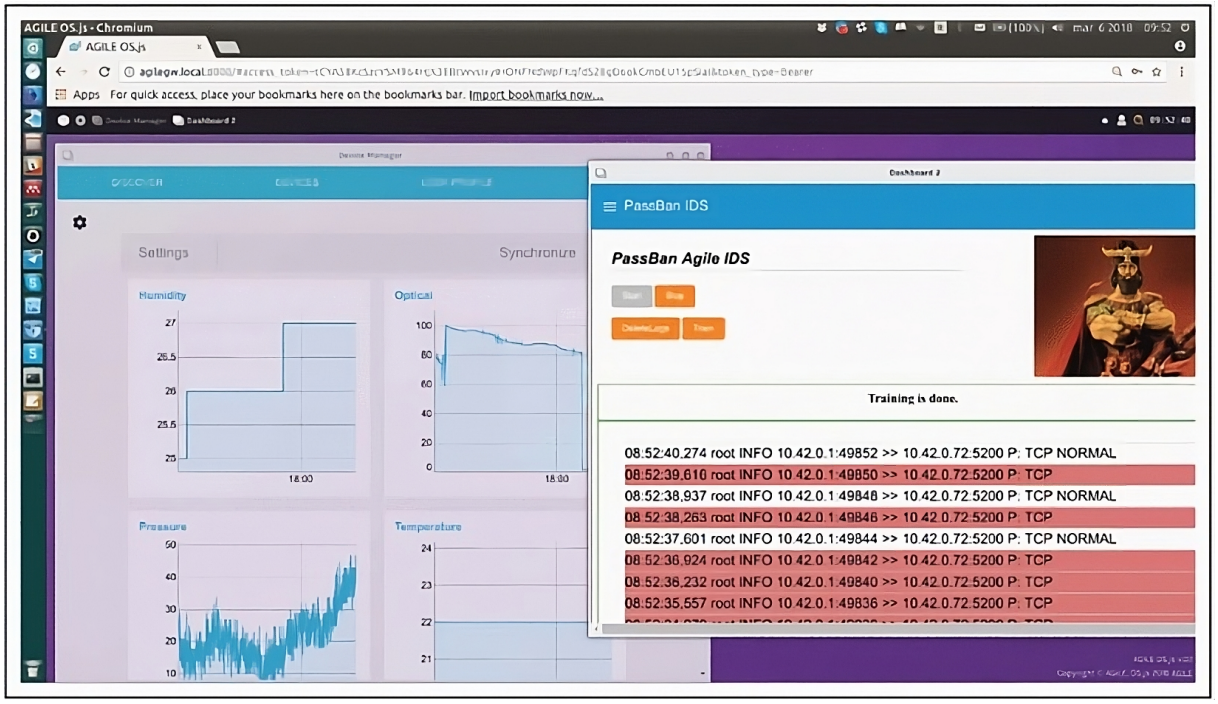

Figure 5.5: The web interface of the Passban IDS is displayed on a popular open-source IoT gateway. Suspicious network flows are in the red rows. [11].

5.3 Passban as an IDS for IoT Gateways

One of the notable projects in open-source hardware for IoT, AGILE, provides a modular hardware and software gateway supporting IoT functionalities such as protocol interoperability, device management, IoT app execution, and cloud communication [13]. To facilitate broader adoption, Passban IDS has been integrated into the AGILE gateway, ensuring compatibility with any IoT gateway, thus achieving platform independence. Figure 5.6 shows the system architecture; Passban is deployed on the IoT gateway, close to the connected IoT devices. As will be discussed next, this proximity is one of Passban's main advantages.

5.3.1 Locality Advantage

Passban captures incoming and outgoing network traffic from IoT devices to build a comprehensive model of normal network conditions (free from attacks or anomalies). For optimal accuracy, placing IDS directly on the IoT devices would be ideal. However, the limited processing power and battery life of these devices pose significant challenges. To address this, Passban is deployed at the IoT gateway, effectively balancing the need for robust threat detection with the resource constraints of IoT devices.

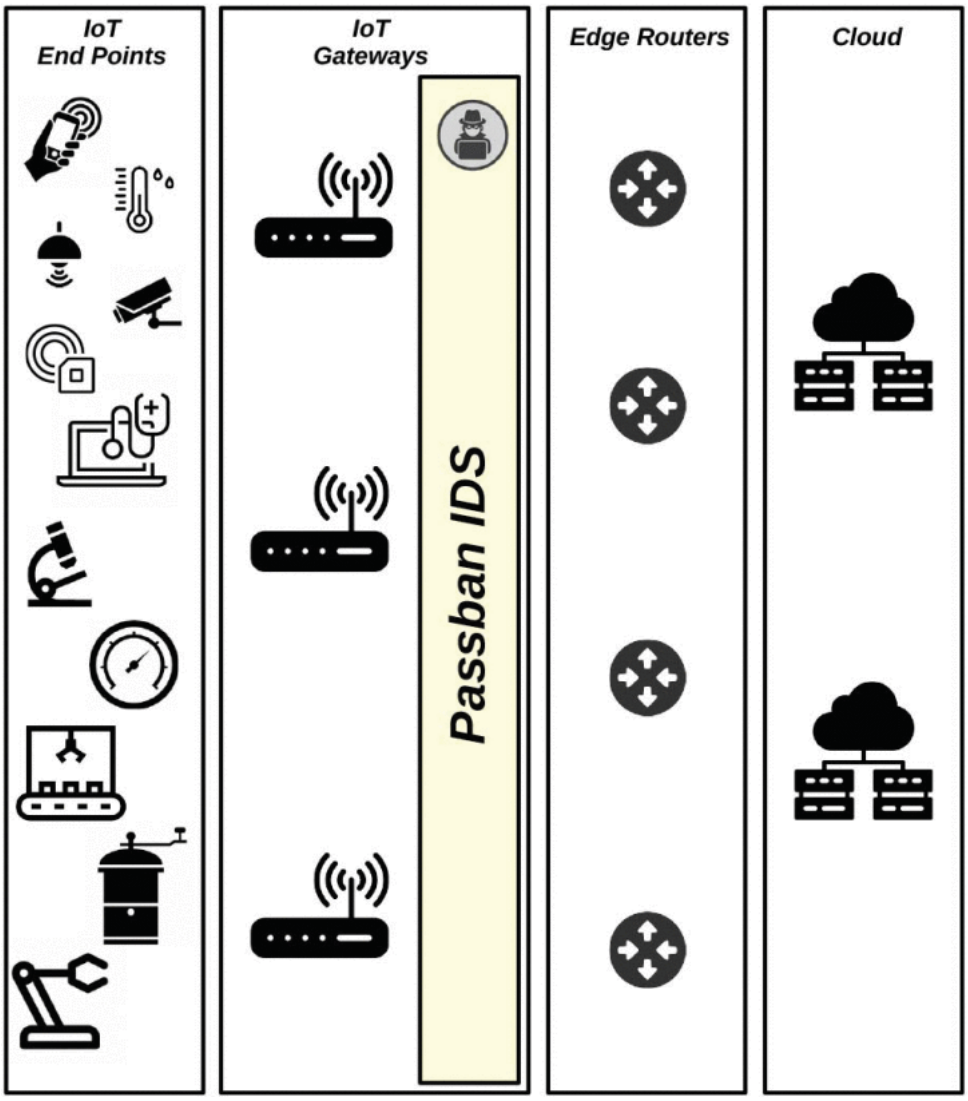

Figure 5.6: Block diagram of the system architecture [11].

5.3.2 Main Limitations

Passban, while being promising for research perspective, lacks full maturity from a development standpoint. Key limitations include its assumption of an attack-free environment during training, potential false positives, the need for frequent retraining due to network changes.

5.4 Real Testbed for Performance Inspection

In order to assess the performance efficiency in terms of threat detection accuracy and computational power required by the proposed IDS, a customized testing environment (i.e., an IoT testbed) is needed- This testbed has been implemented by the authors of the paper, and its details are provided in the following [11].

5.4.1 Testbed Setup and Tools Used

- ***AGILE Gateway Setup:***

  To set up a typical testbed for the home automation scenario, a Raspberry Pi 3 Model B running the AGILE gateway software was utilized. This gateway interacts directly with various IoT devices and a cloud back-end. For the experimental setup, the authors in [11] included two main components:

  1. Texas Instruments BLE SensorTag:
     - *TMP007*: Temperature sensor
     - *BMP280*: Altimeter/Air pressure sensor
     - *OPT3001*: Ambient light sensor
     - *DHC1000*: Humidity sensor
     - *MPU-9250*: 9-axis motion sensor
  2. FosCam FI8910W: WiFi IP Camera

  In Figure 5.7 the architecture of the testbed is shown: Passban IDS monitors all traffic to and from the AGILE gateway.

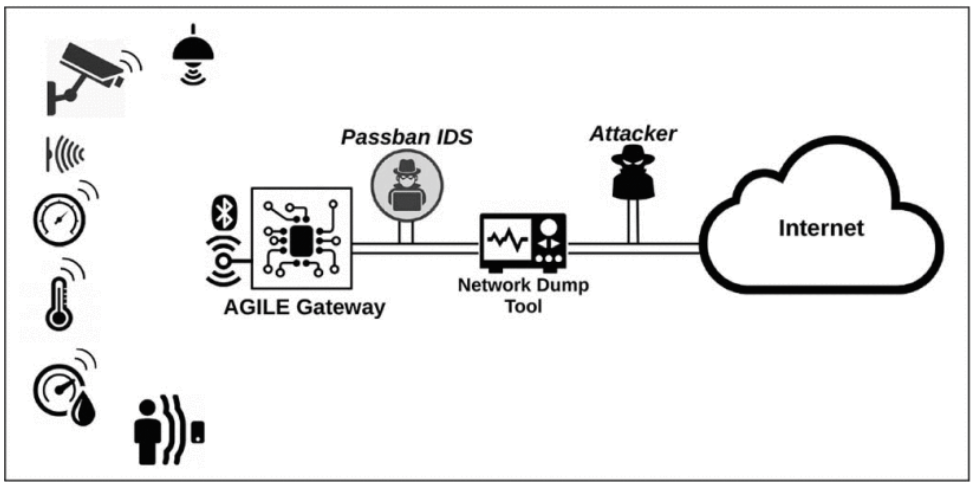

Figure 5.7: Testbed architecture [11].

- ***Communication and Configuration:***

  IoT devices communicate with the AGILE gateway via Bluetooth technology. A *Light Emitting Diode* (LED) actuator on a sensor board was configured to respond to temperature thresholds controlled by the gateway. The gateway was also set to automatically scan for new devices and connect to multiple IP cameras over the local WiFi network, streaming video to a cloud endpoint.

- ***Monitoring and Data Collection:***

  1. *TCPdump*: used as a network dumping tool to capture raw network traffic entering and leaving the AGILE gateway. TCPdump stored this traffic in PCAP files.
  2. *NetMate*: analyzed the PCAP files to extract network statistics and features from the raw network traffic.
  3. *Nmap*: conducted port scanning on the AGILE gateway to identify publicly accessible services. It revealed an open Web server and *Secure Shell* (SSH) port on the gateway.

- ***Attack Simulation:***

  The *Metasploit* framework was configured to execute specific attacks on the AGILE gateway, as reviewed in Section 5.4.2.1

5.4.2 Dataset scenarios

To evaluate Passban's performance, data were gathered by the authors in [11] using two distinct scenarios:

1. In Scenario 1, the tesbed was configured to emulate a typical IoT environment, as illustrated in Figure 5.7.
2. In Scenario 2, Passban was deployed to secure a home automation IoT system.

In this Thesis the focus will be on Scenario 1.

5.4.2.1 Scenario 1

To gather data for this Scenario, the AGILE gateway was configured to interact with an *Internet Protocol* (IP) camera and other sensors, retrieving and sending data to a cloud infrastructure.

During the training phase for Passban IDS, the AGILE gateway operated for 12 hours without any attacks, collecting approximately 17.3 million raw network packets. This data represented the normal operating conditions (benign traffic) of the system. Afterward, four different types of attacks were launched on the AGILE gateway; a brief overview of these attacks is presented below:

- *Port Scanning:*

  allows reconnaissance on the target system to discover potential vulnerable points

- HyperText Transfer Protocol *(HTTP) Brute Force:*

  like most IoT gateways, AGILE offers a web interface for interacting with sensors, applications, and services. This interface, protected by username/password credentials, is a potential attack vector for intruders attempting to brute-force a dictionary of credentials to gain unauthorized access.

- *SSH Brute Force:*

  the SSH protocol opens up vulnerabilities where an intruder can attempt to gain unauthorized access to the gateway via SSH.

- *SYN Flood:*

  is a DoS attack where an attacker floods the target system with SYN (synchronize) requests, exhausting server resources and making it unresponsive to legitimate traffic.

Table 5.1 details the number of attack versus benign flows.

Table 5.1: Summary of normal and attack Flows [11].

Attack Name	**#Normal Flows**	**#Attack Flows**
Normal Traffic	3761	0
Port Scanning	148	57
HTTP Brute Force	106	36
SSH Brute Force	870	389
SYN Flood	117	31

5.5 Machine-Learning-Based Performance Evaluation and Analysis

To assess Passban's ML performance, the following metrics can be emploted: *True Positive*s (TPs) for correctly detected attacks, *True Negative*s (TNs) for accurately identified normal flows, FPs indicating incorrect attack detections, and *False Negative*s (FNs) representing missed attack detections. Typically, these terms are aggregated to compute precision (P), recall (R), and their harmonic mean, known as the F1 score.

Precision measures the proportion of TPs results among all positive predictions made by the model. Recall, on the other hand, measures the proportion of TPs that were correctly identified by the model out of all actual positives in the dataset. The F1 score combines both

precision and recall into a single metric, providing a balanced measure of a model's performance in classification tasks.

The formulas for these metrics are provided in Table 5.2.

Table 5.2: Performance Measures [11].

Performance Measure	**Formula**
Precision (P)	$P = \frac{TP}{TP+FP}$
Recall (R)	$R = \frac{TP}{TP+FN}$
F-Measure (F1)	$F1 = 2 \times \frac{P \cdot R}{P+R}$

Table 5.3 summarizes the main performance indicators computed for Scenario 1, grouped by type of attack.

Attack	**Technique**	**#Normal**	**#Attack**	**FP**	**TP**	**FN**	**TN**	**Precision**	**Recall**	**F1**
Port Scanning	iForest	148	57	1	57	0	147	0.98	1	0.99
	LOF	148	57	10	52	5	138	0.84	0.91	0.87
HTTP Brute Force	iForest	106	36	2	35	1	104	0.95	0.97	0.96
	LOF	106	36	7	35	1	99	0.83	0.97	0.89
SSH Brute Force	iForest	870	389	9	370	19	861	0.98	0.95	0.96
	LOF	870	389	7	302	87	863	0.98	0.78	0.87
SYN Flood	iForest	117	31	2	27	4	115	0.93	0.87	0.9
	LOF	117	31	5	27	4	112	0.84	0.87	0.85

Table 5.3: Attack Detection Metrics [11].

In this Scenario, both LOF and iForest effectively detected all tested attacks with iForest consistently achieving superior precision, recall, and F1 scores across all attack types.

Chapter 6

Conclusion

This Thesis has explored the IoT with a specific focus on the security challenges associated with deploying IDSs that utilize anomaly detection techniques. The Document has provided a comprehensive analysis of IoT's key characteristics, structure, applications, and the associated security concerns that arise from its widespread adoption. Emphasizing the importance of the CIA security model, the various layers of the IoT network and the specific security challenges inherent to each were examined.

Through a detailed review of IDS, particularly those leveraging ML and DL for anomaly detection, the advantages these approaches offer in detecting unknown threats and adapting to evolving network traffic patterns were discussed. The case study on Passban IDS illustrated the practical application of ML algorithms in protecting IoT edge devices, highlighting both the potential and the limitations of IDSs approaches in terms of scalability and robustness.

This Thesis aims to identify the primary security challenges in IoT, show the efficacy of existing anomaly-based IDS and discuss solutions that could be further developed. While substantial progress has been made, the findings illustrated in the Thesis indicate that IoT security remains a dynamic and evolving field that requires ongoing innovation. Future research should focus on optimizing these detection systems for large-scale deployment, integrating them with advanced encryption methods, and developing resilient network architectures.

Acronimi

AI *Artificial Intelligence*

IoT *Internet of Things*

RFID *Radio-Frequency IDentification*

M2M *Machine to Machine*

MQTT *Message Queuing Telemetry Transport*

CoAP *Constrained Application Protocol*

LCD *Liquid-Crystal Display*

GPS *Global Positioning System*

DoS *Denial-of-Service*

DDoS *Distributed Denial-of-Service*

CIA *Confidentiality, Integrity, Availability*

GDPR *General Data Protection Regulation*

MITM *Man-in-the-middle*

IDS *Intrusion Detection System*

SVM *Support vector machine*

PCA *Principal Component Analysis*

ANN *Artificial Neural Network*

ML *Machine Learning*

DL *Deep Learning*

ReLU *Rectified Linear Unit*

CNN *Convolutional Neural Network*

RNN *Recurrent Neural Network*

FP *False Positive*

LOF *Lcoal Outlier Factor*

AM *Action Manager*

API *Application Programming Interface*

LED *Light Emitting Diode*

TP *True Positive*

TN *True Negative*

FP *False Positive*

FN *False Negative*

HTTP *HyperText Transfer Protocol*

SSH *Secure Shell*

IP *Internet Protocol*

Bibliography

[1] Mian Muhammad Ahemd, Munam Ali Shah, and Abdul Wahid. *IoT security: A layered approach for attacks & defenses*. 2017. URL: https://ieeexplore.ieee.org/abstract/document/8065757.

[2] Vladislavs Aleksandrovičs, Eduards Filičevs, and Jānis Kampars. Internet of things: Structure, features and management. *Information Technology and Management Science*, 2016. URL: https://www.researchgate.net/publication/312567480InternetofThingsStructureFeaturesandManagement.

[3] Zainab Ali, Hesham Ali, and Mahmoud Badawy. Internet of things (iot): Definitions, challenges, and recent research directions. *International Journal of Computer Applications*, 2015. URL: https://www.researchgate.net/publication/320532203InternetofThingsIoTDefinitionsChallengesandRecentResearchDirections.

[4] Dr Anitha and L Arockiam. A review on intrusion detection systems to secure iot networks. 2022. URL: https://www.researchgate.net/publication/358903901AReviewonIntrusionDetectionSystemstoSecureIoTNetworks/citation/download.

[5] Kevin Ashton. That 'internet of things' thing. *RFID-JOURNAL*. URL: https://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf.

[6] Avast. Avast security report. Technical report, 2019. URL: https://cdn2.hubspot.net/hubfs/486579/avastsmarthomereportfeb2019.pdf.

[7] Sweta Bhattacharya, Siva Rama Krishnan S, Praveen Kumar Reddy Maddikunta, Rajesh Kaluri, Saurabh Singh, Thippa Reddy Gadekallu, Mamoun Alazab, and Usman Tariq. A novel pca-firefly based xgboost classification model for intrusion detection in networks

using gpu. *Electronics*, 9(2), 2020. URL: https://www.mdpi.com/2079-9292/9/2/219, doi:10.3390/electronics9020219.

[8] Muhammad Burhan, Rana Asif Rehman, Bilal Khan, and Byung-Seo Kim. Iot elements, layered architectures and security issues: A comprehensive survey. *Sensors*. URL: https://www.mdpi.com/1424-8220/18/9/2796.

[9] Shanzhi Chen, Hui Xu, Dake Liu, Bo Hu, and Hucheng Wang. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things Journal*. URL: https://ieeexplore.ieee.org/abstract/document/6851114.

[10] Priya Das and Sohail Saif. *Intrusion Detection in IoT-Based Healthcare Using ML and DL Approaches: A Case Study*, pages 271–294. Springer Nature Singapore, Singapore, 2023. doi:10.1007/978-981-99-2115-712.

[11] Mojtaba Eskandari, Zaffar Haider Janjua, Massimo Vecchio, and Fabio Antonelli. Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices. *IEEE Internet of Things Journal*, 7(8):6882–6897, 2020. doi:10.1109/JIOT.2020.2970501.

[12] INC. Fortinet. Cia triad. URL: https://www.fortinet.com/resources/cyberglossary/cia-triad.

[13] Eclipse Foundation. Agile-iot. URL: https://www.eclipse.org/research/projects/agile/.

[14] Mario Frustaci, Pasquale Pace, Gianluca Aloi, and Giancarlo Fortino. Evaluating critical security issues of the iot world: Present and future challenges. *IEEE Internet of Things Journal*, 2018. URL: https://ieeexplore.ieee.org/abstract/document/8086136.

[15] Octavia Georgiana Dorobantu and Simona Halunga. Security threats in iot. In *2020 International Symposium on Electronics and Telecommunications (ISETC)*, 2020. URL: https://ieeexplore.ieee.org/document/9301127.

[16] A. Holst. Iot connected devices worldwide 2019–2030. URL: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

[17] Irdeto. New 2019 global survey: Iot-focused cyberattacks are the new normal. 2019. URL: https://resources.irdeto.com/global-connected-industries-cybersecurity-survey/new-2019-global-survey-iot-focused-cyberattacks-are-the-new-normal.

[18] Chunggil Jung, Yong-Gwan Lee, Ji-Wan Lee, and Seongjoon Kim. Performance evaluation of the multiple quantile regression model for estimating spatial soil moisture after filtering soil moisture outliers. *Remote Sensing*, 12:1678, 05 2020. doi:10.3390/rs12101678.

[19] Ashvini Kamble and Sonali Bhutad. *Survey on Internet of Things (IoT) security issues & solutions*. 2018. URL: https://ieeexplore.ieee.org/abstract/document/8399084.

[20] Dimitrios Kokkinopoulos. *Data-driven Wind Turbine Performance Analysis*. PhD thesis, 07 2020. doi:10.13140/RG.2.2.10066.99528.

[21] I Lee. Internet of things (iot) cybersecurity: Literature review and iot cyber risk management. *Sensors*, 2020. URL: https://www.mdpi.com/1999-5903/12/9/157#metrics.

[22] Rwan Mahmoud, Tasneem Yousuf, Fadi Aloul, and Imran Zualkernan. *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2015. URL: https://ieeexplore.ieee.org/document/7412116.

[23] Periasamy Nancy, S. Muthurajkumar, S. Ganapathy, S.V.N. Santhosh Kumar, M. Selvi, and Kannan Arputharaj. Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks. *IET Communications*, 14(5):888–895, 2020. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-com.2019.0172, arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-com.2019.0172, doi:10.1049/iet-com.2019.0172.

[24] Keyur Patel, Sunil Patel, P Scholar, and Carlos Salazar. Internet of things-iot: Definition, characteristics, architecture, enabling technologies, application & future challenges. URL: https://www.researchgate.net/publication/330425585InternetofThings-IOTDefinitionCharacteristicsArchitectureEnablingTechnologiesApplicationFutureChallenges.

[25] Ani Petrosyan. Annual number of internet of things (iot) malware attacks worldwide from 2018 to 2022. URL: https://www.statista.com/statistics/1377569/worldwide-annual-internet-of-things-attacks/#:~:text=Global%20annual%20number%20of%20IoT%20cyber%20attacks%202018%2D2022&text=The%20number%20of%20Internet%20of,million%20detected%20cases%20in%202018.

[26] PwC. Managing emerging risks from the internet of things. 2016. URL: https://www.pwc.com/us/en/services/consulting/cybersecurity/library/broader-perspectives/managing-iot-risks.html.

[27] Mouha R. Internet of things (iot). *Journal of Data Analysis and Information Processing*, 2021. URL: https://www.scirp.org/journal/paperinformation?paperid=108574.

[28] Tariq Rao and Ehsan Haq. Security challenges facing iot layers and its protective measures. *International Journal of Computer Applications*, 2018. URL: https://www.researchgate.net/publication/323892938SecurityChallengesFacingIoTLayersanditsProtectiveMeasures.

[29] B. Riyaz and Ganapathy Sannasi. A deep learning approach for effective intrusion detection in wireless networks using cnn. *Soft Computing*, 24, 11 2020. doi:10.1007/s00500-020-05017-0.

[30] Mukaram Safaldin, Mohammed A. Otair, and Laith Mohammad Abualigah. Improved binary gray wolf optimizer and svm for intrusion detection system in wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 12:1559 – 1576, 2020. URL: https://api.semanticscholar.org/CorpusID:220503103.

[31] Leonel Santos, Carlos Rabadao, and Ramiro Gonçalves. Intrusion detection systems in internet of things: A literature review. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018. doi:10.23919/CISTI.2018.8399291.

[32] Eryk Schiller, Andy Aidoo, Jara Fuhrer, Jonathan Stahl, Michael Ziörjen, and Burkhard Stiller. Landscape of iot security. *Computer Science Review*, 2022. URL: https://www.sciencedirect.com/science/article/pii/S1574013722000120.

[33] Aliya Tabassum, Aiman Erbad, and Mohsen Guizani. A survey on recent approaches in intrusion detection system in iots. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019. URL: https://ieeexplore.ieee.org/abstract/document/8766455.

[34] Hatim Mohamad Tahir, Abas Md Said, Nor Hayani Osman, Nur Haryani Zakaria, Puteri Nurul 'Ain M Sabri, and Norliza Katuk. Oving k-means clustering using discretization technique in network intrusion detection system. In *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, pages 248–252, 2016. doi:10.1109/ICCOINS.2016.7783222.

[35] R Vinayakumar, K P Soman, and Prabaharan Poornachandran. Applying convolutional neural network for network intrusion detection. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1222–1228, 2017. doi:10.1109/ICACCI.2017.8126009.

[36] Felix Wortmann and Kristina Flüchter. *Business & Information Systems Engineering*, 2015. URL: https://www.researchgate.net/publication/276439592InternetofThings.

[37] Juan Yang, Zhiwei Ye, Lingyu Yan, Wei Gu, and Ruoxi Wang. Modified naive bayes algorithm for network intrusion detection based on artificial bee colony algorithm. In *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pages 35–40, 2018. doi:10.1109/IDAACS-SWS.2018.8525758.

[38] Min Yang and Jiajie Zhang. Data anomaly detection in the internet of things: A review of current trends and research challenges. *International Journal of Advanced Computer Science and Applications*, 14, 01 2023. doi:10.14569/IJACSA.2023.0140901.

[39] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017. doi:10.1109/ACCESS.2017.2762418.

[40] Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, and Sean Carlisto de Alvarenga. A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, 84, 2017. URL: https://www.sciencedirect.com/science/article/pii/S1084804517300802.