



REPUBLIC OF TURKEY

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

**INTRUSION DETECTION SYSTEM IN IOT
NETWORKS USING MACHINE LEARNING**

Alhasan Kadhim RESEN

Master's Thesis

Supervisor

Asst. Prof. Dr. Ayca KURNAZ

Istanbul, 2022

INTRUSION DETECTION SYSTEM IN IOT NETWORKS USING MACHINE LEARNING

Alhasan Kadhim Resen RESEN

Electrical and Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled INTRUSION DETECTION SYSTEM IN IOT NETWORKS USING MACHINE LEARNING prepared by ALHASAN KADHIM RESEN and submitted on 20\5/2022 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering

Asst. Prof. Dr. Ayca Kurnaz TURKBEN
Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Ayca Kurnaz TURKBEN

Faculty Of Software
Engineering,

Altinbas University

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Faculty of Engineering
and Natural Sciences,

Altinbas University

Asst. Prof. Dr. Serdar KARGIN

Faculty of Biomedical
Engineering,

Beykent University

I hereby declare that this thesis meets all format and submission requirements of a Master`s Thesis.

Submission date of the thesis to Institute of Graduate Studies: ____/____/____

To the best of my knowledge, the material included in this work is correct. This is in accordance with academic standards and ethical behavior. Upon completion of this work, I certify that all materials and discoveries that are not original to this work have been correctly cited and referenced, as required by these standards of conduct.

Alhasan Kadhim RESEN

Signature

DEDICATION

I devote and pledge this research work to my supervisor who is salient for guiding me through the whole research work as well as my family for always assisting me in my hard time.



PREFACE

First and foremost, I would like to thank my supervisor Asst. Prof. Ayça kurnaz turkben for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Asst. Prof. Ayça kurnaz turkben a couple of times every week helped me tremendously in understanding the logic behind the research. It made me better realize the technical need for this research work.



ABSTRACT

INTRUSION DETECTION SYSTEM IN IOT NETWORKS USING MACHINE LEARNING

RESEN, Alhasan

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Ayça KURNAZ

Date: 05/2022

Pages: 62

When we talk about the Internet of Things (IoT), we are referring to any physical object that has been integrated with technological capabilities which let them acquire and distribute information over the Internet. Individuals are at risk of cyber-attacks as a result of the expanding number of security vulnerabilities linked with the Internet of Things, as well as the rapid proliferation of IoT devices in residential settings. Designing a security solution for Internet of Things devices is difficult due to the low power and processing capabilities of these devices. In this paper, we propose use of IDS in the process of developing a data flow classification approach that allows identification of common threats in IOT networks, we utilize CNN and RNN algorithms with simple modifications for further accuracy.

Keywords: ML, CNN, RNN, IOT, IDS.

TABLE OF CONTENTS

	<u>Pages</u>
ABSTRACT	vii
LIST OF FIGURES.....	xi
LIST OF ABBREVIATIONS	xiii
1. INTRODUCTION.....	1
1.1 PROBLEM STATEMENT	4
1.2 THESIS OBJECTIVES.....	5
1.3 CONTRIBUTION.....	6
1.4 THESIS ORGANIZATION.....	7
2. LITRITAURE REVIEW.....	8
2.1 GENERAL INTRODUCTION.....	8
2.2 LITERATURE REVIEW.....	8
2.2.1 Ids Placement Strategy:.....	9
2.3 CONCLUSION.....	14
3. MATERIALS AND METHODS	16
3.1 THE INTERNET OF CONNECTED OBJECTS IN HOMES	16
3.1.1 Home Automation	17
3.1.2 Communication Protocols	18
3.1.3 Short-Range Communication Protocols.....	19
3.1.4 Connected Home Uses	22
3.1.5 The Architecture Of Connected Homes	22
3.1.6 Security Risks Associated with Connected Homes.....	23
3.2 INTRUSION DETECTION SYSTEM	26

3.2.1	Detection by Rule.....	29
3.2.2	Anomaly Detection	31
4.	PROPOSED METHOD.....	33
4.1	CHAPTER INTRODUCTION	33
4.2	DATASET USED	33
4.3	CNN ARCHITECHTURE	36
4.4	TRAINING AND SIMULATION	39
4.5	RESULTS	41
5.	CONCLUSION AND FUTURE SCOPE	47
5.1	CONCLUSION	47
5.2	FUTURE WORK	48
	REFERENCES	50

LIST OF TABLES

	<u>Pages</u>
Table 2.1: application layers of the Internet of Things' architecture	14
Table 4.1: KDD dataset features and its corresponding types.....	34



LIST OF FIGURES

	<u>Pages</u>
Figure 1.1: IDS methods in IOT	3
Figure 1.2: General view of the proposed scheme	6
Figure 3.1: Internet of connected home objects.....	16
Figure 3.2: Smart homes security statistics by 2020	18
Figure 3.3: Diagram of Sigfox protocol stack	19
Figure 3.4: Layers of Zigbee vs layers of 6LoWPAN.....	21
Figure 3.5: A. OVH DNS firewall.....	26
Figure 3.6: HIDS in IO	28
Figure 3.7: Detection by rule in IOT based IDS	31
Figure 4.1: Data preprocessing step before training of the network.....	36
Figure 4.2: Where (a) is the standard CNN net) and (b) is the proposed RNN-CNN modification to the network	37
Figure 4.3: Classes and weights in the network inputs [22]	40
Figure 4.4: CNN layers for classification [23]	40
Figure 4.5: CNN framework [15]	41
Figure 4.6: Simulation results after the ADAM optimizer	43
Figure 4.7: CNN training time and accuracy.....	44
Figure 4.8: CNN accuracy and loss	46

Figure 4.9: Comparing accuracies and loss in CNN, SVM and KN	46
--	----



ABBREVIATIONS

AI : Artificial Intelligence

IoT : Internet of Things

PAN : Personal Area Network

FSK : Frequency Shift Keying

TDMA : Time Division Multiple Access

OFDM : Orthogonal Frequency Division Multiplexing

1. INTRODUCTION

When we speak to the Internet of Things (IoT), we mean any physical device that has been endowed with technical capabilities that allow it to receive and transmit data through the Internet. Individuals are at danger of cyber-attacks as a consequence of the fast adoption of Internet of Things devices in residential settings and the growing number of security vulnerabilities associated with IoT. Due to their small size and poor computational capacity, Internet of Things (IoT) devices constitute a significant security risk [28]. Intrusion detection systems (IDS) may be utilized to aid in cyberattack prevention (IDS). In general, there are two types of IDS: those that look for signs of abuse and those that look for abnormal behaviors. When it comes to distinguishing new attacks from previously found ones, it is necessary for misuse-based filters to create a new attack signature. When it comes to anomaly-based detection, which might unearth new risks by recognizing abnormal traffic, one of the most difficult aspects is preventing a high rate of false positives. This is particularly challenging when dealing with massive volumes of data. Additionally, since the great majority of Internet of Things devices are confined by storage, memory, and computing capabilities, developing an IDS is difficult [30]. N. Chaabouni and colleagues [2] performed study on network intrusion detection systems' operational tools and datasets (NIDS). They are primarily interested with machine learning-based NIDS for the Internet of Things (IoT) (ML). The researchers recognized many research obstacles and future research objectives in the area of Internet of Things device intrusion detection. The capability to discover unknown and zero-day assaults, as well as the usage of incremental machine learning to generate new models, were mentioned in this list of IoT dataset purposes. The participants highlighted that there was no publicly available benchmark dataset, and that success required a focus on developing an IDS capable of identifying both known and emerging threats. A. Thakkar and R. Lohiya [32] study the use of machine learning (ML) and deep learning (DL) to develop intrusion detection system (IDS) models for the purpose of securing the Internet of Things (IoT) network, and they identify a number of challenges and problems that must be addressed. According to researchers' results, currently existing IoT detection systems are incapable of detecting a broad spectrum of assaults. Additional study is needed to find variants of known and unknown

assaults, as well as to construct a model capable of discriminating between them. Because there is no standard dataset for the Internet of Things, this adds another hurdle to overcome. According to a paper published by J. Asharf and colleagues [1], it is difficult to construct an Internet of Things intrusion detection system (IDS) capable of guarding against all threats owing to a paucity of real-world IoT data. Examples include the creation of a representative dataset to quantify validation, the coupling of deep convolutional neural networks (DLN) with reinforcement learning to cope with very high data dimensionality, and the reduction of false positive and negative findings. Due to the breadth of conceivable actions, it is difficult to address these problems effectively. According to A. Khraisat and A. Alazab's new study [12], intrusion detection systems (IDSs) must improve their accuracy and detect a greater range of intrusions with fewer false alarms and other challenges. According to them, intrusion detection systems (IDSs) must improve their accuracy and detect a larger range of threats while producing fewer false alarms and other complications. They believe that an efficient intrusion detection system requires a focus on four critical characteristics. Additionally, it is an excellent fit for Internet of Things communication networks due to its ability to identify zero-day threats and low false alarm rates. N. Koroniotis and colleagues [15] produced a new botnet dataset in response to the absence of a representative dataset for the Internet of Things. The dataset included both legal and simulated network traffic, as well as a variety of assaults. The researchers utilized the gathered data to train and assess three different artificial neural networks, which were then evaluated. The researchers built the dataset by simulating real-world network traffic using their own Vertices tool. They then developed scripts to simulate traffic on MQTT-enabled Internet of Things devices that had been joined to the network after their first connection. The attacks were timed to coincide with the regular level of traffic during the period when the Ostinato campaign was in full swing. They used the Argus program to construct network flows using the pcap data they had previously obtained and saved. Additionally, the R2 and Joint Probability ratings aided the researchers in establishing the dataset's top ten qualities. The dataset may be utilized to train RNN and LSTM models for high-precision classification, and the accuracy rose when the models were trained using the top ten most correct features. Additionally, it was revealed that although SVM had the

maximum accuracy and recall when all features in the dataset were used, it had the highest precision and the least amount of fall-out when just the top ten features were employed.

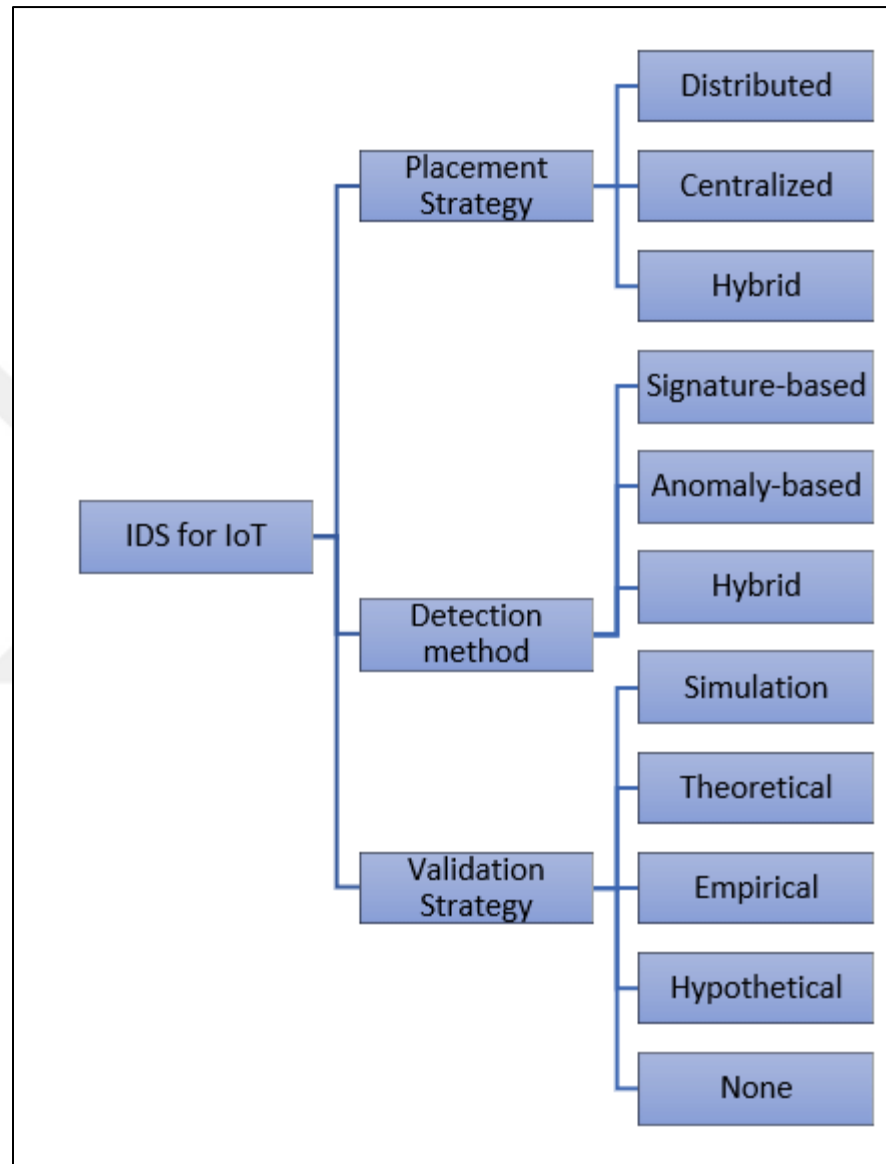


Figure 1.1: IDS methods in IOT

1.1 PROBLEM STATEMENT

The most significant impediment to the full realization of the Internet of Things (IoT) idea is cybersecurity. When opposed to traditional networks, the Internet of Things (IoT) networks lack well-established and standardized design paradigms, such as the client-server architecture. Consequently, a large variety of standard security measures cannot be implemented in Internet of Things (IoT) networks due to this constraint. IoT devices can only supply a few megahertz of processing power, 5-10 megabytes of volatile storage, and almost no permanent storage, which is insufficient for their purposes. Because they rely on batteries for power, their operating duration might range from six to twenty-four hours. Devices connected to the internet are also prevented from using resource-intensive security solutions such as Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and Security Information and Event Management (SIEM). In addition, the growth of Internet of Things (IoT) devices is a major source of worry. By 2020, it is predicted that the number of internet-connected devices will have almost tripled to 20.8 billion [35]. IoT is increasingly becoming a lucrative platform for a diverse range of Internet assaults that may take a variety of shapes and target a variety of resources on a variety of IoT devices. As the number of IoT devices continues to expand, so does the profitability of IoT attacks. The task of keeping track of such a massive number of devices, much alone protect and patch them, would be almost impossible. It is estimated that huge organizations provide the great bulk of Internet of Things (IoT) services, with many of these firms developing their own proprietary IoT communication protocols. Open-source security researchers are unable to access many commercially available Internet of Things (IoT) devices as a consequence of this restriction. Because no two connected devices in an IoT network may utilize the same communication protocol, it is challenging for a security solution to identify a command and control (C&C) server in this heterogeneous network environment. The ineffectiveness of centralized intrusion detection systems (IDS) is exacerbated by peer-to-peer heterogeneous Internet of Things networks. If an intruder's behavior fits within a set of predetermined parameters, the defender will attempt to match the signatures of network traffic to the sets of signatures in an effort to identify the intruder. Researchers have created a variety of rules/signatures-based intrusion detection

approaches [18,] to identify intrusions based on signatures and patterns associated with network traffic in order to identify intrusions. Cybercrime, on the other hand, is constantly evolving and may be very flexible, making it tough to stay up. An attacker can access a command and control (C&C) point from a remote location by employing well-known communication protocols such as Bluetooth, Wi-Fi, and others, and exploit network architectures ranging from more centralized wireless sensor networks to peer-to-peer Internet of Things networks, among other things. [4]. Therefore, a new generation of intrusion detection systems is needed, capable of identifying anomalous activity in Internet of Things networks with pinpoint precision. Due to the fact that it is independent of the kind of IoT device, communication protocol, or network structure being utilized, network topology changes do not pose a difficulty for such an intrusion detection system.

1.2 THESIS OBJECTIVES

The main objective of using this use of IDS is the process of developing a flow approach that allows identifying the way to identify the main objective of the process of developing a network that makes the objective more assertive to identify the main events of development of a network. a network of objectives, for more assertiveness, with the focus of work of the events from real scenarios. The main characteristic of the proposed approach is the ability, using exclusively from the flow, to carry out a detection of even malicious damage in an entire observed network, through the analysis of its segments, with a focus directed to events. To fulfill this objective, the following objectives were achieved:

- a) Through the analysis of attack techniques and sub-techniques: the attack on the MITER ATT&CK framework, whose detection process is performed using data from the network flow.
- b) The main attributes mentioned in the literature were mapped to identify the works mapped from the framework.
- c) A flow base, Packet tracer, was developed from data exported from the virtual network through the IOT protocol, which contains a classification of attacks mapped;

- d) a proposal to evaluate the efficiency of the efficiency of the DS, or an evaluation proposal for the evaluation of a construction strategy with our team, that is, the approach of evaluating an efficiency strategy to our work, that is, the proposal of evaluation in its efficiency not carried out in its analysis carried out in its analysis.
- e) A comparative analysis was carried out on the bases of data flows

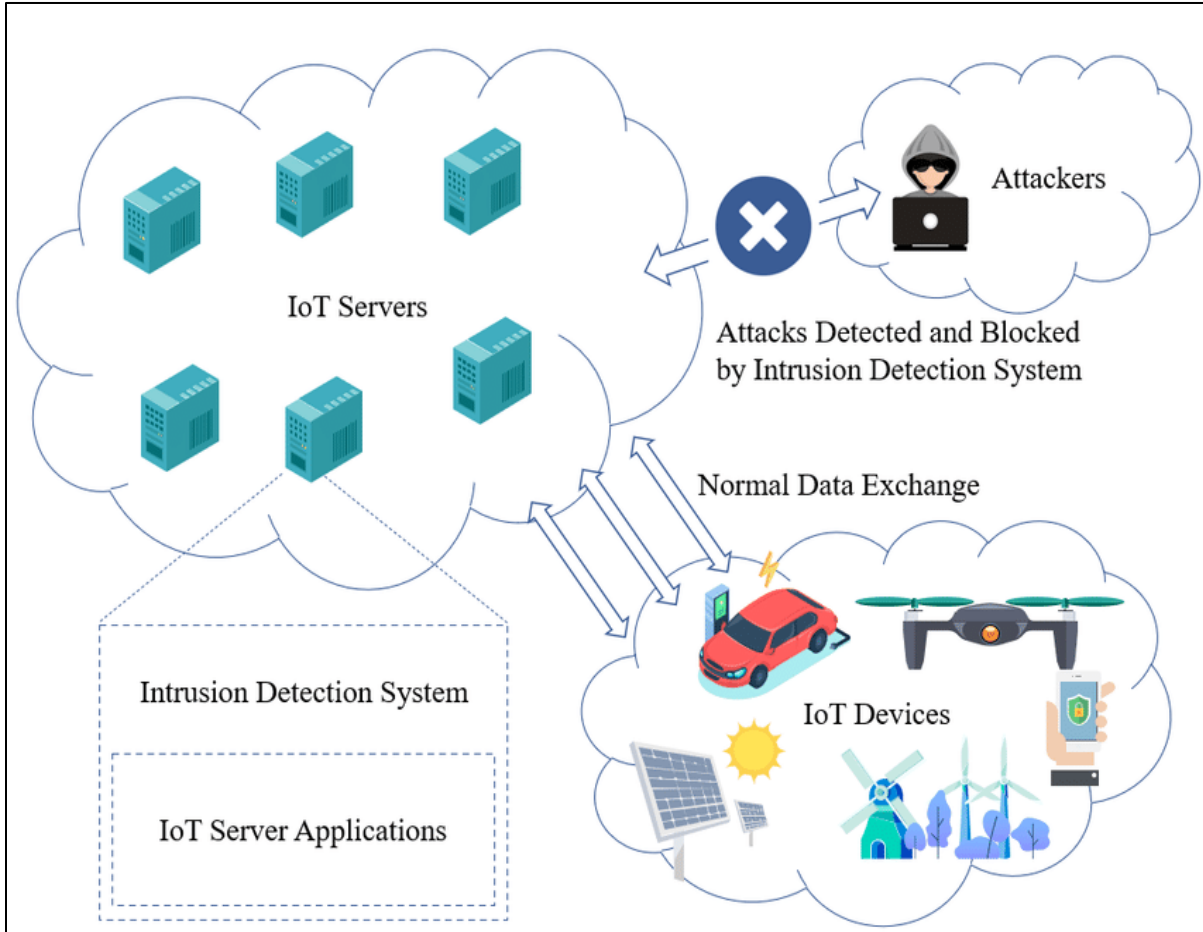


Figure 1.2: General view of the proposed scheme

1.3 CONTRIBUTION

The development and improvement of flow-based IDSs has shown to be a promising field of study, given that, unlike IDSs that perform analyzes considering the payload, the analyzes performed on the flow are not subject to information obscurity, as is the case when the traffic to be analyzed is encrypted - a feature present in networks and systems that increasingly value

security. Another factor that favors the use of flow based IDSs is the non-violation of access privacy. Unlike IDSs that perform the analysis of the content being transmitted and, consequently, lead to the violation of user privacy, flow based IDSs only analyze metadata that describe the traffic between client and server, thus, avoiding the need to know what is happening. being consumed by the user, and in line with legislation that values privacy in the use of the network During the survey of studies related to this work, now directed to the development of flow based IDSs, it was noted a certain scarcity in the exploration of methods that are capable of evaluating the efficiency of these systems regarding the completeness of the analyzes carried out in heterogeneous networks. Approaches to this type of analysis are presented However, the cited studies make use of data different from those intrinsic to the network flow and, in turn, expand the spectrum of information analyzed by including activity records from each device on the network, session information, and other data extracted from the payload. that is, data that does not strictly compose the network flow. Thus, although they have presented results considered good, with accuracy above ninety percent, the aforementioned works carry out a process of training and validation of IDSs based on flow, although they do not use information that is unique and strictly present in it. Therefore, the present work is justified in view of the opening for the study of an IDS evaluation approach that, in addition to considering only data present in the network flow to evaluate its efficiency, proposes the use of the network segmentation strategy as a means to obtain better observability of the elements that compose it and, in this way, obtain improvements in the detection of known attacks.

1.4 THESIS ORGANIZATION

The thesis is structured as follows: Section 2 is where we review some of the previous work and implementation on smart grids. Section 3 is where we give a background about all the components. Section 4 is where we explain our model in details and implementation of that model in details, Section 5 is where we simulate our model and record the results obtained. Section 6 is where we conclude our work and put a future scope into perspective.

2. LITRITAURE REVIEW

2.1 GENERAL INTRODUCTION

This chapter is a critical review of the literature highlights the different approaches proposed to detect intrusions as well as different techniques that could be used to achieve better results the Internet has never connected as many computer systems as today, and the number of objects capable of communicating with each other through this network continues to increase every day. This trend even tends to accelerate with the ever-faster addition of new objects that are constantly diversifying. Although fierce competition between object designers brings a lot of innovation, it also forces manufacturers to reduce the development time of their products and therefore to reduce the security test phase of objects placed on the market. Thus, many vulnerabilities are discovered on smart objects and the associated risks are constantly growing for users. It therefore becomes necessary to be able to secure these different objects, the first step being to be able to detect these malicious activities. Although various works have been proposed, many solutions remain ineffective or very object specific.

2.2 LITERATURE REVIEW

The Internet of Things (IoT) is a relatively new technology that has piqued the interest of academics from across the world. Several notable contributions have been made to the integration of this technology into our daily life. As a complex business, it spans everything from healthcare to automobiles to entertainment to industrial equipment and everything in between. The Internet of Objects (IoT) is pervasive, which makes some everyday chores easier to do, improves how humans interact with their environment, and enriches our social interactions with other people and things. This page provides a comprehensive list of intrusion detection systems (IDS) for the Internet of Things, as well as an in-depth examination of IoT security breaches. Security measures for the Internet of Things (IoT) have been analyzed and organized into a number of categories for your convenience.

2.2.1 Ids Placement Strategy:

Distributed Ids Placement Strategy: - Every physical object was placed by the IDS in a scattered IDS placement, according to their specifications. Oh, and colleagues proposed the use of a distributed lightweight IDS for attack detection. Additionally, they advocated making judgments early in the decision-making process in addition to auxiliary shifting. The development of this IDS was motivated by the need to detect attacks with fewer matches. For the purpose of comparing results, the WuManber algorithm is utilized. The researchers at Lee et al. [2] created lightweight IDs that keep track of the energy consumption of the node in order to detect intrusions. Primary purpose of this identification is to lower the amount of computational resources necessary for intrusion detection. These IDs are also in charge of controlling the energy and traffic flow of the nodes. If the IDS detects an attack, it sends out an alert to all of the nodes.

Centralized Ids Placement Strategy: - IDS is placed in the center of a central IDS installation as a core component. With the use of a border router, Cho and colleagues [3] developed a mechanism for assessing the packets that pass through it. In this study, only communication between border routers is being monitored. Kasinathan et al. [4] made advantage of the centralized location as well, however they deemed the IDS security against a DoS assault to be insufficient (Denial of Service). They decided to create an IDS analytics engine as well as a reporting system on a powerful dedicated server to maximize performance. sensors for the intrusion detection system (IDS) were put in low-power, low-latency networks (LLNs) to monitor network traffic and provide it to the IDS analytics engine. In order to avoid the transmission of IDS data in conjunction with the rest of the wireless network's data, the IDS dedicated host must be linked to the IDS sensors. If the quality of mobile connectivity is reduced as a consequence of a Denial-of-Service attack, the transmission of IDS data will not be affected.

Hybrid Ids Placement: - The concepts of centralized and scattered IDS deployment are blended in a manner that makes the most of their respective advantages while minimizing their respective disadvantages. The first strategy divides the network into regions or clusters,

with just the central node of each region or cluster running an IDS instance. The second technique divides the network into zones. It will be responsible for keeping track of the other nodes in its cluster as a consequence of this development. Because of this, it is possible that IDS for hybrid placement will use more resources than IDS for dispersed placement. According to Amaral et al. [5, this is how they came up with an IDS for the Internet of Things]. In this research, an intrusion detection system (IDS) is placed on a small number of network nodes. These carefully selected nodes (also known as "watchdogs") are hoping to identify attackers on their network by using packet eavesdropping techniques. When deciding whether or not a node has been hacked, the watchdog follows a set of rules. Because each network component is distinct, each watchdog has a unique set of rules to follow. Typical examples are border routers, which receive much more messages than a conventional node. A significant advantage of this strategy is that it allows for the construction of a variety of rules for each segment of the network.

Types Of Ids: - Signature Based Ids:- Attacks are identified by looking for system or network behavior that matches an attack signature stored in the IDS's internal databases. If any system or network activity fits previously recorded patterns or signatures, an alert will be generated. The approaches proposed by Liu et al. in [6] were based on the Artificial Immune System. Datagrams may be categorized as malicious (non-self-element) or normal by detectors that use attack signatures that were modeled after immune cells to distinguish between the two categories (self-element). It is also possible for detectors to alter over time in order to meet changing environmental circumstances.

Anomaly Based Ids: - This approach is also referred to as event-based detection in certain circles. This approach identifies and locates potentially dangerous activities via the use of event-based analysis. First and foremost, it determines the normal behavior of the network's nodes. An incursion is defined as any activity that is out of the norm in the context of the situation. A rogue node may be recognized by comparing the current protocol specification to the protocol state that has been previously specified. This strategy outperforms signature-based intrusion detection systems when it comes to identifying attacks. Thanigaivelan et al. [7] made a short proposal for a distributed anomaly detection system for the Internet of Things

(IoT), which was accepted. When a network inconsistency is detected, the proposed IDS monitors one-hop neighbor node metrics such as packet size and data rate in order to identify and investigate it. In accordance with the writers of this research, a computer system learns and adapts to its environment by continuously monitoring its surroundings.

Specification Based Ids: - In some ways, this technique is similar to anomaly detection. The regular behavior of the network is specified manually in this technique, which decreases the likelihood of false positives. In this paper, an attempt is made to bridge the gap between signature-based and anomaly-based detection approaches by defining abnormalities that are not created by the training data or by the machine learning algorithm in question. As attack or protocol requirements are written by hand, the development process takes longer. There may be certain disadvantages to using this method as a consequence. Network administrators may provide attack detection criteria in the specification-based intrusion detection system presented by Amaral et al. [8]. When one of these rules is breached, the IDS alerts the Event Management System (EMS). Because it functions on an unrestricted node, there are no resource limitations for the EMS.

Hybrid Approaches: - Hybrid techniques combine signature-based, specification-based, and anomaly-based detection ideas to maximize benefits and reduce downsides of each. The IDS assessment approach that Krimmling and Peter [9] established was used to evaluate anomaly and signature based IDSs. Both approaches were shown to be ineffective in identifying specific assaults. According to the authors, a single IDS might handle a broader spectrum of assaults if these methodologies were combined.

Security Threats: - The purpose of this section is to illustrate how different IoT IDS techniques have dealt with different threat categories throughout time. In order to maintain the security and privacy of IoT solutions, it is necessary to apply a number of technologies, services, and standards across the board. Accordingly, we may assume that the Internet of Things paradigm has at least the same security risks as mobile communication networks (for example, wireless sensor networks), cloud services, and other elements of Internet usage. Summerville and colleagues [10] concentrated their attention on conventional attacks. The

performance of Summerville et al IDS. was evaluated using conventional attacks such as worm propagation, tunneling, SQL code injection, and directory traversal, among others.

Cyber-Attacks on Iot Applications: - Attacks such as sinkholes occur when a malicious node attempts to direct network traffic toward itself. It is possible to carry out attacks such as this one by making it seem as if the malicious node's routing cost is at its bare minimum. It is necessary for the attacker to add a fake node into a network in order to execute an attack. When an Intrusion Detection System (IDS) implements the RPL as a routing protocol, sinkhole attacks may be identified and prevented from occurring. In order to validate the Intrusion Ratio (IR) supplied by the intrusion detection agent, the recommended approach takes use of detection metrics such as the number of packets received and delivered, as well as the number of packets received and delivered. It is possible to utilize the IR value to identify whether or not a router node is malicious or not. An IDS system alerts leaf nodes when it detects a rogue node, and the rogue node is isolated in the next data transmission after being identified. One of the objectives of this initiative is to lower the Intrusion Ratio.

Wormhole attack: - In this assault, one node creates a virtual tunnel between two other nodes, which is then attacked by the other nodes. An attacker acts as an intermediary between two genuine nodes by acting as a forwarding node between them. Additionally, the wormhole attack may be used to mislead two distinct nodes into believing they are neighbors by transferring packets back and forth between the two nodes. Using the words of Pavan Pongle: "The technology provided by Gurunath Chavan [12] is an Internet of Things intrusion detection system that can identify a Wormhole assault and its perpetrator." Using the position and neighbor information of the node, as well as the strength of the received signal, it is possible to identify wormhole assaults and prevent them from occurring. This system's development may assist in the prevention of such dangers from affecting the Internet of Things (IoT). In the event that you have a limited quantity of resources, this strategy is the best choice since it employs a preset number of UDP packets to detect potential assaults.

Selective Forwarding Attack: - A malicious node that masquerades as a legal node will discard these packets on a per-packet basis. Black hole attacks are the most basic kind of

selective forwarding attack since they reject all packets that come into contact with the malicious node. Shapla Khanam and colleagues [13] provide an attack model for researching hostile behavior in Internet of Things networks that is based on game theory. Player 1 and player 2 compete to increase the throughput of the network to the greatest extent feasible, while player 3 attempts to maintain it as low as possible. As part of the Internet of Things security architecture, an ACK mechanism for identifying malicious attackers is presented, which may be used to safeguard networks from selective forwarding attacks (IoT).

Sybil Attack: - In this attack, the node assumes a number of different identities. In addition to the routing protocol, a malicious node may attack the detection technique and the co-operation methods as well. Kuan Zhang [14] et al. specify three kinds of Sybil assaults based on the capabilities of the Sybil attacker: SA-1, SA-2, and SA-3. They then provide alternative Sybil defensive measures, including SGSD, BCSD, and mobile Sybil detection, with extensive comparisons. Finally, they discuss Sybil's defense in the face of significant research challenges and future prospects in the Internet of Things.

Denial Of Service (Dos) Attack: - It is possible that the availability of resources may be hampered as a consequence of this attack. Genuine users are unable to access resources as a result of this attack. The term "distributed denial-of-service" refers to an attack in which numerous malicious nodes are involved in the attack (DDoS). Attacks of this kind have the ability to exhaust network resources such as bandwidth and CPU time in a short period of time. Tasnuva Mahjabin et al [15] analyze in detail the nature of distributed denial of service (DDoS) attacks, as well as their prevention and mitigation. A full analysis of the motivations and development of these attacks, as well as the many attacks that have happened to far, the different defenses and mitigations that have been used, as well as the possible limitations and issues of current research, is carried out by the authors. In order to avoid distributed denial of service attacks in the future, certain significant research routes have been identified that need to be given more attention in the near future.

Validation Strategy: - According to D.Chrun[16] Validation is the process of determining whether or not the model developed for the study is accurate enough to achieve the objectives

indicated. Expertise and data are two types of information that may be utilized to distinguish between validation procedures. Generally speaking, when experts are utilized to verify models, the findings are subjective and qualitative; however, when data is used, the conclusions may be quantitative and more objective. The examination of an IoT intrusion detection method validation approach is the primary subject of this paper.

Security Issues In Iot Architecture: - The most challenging aspect of the Internet of Things is ensuring data security and privacy. As a result of the importance of RFID, sensor, embedded, and nanotechnology in the Internet of Things (IoT), construction is a key source of concern. Because the Internet of Things is a collection of different heterogeneous networks, it is difficult to establish a stable link between IoT nodes that are constantly changing. There is a strong interconnection between the sensing, transportation, and application layers of the Internet of Things' architecture (table 2.1)

Table 2.1: application layers of the Internet of Things' architecture

Application layer	Transport layer	Sensing layer
Information availability, user authentication, information privacy, data integrity, IoT platform stability, middleware security, management platform	DOS/DDOS attacks, forgery/middle attack, heterogeneous network attacks, WLAN application Conflicts, capacity and connectivity issues etc.	Interruption, interception, modification, fabrication, uniform coding for RFID, conflict collision for RFID etc.

2.3 CONCLUSION

The rapid expansion of the number of connected objects on the Internet creates many challenges, one of the main ones being the security of its users. Thus, more and more computer attacks involve such objects, which are often very vulnerable compared to traditional computer systems, and the scale of these attacks can be considerable. This insecurity of connected objects creates significant risks for users and companies, it is therefore urgent to propose effective solutions to detect intrusions on such objects. This thesis aims to detail a powerful solution for detecting intrusions, compatible with the majority of smart objects because it is not very intrusive and using machine learning techniques to automate

detection. The tools developed are able to process information at different levels, whether it comes from the kernel or from the user space and rely on powerful tracing techniques to achieve this result.



3. MATERIALS AND METHODS

3.1 THE INTERNET OF CONNECTED OBJECTS IN HOMES

The Internet of connected objects, or IoT (IoT) refers to the interconnection of many embedded systems with each other and with traditional computing. There is thus a convergence between old information networks, cloud computing and new objects capable of communicating across the planet. Our work mainly focusing on connected objects in homes, often called home automation systems, the literature review will mainly detail these aspects. It should however be noted that many other works have been proposed in this field, whether at the level of connected cities or at the level of companies. In addition, large-scale projects are being carried out in several major cities around the world, such as Toronto with the involvement of Google, Dublin and Amsterdam, which have rapid expansion policies and which greatly increase interactions between connected objects, vehicles, people and the global Internet. Connected objects are essentially embedded systems which therefore have limited resources, in terms of computing power, storage space and even power supply since some operate using batteries.

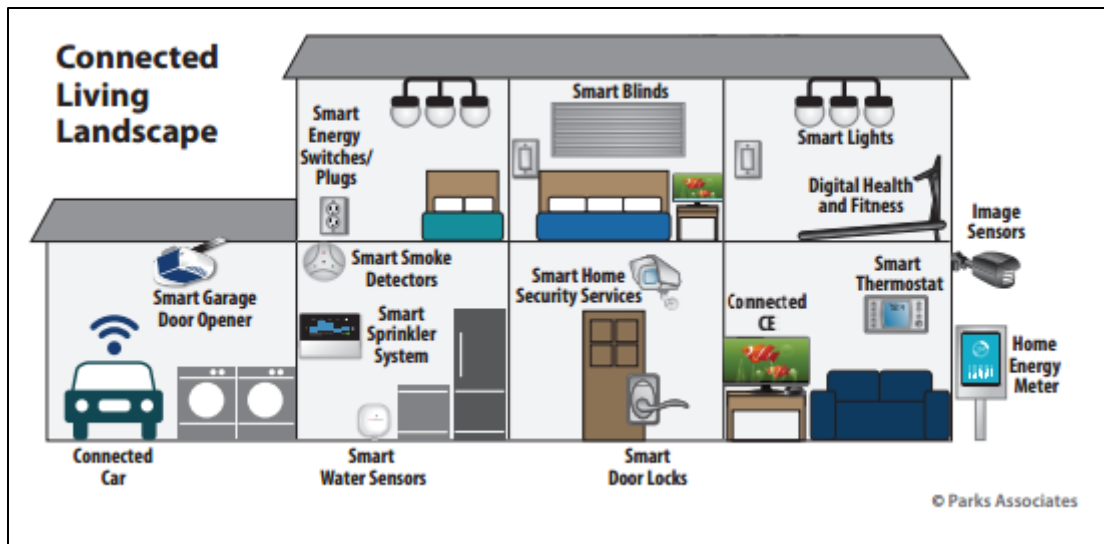


Figure 3.1: Internet of connected home objects

3.1.1 Home Automation

Home automation, or smart home, refers to a house in which connected objects communicate with each other, or with the inhabitants. Thus, the connected home is part of the expansion trend of connected objects and has its place in the smart city projects launched all over the planet. In addition, Gubbi explains that homes will be the first places where connected objects emerge before they appear at the corporate or city level. He also indicates that these objects will be useful to increase the comfort of users, have better control over their energy consumption and even improve their health through various devices. In addition, a report published by n smart homes in 2016 estimates that the connected object market will generate \$7.1 trillion in 2020, while Cisco estimates that at that time there will be 50 billion smart objects in functioning. There is currently no convergence on the technologies used for these objects in homes, whether at the software or hardware level or even at the level of communication protocols. Thus, many manufacturers offer various solutions, sometimes compatible with objects from other manufacturers. This material disparity is however nuanced by the very large presence of ARM processors which are efficient in performing calculations, while consuming few resources, and by the MIPS architectures which have traditionally been used in embedded systems and which offer interesting performances. Similarly, the disparities between different firmwares and different operating systems are very present, but a lot of this software is based on very light distributions of Linux. Thus, manufacturers like Vera rely on modified versions of OpenWRT, a Linux distribution widely used in network routers. However, very few firmwares are open source, and several manufacturers have created their own firmware to optimize the performance of their devices. Finally, the heterogeneity of the different communication protocols, whether long or short range, is underlined in the following paragraphs.

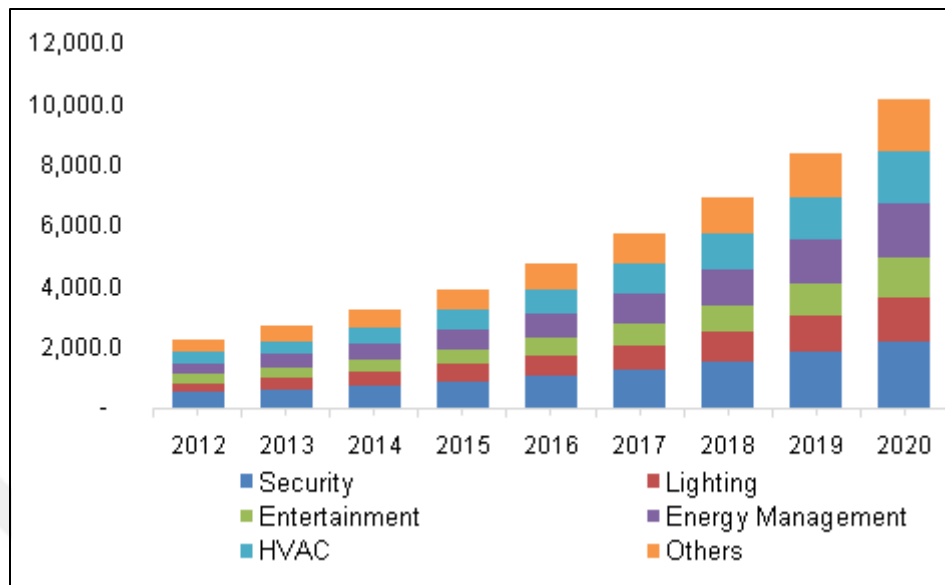


Figure 3.2: Smart homes security statistics by 2020

3.1.2 Communication Protocols

connected objects create new modes of communication: after exchanges between computers, then between computers and humans or humans and humans, new exchanges of information take place between objects and humans and even between the objects themselves. It is the interconnection of these objects for this purpose that is called the Internet of Objects, or IoT for Internet of Things. We can distinguish two categories of communication protocols used by these devices: long-range protocols, such as, NB-IoT and SigFox, as well as the already implemented cellular networks and their new versions. LoRa (acronym for Long Range) is a protocol developed by Semtec and based on a physical layer called LoRA. There are 3 different classes according to (WPL), class A (the nodes speak to the gateway as soon as they need it), class B (the nodes listen at regular intervals in a time window) and class C (the nodes listen at all times). This allows many applications of this protocol. NB-IoT, for Narrowband IoT, comes from an initiative of 3GPP, an organization that develops telecommunications protocol standards. It works on the 10MHz LTE network and has different implementations. Finally, SigFox allows exchanges between connected objects with very low bandwidths. It operates on the 868 MHz bands in Europe and 915 MHz in North America, has a range of up

to 50 km in rural areas and 10 km in urban areas. It works with very small messages (12 bytes for the payload), thus consuming very few energy resources and allowing master-slave or bidirectional (RFW) communications.

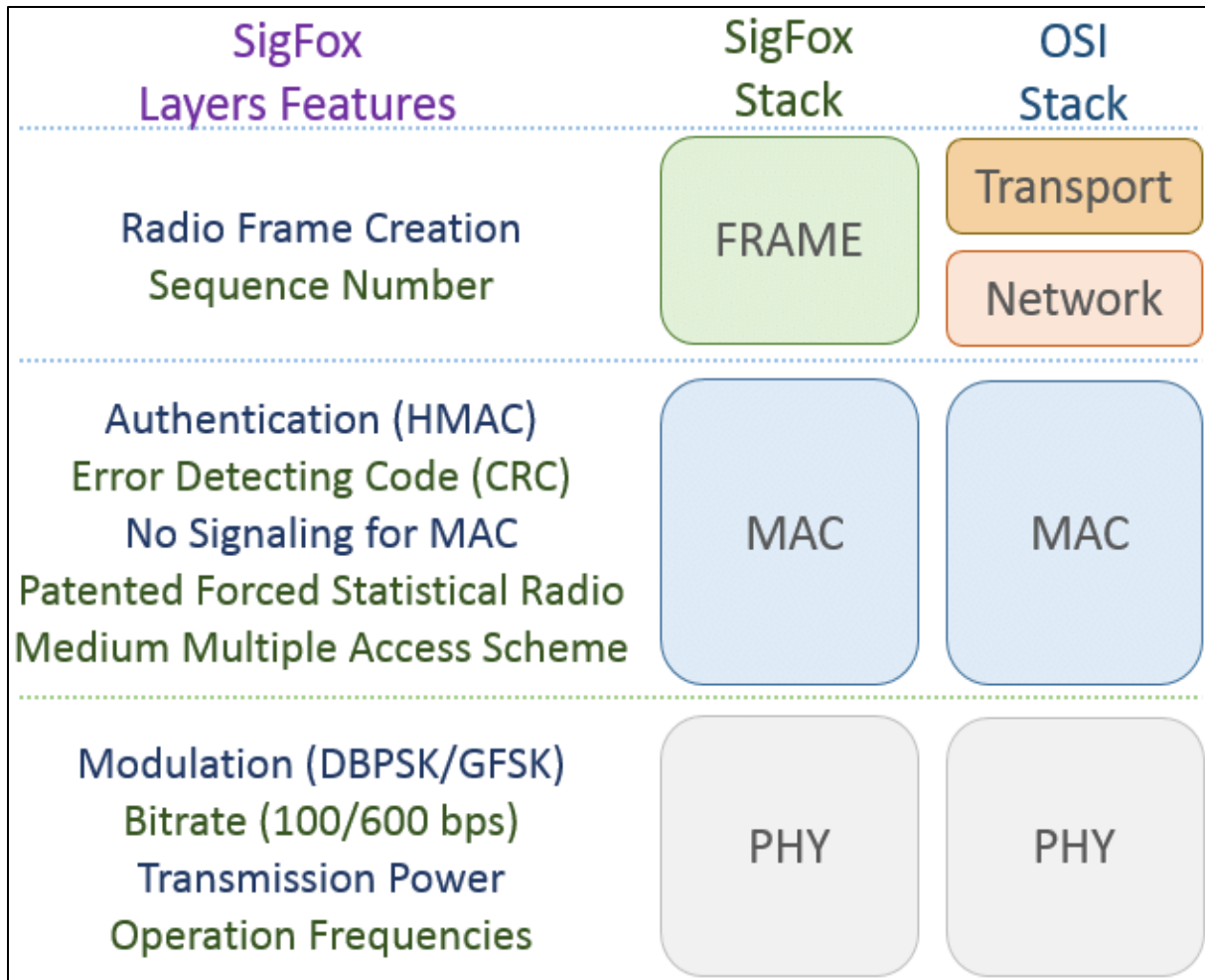


Figure 3.3: Diagram of Sigfox protocol stack

3.1.3 Short-Range Communication Protocols

The Z-wave protocol Z-wave is a protocol developed in 2004 by the company Zensys, operating on the principle of a mesh network, where the nodes of the network can also take care of the routing of the communication. This protocol is supported by the Z-Wave alliance, a consortium of more than 300 companies that are committed to developing Z-Wave compatible

and interoperable products. As a result, this protocol tends to become the leading protocol for smart objects within the home and more and more objects are compatible with this standard. Although in theory this protocol supports up to 232 nodes in the same network, works claim that problems seem to arise when connecting more than 40 nodes (WPH). This document also explains that not all Z-Wave equipment is compatible with each other, since backward compatibility is not guaranteed between older versions of Z-wave and new ones, in particular due to security standards that have evolved. Regarding this last point, Z-Wave relies on 128-bit AES encryption; only Z-Wave chips above the 400 series offer this encryption. This white paper encourages all owners of 200 and 300 series chips to replace them to ensure the security of their network, as well as to regularly change the encryption keys used. In addition, the Z-Wave protocol was developed to facilitate the communication of connected objects in homes. It operates on 908.45 MHz frequencies in North America and 868.42 MHz in Europe. The signal bandwidth is 40 kb/s and two consecutive network nodes must be separated by a maximum of 30 meters. This relatively short distance is not a barrier to the adoption of the protocol in a house, since there are multiple connected objects in different rooms of the dwelling. Z-Waves objects fall into two categories: controllers and slaves. The controllers can give orders and modify the behavior of the slaves and take care of the routing of information, the slaves can then send information back to the controller if they have. It is possible to distinguish four layers to establish a Z-wave communication: the physical layer, made up of radio frequency transmitters and receivers, the transport layer, the routing layer, with the management of the mesh network, and the application with the different classes of Z-Wave commands. There is an Open Source implementation of the protocol, named Open Z-Wave, but the security implementation is not yet supported. In addition, many modules make it possible to transform the classic architectures used in home automation into architectures compatible with the protocol. For example, the Raspberry Pi (Foundation, 3) can transmit Z-Wave signals using a compatible board named Razberry with similar models being compatible with Arduino boards. The Zigbee protocol The ZigBee protocol is also based on a mesh network, but is implemented on the 2.4 GHz frequency bands, such as Wi-Fi or certain Bluetooth bands. Zigbee has a specification based on IEEE 802.15.4 standards and is supported by the Zigbee Alliance. Like Z-wave, ZigBee is compatible with many devices, and its

popularity makes it the direct competitor of Z-Wave. In addition, this protocol offers higher theoretical speeds than Z-Wave. More and more manufacturers are making their products compatible with these two standards, like Samsung with its SmartThings range. The 6LoWPAN protocol It is possible to communicate between items that are linked via the 6LoWPAN protocol, which stands for IPv6 Low Power Wireless Persona Area Network. The Internet Engineering Task Force created it as a "lighter" alternative to the current Internet Protocol (IP) protocols. When employing a mesh network, UDP and TCP are supported to the utmost degree possible. Mulligan states that the packet headers are quite small (2 to 11 bytes) and that it can link 264 nodes at once. This protocol has also been the focus of the vast bulk of research on intrusion detection systems in networked devices. Because it runs in the 2.4GHz frequency band, 6LoWPAN is easier to connect with existing equipment than ZigBee. As the number of connected devices in houses grows, it is getting increasingly difficult to find general solutions for safeguarding them.

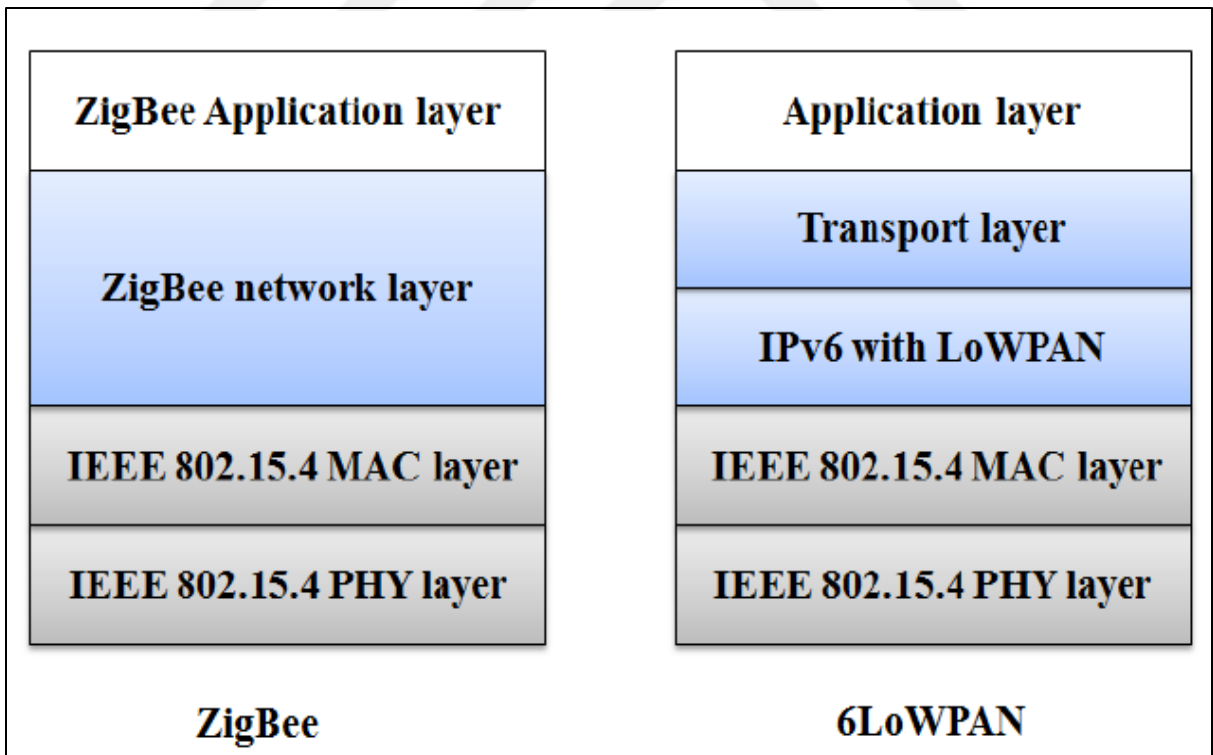


Figure 3.4: Layers of Zigbee vs layers of 6LoWPAN

3.1.4 Connected Home Uses

The appearance of connected objects has created and will continue to create new needs and new uses. The main uses of connected objects in connected homes are entertainment, surveillance and security, resource and energy management, household assistance, as well as light and health management (WPU). However, the authors clearly identified the causes of insecurity of such objects. Indeed, it is explained that most of the vulnerabilities discovered on this new equipment are due to the very design of the products, the use of communication protocols that are not secure or unsuitable authentication methods. In addition, poor implementations or uses of connected objects would be a major cause of the vulnerabilities discovered according to the authors. Additionally, the white paper highlights the lack of available updates or patches for the on-board software. Padilla et al. strongly insisted on the need to be able to update connected objects (Acosta Padilla et al., 2016a). In addition to proposing update strategies adapted to connected objects, the author explains that it would be wise to be able to update the embedded software piece by piece, rather than a global update of this software. In addition, it is necessary to be able to verify the integrity of these new installations. one of the challenges of home automation is to protect the confidentiality and integrity of communications inside and outside the home Accordingly, effective protection techniques must be implemented. Thus, during the exchange of cryptographic keys, connected objects can decrease their transmission power to make it more difficult to intercept packets and thus prevent an attacker from obtaining this crucial information by listening to transmissions outside the house. However, the security of some objects still seems insufficient, since the authors claim that in some objects, the keys useful for encryptions are written directly in the object's memory, and without any particular encryption protection.

3.1.5 The Architecture of Connected Homes

As explained previously, the communication protocols of smart objects are different from the protocols used on the traditional Internet. Consequently, it is necessary to have an object making the link between the protocols of the connected houses and the TCP/IP protocols, the most widespread on the Internet. This is the role of the home automation controller, which

also has the task of initiating actions according to the information it receives from the various sensors present in the house. It is possible to consider this object as the "brain" of the home automation system. Thus, a classic architecture in a connected house In this configuration, connected objects coexist in the home network with traditional devices, such as computers or smart phones. However, some authors recommend splitting the network of smart objects according to their usefulness. who proposes an architecture where each object is partitioned according to the communication protocol it uses According to its criteria, objects using Bluetooth should be distinguished from objects communicating via Z-Wave. However, some objects, typically connected cameras, receive orders via ZWave messages, but can transmit data (here video data) over traditional Wi-Fi with the IP protocol.

3.1.6 Security Risks Associated with Connected Homes

The rapid development of the connected objects market is forcing manufacturers to quickly offer consumers many innovative products. Thus, the security considerations of new objects placed on the market are not sufficiently explored, which creates greater risks for users. Moreover, with the heterogeneity of the proposed solutions, it is complicated for security researchers to address security issues for all smart objects. Conti distinguishes between security challenges related to connected objects and those related to their digital investigation Thus, the author explains that it is necessary to ensure authentication, authorization and access control. It is also necessary to secure user privacy. According to him, this must be done with a secure architecture. Regarding digital forensics, he asserts that the issues related to the identification, collection and preservation of evidence are different from traditional computer systems. It is necessary to reflect on these issues in the context of connected objects in order to be able to analyze the evidence, correlate it and be able to trace the attacks that have occurred. Thus, many authors are concerned about the general lack of security of objects placed on the market and the threats that this introduces to users. the security of objects and respect for the privacy of users appear to be fundamental criteria for the adoption of these objects in homes. The last author identifies the dangers associated with the use of these objects, while Babar makes a taxonomy of threats associated with the emergence of the Internet of Objects (Babar et al., 2010). It identifies five areas of danger for smart objects:

- a) Identification management: this area includes all the security techniques that must be implemented to uniquely identify objects, users or sessions. It is necessary, according to the author, to be able to guarantee a high level of security of authentication, authorization and account management methods.
- b) Communications security: this principle includes all the measures aimed at ensuring the availability of communications.
- c) We could have added to this the integrity of these communications, which should not be able to be modified, as well as the confidentiality of these to avoid putting the privacy of users at risk.
- d) Physical threats: in this category, the author explains that physical access to objects is most of the time very easy, and that this makes it easier for attackers to perform attacks such as reverse engineering of the system or microprobing attacks to find encrypted files
- e) The author does not explain that connected objects can compromise the physical security of their users, this point being detailed below.
- f) Security of embedded systems: the author claims that embedded systems will be subject to attacks similar to traditional systems at the physical and MAC levels. For example, attacks by side channels or against secure environments can be conducted against these objects.
- g) Management of information storage: this principle is essential in the management of cryptographic keys. It is also important to ensure the integrity and confidentiality of the data present on the objects.

Various works have been proposed to ensure the security of data storage in the case of connected objects, such as that of Babar which proposes a global framework for securing such systems. Few articles deal with the risks introduced by connected objects on users, whether it is their physical security or respect for their privacy. Thus, the interconnection of more and more objects such as connected ovens, intelligent alarm or smoke detection systems, or even

gas control systems capable of being operated remotely are becoming more democratic. However, if an attacker obtains sufficient control over these objects, for example by acquiring full access to the home automation controller, it would then be possible for him to open the gas supply and then suddenly turn on the oven, thus possibly causing a explosion and physical damage to the habitat or its inhabitants. Similarly, respecting the privacy of users of smart objects involves securing personal information, such as audio tracks or photos recorded by cameras. It is also necessary to protect the information emitted by the various sensors that may be found in the houses (temperature, movement, light sensors) since they could allow an inhabitant to know the habits of the inhabitants of the house or even to determine their activities. These threats are based on the different waves of attacks involving connected objects. Saxena thus identifies several attacks targeting smart objects while Sikder is interested in attacks specifically targeting sensors in networks of connected objects. Similarly, Babar performs a taxonomy of the attacks encountered and Kozlov shows different attack scenarios depending on the architectures developed. We can summarize the different works of these authors according to the following different principles:

- a) data exfiltration, with the aim of spying on users, stealing information from residents or modifying it. For example, the attacker might want to blackmail the victim, steal secret documents to resell them, or alter documents to deceive his target.
- b) attacks on network messages: the attacker wishes to modify the behavior of the home automation system or even destroy it, to do this he can send false information to connected objects (whether it is erroneous sensor data or orders to actuators), perform replay attacks where it will capture commands sent over the network and replay them or use person-in-the-middle techniques.

the unavailability of systems: the malicious user aims here to make a system inaccessible to its users. This system can be a connected object in itself or a remote server. An example of such an attack where the target was a smart object is the which aims to delete all files of the object it infected and corrupt the memory so that it is not able to be repaired. A famous infection wave targeting a remote server is the botnet campaign which infected smart objects by

attempting default passwords. Once the objects were infected, they could be controlled via a remote server to launch distributed denial of service attempts against remote targets, such as the servers of the DNS provider DYN or the host OVH. It can even create financial risks for owners of connected objects, with waves of ransomware forcing consumers to pay a ransom to be able to decrypt the software of their infected objects and thus be able to use them again intrusion into a network: whether it is a home or business network, the connected objects belonging to it are often less protected than traditional systems, allowing attackers to have under their control a first piece of equipment of such a network so that others can then be targeted. Thus, pivoting techniques can be performed once an object is controlled, sometimes allowing network protections such as external firewalls to be bypassed.

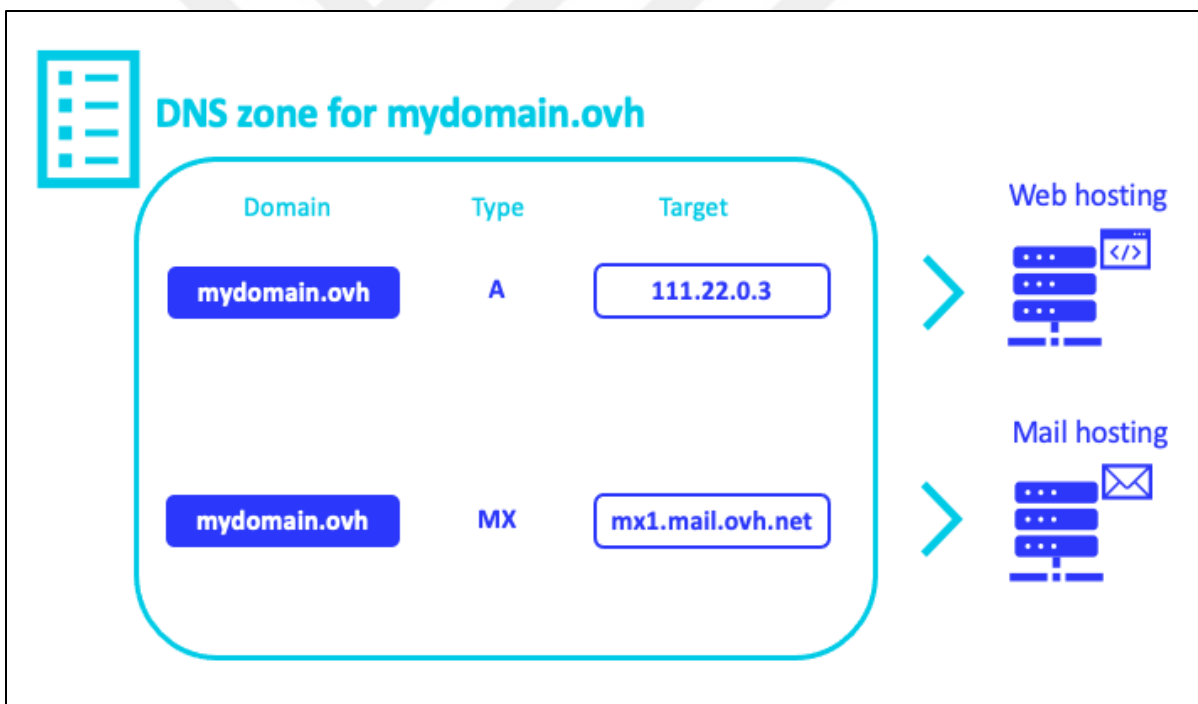


Figure 3.5: A. OVH DNS firewall

3.2 INTRUSION DETECTION SYSTEM

To overcome all these threats, many works have tried to find effective ways to detect intrusions as soon as they occur. Such devices are called IDS, for Intrusion Detection System, and can identify attacks using network traces, it is then a NIDS (Network-based IDS), or using

information on the monitored system is then a HIDS (Host-based IDS). IDSs have been studied for a long time, as Debar explains in his taxonomy on intrusion detection systems and these systems have been deployed in all types of networks as shown by Anandvalli with mobile ad hoc networks says traditional IDSs are not suitable for the IoT because objects have limited resources that make it impossible to use heavy-duty intrusion detection technologies. In addition, the protocols used by these objects are numerous and specific to them, and the objects often evolve in mesh networks, being both a source of data and a system capable of redirecting the data it receives. It is these constraints that have led to new research on IDS in the context of smart objects. However, most of these are based on network information (NIDS), as explained by Sabahi in his literature review on the subject can be explained by the fact that this information can be easily obtained without having to modify the monitored object, which can be complicated with embedded systems and even reduce their performance. Sabahi explains that there are two types of intrusion detectors: those based on rules about the system, and those based on the discovery of system anomalies. The use of an expert system, which is one of the means of carrying out intrusion detection with rules, can be based on known signatures of threats, on rules concerning the data collected or on the state transitions of the system. This type of intrusion detector has the advantage of being precise and working very well on known threats, but it is unable to detect new threats, unlike the second technique. Indeed, anomaly based IDSs monitor the behavior of the system they are protecting and attempt to detect any suspicious variations in that behavior. They therefore require the ability to precisely characterize the behavior of the system they are monitoring, which can sometimes be complicated and require a lot of resources. The various anomaly detection techniques are explained in the following subsection. Sabahi also points out that hybrid methods, using both data from network traffic and characteristics of the monitored host, have been proposed. The architecture of intrusion detectors can also vary greatly, depending on the type of system monitored, the resources needed to detect the intrusion, or the data used. Thus, Sabahi explains that centralized IDSs are the most common, but distributed and hybrid architectures have been proposed. In a centralized architecture, the system that is monitored is the one responsible for detecting anomalies, for example in the case of SNORT where the analysis of network frames is done on one or more nodes of the network, which are the systems that will

collect the information necessary for intrusion detection. On the other hand, distributed architectures are essentially based on agents that will collect information on various hosts and route this information to another system that will take care of detection. An example of such a system is DIDMA, the solution offered by Kannadiga, which consists of static agents on the hosts to be monitored, which transmit information to mobile agents using an agent dispatcher. These mobile agents then transmit the information to alert agents on the IDS console who will then raise the alerts when necessary. As the various articles have pointed out, intrusion detectors are only as effective as the means of collecting the information they need. Thus, several works on HIDS have relied on system calls to detect malicious behavior. However, several articles have shown that this information is not sufficient to prevent attackers from carrying out intrusions without being noticed. Thus, Wagner's work shows how it is possible to evade intrusion detectors based on system calls, and particularly when the arguments of these kernel functions are not taken into account by the intrusion detector. Its method is to imitate the normal behavior of a system by performing illegitimate actions on it. Similarly, Ma points out that it is possible to evade detection from most HIDS by using child processes that will perform few actions, but that the sum of these actions can lead to advanced attacks

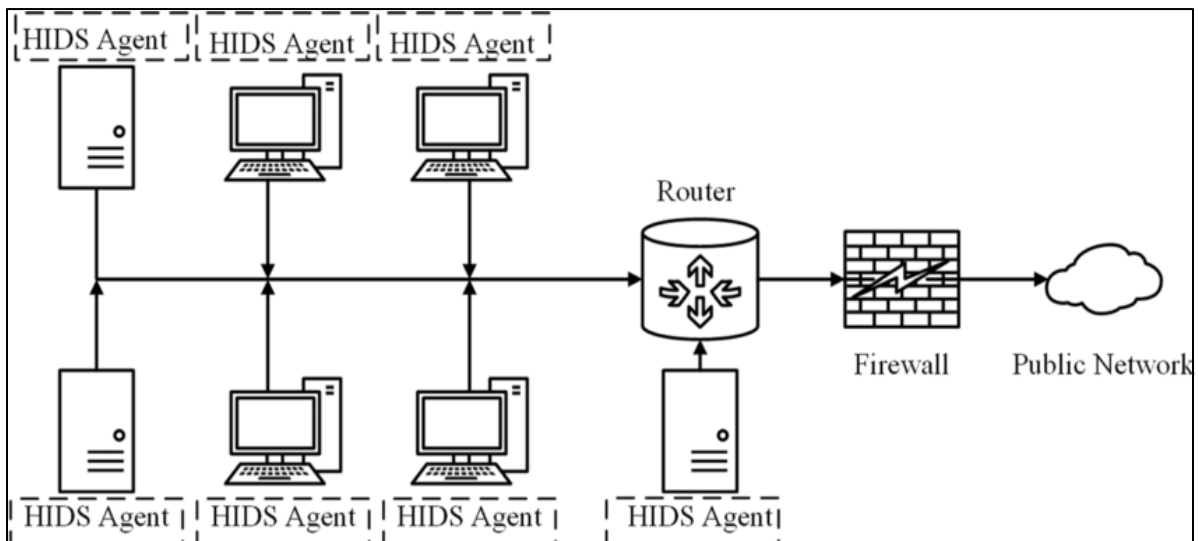


Figure 3.6: HIDS in IO

One of the means used to acquire the information necessary for detection is the use of tracing techniques, which not only allow to have the list of system calls made, but also their arguments and information at different levels, when we trace the kernel, such as network information, the sequence of the actions of the scheduler or even information on the use of the processor. Moreover, Murtaza explains that traces at the kernel level make it possible to acquire much more useful information for intrusion detection. Tracing techniques are detailed in the corresponding section. However, the uses of intrusion detection systems have evolved with the arrival of new computer applications. With regard to cloud computing, very precisely the different intrusion detection techniques that can be used in this context. The author explains that the traditional systems (HIDS and NIDS) can be used, but that new systems have appeared. This is the case of distributed intrusion detection systems (DIDS) which are made up of several IDSs placed in a large network and which communicate with each other (whether with a central server or directly with each other). These systems can combine both HIDS and NIDS. In addition, intrusion detection systems based on virtual machine hypervisors have also been developed. Such systems thus make it possible to monitor the exchanges of information between the virtual machines, between the virtual machines and the hypervisor and even with the hypervisor of the virtual network, according to the author. Likewise, the democratization of smart phones, and particularly of the Android operating system, has created new threats for users. This has justified various works on intrusion detection on such devices. Thus, Borek listed the various intrusion detection techniques employed for objects running Android. He thus explains that several works detect illegitimate applications according to their source code.

3.2.1 Detection by Rule

Many works have been proposed to detect intrusions using rules. Thus, techniques using artificial neural networks, pattern recognition or even tools to identify intrusions in databases have been proposed. Canady has proposed using artificial neural networks to detect network-level attacks. This choice is justified by the fact that setting up an expert system to achieve this goal requires precise knowledge of network attack methods and that it can be difficult to identify intrusions whose method varies slightly. of those known. In addition, the author

explains that expert systems need many updates to remain effective against emerging threats, and that it is necessary to monitor system state transitions rather than the arrival of specific events. in the case of detection by rule. Canady has shown that artificial neural networks can be more flexible than expert systems for network intrusion detection, since they can work even with incomplete data, they are fast, and they can learn from previously analyzed and identified attacks. However, the author explains that this learning phase can be difficult, since it requires a lot of data that can be difficult to obtain, and that it is complicated to understand the exact functioning of the trained model. Kumar was interested in recognizing patterns of certain attacks to recognize intrusions The article lists various pattern recognition techniques that are applicable to attack detection and explains that these methods work well for detecting intrusions on UNIX systems. However, it is necessary to have identified the attacks to extract the patterns, and a variant of an attack could lead to different sequences of events, making the detection ineffective. In addition, rule-based intrusion detection systems have been proposed to protect databases This work proposes to use the log files of the monitored databases to create a profile specific to the operation of the database. Therefore, if this profile tends to change, a potential intrusion can be lifted. The paper proposes an architecture capable of processing database log data, creating an associated profile, and training a model capable of detecting deviations from this model. The profile is created according to various metrics, such as the frequency of the items considered, or the relationships contained in the database.

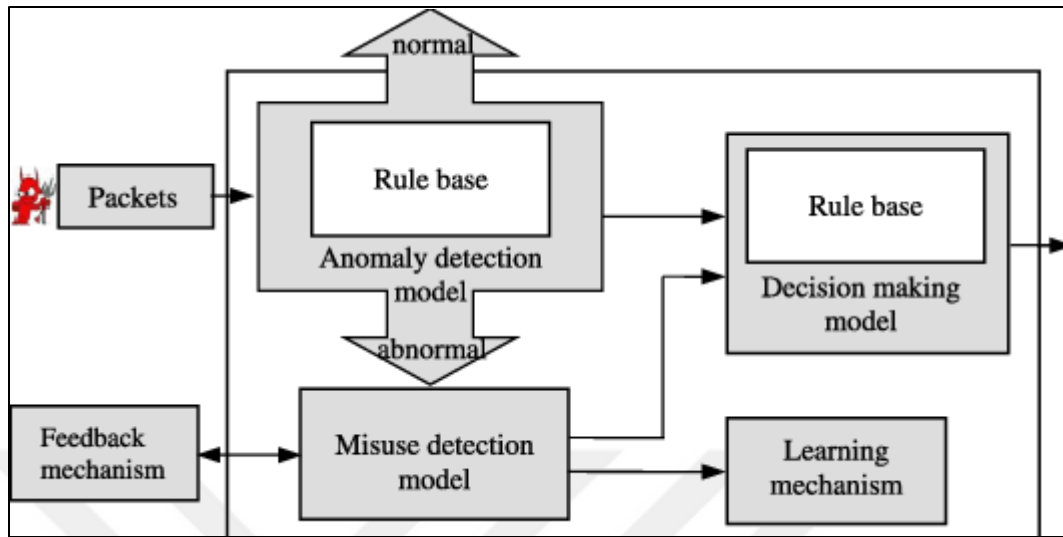


Figure 3.7: Detection by rule in IOT based IDS

3.2.2 Anomaly Detection

anomaly detection is a set of techniques whose goal is to detect unusual behavior on systems (Tsai explains that this can be done with machine learning techniques algorithms can be implemented for this purpose, and several strategies can be used, whether it is data classification, clustering or the use of metrics and statistics on the monitored system. Murtaza explains that anomaly detection using tracing is more effective with kernel-level information than with only user-space level information. Indeed, kernel tracing provides much more information on the state of the system, with information related to running processes such as system calls and the various actions of the scheduler, but also information at the level of the physical components of the system such as the use of the processor or even network information, with the queue of the packets which arrive or which are going to be transmitted. In another work, the same author uses three different algorithms to monitor anomalies in a system: the Kernel State Model, a sequence recognition algorithm and a hidden Markov chain algorithm. All three algorithms rely on system calls that are traced to the monitored machine. However, it is very important to differentiate between the different strategies previously stated, as shown in the following paragraphs.

Statistics It is possible to use statistics on the system to be monitored to detect anomalies when they are easily collectable, and it is not

difficult to characterize the normal behavior of the system. Indeed, if we know the average utilization rate of the RAM and the processor of a computer, as well as the classic occupation of its hard disk, and that these parameters do not evolve much over time, then a sudden change in one of these values could indicate that an anomaly is occurring in the system. On the other hand, if it is very complicated or expensive to collect this information, with systems with very limited resources for example, or if these values vary a lot without any real trend, then the use of statistics to monitor the activity of the system can be difficult. Thus, uses state machines on network packets to detect anomalies on monitored systems



4. PROPOSED METHOD

4.1 CHAPTER INTRODUCTION

There are several security flaws linked with the Internet of Things, which [6] discusses in detail. Certain protocols have a flaw called interdependence, which occurs when a large number of devices are connected and the activities of one device are influenced by the actions of other devices. Interdependence is a defect that may result in complications. Because the Internet of Things employs so many different communication protocols, it is difficult to address all of the security problems. Due to inadequate device resources, such as memory and battery utilization, the app's capabilities are restricted. Myriad refers to the large number of Internet of Things (IoT) devices currently in use; unattended gadgets, which are equipment that cannot be physically touched, are referred to as "unattended" since they may operate for extended periods of time without maintenance. Several disadvantages of gadgets that save personal information about their users include the following: Mobile IoT solutions need devices to communicate with a varied variety of unknown hardware; this is one of the primary reasons for their creation. Finally, Ubiquitous tackles the security and privacy problems raised by the broad use of Internet of Things devices. As a result, it is clear that the Internet of Things (IoT) has serious security issues that must be addressed. This research's deployment of an Intruder Detection System has the ability to effectively resolve security issues linked with Diversity and Myriad [6].

4.2 DATASET USED

KDD'99 data in a collection in order to prevent the appearance of skewed classification results, the categories 4 and 5 were excluded from the dataset. There are several other versions of this data set that may be used for training purposes. KDDTrain+ 20Percent, which has 25192 instances, is one of the many versions of this data set that can be utilized. The KDDTest+ test data set has a total of 22544 instances, which is a large number. However, there are other more data set configurations that may be used, each with a different number of instances, but the total number of characteristics in all of them is 42. The 'class' attribute is

found on line 42 of the data set, and it informs us if a given instance is a legitimate connection or a malicious attempt. Each of the attributes of the KDD data collection, as well as the classes to which they belong, are listed in Table 4.1. Four unique groups are distinguished from one another by the 42 characteristics listed above in the table below: x The most important thing (B) Individual TCP connections have characteristics that are described by attributes that are exclusive to them. Specifically, they are the properties inside a connection that are indicated by domain knowledge (x C traits). In order to determine the attributes known as x Traffic (T) features, a time period of two seconds is utilized. Attacks lasting more than two seconds are evaluated based on a set of parameters known as x Host (H) characteristics.

Table 4.1: KDD dataset features and its corresponding types

Feature Name	Description	Type
hot	number of ``hot" indicators	Continuous
num failed login	number of failed login attempts	Continuous
logged in	1 if successfully logged in; 0 otherwise	Discrete
num compromise	number of ``compromised" conditions	Continuous
root shell	1 if root shell is obtained; 0 otherwise	Discrete
su attempted	1 if ``su root" command attempted; 0 otherwise	Discrete
num root	number of ``root" accesses	Continuous
num file creation	number of file creation operations	Continuous
num shells	number of shell prompts	Continuous
num access files	number of operations on access control files	Continuous
num outbound c	number of outbound commands in an ftp	Continuous
is hot login	1 if the login belongs to the ``hot" list; 0	Discrete
is guest login	1 if the login is a ``guest" login; 0 otherwise	Discrete

It is vital to preprocess data before we can train our classifiers on it, after we have collected the data and analyzed it. In order to do this, we used a five-step approach that involved cleaning and converting data, dividing it into test and validation sets, and creating visual

representations. Organizing and cleansing the data Each of the 43 NetFlow characteristics, as well as the attack type and a label indicating whether or not the entry was malicious, were included in the dataset, as was the attack type. Some considerations should be made when dealing with nonsensical values like "nan" (which stands for "nothing"), "+inf," and "-inf." The following features were removed from all items in the dataset as a preventive step, since they were not useful in characterizing network attacks. In this case, the information comprises IPv4 source and destination addresses, as well as source and destination ports, in addition to the minimum and maximum flow timeouts. Following the cleaning process, we were left with a collection that had 37 characteristics, all of which have been either floating-point or integer numbers. Information Separation and Segregation As a result, we only used 40% of the available data, which allowed us to construct and test models more quickly. We divided the dataset into four sections using stratified splitting to guarantee that the attack distribution was consistent across all of the dataset's segments. We employ a three-way holdout technique with a 60-20-20 split on the dataset to divide it into three groups for our training, testing, and validation sets. To guarantee that each assault is represented in each of the three datasets, stratified splitting is performed. 560927 data points in all, comprising 186976 training samples, 186976 validation samples, and 186976 testing samples, have been gathered. As a general rule, it is less of an issue when training on a large dataset using a three-way holdout strategy than when using k-fold cross validation on a smaller dataset [27]. a decrease in the amount of information and a replication of the amount of information We employ an approach similar to that of J. Kim et al. and S.-N Nguyen et al. to convert the dataset's features to RGB images, which have been shown to perform well in prior research for image classification tasks. Fig. 3.2 illustrates the three steps that occur throughout the process of photographing a subject. To normalize our data, we will use Equation 3.1 to scale each feature in the dataset. The next step creates a 7x7 matrix with padding to guarantee that our array of characteristics is the right size for this phase. Our matrix was multiplied by 255 and then colored using an 8-bit color map to generate this picture. CNNs may benefit from employing color maps rather than grayscale photos to more properly define distributed denial of service (DDoS) assaults and malware. [13,34].

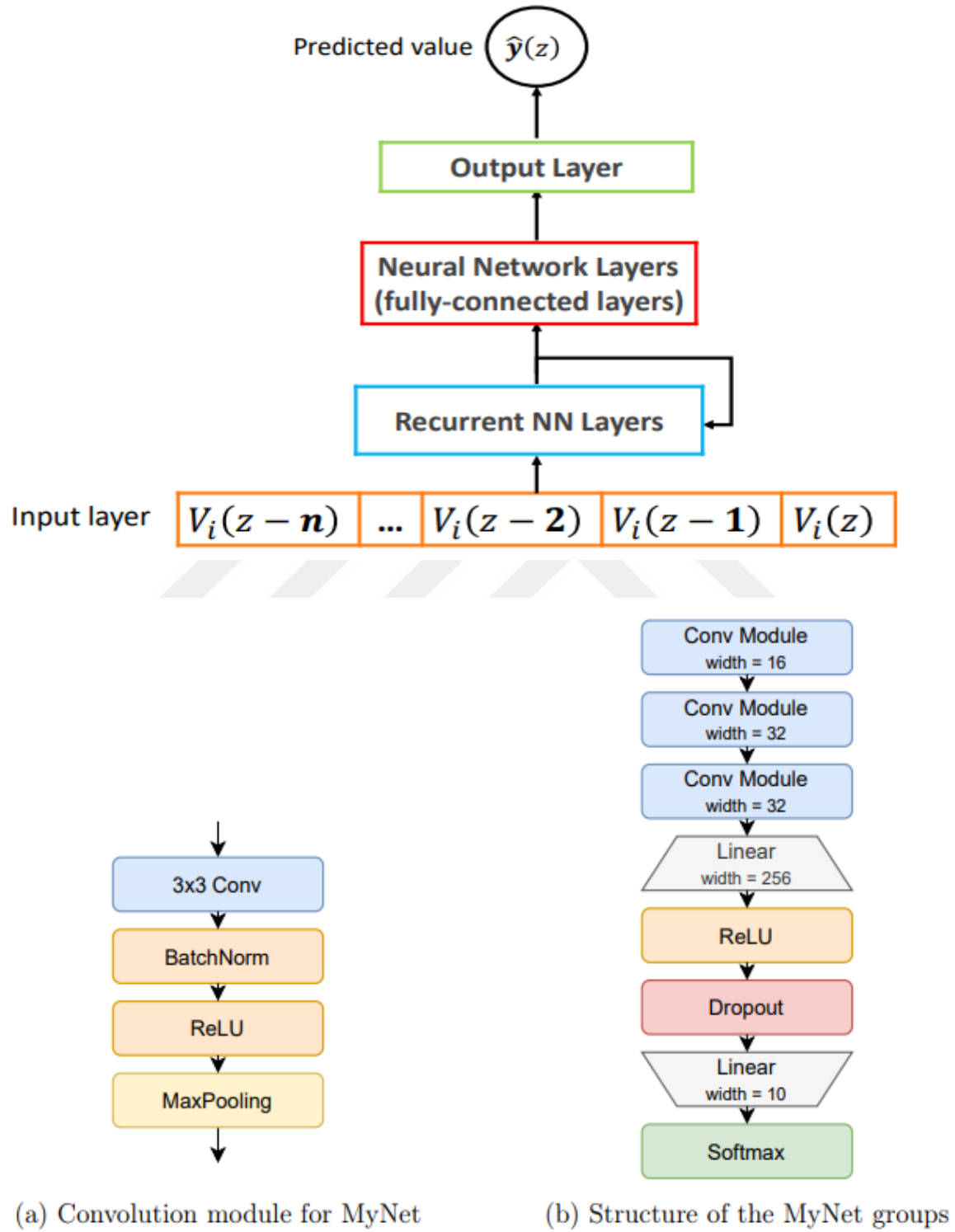


Figure 4.2: Where (a) is the standard CNN net and (b) is the proposed RNN-CNN modification to the network

Looking at Figure 4.2, let's analyze all its levels step by step, starting from the bottom up:

- a) Input layer: for each time-step the model input is obtained from data sequences of length "n", where "n" represents the size of the sample window to be considered, this can be chosen based on constraints on the application, in this case there are no real constraints, however, its value can be adjusted to improve the forecast accuracy. Instinctively, the higher "n" is, the more the forecasts improve, but indirectly the detection time increases, because more samples are required. In our experiments $n = 10$ was set in such a way that it is consistent with the previous steps.
- b) RNN layers + CNN layers, given their complex structure, will be extensively discussed later.
- c) Output layer: An output layer is added to return the forecast to the original data domain. This linear layer uses the "sigmoid" activation functions to scale the values between 0 and 1, in the context under examination 0 and 1 mean respectively no attack or attack.
- d) The final output is a forecast $\hat{y}$, this can be interpreted as a measure of certainty even if it is presented as a real probability.

For example, an expected value of 0.9 indicates that the neural network is predicting with high certainty that the true value is 1, so this is an attack. While values around 0.5 indicate that the model is unsure of its prediction. Consequently, once the prediction has been performed, the approximation function used assumes great importance, that is, finding that value that is able to establish whether given the sequence of "n" input values, a prediction greater than "x" is an attack or less. Initially this value was set to 0.8, then later brought to 0.95. The modification of this threshold value, set only once manually, mainly triggers the occurrence of two events:

- a) First, a too high value of "x" means that the number of false positives is almost nil.
- b) On the contrary, too low values, <0.7 , increase the number of false positives, but it greatly anticipates the detection, since in this situation, only a few samples are enough to say that it is an attack; In reality, in addition to the technical reasons for which this architecture

was chosen, there is also a more logistical one, that is, the neural networks are much simpler than the recurrent ones, their training time (as has been found in the various tests carried out), it has a 5 times lower charge, therefore, given that both networks share the same hyperparameters (essential for adapting the model to the specific case in question), a more conservative approach was opted, in fact, for the CNN tested all the possible configurations of the main parameters, while more targeted and specific tests were performed for the RNN, since in this case testing all the configurations would have been logistically impossible. These tests carried out in two distinct phases made it possible to specialize the second part of the topology as much as possible.

4.4 TRAINING AND SIMULATION

Following the acquisition of the data, whether pre-loaded or taken by a data input, it is essential to classify the region that will be read as characters. It is not essential to extract features from the complete picture in this situation, but simply from the little area containing the plate. This will need cutting the zone where the interpretation will take place. Due to a variety of reasons, the photographs acquired in most instances have several defects such as noise, brightness, and insufficient illumination, which significantly affects the location of the area you need to get. However, as previously stated, several filters were applied to enhance the picture quality and therefore enable further stages. Using these filters in this order addresses most segmentation difficulties and allows for the conversion of the image to gray scale. Continuing with the library functions, the following steps are available. For thresholding or, in the case of the second step, for thresholding. This function accepts the following parameters: The first parameter specifies the source picture, which must be a grayscale image, in this instance the output of the preceding operation.

Class	Weight
Analysis	0.0109
Backdoor	0.0122
Benign	0.0000018287
DoS	0.0012
Exploits	0.0001
Fuzzers	0.0003
Generic	0.0014
Reconnaissance	0.0006
Shellcode	0.0059
Worms	0.0286

Figure 4.3: Classes and weights in the network inputs [22]

The second input specifies the weight value for ranking threat values. The third option, maxVal, specifies the value that should be returned if the threat value is larger than (or equal to) the limit in value:

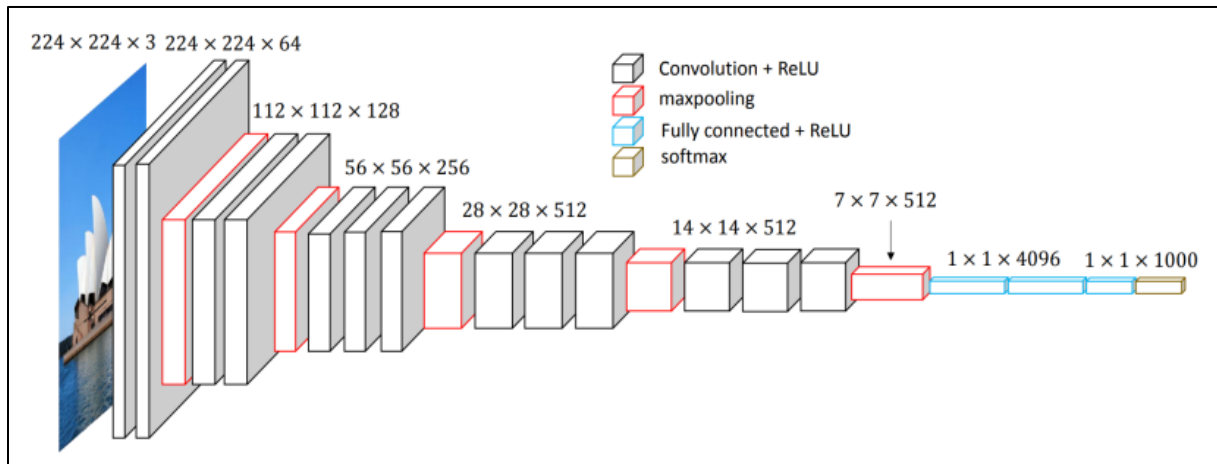


Figure 4.4: CNN layers for classification [23]

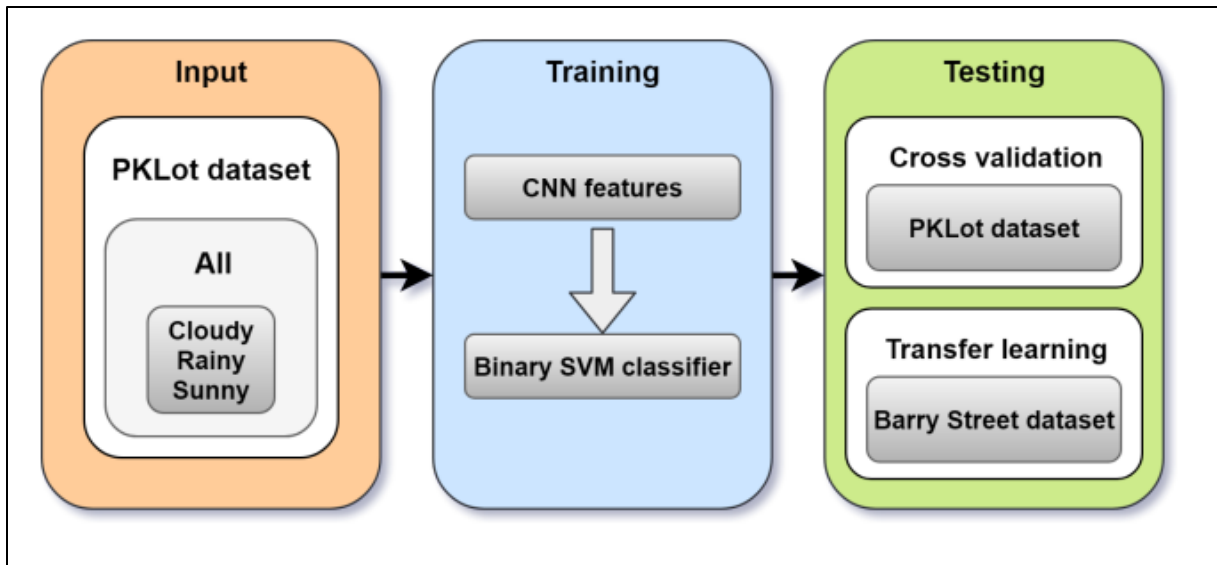


Figure 4.5: CNN framework [15]

For neural networks, the most important phase between training and testing is undoubtedly the first, in fact, it is in this phase that the model, by reading the data from a specially built dataset, is able to recognize from the simplest to the most complex relations that exist between them. This is why this step assumes considerable prestige, especially when the model, as well as its topology, is built step by step, starting from simple levels, up to combinations of parameters and increasingly sophisticated sub-levels. (The first) and practically (the second) how networks should be built, highlighting the fundamental aspects to be taken into account when carrying out a design of this type.

4.5 RESULTS

This subsection contains a complete overview of all the tested hyperparameters indicating also a small description, therefore in this section, in which the complete network will be presented, these will only be mentioned. The final architecture is shown in a simple way, but at the same time, enriched with the various parameters, this was obtained by the design software Netron1, which given the model generated by the python script as input, returns its graphic representation. it is composed of:

- a) An InputLayer in which the input data at the following levels are present, these represent the “n = 10” samples to be classified.
- b) Recurring section, where we find:
 - A first recurring hidden level made up of 128 GRU nodes, with “relu” activation function and uniform initializer;
 - A second recurring level composed, once again, of 128 GRU nodes, with hyperparameters similar to the previous one;
 - Flatten level: this is a level that makes it explicit how to serialize a multidimensional array (typically the input one). This allows the mapping between the input array and the first hidden layer. If the next level is a hidden layer and is "dense", therefore fully connected (completely connected), each element of the input vector (serialized) will be connected with each element of the hidden array. If flatten is not used, the way in which the mapping occurs on the first Dense layer would be ambiguous. This is because Dense layers accept a vector as input, not a matrix as with recurring layers.
- c) Completely connected section, where we find:
 - A first “Dense” level made up of 256 nodes, with “relu” activation function and uniform type initializer of the weights (as for the recurring levels);
 - Subsequently, other 3 “Dense” levels, similar to the previous ones, composed respectively of 128, 64 and 32 nodes.
- d) Finally, the last "Dense" level which represents the previously mentioned outputLayer, with the function of activating the "sigmoid" and initializer similar to the previous connected levels. This last level is the one that then provides the output of the model. All layers in the network are followed by a dropout layer of 20% value. The dropout is a tool used during the training of neural networks that aims to avoid overfitting by randomly removing some units in the network. The whole network is covered each time entirely at each epoch, in such a way as to update the weights in each sub-level from time to time, weights that take on a lot of importance in the recurring levels, since these memorize the past history within them, essential for speeding up the convergence of the internal model. Once the network architecture is complete, the model can be compiled, the compilation phase is characterized by many parameters, here are the main ones:

- e) Loss or cost function (loss): it is able to summarize in a single scalar value all the positive and negative aspects of the model, in order to simplify the comparison. The function selected for this network is the BinaryCrossEntropy (BCE), a choice dictated by the type of classifier, in this case it is a binary classifier, this function assumes a value between 0 and 1, and requires that the last level of the network has the function of activation the "sigmoid" and is composed of a single node.
- f) The type of optimizer: choice made in the "Adam" optimizer.

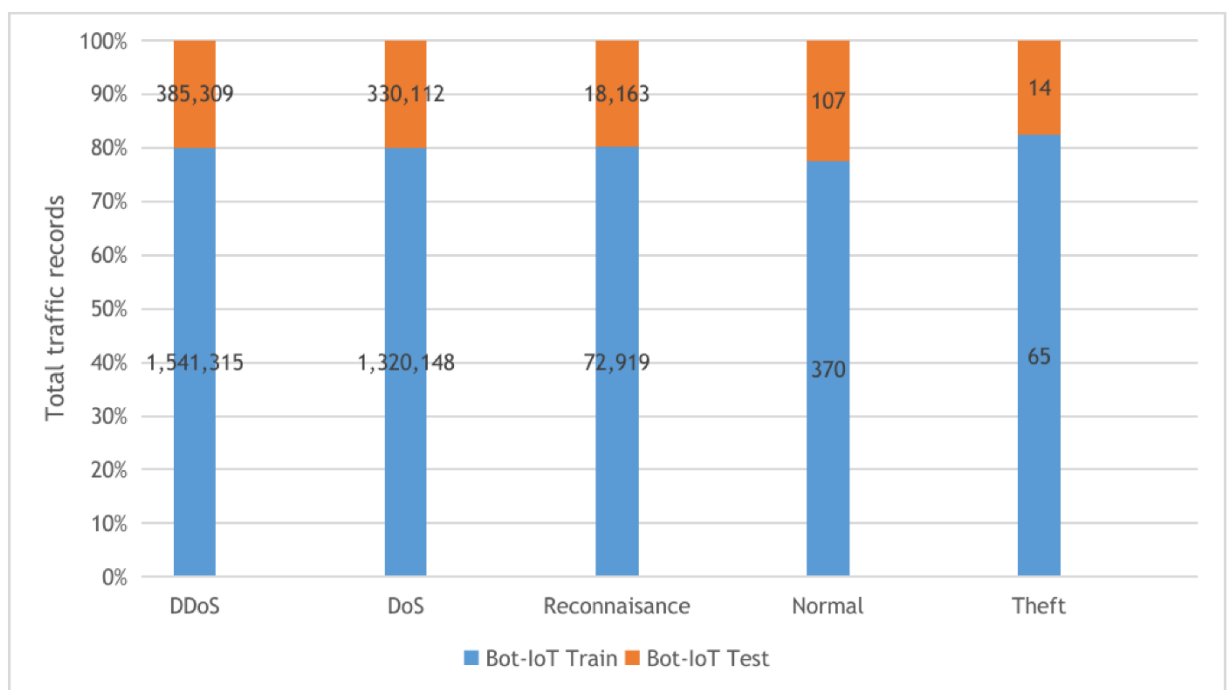


Figure 4.6: Simulation results after the ADAM optimizer

- g) Finally, the metric to consider, being a classification, the selected metric is accuracy, it assumes a value between 0 and 1, and the closer it is to 1, the more the model is training correctly. In any case, this value, in the train phase, assumes minimal importance, the true accuracy of a model must be evaluated on the test dataset; As for the loss function and the metrics, these are updated at each epoch. Once completed, the actual training can start, in which the number of epochs (50/100) and the batch-size to be considered (32) must be

specified. In order to effectively evaluate the performance of the model on non-train data during the training phase, it is possible to divide the input dataset; where a portion (usually 70-80%) is supplied to the network, while the remaining percentage is used to test performance at run-time. However, there are more sophisticated validation techniques where the division is not under the control of the designer but varies with each iteration; an example of a validation algorithm is cross-validation. However, the actual test should take place in a subsequent step to those of the train, with data that the model has never seen while it was training. In fact, before implementing these algorithms within the simulator, various tests were performed following precisely this approach. Then, once the training was completed, carried out in python, the model generated by this phase was exported to then be imported, first in special scripts to perform a validation and confirm its performance, and subsequently, if the results obtained were enough satisfactory was imported into the simulator; where, from time to time, each sample received from both sensors and IOT communications is able to detect attacks.

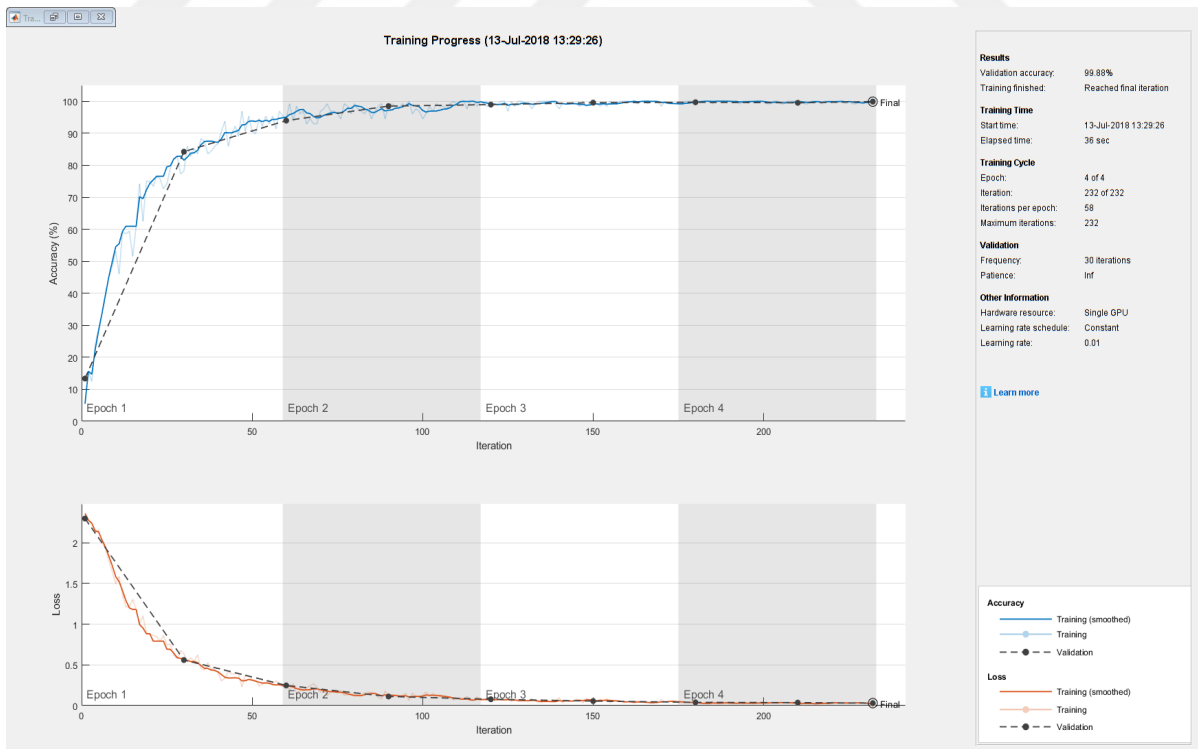


Figure 4.7: CNN training time and accuracy

Two instances should be investigated in order to have a better understanding of interactive techniques. If no ground truth is available, it is required to conduct annotations averaged over all classes as an evaluation measure, and they may be used to replicate them if ground truth is available. The neural network generates an initial inference for each picture on its own, without requiring any annotations, and it does it automatically. As a consequence of simulating a click, the network generates a new hypothesis. This procedure is repeated until a certain number of clicks has been made. The middle of one of the photo's bigger, misclassified parts is duplicated in this picture by comparing predictions to ground truth. Because randomness is used to determine the region and the placement of the click inside it, the simulation of human behavior is more realistic. Where classes in the dataset are underrepresented in comparison to others, we require that clicks be spread evenly among all classes in the dataset, so that the repairs benefit all classes in the dataset. It's all about the old-fashioned manner of doing things. In this instance, the clicks are done manually by a human operator. To address the most serious faults, this operator randomly inserts clicks throughout the software. During this manual review, the CNN confusion matrix is used to test categorization.

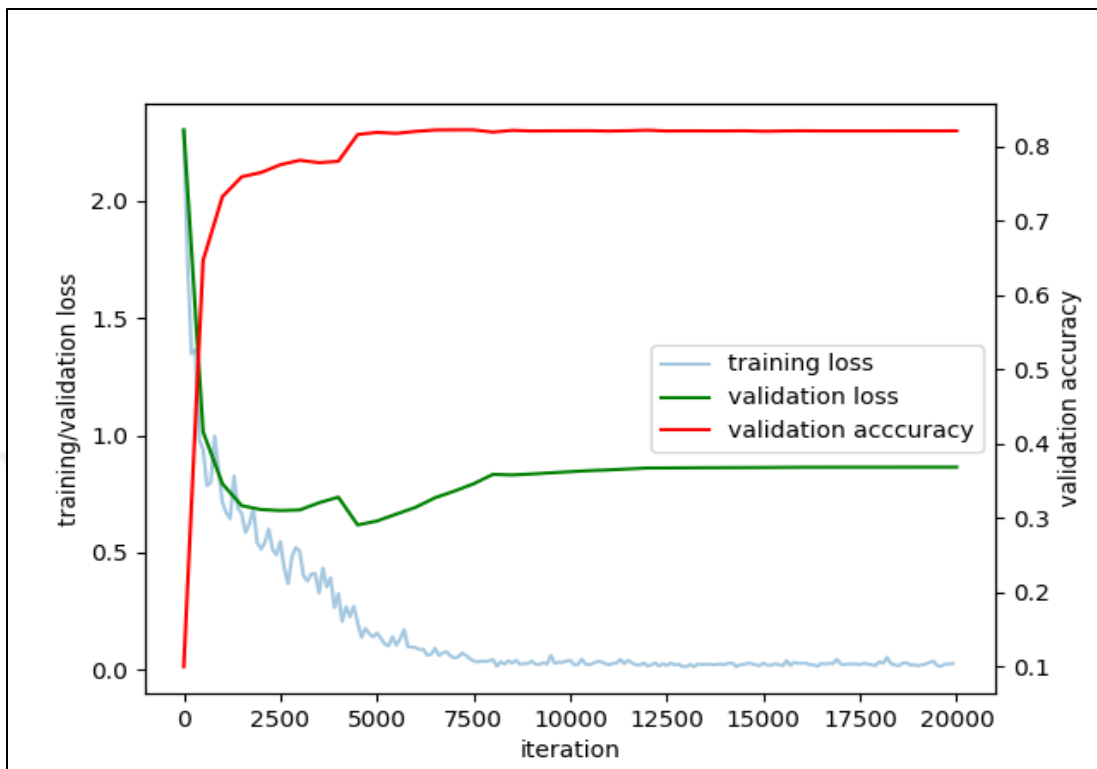


Figure 4.8: CNN accuracy and loss

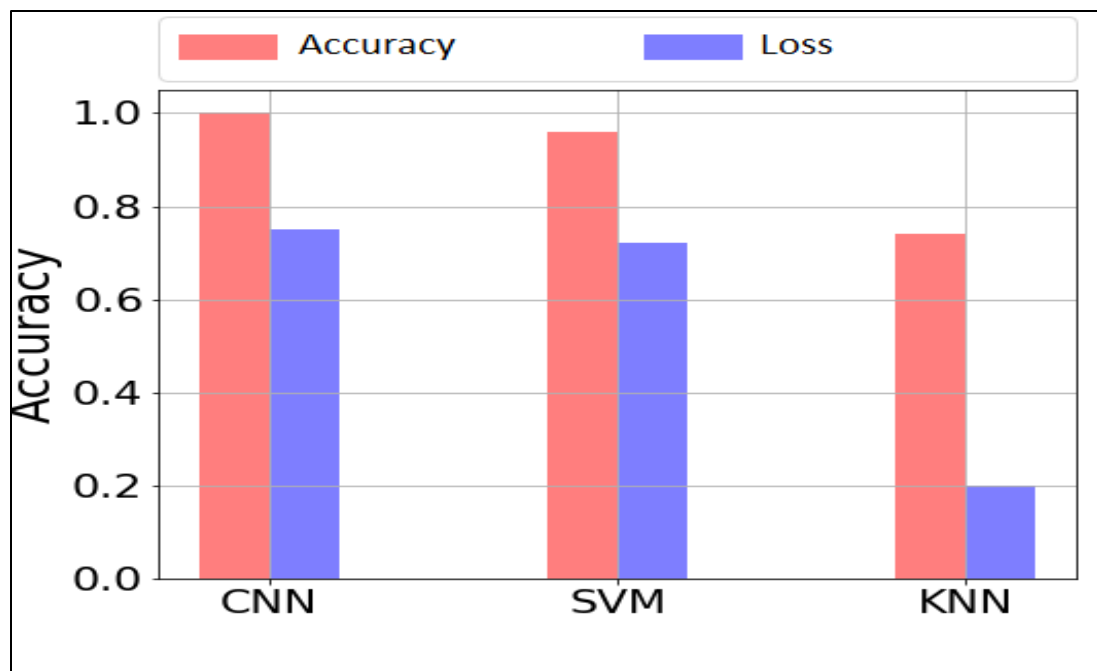


Figure 4.9: Comparing accuracies and loss in CNN, SVM and KN

5. CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

The CNN algorithm is a recent approach applied to anomaly detection. This algorithm also has many qualities that can be explored and many applications where it can be used, including intruder detection, in activities that are subsequent steps to the ping scan, for example. This algorithm also has some innovations:

- a. It does not require a normal and anomalous pattern profile, as in self/nonself based SIA models.
- b. Signal mapping, an essential characteristic of the algorithm, similar to the characteristics of signature detectors, but which are mapped according to the characteristics of the problem.
- c. Concept of signals: input data form the output data that represent the migration of a cell and its maturation.
- d. Weight matrix: relationship between input signals and output signals that makes the algorithm classify the cell in a mechanism similar to that of the immune system.
- e. Way of classifying the antigen and using an anomaly index calculated from the types of cells that collected it.
- f. Cell population-based method, similar to evolutionary and stochastic methods. With these characteristics, the CNN can be considered unique in the detection of anomalies, being able to be used in some complex problems for the classic approach of fault detection, since the use of the mapping of the signals contributes in the aspects justified for the reasons above.

The contributions of this work present a new point of view of the algorithm in relation to works in the literature and a confirmation of the advantages of CNN and the use of signals and antigens in correlation for the detection of anomalies. In the study of ping scans, it was

possible to obtain information about the detection process by CNN. The algorithm has the main advantage of not being tied to models based on dimensions, as in the classical approaches to fault detection, however, a deep knowledge about the context of the problem is necessary for the algorithm's response to be effective. CNN is a non-stationary algorithm, that is, it does not have a long-term trend and does not follow an absolute mean value. This concept is valid both for the analysis of antigens and signals, as well as for the MCAV of the antigens involved. The detection results can be considered interesting, since the detection reached a situation close to the desired one, with about 99% efficiency, a result that is difficult to obtain in the literature. However, this result is an extension to other intrusion situations, especially cases that occur in addition to a ping scan situation. However, intrusion techniques go beyond ping scans and require other parameters to be taken into account by a detector. The use of CNN in these problems is one of the many challenges that could be explored. A question would be: "Would the results obtained in this work be similar in such problems?". Finally, in terms of results, the objectives of the work were met, as the experiments are satisfactory and can be deployed for a test under different conditions, for example, real-time intrusions...

5.2 FUTURE WORK

Machine learning algorithm has been applied to the detection of intrusion principles, especially the detection of inappropriate use of ping scans, including the scan. In this work, the applied scenarios were based on the first tests to detect ping scans. The same can be applied to the scan, from which various information can be extracted. Considering that a simulation model was implemented in this work, a continuity proposal implies testing the detection of the ping scan in the CNN in real time with the changes made and verifying the difference in relation to the model and the results of the literature, in addition to testing the algorithm in other programming languages and comparisons with other methods in the literature, such as Neural Networks, statistical methods, etc. The application of the algorithm in real scenarios, based on tests performed at work, would consist of CNN running on all machines on the network. In this scenario, intrusions must be signaled by the hosts themselves, where the intrusion is taking place. In the future, in more advanced studies, this

detection may be extended to network servers, corresponding to an approach based on NIDS. In addition to these proposals, CNN can be applied to other anomalies in computer networks, such as the detection of denial-of-service attacks or system vulnerabilities; and also fault detection in complex systems, which can be a tool incorporated into mechanisms similar to fault detection. Other improvements can also be applied to CNN, such as adaptability mechanisms, allowing a behavior similar to adaptive systems; decision mechanisms, improving the algorithm's response to the detected anomaly; and memory mechanisms, allowing the same occurrence to be detected under the same conditions, causing the same antigen to be reevaluated in other cases

REFERENCES

- [1] OH, Doohwan; KIM, Deokho; RO, Won Woo. A malicious pattern detection engine for embedded security systems in the Internet of Things. *Sensors*, , 14.12: 24188-24211.2014.
- [2] LEE, Tsung-Han, et al. A lightweight intrusion detection scheme based on energy consumption analysis in 6LoWPAN. In: *Advanced technologies, embedded and multimedia for human-centric computing*. Springer, Dordrecht,. p. 1205-1213.2014.
- [3] E. Cho, J. Kim, C. Hong, Attack model and detection scheme for botnet on 6LoWPAN, in: C. Hong, T. Tonouchi, Y. Ma, C.-S. Chao (Eds.), *Management Enabling the Future Internet for Changing Business and New Computing Services*, Vol. 5787 of *Lecture Notes in Computer Science*, *Springer Berlin Heidelberg*, , pp. 515–518. 4. 2009.
- [4] KASINATHAN, Prabhakaran, et al. Denial-of-Service detection in 6LoWPAN based Internet of Things. *IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)*. IEEE, p. 600-607.2013.
- [5] AMARAL, João P., et al. Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks. *IEEE international conference on communications (ICC)*. IEEE,. p. 1796-1801.2014.
- [6] LIU, Caiming, et al. Research on immunity-based intrusion detection technology for the Internet of Things. *Seventh International Conference on Natural Computation*. IEEE,. p. 212-216.2011.
- [7] THANIGAIVELAN, Nanda Kumar, et al. Distributed internal anomaly detection system for Internet-of-Things. *IEEE annual consumer communications & networking conference (CCNC)*. IEEE, . p. 319-320.2016.
- [8] AMARAL, João P., et al. Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks. *IEEE international conference on communications (ICC)*. IEEE,. p. 1796-1801. 2014.

- [9] KRIMMLING, Jana; PETER, Steffen. Integration and evaluation of intrusion detection for CoAP in smart city applications. *IEEE Conference on Communications and Network Security*. IEEE,. p. 73-78. 2014.
- [10] SUMMERVILLE, Douglas H.; ZACH, Kenneth M.; CHEN, Yu. Ultra-lightweight deep packet anomaly detection for Internet of Things devices. In: 2015 IEEE 34th international performance computing and communications conference (IPCCC). IEEE,. p. 1-8.2015.
- [11] STEPHEN, R.; AROCKIAM, L. Intrusion detection system to detect sinkhole attack on RPL protocol in Internet of Things. *International Journal of Electrical Electronics and Computer Science*, 4.4: 16-20.2017.
- [12] PONGLE, Pavan; CHAVAN, Gurunath. Real time intrusion and wormhole attack detection in internet of things. *International Journal of Computer Applications*, 121.9. 2015.
- [13] Shapla Khanam, Ismail Ahmedy and Mohd Yamani Idna Idris, An efficient detection of selective forwarding attacks in heterogeneous IoT Networks, *FCSIT*, : pp 1-8. 2017.
- [14] ZHANG, Kuan, et al. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal*, 1.5: 372-383.2014.
- [15] MAHJABIN, Tasnuva, et al. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, 13.12: 1550147717741463.2017.
- [16] CHRUN, Danielle. Model-based support for information technology security decision making. University of Maryland, College Park, 2011.
- [17] MAHJABIN, Tasnuva, et al. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, 13.12: 1550147717741463.2017.
- [18] Raviteja gaddam1 & dr. M. Nandhini, Analytical approach to enhance the intrusion detection in internet of things network,international journal of latest trends in engineering and technology, Vol.(9) issue(3), pp.258- 267.

- [19] V.Gayathri, Malatesh S H, Smart Intrusion Detection System for HomeSecurity, *International Journal of Science, Engineering and Technology*. 20. Aastha Puri¹, Nidhi Sharma, A Survey On Intrusion Detection System, *IJEDR* | Volume 5, Issue 2 | ISSN: 2321-9939.2017.
- [20] Szarvas, M., Yoshizawa, A., Yamamoto, M., and Ogata, J. Pedestrian detection with convolutional neural networks. *IEEE Proceedings. Intelligent Vehicles Symposium*, ,pages 224–229. 2005.
- [21] True, N. . Vacant parking space detection in static images. University of California, San Diego, 17. 2007.
- [22] TSAI, Luo-Wei; HSIEH, Jun-Wei; FAN, Kuo-Chin. Vehicle detection using normalized color and edge map. *IEEE transactions on Image Processing*, , 16.3: 850-864.2007.
- [23] RASHID, Muhammad Mahbubur, et al. Automatic parking management system and parking fee collection based on number plate recognition. *International Journal of Machine Learning and Computing*, 2.2: 94. 2012.
- [24] JIAN, Ming-Shen; YANG, Kuen Shih; LEE, Chung-Lun. Modular RFID parking management system based on existed gate system integration. *WSEAS Transactions on systems*, 7.6: 706-716. 2008.
- [25] TSIROPOULOU, Eirini Eleni, et al. RFID-based smart parking management system. *Cyber-Physical Systems*, , 3.1-4: 22-41.2017.
- [26] WEI, Lanxin, et al. Design and implementation of smart parking management system based on rfid and internet. *International conference on control engineering and communication technology*. IEEE. p. 17-20. 2012.
- [27] BI, Y., et al. A parking management system based on wireless sensor network. *Acta Automatica Sinica*, 2006, 32.6: 968.
- [28] VERA-GÓMEZ, Juan A., et al. An intelligent parking management system for urban areas. *Sensors*, 16.6: 931.2016.

- [29] GANDHI, BM Kumar; RAO, M. Kameswara. A prototype for IoT based car parking management system for smart cities. *Indian Journal of Science and Technology*, 9.17: 1-6.2016.
- [30] SADHUKHAN, Pampa. An IoT-based E-parking system for smart cities. *International conference on advances in computing, communications and informatics (ICACCI)*. IEEE. p. 1062-1066.2017.
- [31] ABDULKADER, Omar, et al. A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT. *Learning and Technology Conference (L&T)*. IEEE. p. 102-106.2018.
- [32] LOU, Liangliang, et al. An IoT-driven vehicle detection method based on multisource data fusion technology for smart parking management system. *IEEE Internet of Things Journal*, 7.11: 11020-11029. 2020.
- [33] LOU, Liangliang, et al. An IoT-driven vehicle detection method based on multisource data fusion technology for smart parking management system. *IEEE Internet of Things Journal*, 7.11: 11020-11029.2020.
- [34] Chandra, H., Hadisaputra, K. R., Santoso, H., Anggadajaja, E., Smart parking management system: An integration of RFID, ALPR, and WSN. In: *IEEE International Conference on Engineering Technologies and Social Sciences (ICETSS)*, pp. 1–6 2017
- [35] Al-Kharusi, H., Al-Bahadly, I., Intelligent Parking Management System Based on Image Processing. *World Journal of Engineering and Technology*, vol.2014
- [36] LIN, Sheng-Fuu; CHEN, Yung-Yao; LIU, Sung-Chieh. A vision-based parking lot management system. *IEEE International Conference on Systems, Man and Cybernetics*. IEEE. p. 2897-2902.2006.
- [37] HOUBEN, Sebastian, et al. On-vehicle video-based parking lot recognition with fisheye optics. In: *International IEEE Conference on Intelligent Transportation Systems (ITSC) IEEE*, 2013. p. 7-12. 2013.
- [38] HAMADA, Kazukuni, et al. Surround view based parking lot detection and tracking. In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE,. p. 1106-1111.2015.

- [39] Jayakshei Dadaji Bachhava, Mechkul “Smart Car Parking System”, *International Research Journal of Engineering and Technology (IRJET)* Volume: 04 Issue: 06, June - 2017.
- [40] RANE, Sagar; DUBEY, Aman; PARIDA, Tejisman. Design of IoT based intelligent parking system using image processing algorithms. *International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE. p. 1049-1053. 2017.
- [41] KHANNA, Abhirup; ANAND, Rishi. IoT based smart parking system. *International Conference on Internet of Things and Applications (IOTA)*. IEEE,. p. 266-270. 2016.
- [42] Chungsan Lee, Youngtak Han, Soobin Jeon, Dongmahn Seo*, Inbum Jung “Smart Parking System for Internet of Things”,*IEEE International Conference on Consumer Electronics*. 2016.
- [43] SR, Mr Basavaraju. Automatic smart parking system using Internet of Things (IOT).*International Journal of Scientific and Research Publications*, 5.12: 629-632.2015.
- [44] FastPark System website, <http://www.fastprk.com>.
- [45] ZHOU, Fengli; LI, Qing. Parking guidance system based on ZigBee and geomagnetic sensor technology. In: 2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science. IEEE, p. 268-271.2014.
- [46] JIAN, Ming-Shen; YANG, Kuen Shih; LEE, Chung-Lun. Modular RFID parking management system based on existed gate system integration. *WSEAS Transactions on systems*, 7.6: 706-716.2008.
- [47] TSIROPOULOU, Eirini Eleni, et al. RFID-based smart parking management system. *Cyber-Physical Systems*, 3.1-4: 22-41.2017.
- [48] WEI, Lanxin, et al. Design and implementation of smart parking management system based on rfid and internet. In: 2012 International conference on control engineering and communication technology. IEEE, p. 17-20.2012.
- [49] BI, Y., et al. A parking management system based on wireless sensor network. *Acta Automatica Sinica*, 32.6: 968.2006.

- [50] VERA-GÓMEZ, Juan A., et al. An intelligent parking management system for urban areas. *Sensors*, 16.6: 931.2016.
- [51] GANDHI, BM Kumar; RAO, M. Kameswara. A prototype for IoT based car parking management system for smart cities. *Indian Journal of Science and Technology*, 9.17: 1-6. 2016.
- [52] SADHUKHAN, Pampa. An IoT-based E-parking system for smart cities. In: *International conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2017. p. 1062-1066.2017.
- [53] ABDULKADER, Omar, et al. A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT. *Learning and Technology Conference (L&T)*. IEEE, p. 102-106.2018.
- [54] LOU, Liangliang, et al. An IoT-driven vehicle detection method based on multisource data fusion technology for smart parking management system. *IEEE Internet of Things Journal*, 7.11: 11020-11029.2020.
- [55] UR REHMAN, Shafiq; GRUHN, Volker. Recommended architecture for car parking management system based on cyber-physical system. In: *International Conference on Engineering & MIS (ICEMIS)*. IEEE, p. 1-6. 2017.
- [56] FARIZA, Ms Anis. BACHELOR IN ELECTRONIC & ELECTRICAL ENGINEERING COURSEWORK SUBMISSION COVER SHEET. 2019.
- [57] AL-KHARUSI, Hilal; AL-BAHADLY, Ibrahim. Intelligent parking management system based on image processing. *World Journal of Engineering and Technology*, 2014.
- [58] LIN, Sheng-Fuu; CHEN, Yung-Yao; LIU, Sung-Chieh. A vision-based parking lot management system. *IEEE International Conference on Systems, Man and Cybernetics*. IEEE, p. 2897-2902.2006.
- [59] HOUBEN, Sebastian, et al. On-vehicle video-based parking lot recognition with fisheye optics. In: *International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE,p. 7-12.2013.