FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

DETECTION AND ANALYSIS OF CYBER-ATTACKS ON IOT NETWORK DEVICES

Bashir Zak ADAMU

Master's Thesis

DEPARTMENT OF DIGITAL FORENSIC ENGINEERING

FEBRUARY 2022

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Digital Forensic Engineering

Master's Thesis

DETECTION AND ANALYSIS OF CYBER-ATTACKS ON IOT NETWORK DEVICES

Author
Bashir Zak ADAMU

Supervisor
Assoc. Prof. Fatih ERTAM

FEBRUARY 2022
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

TÜRKİYE

Department of Digital Forensic Engineering

Master's Thesis

Title:	Detection and Analysis of Cyber-attacks on IoT Network Devices
Author:	Bashir Zak ADAMU
Submission Date:	03 January 2022
Defense Date:	03 February 2022

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

		Signature
Supervisor:	Assoc. Prof. Fatih ERTAM Firat University, Faculty of Technology	Approved
Chair:	Assistant Prof. Mustafa KAYA Firat University, Faculty of Technology	Approved
Member:	Assistant Prof. Hafzullah İŞ Batman University, Faculty of Engineering	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / ..... / 20 .....

Signature

Prof. Dr. Kürşat Esat ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled “Detection and Analysis of Cyber-attacks on IoT Network Devices” in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

03 February 2022

Bashir Zak ADAMU

PREFACE

In the near future, almost every object will be connected to the internet thanks to IoT technology. These objects can be made targets for cyber-attacks by adversaries since there will be a lot to choose from. Developing systems capable of detecting and preventing such attacks will give cyber security and forensic experts an upper hand in the war of cyber-attacks.

The National Information Technology Development Agency (NITDA) has supported me financially while my able supervisor Associate Professor Fatih Ertam has supported me technically, financially, and spiritually.

Much gratitude also goes to my Dad Major. Z.Y Adamu (Rtd) and Mom Umma Jummai Zakar for the prayers and never-ending support they have been giving to me. The same goes for my wife Hauwa Isah and daughters Humairah, Rumaisa, and Amnah for cheering me up and bearing with my absence.

Bashir Zak ADAMU
ELAZIG, 2022

TABLE OF CONTENTS

Page

ABSTRACT

Detection and Analysis of Cyber-attacks on IoT Network Devices

Bashir Zak ADAMU

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Digital Forensic Engineering

February 2022, Page: xi + 58

One of the most pressing concerns in network forensics is the detection of cyber-attacks in the IoT networks and their devices. Traditional intrusion detection systems based on signature rules are unable to detect current attack types. Hence, the need to urgently develop advanced methods for classifying IoT network traffic that can swiftly detect cyber-attacks becomes inevitable. This research aims to develop machine learning algorithms for cyber-attack detection in IoT-based networks, by analyzing the traffic data composed from the network itself. An ideal IoT network was implemented solely for the attack scenarios and generation of a dataset. The self-generated dataset from the IoT network was utilized for the comparison of machine learning algorithms suitable for the classification of attack-to-normal network traffic data. The algorithms are expected to classify the multi-class data into nine classes (normal traffic inclusive).

Keywords: Intrusion Detection, Cyber-attack, IoT, Machine Learning

ÖZET

IoT ağ cihazlarındaki Siber saldırıların Tespiti ve Analizi

Bashir Zak ADAMU

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Adli Bilişim Mühendisliği Anabilim Dalı

Şubat 2022, Sayfa: xi + 58

Adli bilişim ve siber güvenlik alanındaki en önemli zorluklardan birisi de nesnelerin interneti (IoT) ağlarındaki siber saldırıların tespit edilmesidir. İmza tabanlı geleneksel saldırı tespit sistemleri tüm saldırıları tespit edemez. Bu nedenle, özellikle IoT ağlarındaki siber saldırıları hızlı bir şekilde tespit etmek ve ağ trafiğini sınıflandırmak için çeşitli yöntemlerin geliştirilmesi çok önemlidir. Bu tez çalışmasında bir IoT ağı oluşturularak, çeşitli saldırı senaryoları oluşturulmuş ve ağ trafiği kayıt altına alınmıştır. Kayıt altına alınan ağ trafiği çeşitli makine öğrenmesi yaklaşımları ile analiz edilmiş ve yüksek başarım değerlerine ulaşılmıştır.

Anahtar Kelimeler: İzinsiz giriş Tespiti, Siber Saldın, Nesnelerin İnterneti, Makine Öğrenmesi

LIST OF FIGURES

Page

LIST OF TABLES

Page

SYMBOLS AND ABBREVIATIONS

Abbreviations

IoT	: Internet of Things
IDS	: Intrusion Detection System
SBC	: Single Board Computer
SVM	: Support Vector Machine
KNN	: K-Nearest Neighbor
FFNN	: Feed Forward Neural Network
PCA	: Principal Network Analysis
WSN	: Wireless Sensor Networks
RFID	: Radio Frequency Identification

1. INTRODUCTION

For more than a century, mankind has been using automatic sensors and controls, calling them sensors (from the word feel and measure), and has been using them to connect computers. Furthermore, we are currently amidst a remarkable technological revolution that offers unprecedented opportunities but also poses the huge risks associated with these changes. It is now possible to use extremely inexpensive controllers and sensory devices with amazingly exponential capability, availability, and growing varieties. Following their unconditional usefulness and economic feasibility, technologies based on IoT in green buildings, environmental monitoring, health and safety monitoring of facilities have already been deployed. We buy cars with already built-in sensors and controls or, if you prefer, with the IoT (although cars, for the most part, are not directly linked to the web). With the continuous unprecedented widespread of IoT technology, they became a soft target for cyber-attacks due to their nature, mode of operation, and availability. As such, cyber-attacks on IoT networks and devices became prevalent everywhere, making it a difficult concern for forensic experts. Hence, the best way for tackling these attacks is to detect them in real-time while they occur, a terminology known as “live forensics”[1].

Intrusion Detection Systems (IDSs) are one promising component of the live forensic approach for monitoring IoT networks and are essentially effective at the cyber level. IDSs deployed in IoT environments analyse network traffic packets and produce real-time responses [2]. But for these IDSs to be effective, they must operate under rigorous IoT conditions of low energy, low process capacity, quick response, and eminently gigantic volumes of information handling. Hence, augmenting IoT embedded IDSs significantly requires a continuous and serious understanding of the security vulnerabilities posed by IoT networks

1.1. Problem Definition

Firstly, according to 2019 data from PwC (an international network of companies offering consulting and audit services), the IoT is currently an actively developing technology amassing trust for changing the business models of both companies and entire industries [3]. In the presented rating of Forbes [4], and the publication [5], "IoT" is ahead of technologies such as artificial intelligence, robotics, augmented and virtual reality, with a projected number of connected device base of approximately 10.07 billion by 2021 which is expected to increase to 25.44 billion by 2030 as shown in Figure 1.1. According to [6], the machine-to-machine (M2M) connections count is expected to reach 27 billion by 2024.

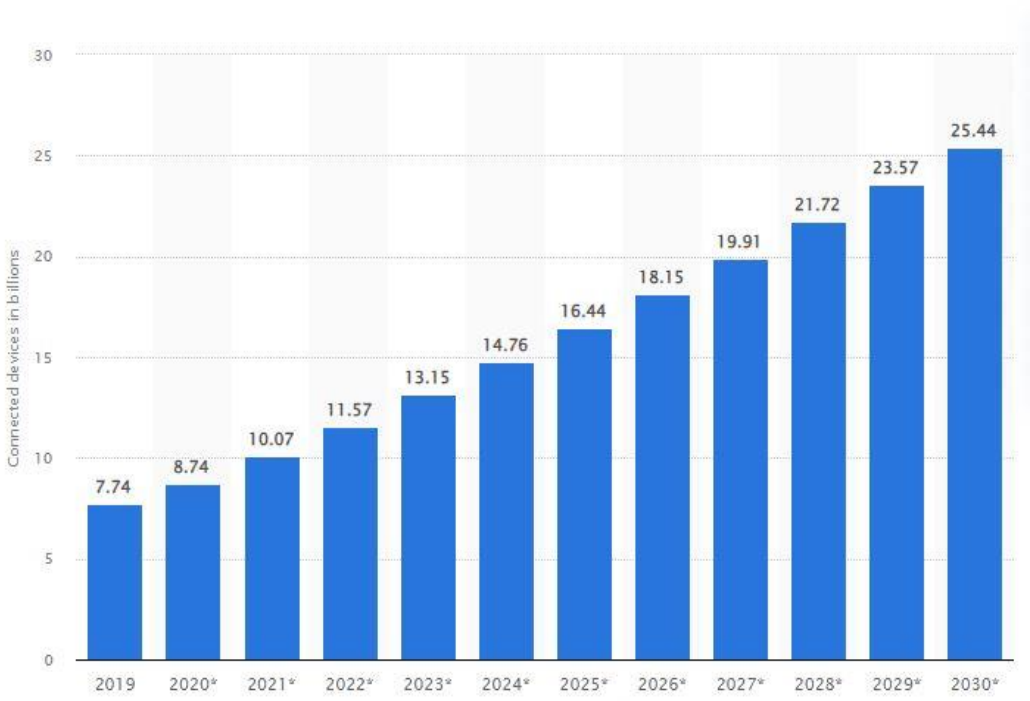

Figure 1.1. Projected Estimate of IoT Connected Devices from 2019 to 2030 [7]

Secondly, with the hike in I.o.T devices globally, all objects that are part of the IoT become a target and are under threat: objects of the Department of Defence and critical infrastructures such as employees, equipment, and other properties may be used to initiate IoT security attacks. If not looked into appropriately, adversaries who are determined may end up undermining the security and processes that will bring down systems like power grids, medical access, communication e.t.c. One way or another, all these reputable departments are closely studying the applications of IoT in various systems. If no action is taken to get around the hike in the IoT device base, the problem will grow exponentially further, and the growth of IoT could sweep these structures like a tidal wave.

Furthermore, IoT attack detection has recently proved to be a pressing issue in the atmosphere of securing corporate networks. Large-scale outbreaks of network worms, DDoS attacks from botnets, automated tools for finding vulnerabilities in networks - all this makes securing IoT networks a very laborious task. Nowadays it is difficult to find a network that lacks such active means of preventing attacks as antivirus, firewall, and intrusion prevention systems. Unfortunately, active countermeasures alone are not enough. Therefore, in addition to them, passive means of fighting attacks are used - network intrusion detection systems.

1.2. Motivation

Connecting every smart object as well as the internet results in numerous security and privacy challenges. Confidentiality, Authenticity and Data Integrity as basic security elements are threatened because most IoT devices lack security by design. A malicious adversary may not even need to hack most components of the IoT because they lack authentication components. Hence, the need to detect and prevent such attacks in real-time becomes paramount.

Furthermore, Intrusion detection tools that use only the signature-based intrusion detection method in their operation cannot detect new types of attacks or modifications of old ones; therefore, the task of developing better algorithms for detecting IoT network traffic anomalies becomes paramount. Also, some researchers suggest the application of machine learning principles to develop up-to-the-task IDSs that can eliminate the difficulties faced by the traditional approaches.

1.3. Contribution

- The design and implementation of a real-time IoT network.
- Practical execution of various IoT cyber-attacks on the IoT network devices.
- Self-generation of the dataset used for classification
- Design and implementation of an automated Machine learning algorithm for detecting and classifying the experimental attacks for security and forensic purposes

1.4. Concept & Attributes of the IoT

The term "Internet of Things" (hereinafter abbreviated as IoT) is a concept of a data transmission network between physical objects equipped with various modules to interact with each other through the use of data transmission networks. IoT technology can therefore be explained as a collection of tools and devices that collate data, transfer, process, and analyze it. The main components of this technology are sensors with transmitters, base stations, network servers, and application servers. The sensor collects information from the external environment and periodically sends the data to the base station via the wireless module. A feature of IoT technology is the combination of wireless sensors and very low-power transmitters. The useful life of the transmitter is measured in a few years. Low power consumption has made it possible to introduce IoT technology. If you need to connect more IoT devices to a larger area, you can connect using an existing cellular network or using an LPWAN (that is, a Low Power Wide Area Network).

The battery life of LoRaWAN devices can reach several years, which makes them ideal for use in distant locations or volatile systems [8]. The network server amasses signals from the sensors, "extracts" valuable data, and subsequently conveys it to the application server. Here, the server allows you to interpret the data received from the sensors and provide it to the user. It is the

application server that is the analytical center that interprets data for the end-user. Today there are already cloud services that provide application server functions. In developed countries, IoT technologies are magnificently deployed in industry, transport, agriculture, healthcare, and defense, where a large number of successful public projects are also known [9].

The IoT is a methodology for connecting information originating from different sources on a virtual platform/prevailing Internet infrastructure. The concept of the IoT emerged in 1982 when an altered soda machine with an Internet connection reported the presence of drinks and their temperature [10]. Later, in 1991, Mark Weiser gave the first modern assessment of the IoT. Besides, Bill Joy hinted about possible communication between devices in his Internet taxonomy in the year 1999 [11]. That same year, Kevin Ashton later formulated the term "IoT" for connected devices. The International Telecommunications Union (ITU) [12] in 2005 heralded the era of ubiquitous networks, the main feature of which is the connection between networks (IoT).

The idea behind IoT technology is to enable automated information exchange amongst distinctively identifiable devices in the real world. The devices can listen to control and data which can be processed to achieve the desired task through device training. The practical implementation of IoT is well demonstrated in “Twine” low-power compact hardware that works in real-time with networking software to actualize the reality of the concept [13]. Nonetheless, different people/ organizations perceive the concepts of the IoT differently. For the effective IoT deployment in environments, some attributes have to be considered like the ones entailed below;

First, sensors (sensors) are the main attribute of the IoT. A sensor is an electronic device that measures a physical state or chemical composition and transmits a signal consistent with the observed characteristic. Examples include thermometers, microphones, cameras, etc.

Secondly, actuators act as a trigger, which initiates physical activity after a signal from the sensors. A wide variety of devices can fulfill this role: relays, chemical dispensers, power generators, speakers, electronic locks, etc.

Naturally, for these elements to interact with each other, computers, servers, and computing cores are required, which will provide the logic of interaction between these devices. An important fact is that this logic is provided by a person, but given the active development of information technologies, a situation will arise in which algorithms will perform the role of a person much more efficiently.

For the devices that read information and respond to the received signal to be interconnected with the computing core, data transmission networks are needed that determine the methods of data exchange between objects.

The most important attribute that makes the “IoT” technology not just an automated system, but a “smart” platform is code. Code is programmed algorithms in the core of the system, which determines the meaning of the interaction between sensors and actuators.

All of the above attributes taken together, forming the "IoT" technology, are capable of providing fast access to information that can affect both the speed of sales, the quality of service, and the development of more efficient devices and applications. The concept can be understood more from Figure 1.2 below;

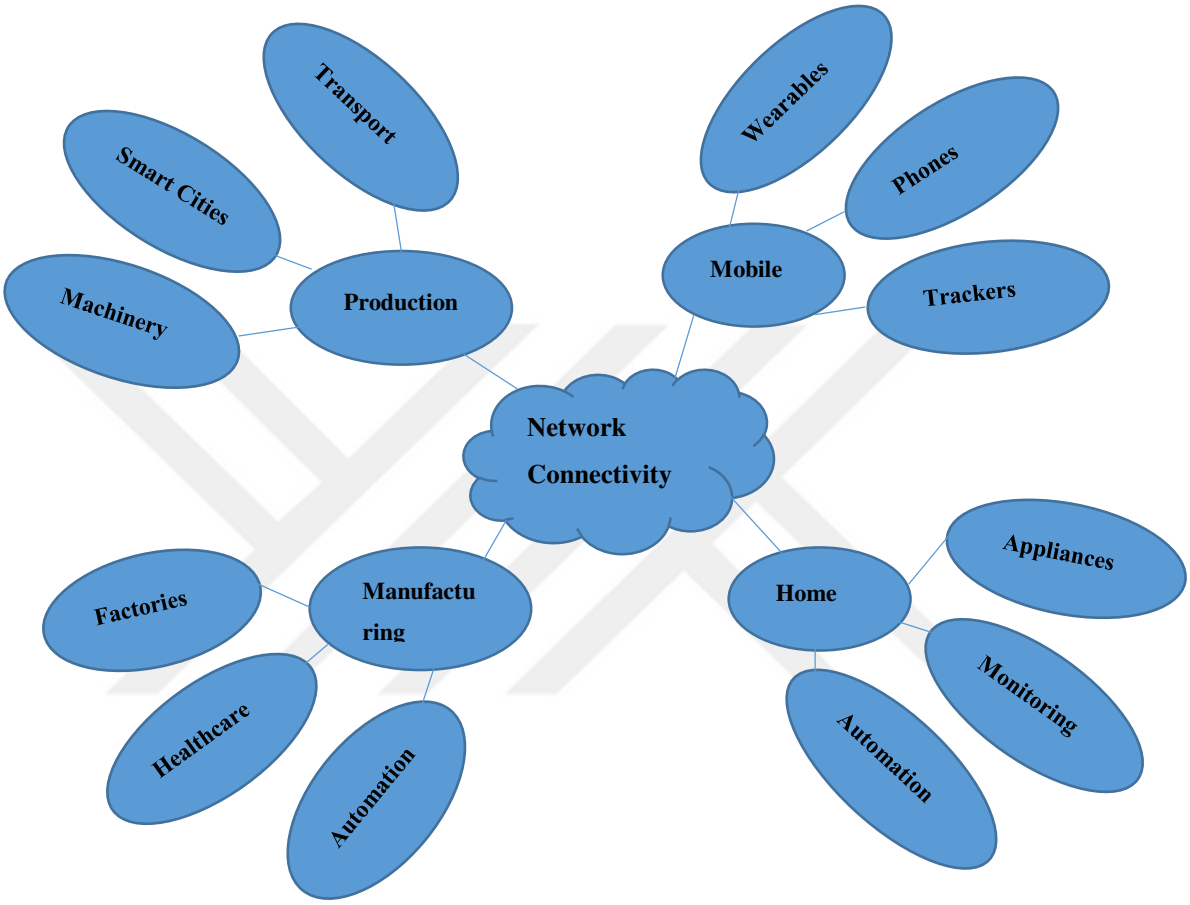

Figure 1.2. The concept of the IoT

2. LITERATURE REVIEW

2.1. Review of Related Works

Most IoT security researchers see machine learning techniques as part of their approach to detecting attacks and anomalous device behavior [14]. The main advantages of machine learning over traditional signature techniques are high performance and scalability for growing amounts of data, and the ability to automatically select useful features from raw data.

Similarly, the paper [15] explores the potential of recurrent neural networks with LSTMs for detecting IoT malware. The developed model is compared to classifiers based on traditional machine learning techniques: Support Vector Machine (SVM), Naive Bayes Classifier, Random Forest, Adaptive Boosting (AdaBoost), K-Nearest Neighbors (KNN). Analysis shows that the deep learning approach gives the best possible results. However, no comparison with other deep learning models was made.

In a study [16], The authors propose a unique anomaly detection system for industrial IoT systems that uses an autoencoder and a deep feedforward neural network. The developed model is compared to the properties of several developed anomaly detection methods such as Deep Trust Web, Recurrent Network, DNN, and EnsembleDNN. At the same time, the presented model was evaluated on different hardware and software with different subsets of initial data.

The article [17] proposed a decentralized cloud-based deep learning environment for the discovery prevention of phishing and botnet attacks on smart devices. The developed RNN-LSTM model is compared to the deep learning model developed by other researchers. The main drawback of comparative analysis in this task is that the DNN model under consideration is not evaluated on the same dataset.

The authors in [16] analyzed several deep learning methods for detecting DDoS attacks. Multilayer perceptron, convolutional neural network, RNN-LSTM, CNN + LSTM-ensemble. These are further compared to machine learning techniques. Support vector machine, Bayes classifier, and random forest were considered. The authors concluded that deep learning techniques, especially recurrent networks, are more effective.

In the study by [18], methods were identified to prevent attacks on IoT networks by using an IoT monitoring system. At the same time, the study considered a special module for monitoring traffic within the IoT network, which allows detecting attempts to carry out attacks on devices of this network. A machine learning approach was also employed.

2.2. IoT Architectures

For IoT, three main characteristics are defined - comprehensive knowledge (as a result of obtaining information about an object, anywhere and at any time), reliable transmission (using communication protocols, routing, encryption, network security, with real-time high accuracy), Intelligent Processing (taking into account a lot of calculations, fuzzy identification and other technologies for analyzing and handling Big data by various users). Following these features, the IoT structure was primarily classified into the perceptual layer, network layer, and application layer [19]. Other classifications can be seen in Figure 2.1.

2.2.1. The Perception Layer

This is a layer of IoT devices that gives each object a physical denotation. It comprises several kinds of data sensors like IR sensors and RFID tags, as well as other sensory networks that can read the location, speed, temperature, and humidity of objects. Moreover, it is also responsible for the collection of valuable data about objects from the sensors connected to them, thereby converting the information into digital signals ready to be transmitted for further processing to the network stage.

2.2.2. The Network Layer

This layer is responsible for providing pervasive access, information processing, transfer, and storage by receiving digital signals of valuable information from the perceptual level, thereby transferring it to the processing systems. It consists of an access layer (mobile communication networks), and a basic exchange layer (Internet, NGN next-generation networks, virtual private networks). Most sensor networks use wireless communication networks like Bluetooth (WPAN), Wi-Fi (WLAN), WiMAX (WMAN), 4G networks (WWAN), GPS (satellite network). Sensor networks in the IoT use IP-based communication protocols such as IPv6 or MQTT.

2.2.3. The Application Layer

This is the layer responsible for using processed data to implement all kinds of IoT-based applications. It manages and controls applications as well as services by taking the right decisions after analyzing/processing the received information from the network layer. At this same level, functions are performed to collect and store data, for ensuring the supply of efficient energy and other related processes. This layer is also useful for the large-scale development of IoT networks.

With the current hike in IoT deployment, the current architecture of the Internet, as well as its protocols (TCP / IP), can’t cope with IoT as a large network [20]. Hence, a new uncluttered

architecture that is capable of reporting security issues and ensuring Quality of Service (QoS) while supporting current network applications using open protocols [21], [22].

The implementation of IoT must embed proper security for guaranteed results. Hence, privacy and protection of data become paramount for IoT adoption to prevent further challenges. Many layered security-wise architectures have been suggested for the further development of the IoT. Some architectures including the above-stated structure are depicted below;

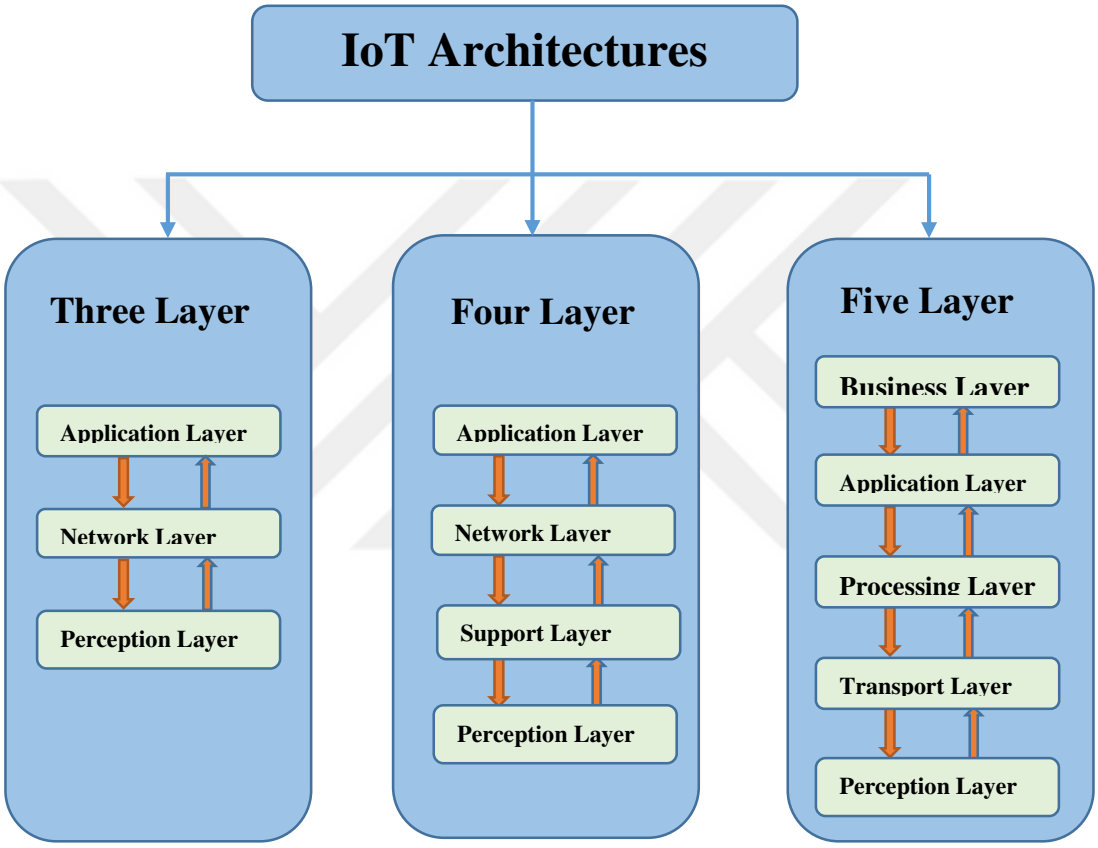

Figure 2.1. IoT Architectures Based on hierarchical network structure

2.3. Technologies Related to IoT

The advance of pervasive computing systems capable of reasoning, interacting, and identifying digital objects uniquely to collect data based on automated action requires the latest state-of-the-art technologies. This is only possible with the combination of different technologies capable of identifying objects and making them interact with each other. Figure 2.3 portrays some technologies widely used in IoT implementations.

With the increase in the evolution of IoT technologies, the IoT domain will evolve on a massive scale. From smart homes to smart healthcare, the networking paradigm will affect every bit of our daily lives (embedding intelligence in everything around us). The following technologies can help in large-scale IoT development;

2.3.1. Radiofrequency Identification (RFID)

The RFID is a crucial device uniquely designed for object identification. It is a low-cost small-sized tag that can easily be integrated into various objects. Depending on the situation, the microchip-like tag can be stuck to objects as passive or active transceivers. With the active tags constantly emitting signals, they are manufactured with built-in batteries to support longevity, whereas the less-expensive passive tags are only active when they are activated [23], [24]. The RFID system comprises readers/RFID-linked tags used in generating data (identification, topographic, e.t.c) about objects when an appropriate signal is captured. Using radio waves, the data signals emitted are transmitted to a reader and then analyzed by the processors subject to the application type [11]. RFID frequencies are divided into 4 frequency ranges:

1. Low Frequency
2. High Frequency
3. Ultra-High Frequency
4. Microwave Frequency

2.3.2. Wireless Sensor Networks (WSN)

The WSN is a two-way wireless sensory network built of multiple nodes scattered across a field of sensors connected to one or more sensors capable of capturing data of objects like speed, humidity, and temperature [25]. The data is further transmitted to other types of equipment for processing. Mostly, every transceiver sensor is built with an antenna, a micro-controller, and interface circuits [26]. Memory can also be added to store data.

2.3.3. Cloud Computing

The cloud is an intelligent computing technology capable of analyzing and storing huge data efficiently. It is currently considered the only technology to do that effectively. Multiple servers can be connected in a single cloud platform to communicate and share resources anytime, anywhere [27], [28]. The application of cloud computing into the IoT sphere can be seen in Figure 2.2, it also processes data/information at an increased capacity to deduce useful information [29]. However, the true potentials of this technology are yet to be realized. Cloud computing with a smart initiative leveraging tons of potential sensors, has the potential to support the extremely large-scale

development of IoT [30], so research will only begin when the IoT is fully dependent on cloud computing.

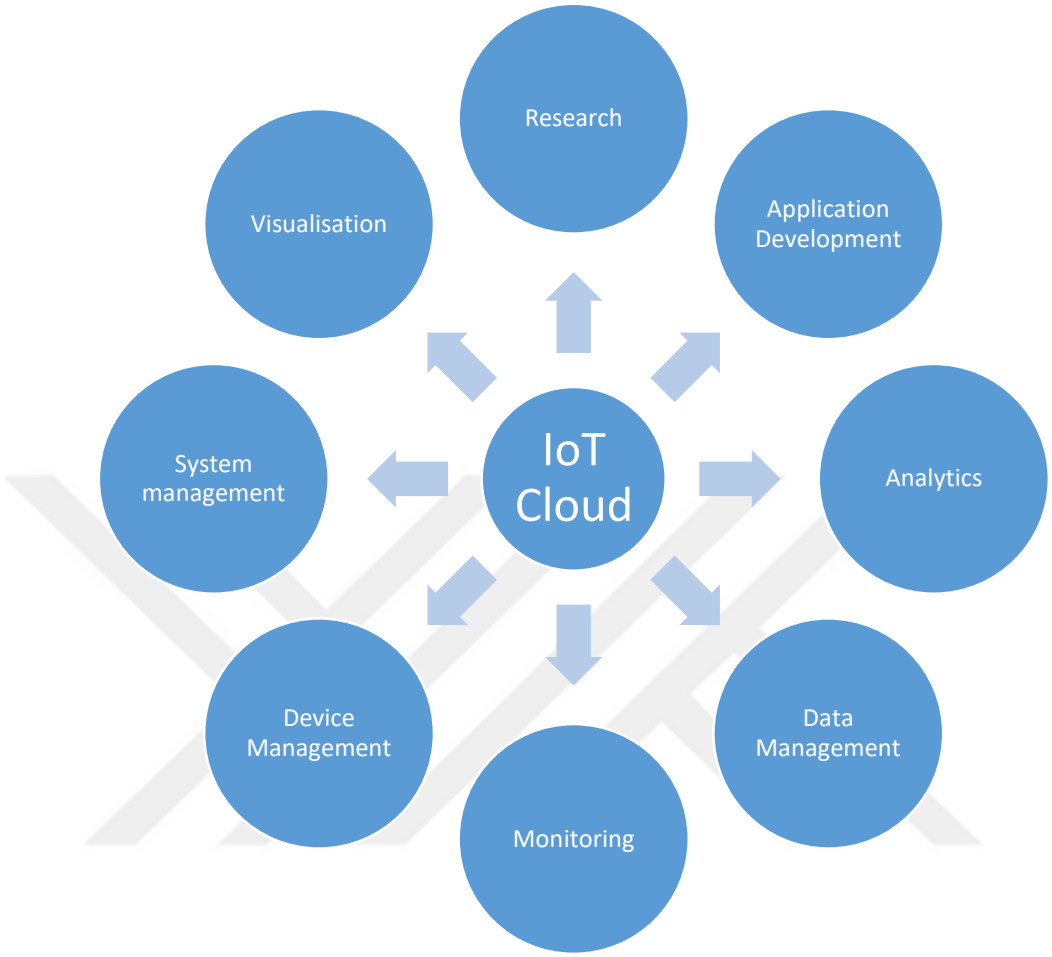

Figure 2.2. Application domains of IoT cloud platforms

2.3.4. Network technologies

These are technologies for establishing communication between different objects. The widespread of IoT devices brought about the need for fast and efficient network solutions capable of handling the huge number of potential devices. Even though technologies like Bluetooth and WiFi can be used for short-range networks, broadband networks like the 3G and 4G are usually used, but as you know, mobile traffic is becoming more and more unpredictable making it necessary to adopt super-fast technologies like the 5G.

2.3.5. Nanotechnology

Nanotechnology involves the development of tiny versions of objects. It can improve the IoT product life cycle by reducing power consumption. The nano-scaled devices can be used either as a sensor or as an active element in the same manner as conservative devices.

2.3.6. Optical Technologies

Technologies such as BiDi and Li-Fi from Cisco are developing swiftly presenting optical technology as a major breakthrough in IoT development. The Li-Fi is a landmark that uses VLC (Visible Light Communication) technology to provide outstanding connectivity spanning a wide range of frequencies for objects connected in the IoT concept. Likewise, BiDirectional (BiDi) technology allows a 40-gigabyte Ethernet channel to be used for large amounts of information from a variety of IoT devices.

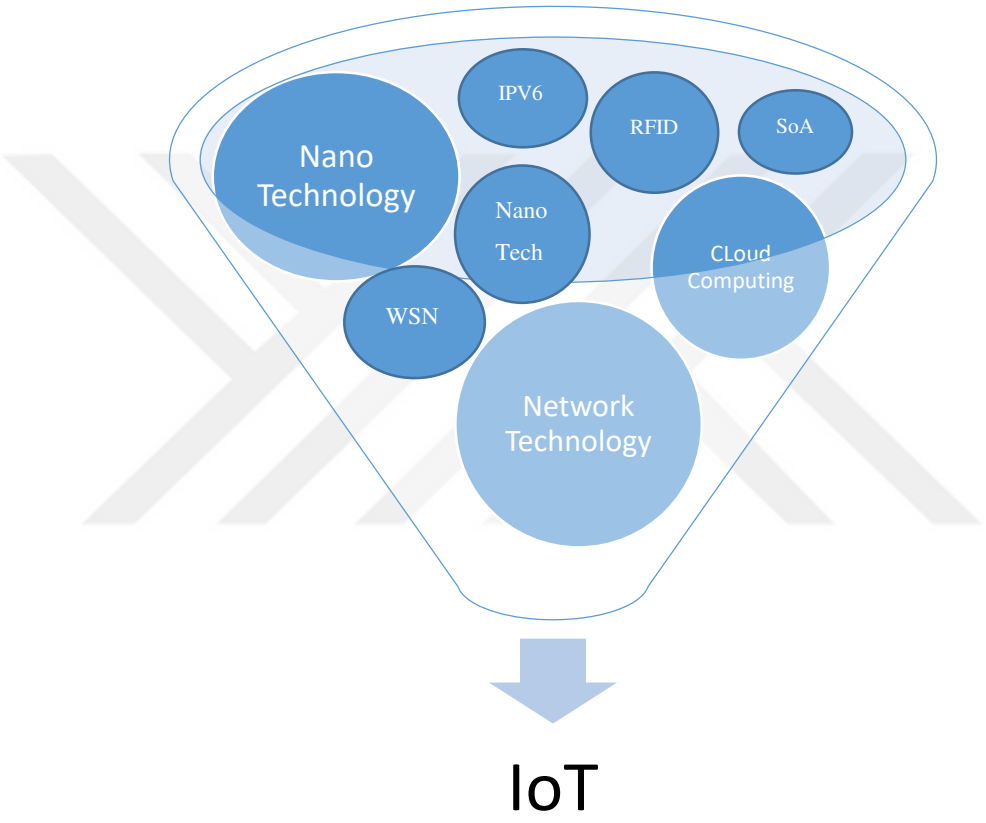

Figure 2.3. Technologies associated with the IoT

2.4. Application Areas of IoT

Most of the everyday applications we see are already in the "smart" category, though they might not be able to interrelate with each other. To make them interact, new technological breakthroughs need to be developed that can increase their potential for interaction and sharing of resources, as well as vital information. Such applications of IoT with a bit of some offline capabilities will undoubtedly improve the way we live. Here are some examples of possible future applications that can provide tremendous benefits.

2.4.1. Smart Road Network

The road network is an IoT application established to support the smart transport initiative. It comprises a system that leverages the potentials of IoT devices to collect traffic data for the sole purpose of improving the road situation. The system is more of a smart traffic monitoring system that identifies vehicles and varying objects/factors autonomously in an intelligent way that surpasses the capabilities of image recognition systems [31]. A smart traffic monitoring system will facilitate transportation by eliminating congestion. This system also has such features as theft detection, traffic incident reporting, less pollution [32]. Smart city roads will also offer detours due to bad weather or traffic jams as all processes will be enhanced to adapt to varying conditions, saving energy as well. Also, each consumer will have access to data on the possibilities of parking spaces.

2.4.2. Smart Environment

This is an application of IoT that aids in monitoring environmental situations. The acquired data within the smart environment can be used to predict natural disasters like fire-outbreak and earthquakes [33]. They will also allow better control over floods as well as environmental pollution.

2.4.3. Smart House

This initiative provides non-industrial home automation solutions that can remotely control home appliances based on the user's needs. Close monitoring of utility meters like electricity and water supplies will help in conserving resources as well as detecting unexpected overloads or outages [34]. The improved intrusion detection in such systems will prevent theft. The sensors used here can measure lighting, humidity, temperature, and other parameters that make up a smart house.

2.4.4. Smart Hospitals

These are hospitals equipped with smart IoT technology. In this aspect, reconfigurable devices equipped with RFID are issued to patients on arrival, to monitor the temperature, blood pressure, pulse, and other parameters [35],[36]. The monitoring is done even after discharging some patients to track progress and avoid critical emergencies like heart attack and speed up time for an ambulance to react and get to a patient [37]m. There are already ambulance drones on the market that can track a patient or scene thereby rendering first aid kits before the vehicular ambulance arrives.

2.4.5. Smart Farming

This application will be able to control soil nutrition, lighting, humidity and improve the landscaping of farms, automatically increasing the temperature for maximum results [38]. It will foster adequate irrigation and fertilization to maximize profit by conserving resources.

2.4.6. Smart Retail and Supply Chain Management

It has been explained in [39] that the IoT application coupled with RFID is already providing huge paybacks for merchants. The products are coupled with RFID to able to track inventory and detect theft. Moreover, the salesperson can compile tops of sales and charts for effective strategies.

2.4.7. Smart Defense

Graswald and Markus in [40] explained this IoT application as a technology that involves incorporating weapons and other security components with IoT technologies to aid tracking, surveillance, and real-time monitoring of operations to limit casualties and maximize success in missions.

2.5. International Standardization of the IoT

The most fundamental work in the field of IoT standardization is being done at NIST. In the summer of 2016, a sensational document was released [41], [42]. The document discussed the technology as IoT and NoT (Network of Things) which are sometimes interchanged, but they are slightly different. In [42], [43], many new concepts are introduced, the basis of which is the concept of a primitive. Primitives are building blocks that provide an opportunity to build larger blocks from smaller elements [42].

In [42] there is no definition of what is or is not a “thing”. Each primitive is considered to introduce behavior that represents that "thing", workflow, and data flow. "Things" can occur in physical spaces like people, computers, vehicles, and other smart devices or virtual spaces like virtual machines, software, virtual networks, etc.

The primitives are also capable of offering unified vocabularies that allow information to be assembled and exchanged between different networks. They offer clarity about the intricacies, including interoperability, layout, and continuously linking proactive processes. In [42] the prerequisites for expressing the behaviors of IoT were revealed without the need of worrying about the definition of the IoT. Insight on trust issues and systems security can be deduced from such an approach.

We believe that [42] is an important work in the standardization of the “IoT” Phenomenon that is currently adopted globally, the first three primitives sensor, aggregator, and communication channels are portrayed below;

2.5.1. Sensor

A sensor is an electronic device that measures the properties (physical) of an object such as presence, identity, contact, location, weight, sound, contact, location, identity, etc [44]. There are various kinds of sensors but they all utilize electronic, mechanical, chemical, optical, or other various effects at a controlled process interface or open environment.

2.5.2. Aggregator

This is software that transforms a group of raw data into intermediately aggregated data. It is implemented based on mathematical functions that help to manage big data [45]. It works with an abstract group (cluster) of sensors that can instantly appear or disappear. The aggregator also measures the degree to which data from specific sensors affects the aggregator's calculation, which is known as weight.

2.5.3. Communication channel

The communication channel is a medium through which data transmission is carried out from one point to the other across the overall IoT architecture [46]. The medium could be physical like a USB (Universal Serial Bus) or wireless like the router.

NIST is the world's largest research institution for standardization in all areas of the national economy. It is part of the US Department of Commerce (located in the suburbs of Washington). Only a separate item of the IoT required a huge amount of work. It even took a publication titled "Demystifying the IoT" [41] to cool the hot marketing heads. Here are just a few of the fundamental works of NIST:

[47]- January 2017, on risk management of engineering works;
[48]- July 2020, NIST Advanced Manufacturing Series 200-2;
[49] – May 2020, Defining Actionable Rules for Verifying IoT Security;
[50]- May 2020, IoT Device Cyber security Capability Core Baseline
[51]- November 2016, Cyber-physical Systems and the IoT

3. IOT SECURITY

For more than a century, mankind has been using automatic sensors and controls, calling them sensors (from the word feel and measure), and has been using them to connect computers. Furthermore, we are currently amidst a remarkable technological revolution that offers unprecedented opportunities but also poses the huge risks associated with these changes. It is now possible to use extremely inexpensive controllers and sensory devices with amazingly exponential capability, availability, and growing varieties [22]. Following their unconditional usefulness and economic feasibility, technologies based on IoT in green buildings, environmental monitoring, health and safety monitoring of facilities have already been deployed. We buy cars with already built-in sensors and controls or, if you prefer, with the IoT (although cars, for the most part, are not connected to the Internet).

Therefore, all objects that are part of the IoT are under threat: objects of the Department of Defense and critical infrastructures such as employees, equipment, and other properties may be used to initiate IoT security attacks. If not looked into appropriately, adversaries who are determined may end up undermining the security and processes that will bring down systems like power grids, medical access, communication e.t.c. One way or another, all these reputable departments are closely studying the applications of IoT in various systems, If no action is taken to get around the hike in IoT device base, the problem will grow exponentially further, and the growth of IoT could sweep these structures like a tidal wave. [52] provides guidance and policy recommendations for addressing vulnerabilities (and exploiting opportunities) “associated with the increasingly pervasive and semi-autonomous devices that support the Internet, which is the so-called IoT. Figure 3.1 further identifies some common IoT security issues as well as feasible solutions for each issue.

3.1. Information security in IoT

One of the challenges in the development of the concept of the IoT in many applications is the complex problem of ensuring information security in a wide range of protection against malicious threats. These challenges are particularly relevant as user demand for IoT is projected to rise. For the analysis of the problem posed above, one of the most common IoT architectures consisting of three layers was adopted. The layers include the perceptual layer, the application layer, and finally the network layer. For each of these layers, the main problems of information security are given. The IoT receives data from numerous devices, collects big data of various formats from multiple sources with heterogeneous characteristics. The result is DoS failures due to network congestion, software bugs due to the difficulty of real-time debugging with an external

load simulator. The main reasons for the complexity of providing information security at all the layers are noted below;

3.1.1. Information security problems at the levels of the IoT structure

It should be noted that in some papers more than three-tier IoT architecture is considered. In [53], a five-tier IoT architecture was adopted, including, for example, an intermediate layer (Middleware) amid the application and network layers. This level performs the function of processing information messages of interacting sensor sensors of the same type.

In this work, we will restrict ourselves to analyzing the problems of informational IoT at each of the three levels - the level of perception, network, and application levels.

3.1.2. Information Security Problems at the perceptual layer

The main security problem at the perceptual level is the security of physical devices as well as information collection security. Most of the perceptual nodes, which are characterized by deployment in an unattended environment with no standards, variety, simplicity, limited power supply, and weak ability to protect security. Therefore, providing unified security is almost not possible making it vulnerable to attacker threats.

Security issues at this level include physical hijacking of sensor nodes, hijacking of a gateway node, power depletion, leakage of sensor information, threats to data integrity, congestion threats, DoS (denial of service) attacks, and routing threats by installing illegitimate sensors into the network, and node copying threats.

3.1.3. Information Security Problems at the Network Layer

These are existing threats within the communication networks, like data interception, unauthorized access, integrity, viruses, exploits, rootkits, network worms, e.t.c. [54]. There are more complex security challenges in the IoT than previously encountered, this is due to two reasons; the heterogeneous nature of its structure and a large number of objects [55]. The IoT receives information of various formats from a vast amount of devices, thereby creating more complex problems at the network layer.

Software vulnerabilities receive too much attention, leading to a breach of information security after implementation. Software vulnerabilities can be caused by inevitable mistakes made by developers of complex multilayered software, program kernel errors, unprotected code implementation, poor handling of exceptions, and poor indexing or pinning of database queries

[56]. Others also include web vulnerabilities, insufficient software performance or scalability, errors in distributed application operation, as well as virtual platforms and clouds [57]. It should be noted the complexity of software in the IoT, caused by a wide variety of used hardware platforms and operating systems.

Another reason for a software vulnerability can be backdoors - these are pieces of code introduced for subsequent usage by developers to remotely control a computer. Backdoors are also installed on equipment by manufacturers for management or testing purposes. However, this "back door" can be detected and used by an intruder [58].

3.1.4. Information Security Problems at the Application layer

The integration of computer and communication technology brought about the widespread of IoT and various areas of the industrial sector. Moreover, the violation of information security of traditional communication networks (as a result of threats of replay, eavesdropping, information distortion, information disclosure, etc.) also contributed a lot [59]. There are also security issues when using cloud services, privacy protection, information processing, etc.

3.2. IoT Security and Privacy Objectives

IoT makes it possible to find anyone, making our lives easier; however, without due confidence in the security and privacy of user data, this system will not be accepted by many. Therefore, it is a must for IoT to have secured infrastructures for better widespread implementations. IoT must have strong protective infrastructures. Nonetheless, the problems and tasks of security of IoT-based networks have been discussed in detail in [55]. Here are just some specific potential problems related to IoT security and the use of technologies described above.

3.2.1. Unauthorized RFID Access

When an attacker acquires access to an RFID tag containing identification details, confidential user information will be exposed [24], so it must be addressed first. The tag can not only be read by an intruder's reader but even modified or damaged. There are several real threats to RFID, which include RFID viruses, Mobile Phone Attacks, and SpeedPass Hack.

3.2.2. A security breach of sensor assemblies.

A WSN is a two-way sensor that can transmit/capture data. Its ability to capture can be utilized to initiate attacks such as plugging the channel, tampering, Sybil's attack (a network attack in which one of the nodes can have multiple identifiers, thereby disrupting the system), interference, padding, clogging, e.t.c.

3.2.3. Cloud Computing Abuse.

As cloud computing offers the sharing of resources on a large network of servers, sharing resources can introduce many security threats such as data loss, theft of accounts, insider attacks, phishing, botnet attack, e.t.c.

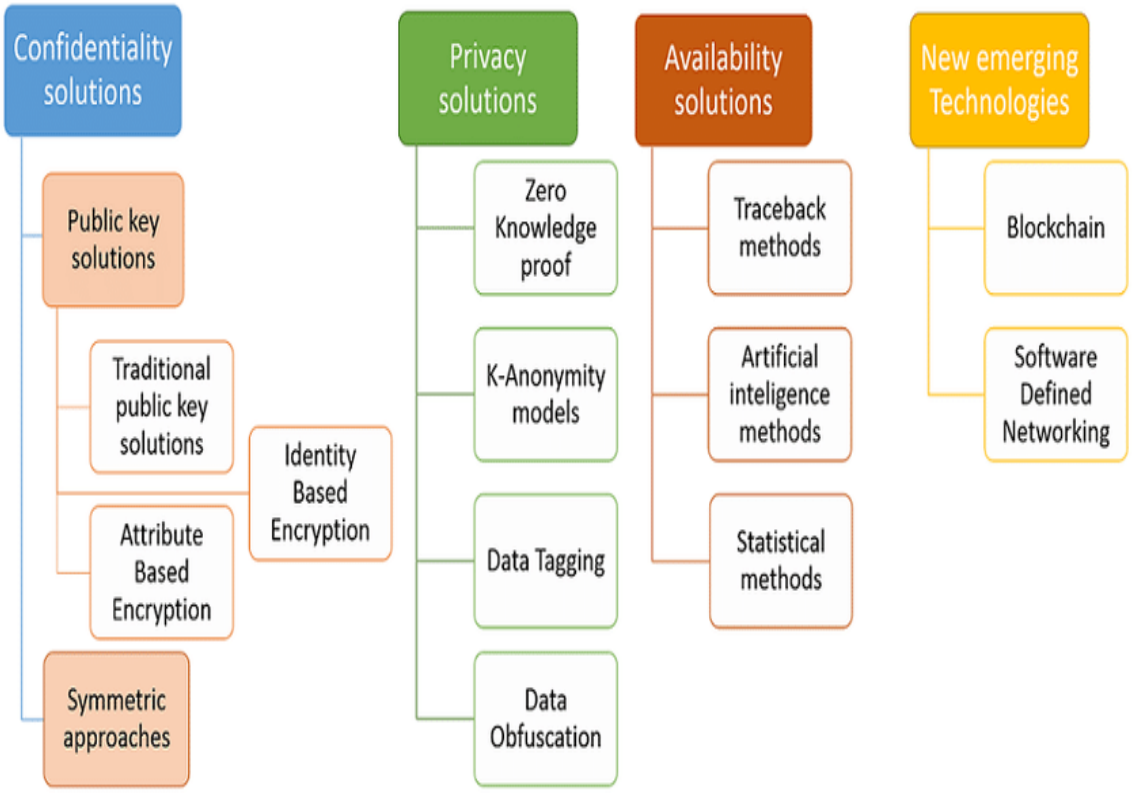

Figure 3.1. Ideal IoT Security Issues and Proposed Solutions

3.3. Security Monitoring in Iot-Networks

Among a large number of smart home devices (including security systems), there are numerous severe security problems. Statistics show that 65% of IoT devices have easily implemented vulnerabilities [60]. To prevent the implementation of these vulnerabilities, it is recommended to use the concept of IoT network monitoring.

Network monitoring begins with collecting data from the corresponding network, which will subsequently allow you to control remote devices and equipment [61]. All of these smart connected devices come together and work together to automate and control other equipment from the data center.

To be truly effective, an IoT monitoring system must have certain functionality. These functions should include the ability to accumulate and process significant data and information

about the network. This functionality includes instant notification and alarm filtering [58]. Instant alerting means getting timely notifications of important changes in status or conditions. They must be informative enough for a correct and quick reaction to threaten a security breach.

There are two main ways of collecting data: push-based or polling [62]. The push-based method may be more convenient for IoT monitoring systems, but there are some trade-offs to consider. These trade-offs are usually associated with the existing communication protocol. It is important to ensure that the protocols supported by the devices on your network provide an efficient way to collect data. But it is also important to use open protocols to ensure interoperability between many devices. An up-to-date protocol will provide efficient data collection and full network visibility.

IoT security monitoring is different from general IT security monitoring. In fact, the security approach for IoT devices is radically different from traditional devices because it relies on a variety of software, hardware, other components, and communication protocols [63]. Intrusion detection systems (IDS) are one of the most popular solutions to support security monitoring.

The paper [63] proposes the concept of a device, which is a small module that can be embedded both in an extensive IoT network and in a network with limited resources, thereby becoming a monitoring system and allowing to organize an uninterrupted check of the IoT for an attempted attack on a given network. This module consists of four elements responsible for capturing traffic inside the IoT network, filtering it, and issuing a verdict on an attempt to implement one or another attack on the devices of this network. Its main components include the following elements:

- Data storage device - permanently stores attack signature templates used in data analysis.
- Data analysis device – used to identify anomalous and suspicious in a network.
- Data pre-processing and filtering device - receives data from various monitoring agents available to provide an initial pre-processing as well as data aggregation.
- Incident detection tool – it collects and analyses processed data to trigger alerts whenever an incident is suspected.

The schematic structure of this module is shown in Figure 3.2.

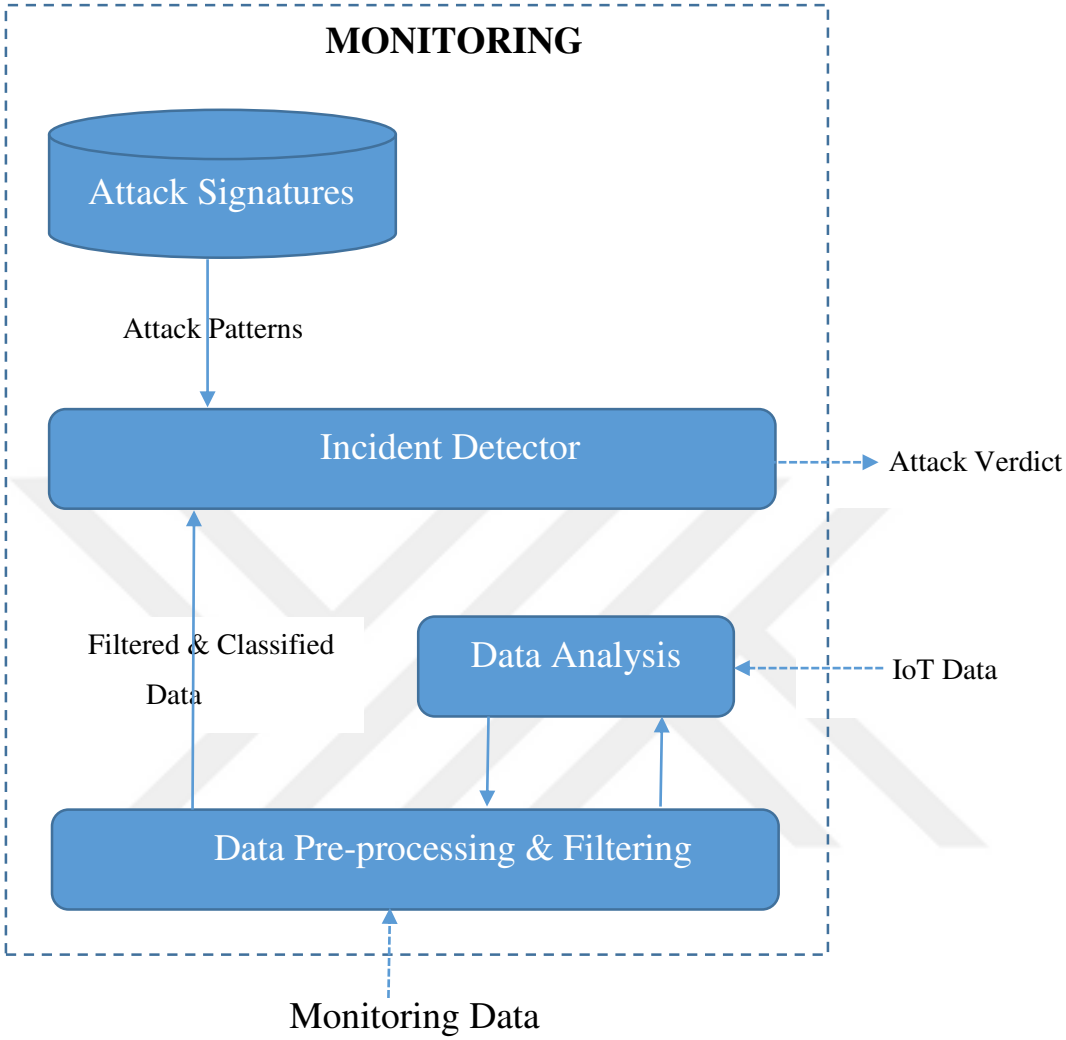

Figure 3.2. IoT Network Monitoring Module

In this case, the data analysis component can be represented as an intrusion detection system based on anomalies. This tool creates a profile or model that is considered to be a model of "normal" behavior using supervised or unsupervised machine learning algorithms [63]. In this case, any deviation from the prescribed model may be identified as potential threats and tagged accordingly.

According to [18], ensuring IoT security is very difficult as many of the constraints on computing resources and power prevent proper countermeasures from being taken. These limitations make traditional IDS unsuitable for full monitoring of the IoT infrastructure.

3.4. Intrusion Detection Systems in IoT Networks

IoT Intrusion Detection System (IDS) looks at all IoT network traffic (or traffic of a specific section of the network) and, if any deviations in it are detected, they signal it. Formal SSOBs use signature rules - packets arriving at the sensors are compared with the signature database and, if a match is found, an alarm is announced. Unfortunately, even formal SSBNs are becoming insufficient for reliable network protection. According to CERT, the number of known new intrusion methods in 2010 alone exceeded 25,000. This means that on average, about 70 new attacks appear every day. It is physically impossible to update the signature database of formal SSOBs for such time intervals. In addition, increasing the volume of signatures negatively affects the performance of systems.

Therefore, the urgent task is to develop systems that can classify IoT network traffic quickly for attack detection. The solution can either be formulated as a binary classification problem (normal or abnormal traffic) or as a more complex multiclass classification problem, when abnormal traffic, in turn, is classified according to pre-identified attack types.

According to [64], [65] an IDS in IoT is a tool that monitors IoT networks by observing data (packets) in the IoT's network traffic for identification and protection against intrusions that could jeopardize the security of information systems within the network architecture. Such systems are used by practitioners for the discovery of security vulnerabilities as well as anomalous/malicious activities. Most of these IDS systems accomplish their task by using passive traffic collection and analysis. Since the presence of IDSs does not invalidate the use of primary security components that serve as the first line of network defense, IDSs are tagged by security practitioners as a second line of network defense [66].

3.4.1. Taxonomy of Intrusion Detection Systems in IoT Networks

IoT IDSs are significant countermeasures against numerous kinds of cyber-attacks. However, most IDSs existing are resource intensive and difficult to use with small IoT devices. Hence, there is a need for a special IDS solution that is lightweight and has a high protection class. These lightweight IDSs capitalize on a general algorithm called “principal component analysis (PCA)”. The PCA is a lightweight algorithm that uses several identification techniques for intrusion detection. In the paper [67], they focused their research on the application of PCA on anomaly-based IoT IDSs.

The researchers in [68] recommended PCA for creating anomaly-based IDSs for statistics and data mining. Their proposal depended on dividing the PC into two (most important and least important PC). Payload-based PCA intrusion detection systems were discussed in [69], the authors

of [70] worked on models based on statistics, a machine-learning-based model was explored in [71], and [72] explored data mining approaches in detail.

IDS security modules are typically provided in two formats: host-based IDS (HIDS) and network-based IDS (NIDS). The Hybrid IDSs monitors server activities, while NIDS monitor communications and activities in a network. Since normal and malicious behaviors are expected to be different, IDSs can monitor host and network activity behavior for signs of attack [64].

With regards to the architectures of IDSs, they were categorized by [73] into hierarchical, centralized, and distributed IDSs. The centralized IDS monitors data that is mostly from a remote or host-based location, while the distributed IDS is mostly shared between multiple nodes, with an equal allocation of responsibilities. Considering the implementation strategies of IoT IDSs, they were further classified by [74] into 9 classes as shown in Table 3.1. With regards to this, several implementations were compared from several publications as seen in Table 3.2.

Table 3.1. Implementation schemes for IDSs in IoT networks

Implementation Strategy	Energy Consumption	Processor Requirements in a Net With Powerful Nodes	Processor Requirements in a Net Without Powerful Nodes	Detection Accuracy	Implementation on Resource-constrained Nodes
Hierarchical	B	A	C	C	A
Distributed and Collaborative	C	B	B	C	C
Statistical Detection	D	B	E	B	E
Voting	A	A	A	D	A
Cross Layer	E	E	E	A	D
Game Theory	D	D	D	B	E
Reputation	D	A	A	C	A
Mobile Agent	D	D	D	C	E
Machine Learning	D	B	E	A	E

Methodology-wise, IDSs are categorized as misuse-based (signature), anomaly-based, and Hybrid IDSs. While the misuse-based approach uses a database of known patterns and signatures to detect suspicious attacks, anomalous-based systems use normal traffic data patterns based on data from the user's established normal behavior. A profile is created and then the operation of the current traffic data pattern is analyzed to detect anomalies in real-time [66]. For the hybrid IDS, it utilizes both the signature-based approaches as well as the anomalous approaches. It functions by utilizing two modules, one is responsible for detecting attacks by comparing with signatures, and the other works by discovering deviations from normal traffic files [75]. In this research, we will be focusing on anomaly-based IDSs.

3.4.2. Anomaly-based IDSs

The term “Anomaly detection” refers to the process of detecting deviations from the normal profile of a system. Binary classification is mostly used to detect anomalies in IoT network traffic, and link anomaly criteria are used as a class specification, where 0 corresponds to the normal profile and 1 corresponds to the anomalous. Generally, such IDS uses heuristic techniques for the detection of anomalies in the behavior of a network with the aid of a certain set of rules. To identify whether the activity is an intrusion or not, certain behavioral pattern thresholds are used.

Anomaly detection methods are more promising because they can detect previously unknown attacks without the need to first generate intrusion signatures for each new attack. One of the most relevant research areas in the anomaly detection sphere is the detection of anomalies in IoT network traffic.

3.4.3. Methods for detecting network anomalies.

Anomalies are objects or incidents that differ by more than the maximum permissible deviation (error) from the accepted value. Thus, the detection of network anomalies is reduced to their identification. It is proposed to detect anomalies using constructed triggers based on links to rare events when monitoring traffic. Such connections are suspicious of analysts since they differ significantly from most of the identical states of legitimate network traffic.

In machine learning, anomaly detection is applied in a variety of areas, including intrusion detection, fraud detection, and ecosystem disruption detection.

There are three broad categories of anomaly detection: controlled, uncontrolled, and semi-controlled [15]. In terms of the machine learning approach, some of the popular detection methods include, among others, density-based k-nearest neighbor (KNN), one-class Support Vector Machine (SVM), Bayesian networks, outlier detection based on cluster analysis [76]. Several analysis systems use the above or similar machine learning detection methods.

Table 3.2. Comparative analysis of IDSs implemented for IoT networks

IDSs	Implementation	Methods	Attacks
Azmoodeh et al. [77]	Machine Learning	Anomaly	Junk code insertion attacks
Arrington et al. [78]	Statistical	Anomaly	Not Assigned
Anthi et al. [79]	Machine Learning	Anomaly	Sinkhole, DoS, Sybil, Hello Flood
Arshad et al. [80]	Distributed and Collaborative	Anomaly	Routing and application-specific
Bao et al. [81]	Reputational	Signature	Sinkhole, DoS, Sybil, Jamming
Cervantes et al. [82]	Reputational	Anomaly	Sinkhole
Fu et al. [83]	Statistical	Anomaly	DoS, Junk Data Injection
Kasinathan et al. [84]	Distributed and Collaborative	Rule	DoS
Khan and Herrmann [85]	Reputational	Rule	Sinkhole, Selective Forwarding
Khan et al. [86]	Reputational	Rule	Ballot Stuffing, Self-Promoting, Bad Mouthing
La et al. [87]	Game Theory	Rule	Not Assigned
Li et al. [88]	Machine Learning	Anomaly	DoS, Probing
Liu et al. [89]	Distributed and Collaborative	Rule	Not Assigned
Liu and Wu [90]	Statistical Detection	Anomaly	Not Assigned
Liu et al. [91]	Machine Learning	Anomaly	Not Assigned
Raza et al. [92]	Distributed and Collaborative	Hybrid	Sinkhole, Spoofing, Selective Forwarding
Sedjelmaci and Feham [93]	Machine Learning	Hybrid	Routing Disruption
Sedjelmaci et al. [94]	Game Theory	Anomaly	DoS
Strikos [95]	Hierarchical	Signature	DoS, Routing Disruption
Summerville et al. [69]	Statistical Detection	Anomaly	Wormhole, Junk Data Injection
Shin et al. [96]	Hierarchical	Signature	Selective Forwarding
Wang et al. [97]	Reputational	Signature	Selective Forwarding
Xiao et al. [98]	Machine Learning	Anomaly	Malware, Offloading, Identity attacks
Yang et al. [99]	Machine Learning	Anomaly	Eavesdropping, Packet dropping, hole attacks

3.5. IoT Attack Types

Since IoT devices became an integral part of our lives, the frequent use of these devices has made it a general knowledge to almost everyone that IoT devices are mostly imperfect and vulnerable, leaving their users with a false sense of security. Furthermore, certain IoT attacks like Brute Force and DoS will most likely continue to be huge problems for this technology even in the near future, this is because such attacks have seen tremendous usage and upgrade by several IoT network adversaries. There are several types of IoT attacks used by adversaries to compromise or inflict damage to the network, data, or its devices. For this research, some common and basic IoT attacks were used to generate the dataset as described in Table 5.5. These attacks are further explained below.

3.5.1. Denial of Service DoS Attacks

A denial of service (DoS) attack deliberately attempts to overload capacity on the target system by sending multiple requests. Unlike phishing and brute force attacks, denial of service attackers do not seek to steal critical data. However, DoS can be used to slow down or shut down a service to damage the reputation of a business. DoS attacks are also easy to carry out, even by a novice hacker.

Three types of DoS attacks were used for this research, the first one is the “DoS ICMP flood” (Internet Control Message Protocol) which was launched to overwhelm the targets with ICMP ping requests. “DoS SYN flood” was the second attack type used to rapidly send a series of synchronized messages to the targets. Finally, “DoS Http get Flood” was also used to exploit seemingly-legitimate HTTP GET or POST.

3.5.2. Brute Force Attacks

Brute force attacks are cryptanalytic attacks that consist of sending multiple passwords or passphrases in the hope that the attacker will guess (a possible combination) the password correctly. All possible passwords and passphrases are steadily checked in systematic order until a suitably matching password is found. IoT networks and devices are generally vulnerable to Brute Force attacks since most IoT devices use default authentication passwords while a majority use none at all.

Review and analysis of IoT brute force attacks reveal that attackers are constantly improving existing malware by extending the standard password database (wordlist) and attempting to attack additional HTTP, SSH, and Telnet ports. In this research, “Brute Force SSH” was performed to guess the target password via the SSH ports, “Brute Force RDP” was also performed via the RDP

ports to acquire access to the target, and “Brute Force Telnet” was finally performed to compromise some targets via the Telnet protocols.

3.5.3. Fuzzing Attacks

Fuzzing attacks are cyber-attacks performed to stress a network, system, or devices by randomly feeding them with crafted invalid or unexpected data inputs. A successfully executed fuzzing attack on an IoT network, devices, or its firmware will most likely cause it to trigger an unexpected behavior thereby exposing vulnerabilities that could have easily been unnoticed. The overall process works in a few basic steps that include execution, monitoring, analysis, and bug reporting.

For the research at hand, carefully crafted invalid data were randomly fed to the overall network thereby stressing all network components and vulnerabilities were identified.

3.5.4. SYN Scan Attacks

An SYN Scan attack is a half-open scanning performed without the establishment of a full connection to a target network. It is ideally a faster means of determining the status of communication ports (TCP). The general aim is to locate open ports that can be easily used to further cause mayhem on the targeted IoT network and its devices.

The SYN Scan was also performed at the beginning of the attack simulations to probe for open ports by determining the status of the communication ports in a stealth state without establishing a full connection and acquiring access to the overall network devices.

4. MACHINE LEARNING ALGORITHMS FOR IOT IDSS

Real-time forensic response to security threats requires analysis tools that are capable of handling numerous events within the IoT systems. These events are contained in logs, traffic, and other data containers with huge data volumes. Furthermore, source and information database heterogeneity also leads to a high degree of heterogeneity in the analyzed data. Machine learning is now a powerful technology for attack and anomalous behavior as well as the analysis of security events on smart devices [100]. Nonetheless, comparing different machine learning algorithms in the scientific literature with each other is quite challenging - in assessing the effectiveness of using the models, various kinds of datasets or subsets of a particular varying set are used by researchers.

As the statistics published in the annual report of the "Microsoft Digital Defense" show, the number of known attacks is growing rapidly every year [101]. To counter this threat, it is necessary to use effective defenses, such as intrusion detection systems (IDS). These protections typically use signature analysis for intrusion detection and require regular updates to the intrusion signature databases. They are not capable of detecting attacks whose signatures are not present in the signature databases, and this is where the machine learning approach becomes handy.

Machine learning is a data analysis tool that is used to efficiently perform specific tasks without using explicit instructions but relies on patterns and inferences. It is a data analysis technique that allows computers to learn on their own by solving an array of similar problems. Machine learning capabilities are employed in solving various problems, especially in issues relating to cyber security and forensics. Continuous analysis and training will allow generating various forecasts of the development of cyber-attacks, simulating various situations, and detecting abnormal activities in the network. These overall processes are achieved by following a set of procedural stages referred to as the “Machine Learning Life Cycle” which can be seen in Figure 4.1.

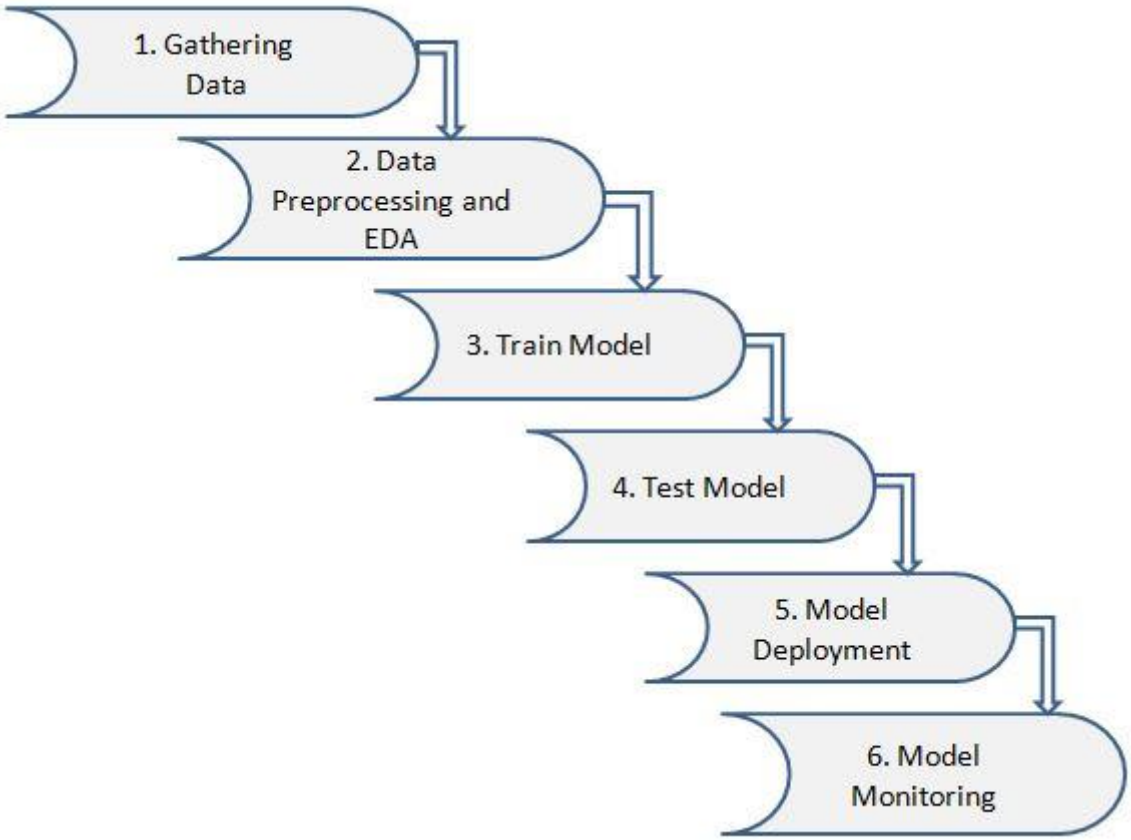

Figure 4.1. Machine Learning Life-cycle

Machine learning has deservedly attracted the interest of cyber security professionals. Since hardware and computing power is becoming more available, machine learning methods can be used to analyze and classify the nature of the occurrence of anomalies, malicious activities from aggregated metadata.

Large domestic providers use machine learning integrated into cloud-based intelligent systems to identify malicious and illegitimate content, isolate infected hosts, and display informative graphs and dashboards about the general state of a particular segment [102]. A major challenge of developing machine learning algorithms is building intelligent systems capable of learning sequential tasks and then transferring knowledge from the previously learned "foundation" to solve new problems. This capability is called continuous machine learning or intelligent lifelong learning systems [103].

There are hundreds of algorithms and approaches to machine learning, which are majorly grouped into supervised, semi-supervised, and unsupervised learning. The supervised learning approaches are performed in the context of classification, (where input matches output) or regression (where input maps to continuous output). Self-directed learning is mainly achieved through clustering and is used for exploratory analysis, reducing the number of measurements of objects. Both of these approaches can be applied in the field of cyber security to analyze malicious

activities in real-time, which eliminates the disadvantages of traditional methods of detecting such activities. The classification and scope of machine learning systems is further shown in Figure 4.2 below;

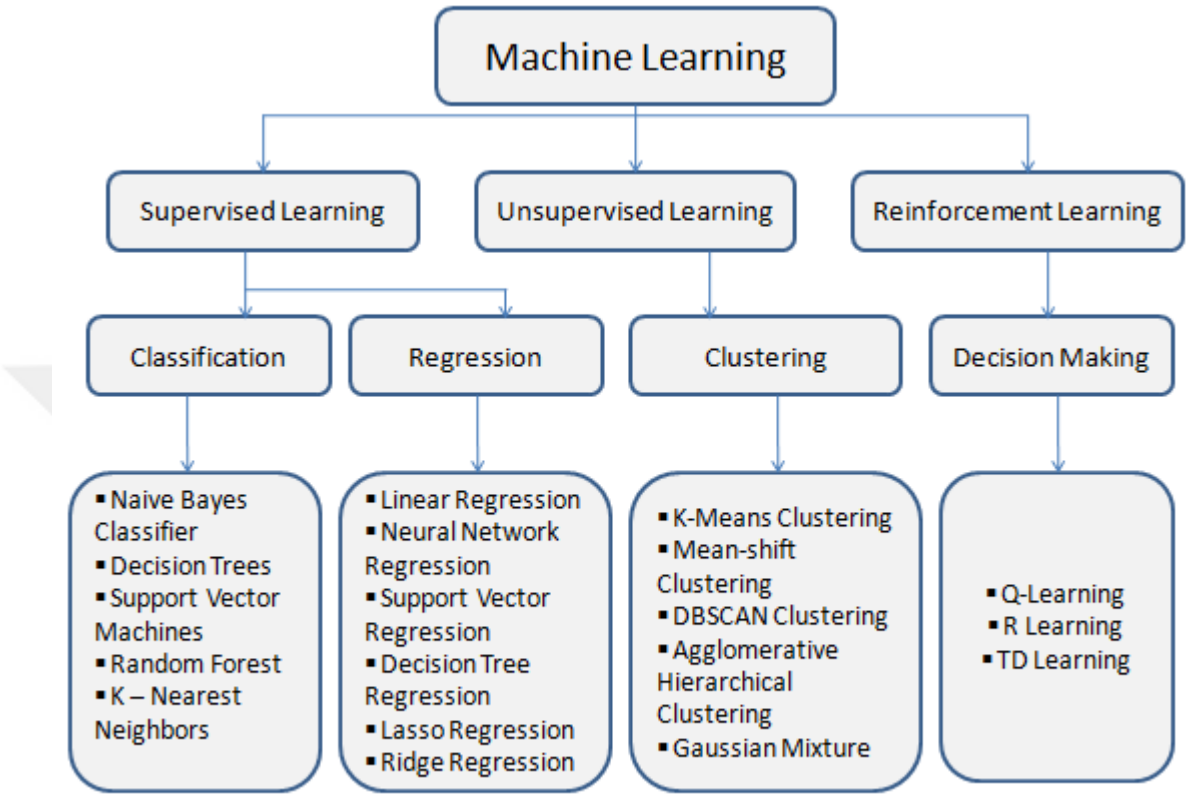

Figure 4.2 Machine Learning Algorithms

4.1. Supervised Learning

Supervised learning is an algorithm mostly used for solving regression problems to predict the target numeric value of a variable given a set of characteristics or attributes. It is particularly good in solving classification problems based on time series. Splitting the dataset into training and test sets is essentially necessary for all supervised learning algorithms.

For supervised learning, it is necessary to split the dataset into training and test sets. The machine learning algorithm takes the training set as input to generate the model as output, and then takes in the test set as input for the model to provide predictions as the output. Every machine learning model works on three basic components which are:

1. A family of models, which describes a variety of models to choose from
2. A cost function that compares various models quantitatively
3. An optimization technique that chooses the best model from the set.

The training data includes desirable solutions called labels. The most important algorithms for supervised learning can be as identified:

- Decision Trees
- Random Forests
- Linear Regression
- Logistic Regression
- Support Vector Machine
- K-Nearest Neighbors Method
- Neural Network

4.1.1. Linear regression

This is a supervised learning model that works based on the principle of determining the "line of best fit" between points of datasets. It also serves as a simple preliminary support for non-linear methods used in training neural networks. Wholly, the process of linear regression depends on the model of the variable x via one or more other variables like factors, regressions, and other independent variables that have linear functional dependencies.

4.1.2. Logistic regression

This is a supervised learning algorithm that is used as a technique for binary classification. The term itself is borrowed from statistics. The method is based on logistic functions - the sigmoid curve, which is useful for several areas, including neural networks. Logistic regression models the likelihood of classification problems with two possible outcomes and can be used to identify network traffic as malicious (true) or not (false).

4.1.3. Support Vector Machine (SVM)

An SVM is also a linear classifier similar to logistic regression that aims to separate two hyperplane dataset classes in a vector space. It is very different from logistic regression as it does not deal with a cost function. The SVM classifier tries to find the hyperplane with the maximum margin to separate the two classes. The "Margin" indicates the distance from the split level to the next training pattern on both sides.

4.1.4. K-Nearest Neighbor (KNN)

The KNN algorithm is among the frequently used nonparametric classification methods. It consists of mainly three steps which are; selecting the K number/distance metric, determining the

KNN of the sample, and allocation of the class label through popular vote. Its major disadvantage is the creation of large models that must store all training data containing feature vectors and labels.

4.1.5. Decision tree (DT)

Decision trees are universal supervised learning algorithms widely used because they are generally easy to interpret. This algorithm can be used for both classification and forecasting. It is termed “Tree” because its nodes are structured in a tree-like form. In many problems, a binary tree is used, each node of such a tree splits the data into two sets using the threshold value of one of the informative features. Based on certain conditions, the child subset is then recursively divided into smaller subsets by automatic selection of certain conditions.

4.1.6. Random Forest (RF)

A random forest is a model that consists of many decision trees. Rather than simply averaging the predictions of different trees (this concept is simply called “forest”), this model trains a group of classifiers based on decision trees, using a different random subset of the training set for each. Individual decision trees tend to adapt to the training set, and Random Forest mitigates this effect by averaging multiple decision trees. This usually improves the performance of the model. Additionally, each tree in a random forest can be trained independently of all other trees, making training algorithms parallelization very easy. This phenomenon generally makes random forests extremely more effective for training.

4.1.7. Neural networks

This is another algorithm for predicting time series data using artificial neural networks, particularly Recurrent Neural Networks (RNNs) uniquely designed for studying styles and patterns relating to classification or forecasting. Neural networks are used to build very complex models, especially in the case of large datasets. However, firstly, they are sensitive to data scaling, and secondly, it takes a lot of time to train such models

4.2. Unsupervised Learning

Among machine learning algorithms, a special place is occupied by machine learning algorithms, where the algorithm learns to solve a task on its own without human intervention. This is possible by interacting directly with the environment in which it is trained. Such algorithms are collectively called unsupervised learning algorithms. For such algorithms, you do not need to collect databases, classify, or mark them up. However, an algorithm for a student without a teacher is only enough to give feedback to its actions or decisions - whether they were good or not.

Unsupervised Learning can therefore be regarded as Machine Learning (ML) technique in which the Model is trained on Unlabeled Data.

The task of Unsupervised Learning can be to find groups of similar patterns in the data, and this is called Clustering, or determining how the data is distributed in space, and this is known as Density Estimation. These algorithms allow for more complex processing tasks than supervised learning, although they are less predictable. Some of its properties include finding unknown patterns in data and finding features useful for categorization, it costs less (i.e. preparing untagged data), and does not oblige to determine the number of classes.

4.2.1. Principal Component Analysis (PCA)

The PCA algorithm is designed to search and identify parameters that have a linear relationship with each other, as well as to isolate differing components for reducing the dimension of the data. The PCA can further be viewed as a sequence of optimization problems; it transforms the dataset into artificial parameters that cover the maximum variance of the original ones. The point is to explain the correlation between continuous explanatory variables using linear combinations of new ones called principal components.

4.2.2. K-mean clustering

Clustering refers to dividing a dataset into multiple classes so that the data within the classes is as similar as possible, and the difference in data similarity between different types is as large as possible. Cluster analysis is based on the clustering similarity of the dataset, which makes it unsupervised learning. As a simple iterative clustering algorithm, the k-mean algorithm uses distance as its similarity indicator for finding the K-classes in a given dataset. The class center of each class is the average value of the overall class.

4.3. Reinforcement Learning

Reinforcement Learning (RL) is an approach that sits between Supervised Learning and Unsupervised Learning. It is not tightly controlled because it does not rely only on a set of labeled training data, but it is also not unsupervised learning because there is a maximized reward. The Model has to find the "right" actions in various situations to succeed.

Reinforcement learning can also be considered as a science focused on decision making, there is no supervisor here - a living person who once marked the Dataset - and the model only uses the reward signal to determine if it is doing well or not. Time is a key component: the process is sequential with delayed feedback. And each action of the model affects the next data it receives.

So far, we have said that the agent must find the "right" action, which depends on the reward. The reward R_t is a scalar feedback signal that indicates how well the Algorithm is performing at a time step t. A good application of reinforcement learning is in gaming technology.

Having seen the three major categories of machine learning, Figure 4.3 depicts their major differences and mode of operation in comparison to each other.

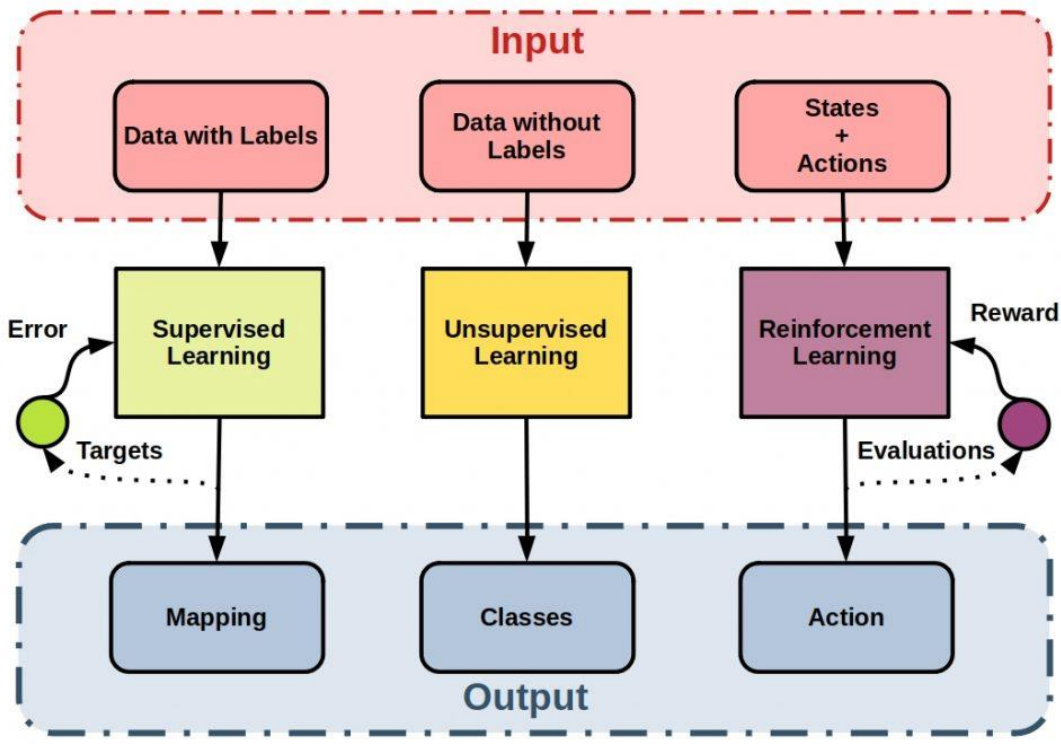

Figure 4.3. Supervised, Unsupervised vs Reinforcement Learning

5. MATERIALS AND METHOD

5.1. Materials Used:

Used hardware: Because a network has to be solely designed for the successful implementation of attacks and generation of the dataset that will serve as input for the machine learning algorithms of the intrusion detection system, various hardware and software were used. Both hardware and software used are elucidated below;

5.1.1. PC Features

Three PCs were used, one served as the attacking system while the second and third served as the victim and intercepting PCs respectively. The attacking system is a Fujitsu desktop PC with Kali Linux OS, the victim PC is also a Fujitsu desktop with windows 10 pro OS and five virtual machines as shown in the tables below. The intercepting PC is a workstation (HP) with windows 10 pro OS and Wireshark installed to serve as the network sniffer for generating the pcap files. The PCs can be clearly seen in Table 5.1 below;

Table 5.1. Personal Computers Used

PC Name	Processor	RAM	Graphics	Storage	OS
Fujitsu ESPRIMO_P400 E85+	Intel Core i5-3470 @ 3.20 GHz x 2	8.00 GB (7.89 GB usable)	Intel HD	500 GB	Windows 10 Pro
Fujistu ESPRIMO_PH556	Intel Core i5-6400 @ 2.70 GHz x 4	7.7 GB	Mesa Intel HD 530 (SKL GT2)	620.1GB	Kali GNU/Linux Rolling
HP Z840 Workstation	Intel Xeon i5-6400 CPU E5=2620 v4 @ 2.10 GHz x2	32 GB	Intel HD Graphics	1.2 TB	Windows 10 Pro

5.1.2. Single Board Computers (SBCs) Features

Single Board Computers were also used to connect the IoT sensors and devices to the network. As shown in Table 5.2, the SBC's used included Arduino, RaspberryPi, and Odroid. Other attachment boards like the WIZnet Ethernet and wemos Wifi/Bluetooth were also used.

Table 5.2. Used SBCs

SBC	Processor	OS
Arduino MEGA 2560	Atmega 2560 micro controller	Needs coding via Arduino IDE
RaspberryPi 3 B v1.2	Quad Core 1.2 GHz Broadcom BCM2837 64bit	Raspbian
Odroid -XU4	Samsung Exynos5422 ARM Cortex -A15 Quad 2.0 GHz + Cortex™-A7 Quad 1.4GHz	Android

5.1.3. Virtual Machine Features

Some virtual machines were also deployed to provide more number of devices (nodes) within the network as well as more vulnerable devices for easy execution of attacks. The virtual machines used are depicted in Table 5.3 below;

Table 5.3. Virtual Machines Used

Virtual Machine	CPU Core	RAM	Storage
Kali Linux/Rolling 2020	4	2 GB	80 GB
Metasploitable2 (Linux)	1	512 MB	8 GB
Ubuntu	2	2 GB	30 GB
Vulnerable XP (SP1)	1	1.4 GB	12 GB
Bee-box (bWAPP)	1	1 GB	20 GB

5.1.4. Software Features

It is also noteworthy to identify the key software used during this experimentation. We used Wireshark to intercept and sniff the network for several attacks as well as normal traffic for the successful capture and generation of the pcap files. The Wireshark software originally “Ethereal” is an open-source free-to-use packet sniffer and analyzer used in software and communication protocol developments as well as for educational purposes.

The pcap files were later preprocessed and parsed with the help of the CICFlowmeter software to generate ".csv" files as the dataset. The CICFlowmeter formerly known as ISCXFlowMeter is a network traffic flow generator and analyzer developed by the Canadian Institute for Cyber-security[104]. It is used in generating to-and-from (bidirectional) flows where the first packet defines the source-to-destination packets and destination-to-source packets[105].

The ".csv" files were later imported into the MATLAB application software for machine learning classification, training, and prediction. MATLAB is a programming platform for technical computing and a wide range of engineering and scientific problems of any complexity in any industry[106]. It is specifically designed for scientists and engineers to design and analyze products and systems used to transform our world [107]. The aforementioned software are further described in Table 5.4 below;

Table 5.4. Used Software

Software	Version	Platform	Category	Used For	Output Format
Cicflowmeter	v.4.0	Windows	Network traffic flow generator and analyser	Generating Biflows from pcap files	csv
Wireshark	v3.6.0	Windows	Network Protocol analyser	Packet sniffing to generate offline data	pcap
MATLAB	R2021b	Windows	Programming and Numeric computing	Classification Learning	

5.2. METHOD

Concerning the review carried out on different approaches to intrusion detection in IoT networks as seen in this paper, we decided to employ the anomaly-based intrusion detection technique and machine learning methods for this research methodology. Furthermore, to evaluate the performance of an IoT anomaly IDS, raw data is highly essential. The data should encompass various network features like flow, behavior, payload, source, and destination port number.

Having acquired the hardware, software, and SBCs earlier mentioned in this research, the network architecture was designed as seen in Figure 5.2. The devices were all connected and configured as per the research requirement. The cyber-attack categories chosen were launched on different nodes within the network and the packets were sniffed via Wireshark to generate the pcap files. The overall process starting from the attack launching stage to the final stage of prediction can be seen in Figure 5.1. For further understanding of the whole methodology, the network structure, selected attacks, dataset generation, preprocessing, training, and prediction are explained herein below;

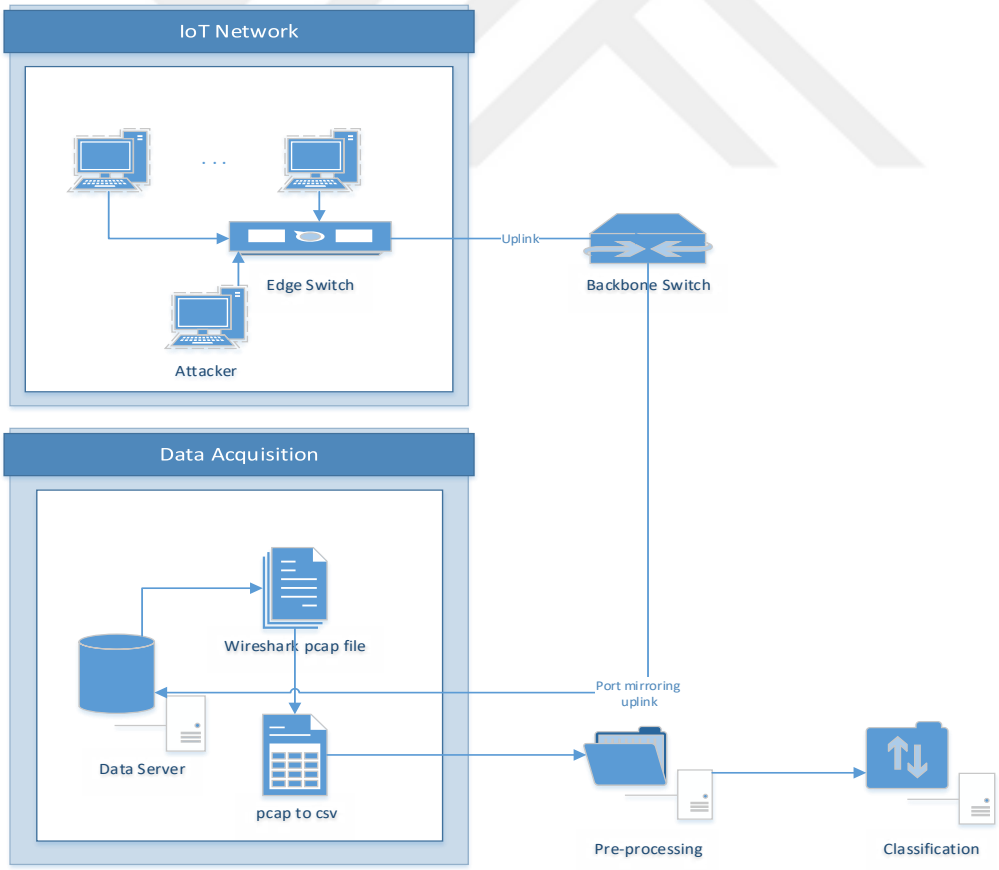

Figure 5.1 Proposed Method

5.2.1. Network Structure:

The whole network is centralized on an Enterasys switch (A2H124-24) whilst the switch is connected to the university's internet. The virtual machines were deployed on the Fujitsu victim PC with the required OS installed on each machine. The SBCs were also connected to the switch and configured. The default RaspberryPi OS was installed on the RaspberryPi SBC and the Arduino was connected to the PC via USB with the Arduino IDE application as the mediator. The Arduino compatible Ethernet board (WIZnet Ethernet WS100) was then connected to the Arduino and configured via the IDE as well. From this point, the Arduino was assigned an IP and server address for accessibility via browser. Some sensors and cameras were then connected to the Arduino as well as the RaspberryPi. The Odroid was also connected to the network just like the Arduino and some sensors were also connected to it. The network topology for the IoT network is shown in Figure 5.2 below;

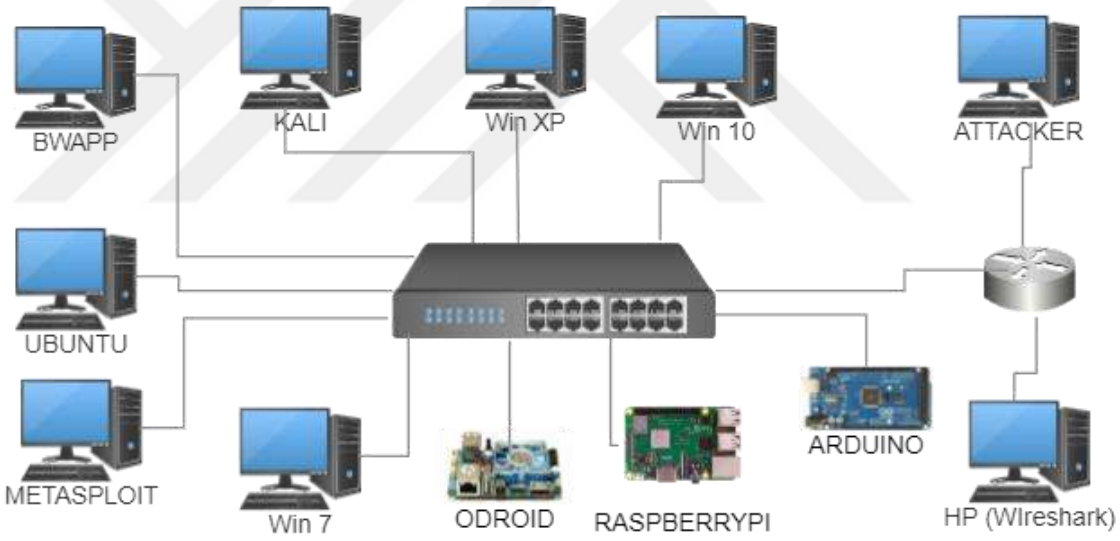

Figure 5.2. The Ideal IoT Network Model Architecture

5.2.2. Executed Attack Categories

Cyber-attacks were the central issue that brought about this research in the first place. The whole research is focused on finding reliable ways of detecting cyber-attacks while they happen so as to be able to prevent such attacks instantly or in the future. Because this research suggested the use of a self-generated dataset, some cyber-attacks were selected for the research at hand. Eight attack categories were selected and launched successfully on the IoT network and its devices. The attacks were Brute force RDP, Brute force SSh, Brute force Telnet, DoS ICMP Flood, DoS SYN

Flood, DoS Http get Flood, Fuzzing, and SYN Scan. An additional class of normal traffic is also included making the overall class total to be nine, as shown in Table 5.5 below.

Table 5.5. Executed Attack Description

Attack Class	Description
Dos_Http_get_flood	Exploits seemingly-legitimate HTTP GET or POST requests
Fuzzing	Randomly feeds invalid and unexpected data inputs to stress the network
SYN_Scan	Determines the status of communication ports without a full connection
Brute_Force_SSH	Brute force guessing via SSH ports
Brute_Force_RDP	Brute force guessing via RDP ports
Brute_Force_Telnet	Brute force guessing via Telnet protocol lines
DoS_ICMP_FLOOD	Overwhelms the target with ICMP ping requests
DoS_SYN_FLOOD	Rapidly sends a series of synchronized messages to the target
Normal	Normal network traffic without any compromise

5.2.3. Dataset Acquisition

As explained earlier in this chapter, raw data is the primary ingredient for the forensic evaluation of any IoT anomaly IDS. As such, the acquisition of the dataset for our experimentation and research became paramount since it has been proposed to be self-generated. As part of the contribution for this thesis, the dataset used was self-generated from the implemented IoT network.

Before launching any cyber-attack, the normal network traffic of the connected IoT devices and the overall network traffic was sniffed via Wireshark, and the pcap files saved were labeled as normal. Since eight attacks were required, the network traffic containing those attacks was also sniffed concurrently whilst the attacks were successfully executed on the run. The pcap files saved after successful execution of every attacking category were also labeled according to the name of the attack. When the pcap files for all the attacks required were saved, they were all parsed using the CICFlowmeter software and the ".csv" file for the dataset was generated. The collective attack dataset can be seen pictorially in Figure 5.3.

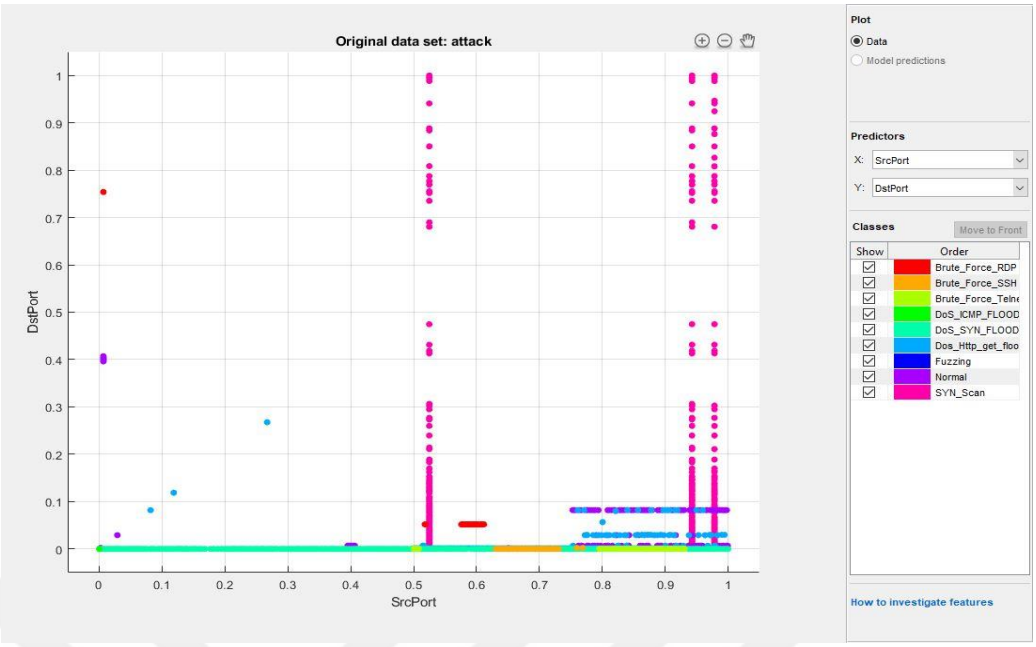

Figure 5.3. Final Attack Dataset

5.2.4. Preprocessing

Data preprocessing here involves transforming the input dataset: which was made up of 84 network features (classes) into an input compliance format for algorithms to be analyzed. To represent the categorical variables in binary vector formats, the "one-hot encoding" was applied to nominal features like IP address and protocol name. To avoid instability and other unforeseen circumstances, data normalization was also passed out on the remaining features due to the imbalance between them. The values were normalized ranging from 0 to 1. Furthermore, 75% of the original dataset (6750 records) were selected as training data for all the models, while the remaining 25% (2250 records) were used for testing. At this investigative phase, a high imbalance between normal and abnormal network class was encountered, which happens to be a common experience when anomalous traffic flow. Both the training and test data were homogenous with an abnormal-to-normal ratio of 1:4. The equation for the normalization can be seen in Equation 5.1

$$Xnorm = \frac{X - Xmin}{Xmax - Xmin} \tag{5.1}$$

5.2.5. Classification and performance matrices

The machine learning methods utilized were also selected based on the systematic analysis of existing approaches used to identify network anomalies in cyber-security. The analysis of indicators for detecting anomalies in network traffic was further done on models from different classes as well as models from the same class that have a varying number of neutrons and layers.

This allowed us to experimentally determine the relationship between the structure of the models and their performance. Based on the above, the following models of machine learning were analyzed: Decision Trees (DT), Support Vector Machine (SVM), k-Nearest Neighbors (KNN), and Ensemble.

The indication metrics for measuring the efficiency of detection algorithms used were: level of accuracy, prediction speed, training time, and the training time of all algorithms. For the classification, multi-class classification was used instead of binary classification to ascertain the attack category when the anomaly is detected. The multi-class classes summed up to nine classes (8 attacks + Normal). For each classification algorithm, 75% of the random records of the dataset converted to ".csv" via CICFlowmeter were used for training, while the remaining 25% were used for the accuracy assessment using 5 block cross-validation.

5.2.6. Step-by-step Method Flow:

Step 1: A literature review was firstly conducted in order to ascertain the situation with the topic of research at hand.

Step 2: An ideal IoT network was then designed and implemented solely for the simulation and experimentations with respect to the research at hand.

Step 3: Various attacks were practically launched on the IoT network (PC's and IoT devices).

Step 4: The packets within the network were sniffed by listening to switch ports via Wireshark and pcap files saved.

Step 5: The pcap files were parsed and the structured data created

Multi-class classification learning algorithms were employed on the dataset for training and prediction via MATLAB.

6. EXPERIMENTAL RESULTS AND DISCUSSION

The dataset (.csv file) generated from the CICflowmeter initially contained 9002 records and 84 features, the 9002 records represent the attack traffic as well as the normal traffic with a total size of 11.7MB. The file was later preprocessed by deleting unnecessary features that might slow the classification process or even compromise it. The deleted features were mostly deleted because they are strings (not numbers) or have zero values all through. The deleted features and used features can both be seen in Table 6.1 and Table 6.2 respectively. All values were later normalized using the minimum to maximum normalization between 1 and 0 for faster training and classification by the models. After the normalization, the file shrinks to 4.13MB with 74 features (classes) and 9000 records, where each class has 1000 records. The last feature (class) contains the labeled name of each attack traffic, hence the classes to be used for the multi-class classification algorithms.

The dataset was then successfully imported into the MATLAB environment without any error and all features and structures were intact. At this stage, the proposed classification models were applied to the dataset after the training and testing dataset were separated. The training went smooth and good results were achieved as anticipated from the selected machine learning algorithms. The result of the general prediction can be seen in Figure 6.1. The training and testing were further reinitiated twice to see if the results will differ variably. The results of the re-run came out almost similar with differences of less than 1%. Here, the average of the results for each algorithm was taken as seen in Table 6.3.

Table 6.1. Obtained Features

Features	Description	Features	Description
Src Port	Source port address	Bwd URG Flags	No. of times the URG flag was set in packets travelling in the backward direction
Dst Port	Destination Port address	Fwd PSH Flags	No. of times the PSH flag was set in packets travelling in the forward direction
Protocol	Network Protocol	FIN Flag Count	Number of packets with FIN
Flow Duration	Flow Duration	SYN Flag Count	Number of packets with SYN
Total Fwd Packet	Total Packets in the forward direction	RST Flag Count	Number of packets with RST
Total Bwd packets	Total Packets in the backward direction	PSH Flag Count	Number of packets with PSH
Total Length of Fwd Packet	Total size of Packets in the forward direction	ACK Flag Count	Number of packets with ACK
Total Length of Bwd Packet	Total size of Packets in the backward direction	URG Flag Count	Number of packets with URG
Fwd Packet Length Max	Maximum size of Packets in the forward direction	CWR Flag Count	Number of packets with CWR
Fwd Packet Length Min	Minimum size of Packets in the backward direction	ECE Flag Count	Number of packets with ECE
Fwd Packet Length Mean	Average size of Packets in the forward direction	Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
Fwd Packet Length Std	Standard size of Packets in the forward direction	Average Packet Size	Average Packet Size
Bwd Packet Length Max	Maximum size of Packets in the backward direction	Fwd Segment Size Avg	Average size observed in the forward direction
Bwd Packet Length Min	Minimum size of Packets in the backward direction	Bwd Segment Size Avg	Average size observed in the backward direction
Bwd Packet Length Mean	Average size of Packets in the backward direction	Bwd Bytes/Bulk Avg	Average number of bytes bulk rate in the backward direction
Bwd Packet Length Std	Standard size of Packets in the backward direction	Bwd Packet/Bulk Avg	Average number of packets bulk rate in the backward direction
Flow Bytes/s	Flow byte rate; that is number of packets transferred per second	Bwd Bulk Rate Avg	Average number of bulk rate in the backward direction
Flow Packets/s	Flow packet rate; that is number of packets transferred per second	Subflow Fwd Packets	Average number of packets in a sub flow in the forward direction
Flow IAT Mean	Average time between two flows	Subflow Fwd Bytes	Average number of bytes in a sub flow in the forward direction
Flow IAT Std	Standard deviation time of two flows	Subflow Bwd Bytes	Average number of bytes in a sub flow in the backward direction
Flow IAT Max	Maximum time between two flows	FWD Init Win Bytes	Number of bytes sent in initial window in the forward direction
Flow IAT Min	Minimum time between two flows	Bwd Init Win Bytes	Number of bytes sent in initial window in the backward direction
Fwd IAT Total	Total time between two packets sent in the forward direction	Fwd Act Data Pkts	N0. of packets with atleast 1byte of TCP data payload in the forward direction
Fwd IAT Mean	Average time between two packets sent in the forward direction	Fwd Seg Size Min	Minimum segment size observed in the forward direction
Down/Up Ratio	Download and upload ratio	Active Mean	Average time a flow was active before becoming idle
Fwd IAT Max	Maximum time between two packets sent in the forward direction	Active Std	Standard deviation time a flow was active before becoming idle
Fwd IAT Min	Minimum time between two packets sent in the forward direction	Active Max	Maximum time a flow was active before becoming idle
Bwd IAT Total	Total time between two packets sent in the backward direction	Active Min	Minimum time a flow was active before becoming idle
Bwd IAT Mean	Average time between two packets sent in the backward direction	Idle Mean	Average time a flow was idle before becoming active
Bwd IAT Std	Standard time between two packets sent in the backward direction	Packet Length Max	Maximum length of a flow
Bwd IAT Max	Maximum time between two packets sent in the backward direction	Packet Length Mean	Average length of a flow
Bwd IAT Min	Minimum time between two packets sent in the backward direction	Packet Length Std	Standard deviation length of a flow
Fwd Header Length	Total bytes used for headers in the forward direction	Packet Length Variance	Minimum inter-arrival time of packet
Bwd Header Length	Total bytes used for headers in the backward direction	Idle Std	Standard deviation time a flow was idle before becoming active
Fwd Packets/s	Number of forward packets per second	Idle Max	Maximum time a flow was idle before becoming active
Bwd Packets/s	Number of backward packets per second	Idle Min	Minimum time a flow was idle before becoming active
Packet Length Min	Minimum length of a flow	Label	Attack class name

Table 6.2. Reformed Features

Flow ID	Flow Identifier
Src IP	Source IP address
Dst IP	Destination IP Address
Timestamp	Timestamp
Bwd PSH Flags	Number of times the PSH flag was set in packets traveling in the backward direction (0 for UDP)
Fwd URG Flags	Number of times the PSH flag was set in packets traveling in the backward direction (0 for UDP)
Fwd Bytes/Bulk Avg	Number of times the PSH flag was set in packets traveling in the backward direction (0 for UDP)
Fwd Packet/Bulk Avg	Number of times the PSH flag was set in packets traveling in the backward direction (0 for UDP)
Fwd Bulk Rate Avg	Number of times the PSH flag was set in packets traveling in the backward direction (0 for UDP)
Subflow Bwd Packets	Number of times the PSH flag was set in packets traveling in the backward direction (0 for UDP)

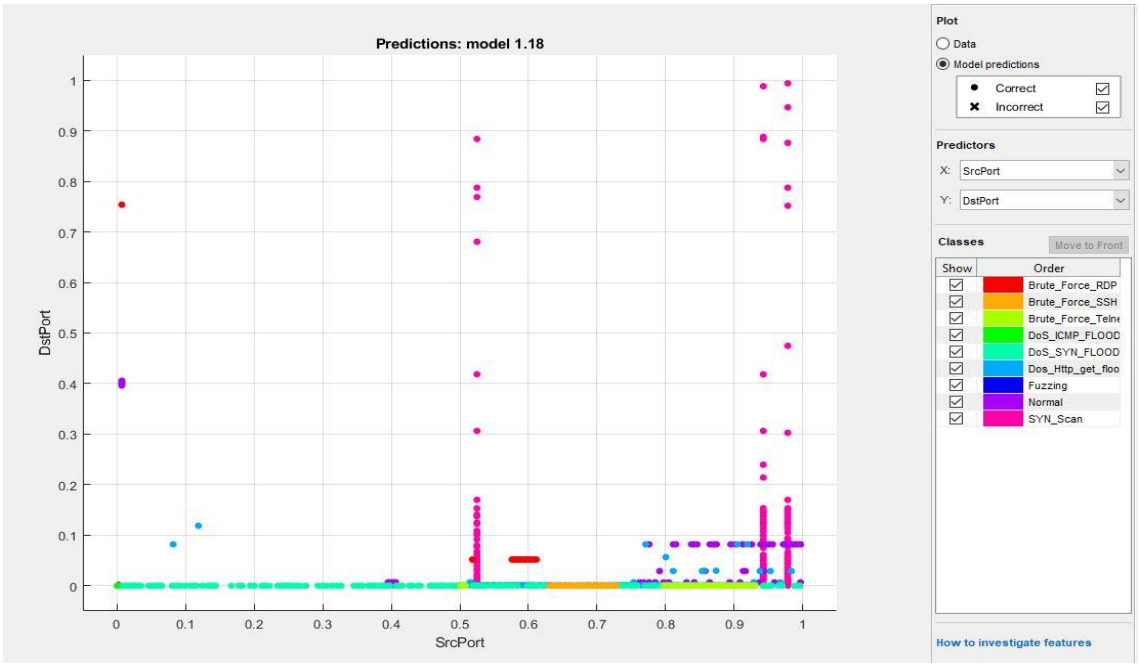

Figure 6.1. Predicted Results

For the Tree algorithms, the fine tree yielded more accuracy (99.8%) followed by the medium tree (99.7%) as can be seen in Figure 6.2, and finally the coarse tree which produced a very poor result (55.6%). For training time, the fine tree took longer (1.4451 sec) and the medium tree took an average time (0.97806 sec) while the coarse tree took the lesser time (0.80334).

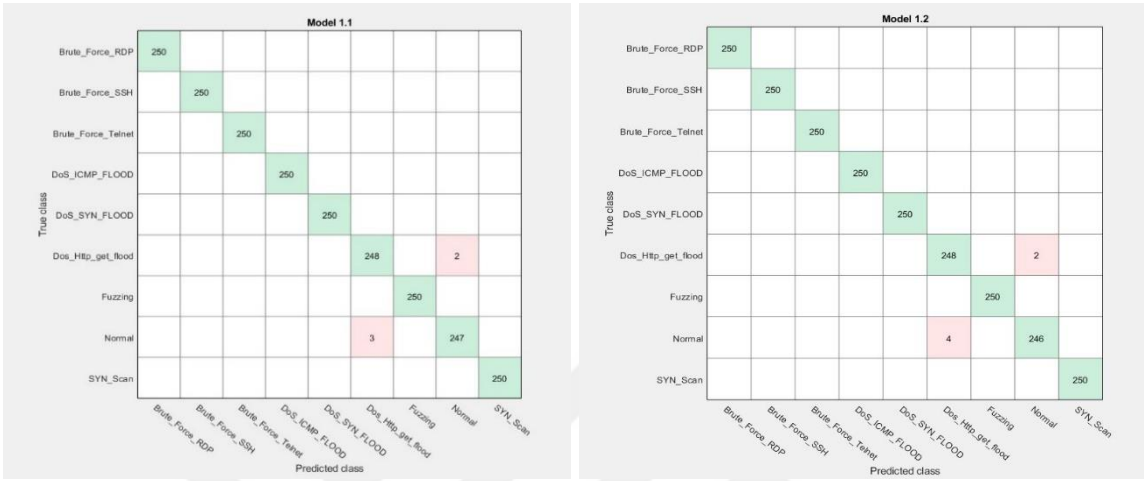

Figure 6.2. Best results of the Tree Algorithms

From Figure 6.3, we can see that the cubic and quadratic SVMs portrayed the highest accuracies of 98.5%, followed by the linear SVM (98.4%). The fine Gaussian yielded an accuracy of 95.8%, the medium Gaussian 98.6%, and finally the coarse Gaussian 96.8%.

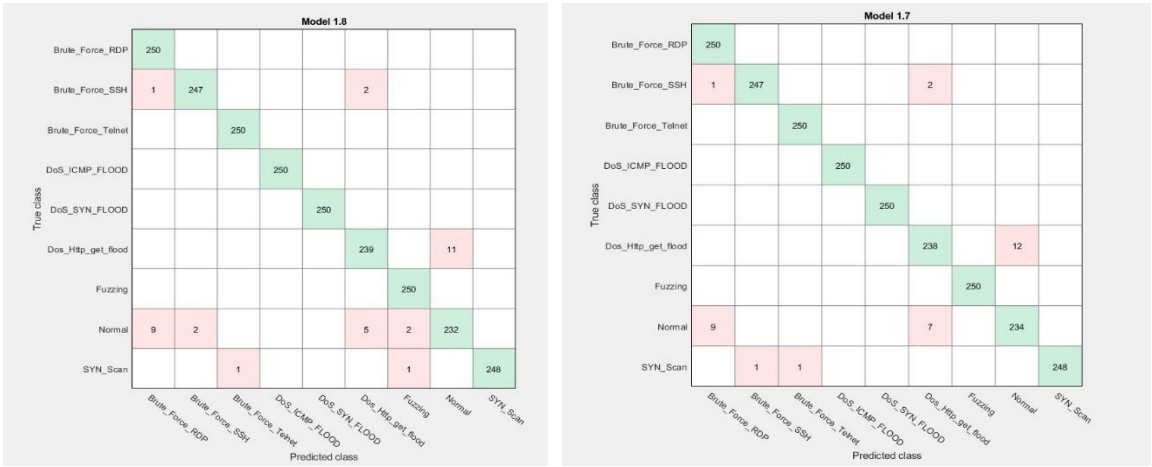

Figure 6.3. Best results of the SVM algorithms

Out of the KNN models, the fine KNN yielded the best result (97.3%), followed by the weighted KNN (97.2%), this can be clearly seen pictorially in Figure 6.4. The medium KNN also yielded a good result (96.2%) while the coarse KNN is 89.2%, the cosine KNN finished at the accuracy of 96.8%, and finally the cubic KNN (95.8%).

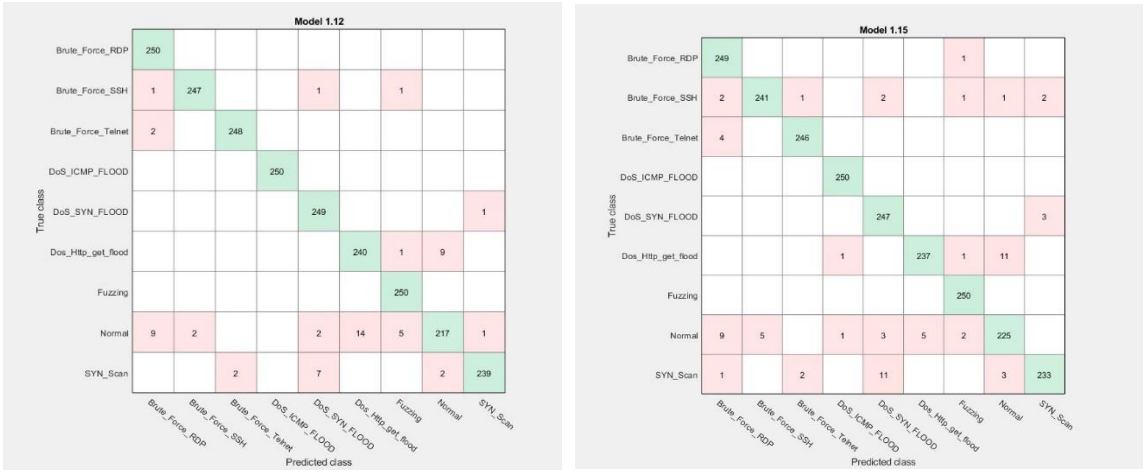

Figure 6.4. Best results of the KNN

All the ensemble models produced higher accuracy rates when compared to the other models mentioned above. The boosted and bagged trees all yielded an accuracy of 99.9% each followed by the RUSBoosted trees (99.7%) while the subspace discriminant and subspace KNN yielded 96.0% and 99.3% respectively. The success rate of this algorithm over the other algorithms is clearly visible in Figure 6.5 since the boosted trees represented in model 1.18 mispredicted one instance only and the bagged trees mispredicted two instances only.

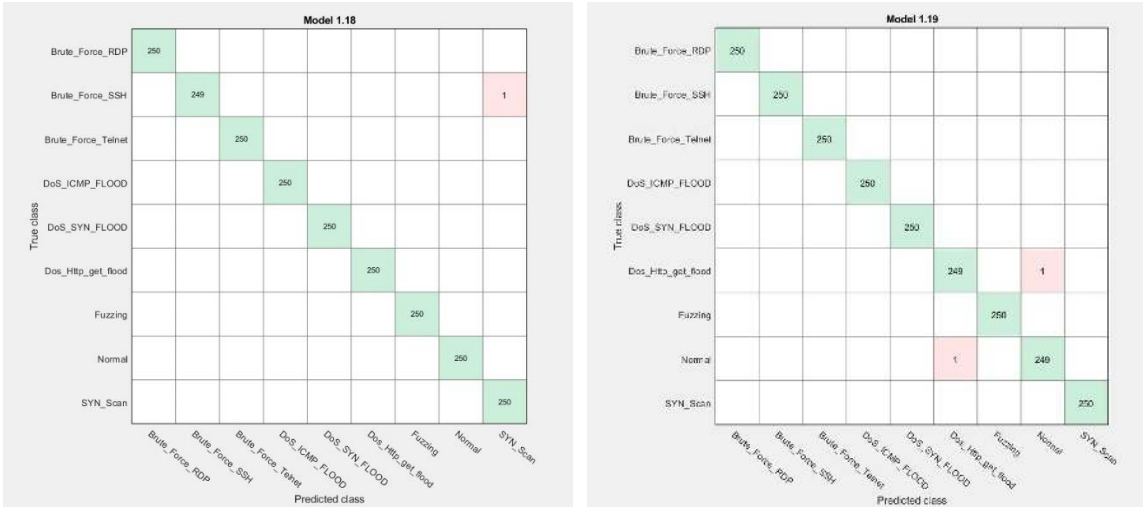

Figure 6.5. Best results of the Ensemble

The results of the experiments performed suggest that most of the machine learning algorithms used have a high level of accuracy in detecting large heterogeneous traffic anomalies for real-world applications. This was further confirmed by comparing the results we obtained with

the results obtained from similar experimentations by other authors in various research publications.

Table 6.3. Average results for each algorithm

Main Algorithm	Sub Algorithm	Accuracy (%)	Prediction Speed obs/sec	Training Time (Sec)
DT	Fine Tree	99.8	110000	1.4451
	Medium Tree	99.7	130000	0.97806
	Coarse Tree	55.6	160000	0.80334
SVM	Linear SVM	98.4	11000	6.0446
	Quadratic SVM	98.5	7800	8.8809
	Cubic SVM	98.5	10000	207.22
KNN	Fine KNN	97.3	1400	3.937
	Medium KNN	96.2	1300	4.3359
	Cosine KNN	96.8	1200	7.8711
Ensemble	Boosted Trees	99.9	21000	19.009
	Bagged Trees	99.9	18000	11.013
	Subspace KNN	99.3	120	48.567

The results obtained were further analyzed and compared in Table 6.4 according to their accuracies, precision, f-measure, and geometric progression. The formulae for calculating each metric are given in the equations 6.1-6.4.

$$Accuracy = \frac{TP + TN}{(TP + TN) + (FP + FN)} \tag{6.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

$$Geometric\ Mean = \sqrt{\frac{TP * TN}{(TP + FN) * (TN + FP)}} \tag{6.3}$$

$$F - Measure = \frac{2TP}{2TP + FP + FN} \tag{6.4}$$

Table 6.4. Results Comparison

Main Algorithm	Sub Class	Percentage Accuracy	Percentage Precision	Geometric Mean	F-Measure
DT	Fine Tree	0.9982	0.9975	0.9975	0.9976
	Medium Tree	0.9978	0.9975	0.7983	0.9976
	Coarse Tree	0.5564	0.5562	0.5542	0.5562
SVM	Linear SVM	0.9843	0.9764	0.9623	0.9714
	Quadratic SVM	0.9851	0.9802	0.9782	0.9842
	Cubic SVM	0.9852	0.9765	0.9736	0.9758
KNN	Fine KNN	0.9734	0.9544	0.9476	0.9553
	Medium KNN	0.9621	0.9453	0.9365	0.9437
	Cosine KNN	0.9683	0.9568	0.9384	0.9586
Ensemble	Boosted Trees	0.9998	0.9995	0.9996	0.9996
	Bagged Trees	0.9998	0.9994	0.9994	0.9994
	Subspace KNN	0.9934	0.9932	0.9931	0.9931

Comparing the results above with the reviewed literature, we can conclude that the machine learning algorithms proposed yielded similar results with papers that employed similar models like [100], [108].

7. CONCLUSIONS

In summary, due to a large amount of metadata to be analyzed, today's intrusion detection and prevention methods require modification using various machine learning approaches to detect malicious activities in IoT network traffic or devices. A reference solution designed for the forensic purpose of detecting and preventing cyber-attacks should analyze the incoming data set of any size, quality, and depth, using artificial intelligence methods to classify traffic as legitimate or as malicious. In the latter case, the software must provide a list of threats, with their classification, probabilities, and attack trajectories to support cyber-forensic investigations. At this stage, it is noteworthy to state the following;

- The IoT network designed for the experimentations was physically implemented and running.
- All cyber-attacks presented were executed real-time on the functional network
- The dataset used for the research was self-generated from the live traffic of the implemented network.
- The Proposed approaches yielded excellent results as anticipated, making the entire research a successful one.

RECOMMENDATIONS

Staying at least one step ahead of cyber-attack perpetrators is the only way to win in the war against cyber-attacks. Researchers and practitioners in the field of forensics, cyber security, and IoT security must continue to update their tools of the trade. As such, the journey to attaining maximum security should be a never-ending one because deploying more IoT technologies increases the number of threats since the adversaries will have more targets especially when those targets have the low processing power and low-level security interfaces.

REFERENCES

[1] B. Z. Adamu, M. Karabatak, and F. Ertam, "A Conceptual Framework for Database Anti-forensics Impact Mitigation," 2020, doi: 10.1109/ISDFS49300.2020.9116375.

[2] H. Tahaei, F. Afifi, A. Asemi, F. Zaki, and N. B. Anuar, "The rise of traffic classification in IoT networks: A survey," *J. Netw. Comput. Appl.*, vol. 154, no. September 2019, 2020, doi: 10.1016/j.jnca.2020.102538.

[3] PWC, "2019 IoT Survey: Speed operations, strengthen relationships and drive what's next," *Emerging Technology*, 2019. https://www.pwc.com/us/en/services/consulting/technology/emerging-technology/iot-pov.html (accessed Jan. 15, 2021).

[4] Forbes, "Roundup Of Internet Of Things Forecasts," *Roundup Of Internet Of Things Forecasts*, 2017. https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/?sh=697c72211480.

[5] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT Scenarios," *IEEE Access*, vol. 8, no. February, pp. 23022–23040, 2020, doi: 10.1109/ACCESS.2020.2970118.

[6] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0268-2.

[7] A. Holst, "IoT connected devices worldwide 2019-2030," *Statista*, 2021. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed Jan. 19, 2022).

[8] T. H. E. Smart, U. Revolution, R. F. Mesh, T. International, and E. Agency, "WHY UTILITIES ARE CHOOSING," no. January, 2021.

[9] UNITED NATIONS CONFERENCE ON TRADE AND DEVELOPMENT., *DIGITAL ECONOMY REPORT 2019 : value creation and capture - implications for developing countries.* 2019.

[10] M. P. Mahajan, R. R. Nikam, V. P. Patil, and R. D. Dond, "Smart Refrigerator Using IOT," *Int. J. Latest Eng. Res. Appl.*, pp. 86–91, 2017, [Online]. Available: www.ijlera.com.

[11] M. U.Farooq, M. Waseem, S. Mazhar, A. Khairi, and T. Kamal, "A Review on Internet of Things (IoT)," *Int. J. Comput. Appl.*, vol. 113, no. 1, pp. 1–7, 2015, doi: 10.5120/19787-1571.

[12] T. Internet, "ITU Internet Reports 2005: The Internet of Things – Executive Summary," [Online]. Available: http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf.

[13] D.-L. Yang, F. Liu, and Y.-D. Liang, "A Survey of the Internet of Things," 2010, doi: 10.2991/icebi.2010.72.

[14] B. Mahbooba, R. Sahal, W. Alosaimi, and M. Serrano, "Trust in Intrusion Detection Systems: An Investigation of Performance Analysis for Machine Learning and Deep Learning Models," *Complexity*, vol. 2021, no. 1, 2021, doi: 10.1155/2021/5538896.

[15] J. Vávra, M. Hromada, L. Lukáš, and J. Dworzecki, "Adaptive anomaly detection system based on

machine learning algorithms in an industrial control environment," *Int. J. Crit. Infrastruct. Prot.*, vol. 34, p. 100446, 2021, doi: https://doi.org/10.1016/j.ijcip.2021.100446.

[16] C. Wang, B. Wang, H. Liu, and H. Qu, "Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network," *Wirel. Commun. Mob. Comput.*, vol. 2020, pp. 1–10, 2020, doi: 10.1155/2020/8897926.

[17] G. Parra, P. Rad, K.-K. R. Choo, and N. Beebe, "Detecting Internet of Things attacks using distributed deep learning," *J. Netw. Comput. Appl.*, vol. 163, p. 102662, 2020, doi: 10.1016/j.jnca.2020.102662.

[18] S. Li, T. Tryfonas, and H. Li, "The Internet of Things: a security point of view," *Internet Res.*, vol. 26, pp. 337–359, 2016, doi: 10.1108/IntR-07-2014-0173.

[19] P. Goyal, A. K. Sahoo, and T. K. Sharma, "Internet of things: Architecture and enabling technologies," *Mater. Today Proc.*, no. xxxx, 2020, doi: 10.1016/j.matpr.2020.04.678.

[20] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018, doi: 10.1016/j.jksuci.2016.10.003.

[21] S. Naveen and S. Hegde, "Study of IoT: Understanding IoT Architecture, Applications, Issues and Challenges," *Int. J. Adv. Netw. Appl.*, no. January, pp. 477–482, 2016, [Online]. Available: http://www.ijana.in/Special Issue/S105.pdf.

[22] A. H. Srinivasa and Siddaraju, "A comprehensive study of architecture, protocols and enabling applications in internet of things (Iot)," *Int. J. Sci. Technol. Res.*, vol. 8, no. 11, pp. 1767–1779, 2019.

[23] X. Jia, Q. Feng, T. Fan, and Q. Lei, "RFID technology and its applications in Internet of Things (IoT)," *2012 2nd Int. Conf. Consum. Electron. Commun. Networks, CECNet 2012 - Proc.*, no. July, pp. 1282–1285, 2012, doi: 10.1109/CECNet.2012.6201508.

[24] Kubo, "The research of IoT based on RFID technology," *Proc. - 7th Int. Conf. Intell. Comput. Technol. Autom. ICICTA 2014*, pp. 832–835, 2015, doi: 10.1109/ICICTA.2014.199.

[25] A. Flammini and E. Sisinni, "Wireless sensor networking in the internet of things and cloud computing era," *Procedia Eng.*, vol. 87, no. December, pp. 672–679, 2014, doi: 10.1016/j.proeng.2014.11.577.

[26] P. N. Borza, M. Machedon-Pisu, and F. G. Hamza-Lup, "Design of Wireless Sensors for IoT with Energy Storage and Communication Channel Heterogeneity," *arXiv*, 2019.

[27] S. C. Abraham, "Internet of Things (IoT) with Cloud Computing and Machine-to-Machine (M2M) Communication," *Int. J. Emerg. Trends Sci. Technol.*, no. September 2016, 2016, doi: 10.18535/ijetst/v3i09.13.

[28] J. Osborne, "Internet of things and cloud computing," *Internet Things Data Anal. Handb.*, pp. 683–698, 2017, doi: 10.1002/9781119173601.ch42.

[29] M. G. Avram, "Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective," *Procedia Technol.*, vol. 12, pp. 529–534, 2014, doi: 10.1016/j.protcy.2013.12.525.

[30] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018, doi: 10.1016/j.jksuci.2016.10.003.

[31] J. Jeong *et al.*, "A comprehensive survey on vehicular networks for smart roads: A focus on IP-based approaches," *Veh. Commun.*, vol. 29, p. 100334, 2021, doi: 10.1016/j.vehcom.2021.100334.

[32] K. Aishwarya and R. Manjesh, "A Novel Technique for Vehicle Theft Detection System Using MQTT on IoT," 2020, pp. 725–733.

[33] R. Thapliyal, R. Kumar Patel, A. Kumar Yadav, and A. Singh, “Internet of Things for Smart Environment and Integrated Ecosystem,” *Int. J. Eng. Technol.*, vol. 7, no. 3.12, p. 1218, 2018, doi: 10.14419/ijet.v7i3.12.17841.

[34] F. Al-Turjman and C. Altrjman, “IoT Smart Homes and Security Issues: An Overview,” 2019, pp. 111–137.

[35] P. Nhan, D. Nguyen, and Y. Le, “BUILDING SMART HOSPITAL SYSTEM,” *Sci. J. Tra Vinh Univ.*, vol. 1, pp. 48–55, 2019, doi: 10.35382/18594816.1.1.2019.87.

[36] K. E. Psannis and T. Xifilidis, “Smart Hospitals : The future for care provision ? Speakers : Mr . Vasileios A . Memos Contributors : Mr . Andreas Plageras University of Macedonia (Greece) 5 th International Summer School of Medical Law in Bioethics on ‘ Human Rights in Health ’ July 2,” no. July, 2020, doi: 10.13140/RG.2.2.15302.29761.

[37] B. Murthy and S. Peddoju, “IoT-Based Patient Health Monitoring: A Comprehensive Survey,” 2021, pp. 349–356.

[38] A. D. Boursianis *et al.*, “Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review,” *Internet of Things*, no. xxxx, p. 100187, 2020, doi: 10.1016/j.iot.2020.100187.

[39] N. Bansal, “IoT Applications in Retail,” 2020, pp. 217–238.

[40] M. Graswald, “IoT Applications in Defense: Research and Implementation of Intelligent Systems.” 2017.

[41] J. Voas, L. Feldman, and G. Witte, “Itl Bulletin for September 2016 Demystifying the Internet of Things,” no. September, pp. 1–4, 2016.

[42] J. M. Voas, “NIST Special Publication 800-183 - Networks of ‘things,’” *NIST Spec. Publ.*, p. 30, 2016.

[43] P. A. Laplante, J. Voas, and N. Laplante, “Standards for the Internet of Things: A Case Study in Disaster Response,” *Computer (Long. Beach. Calif).*, vol. 49, no. 5, pp. 87–90, 2016, doi: 10.1109/MC.2016.137.

[44] J. J. Amalraj, S. Banumathi, and J. J. John, “IOT sensors and applications: A survey,” *Int. J. Sci. Technol. Res.*, vol. 8, no. 8, pp. 998–1003, 2019.

[45] B. Pourghebleh and N. J. Navimipour, “Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research,” *J. Netw. Comput. Appl.*, vol. 97, no. August, pp. 23–34, 2017, doi: 10.1016/j.jnca.2017.08.006.

[46] F. K. Shaikh, B. S. Chowdhry, H. M. Ammari, A. Shah, and M. A. Uqaili, “Preface,” *Commun. Comput. Inf. Sci.*, vol. 366 CCIS, no. December, 2013, doi: 10.1007/978-3-642-41054-3.

[47] National Institute of Standards and Technology, “NISTIR 8062: An introduction to privacy engineering and risk management in federal systems,” p. 49, 2017, [Online]. Available: https://doi.org/10.6028/NIST.IR.8062.

[48] T. Hedberg and T. Hedberg, “NIST Advanced Manufacturing Series 200-1 Design and Confguration of the Smart Manufacturing Systems Test Bed NIST Advanced Manufacturing Series 200-1 Design and Confguration of the Smart Manufacturing Systems Test Bed,” *Nist*, pp. 200–1, 2017, [Online]. Available: https://doi.org/10.6028/NIST.AMS.200-1.

[49] K. E. Ibrahim, S. Majumdar, D. Bastos, and A. Singhal, “Poster : Defining Actionable Rules for Verifying IoT Security.”

[50] M. Fagan, K. N. Megas, K. Scarfone, and M. Smith, “IoT device cybersecurity capability core baseline,” 2020, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259A.pdf.

[51] M. Burns, “Cyber Physical Systems and Internet of Things in Industry,” pp. 2839–2840, 2019, doi: 10.1109/iecon.2018.8591610.

[52] K. K. R. Choo, K. Gai, L. Chiaraviglio, and Q. Yang, “A multidisciplinary approach to Internet of Things (IoT) cybersecurity and risk management,” *Comput. Secur.*, vol. 102, pp. 2020–2022, 2021, doi: 10.1016/j.cose.2020.102136.

[53] S. Naveen and S. Hegde, “Study of IoT: Understanding IoT Architecture, Applications, Issues and Challenges,” *Int. J. Adv. Netw. Appl.*, no. January, pp. 477–482, 2016.

[54] M. Khalifa, F. Algarni, M. Ayoub Khan, A. Ullah, and K. Aloufi, “A lightweight cryptography (LWC) framework to secure memory heap in Internet of Things,” *Alexandria Eng. J.*, vol. 60, no. 1, pp. 1489–1497, 2020, doi: 10.1016/j.aej.2020.11.003.

[55] J. Li *et al.*, “A fast and scalable authentication scheme in IOT for smart living,” *Futur. Gener. Comput. Syst.*, vol. 117, pp. 125–137, 2021, doi: 10.1016/j.future.2020.11.006.

[56] A. D. Jurcut, P. Ranaweera, and L. Xu, *Introduction to IoT Security*, no. December. 2019.

[57] R. K. Jha, Puja, H. Kour, M. Kumar, and S. Jain, “Layer based security in Narrow Band Internet of Things (NB-IoT),” *Comput. Networks*, vol. 185, no. August 2020, p. 107592, 2020, doi: 10.1016/j.comnet.2020.107592.

[58] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, “Network-Level Security and Privacy Control for Smart-Home IoT Devices,” no. June 2016, 2015, doi: 10.1109/WiMOB.2015.7347956.

[59] J. Norman and D. P. Joseph, “Security in Application Layer Protocols of IoT:,” 2017.

[60] A. Maizate, S. Aouad, and zakari abdelouahed, “Cyber Security and the Internet of Things : vulnerabilities and Security requirements,” 2019.

[61] B. Mostafa, “Monitoring Internet of Things Networks,” 2019, pp. 295–298, doi: 10.1109/WF-IoT.2019.8767203.

[62] R. Sofia and P. Mendes, “ICN applicability in IoT: The Need for Push-based Communication.” 2018.

[63] V. Casola, A. De Benedictis, A. Riccio, D. Rivera, W. Mallouli, and E. de Oca, “A security monitoring system for Internet of Things,” *Internet of Things*, vol. 7, p. 100080, 2019, doi: 10.1016/j.iot.2019.100080.

[64] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, “Intrusion detection systems for IoT-based smart environments: a survey,” *J. Cloud Comput.*, vol. 7, no. 1, pp. 1–20, 2018, doi: 10.1186/s13677-018-0123-6.

[65] S. Das, M. Ashrafuzzaman, F. T. Sheldon, and S. Shiva, “Network Intrusion Detection using Natural Language Processing and Ensemble Machine Learning,” *2020 IEEE Symp. Ser. Comput. Intell. SSCI 2020*, no. Ml, pp. 829–835, 2020, doi: 10.1109/SSCI47803.2020.9308268.

[66] K. Albulayhi, A. A. Smadi, F. T. Sheldon, and R. K. Abercrombie, “Iot intrusion detection taxonomy, reference architecture, and analyses,” *Sensors*, vol. 21, no. 19, 2021, doi: 10.3390/s21196432.

[67] J. Camacho, R. Theron, J. M. Garcia-Gimenez, G. MacIa-Fernandez, and P. Garcia-Teodoro, "Group-Wise Principal Component Analysis for Exploratory Intrusion Detection," *IEEE Access*, vol. 7, pp. 113081–113093, 2019, doi: 10.1109/ACCESS.2019.2935154.

[68] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Flow-based features for a robust intrusion detection system targeting mobile traffic," *2016 23rd Int. Conf. Telecommun. ICT 2016*, 2016, doi: 10.1109/ICT.2016.7500483.

[69] D. H. Summerville, K. M. Zach, and Y. L. Chen, "Ultra-lightweight deep packet anomaly detection for Internet of Things devices," *2015 IEEE 34th Int. Perform. Comput. Commun. Conf.*, pp. 1–8, 2015.

[70] N. V. Abhishek, T. J. Lim, B. K. Sikdar, and A. Tandon, "An Intrusion Detection System for Detecting Compromised Gateways in Clustered IoT Networks," *2018 IEEE Int. Work. Tech. Comm. Commun. Qual. Reliab.*, pp. 1–6, 2018.

[71] B. Arrington, L. Barnett, R. Rufus, and A. C. Esterline, "Behavioral Modeling Intrusion Detection System (BMIDS) Using Internet of Things (IoT) Behavior-Based Anomaly Detection via Immunity-Inspired Algorithms," *2016 25th Int. Conf. Comput. Commun. Networks*, pp. 1–6, 2016.

[72] L. Deng, D. Li, X. Yao, D. Cox, and H. Wang, "Mobile network intrusion detection for IoT system based on transfer learning algorithm," *Cluster Comput.*, vol. 22, pp. 9889–9904, 2018.

[73] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018, doi: 10.1109/COMST.2018.2844742.

[74] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. \& Secur.*, vol. 28, no. 1–2, pp. 18–28, 2009.

[75] Z. AliKhan and P. Herrmann, "Recent advancements in intrusion detection systems," *Secur. Commun. Networks*, vol. 2019, 2019.

[76] S. Chen, M. Peng, H. Xiong, and X. Yu, "SVM Intrusion Detection Model Based on Compressed Sampling," *J. Electr. Comput. Eng.*, vol. 2016, 2016, doi: 10.1155/2016/3095971.

[77] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust Malware Detection for Internet of (Battlefield) Things Devices Using Deep Eigenspace Learning," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 88–95, Jan. 2019, doi: 10.1109/TSUSC.2018.2809665.

[78] B. Arrington, L. E. Barnett, R. Rufus, and A. Esterline, "Behavioral modeling intrusion detection system (BMIDS) using internet of things (IoT) behavior-based anomaly detection via immunity-inspired algorithms," *2016 25th Int. Conf. Comput. Commun. Networks, ICCCN 2016*, pp. 1–6, 2016, doi: 10.1109/ICCCN.2016.7568495.

[79] E. Anthi, L. Williams, and P. Burnap, "Pulse: An adaptive intrusion detection for the Internet of Things," in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, Mar. 2018, pp. 1–4, doi: 10.1049/cp.2018.0035.

[80] J. Arshad, M. A. Azad, M. M. Abdellatif, M. H. Ur Rehman, and K. Salah, "COLIDE: A collaborative intrusion detection framework for Internet of Things," *IET Networks*, vol. 8, no. 1, pp. 3–14, 2019, doi: 10.1049/iet-net.2018.5036.

[81] F. Bao, I.-R. Chen, M. Chang, and J.-H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 9, no. 2, pp. 169–183, Jun. 2012, doi: 10.1109/TCOMM.2012.031912.110179.

[82] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 606–611.

[83] R. Fu, K. Zheng, D. Zhang, and Y. Yang, "An intrusion detection scheme based on anomaly mining in Internet of Things," 2011.

[84] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," *2013 IEEE 9th Int. Conf. Wirel. Mob. Comput. Netw. Commun.*, pp. 600–607, 2013.

[85] Z. A. Khan and P. Herrmann, "A Trust Based Distributed Intrusion Detection Mechanism for Internet of Things," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, Mar. 2017, pp. 1169–1176, doi: 10.1109/AINA.2017.161.

[86] Z. A. Khan, J. Ullrich, A. G. Voyiatzis, and P. Herrmann, "A Trust-Based Resilient Routing Mechanism for the Internet of Things," 2017, doi: 10.1145/3098954.3098963.

[87] Q. D. La, T. Q. S. Quek, J. Lee, S. Jin, and H. Zhu, "Deceptive Attack and Defense Game in Honeypot-Enabled Networks for the Internet of Things," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1025–1035, Dec. 2016, doi: 10.1109/JIOT.2016.2547994.

[88] J. Li, Z. Zhao, R. Li, and H. Zhang, "AI-Based Two-Stage Intrusion Detection for Software Defined IoT Networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2093–2102, Apr. 2019, doi: 10.1109/JIOT.2018.2883344.

[89] C. Liu, J. Yang, R. Chen, Y. Zhang, and J. Zeng, "Research on immunity-based intrusion detection technology for the Internet of Things," *2011 Seventh Int. Conf. Nat. Comput.*, vol. 1, pp. 212–216, 2011.

[90] Y. Liu and Q. Wu, "A lightweight anomaly mining algorithm in the Internet of Things," in *2014 IEEE 5th International Conference on Software Engineering and Service Science*, 2014, pp. 1142–1145.

[91] L. Liu, B. Xu, X. Zhang, and X. Wu, "An intrusion detection method for internet of things based on suppressed fuzzy clustering," *EURASIP J. Wirel. Commun. Netw.*, vol. 2018, no. 1, pp. 1–7, 2018.

[92] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013, doi: 10.1016/j.adhoc.2013.04.014.

[93] H. Sedjelmaci and M. Feham, "Novel Hybrid Intrusion Detection System For Clustered Wireless Sensor Network," *Int. J. Netw. Secur. Its Appl.*, vol. 3, 2011, doi: 10.5121/ijnsa.2011.3401.

[94] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6, doi: 10.1109/ICC.2016.7510811.

[95] A. Strikos, "A full approach for Intrusion Detection in Wireless Sensor Networks," 2007.

[96] W. Shin, "Countermeasures against Anti-forensics by Analyzing Anti-forensics Techniques," *J. Secur. Eng.*, vol. 11, no. 6, pp. 605–614, 2014, doi: 10.14257/jse.2014.12.05.

[97] C. Wang, T. Feng, J. Kim, G. Wang, and W. Zhang, "Catching Packet Droppers and Modifiers in

Wireless Sensor Networks," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Jun. 2009, pp. 1–9, doi: 10.1109/SAHCN.2009.5168914.

[98] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 41–49, 2018, doi: 10.1109/MSP.2018.2825478.

[99] K. Yang, J. Ren, Y. Zhu, and W. Zhang, "Active Learning for Wireless IoT Intrusion Detection," *IEEE Wirel. Commun.*, vol. 25, no. 6, pp. 19–25, Dec. 2018, doi: 10.1109/MWC.2017.1800079.

[100] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Comput. Networks*, vol. 188, no. October 2020, p. 107840, 2021, doi: 10.1016/j.comnet.2021.107840.

[101] "Microsoft Digital Defense Report," *Netw. Secur.*, vol. 2020, no. 10, p. 4, 2020, doi: 10.1016/s1353-4858(20)30114-8.

[102] K. Geetha and S. H. Brahmananda, "Securing IoT Using Artificial Intelligence and Feature Engineering," in *Soft Computing for Security Applications*, 2022, pp. 107–114.

[103] B. Liu, "Lifelong machine learning: a paradigm for continuous learning," *Front. Comput. Sci.*, vol. 11, 2016, doi: 10.1007/s11704-016-6903-6.

[104] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," *ICISSP 2016 - Proc. 2nd Int. Conf. Inf. Syst. Secur. Priv.*, no. Icissp, pp. 407–414, 2016, doi: 10.5220/0005740704070414.

[105] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," *ICISSP 2017 - Proc. 3rd Int. Conf. Inf. Syst. Secur. Priv.*, vol. 2017-Janua, no. Cic, pp. 253–262, 2017, doi: 10.5220/0006105602530262.

[106] P. Goyal, "MATLAB for Machine Learning."

[107] D. Houcque, "Introduction To Matlab for Engineering Students," no. August, 2005.

[108] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *J. Netw. Comput. Appl.*, vol. 161, no. March, 2020, doi: 10.1016/j.jnca.2020.102630.

CURRICULUM VITAE

Bashir Zak ADAMU

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]’

ACADEMIC ACTIVITIES

Paper:

1. B. Z. Adamu, M. Karabatak and F. Ertam, "A Conceptual Framework for Database Anti-forensics Impact Mitigation," 2020 8th International Symposium on Digital Forensics and Security (ISDFS), 2020, pp. 1-6, doi: 10.1109/ISDFS49300.2020.9116375.