**DEVELOPMENT OF IOT THREAT INTELLIGENCE SYSTEM**

**BY**

**MUHAMMED HARUNA**

**(200137)**

**BEING A PROJECT REPORT SUBMITTED**

**TO**

**DEPARTMENT OF CYBER SECURITY SCIENCE, FACULTY OF COMPUTING AND INFORMATICS, LADOKE AKINTOLA UNIVERSITY OF TECHNOLOGY, OGBOMOSO, OYO STATE, NIGERIA.**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF THE BACHELOR OF TECHNOLOGY (B.TECH) DEGREE INCYBER SECURITY SCIENCE**

**AUGUST, 2025**

# CERTIFICATION

This is to certify that this project work and report with the title Development of IoT Threat Intelligence System, submitted by **MUHAMMED Haruna**, with matric number 200137, was carried out under my supervision in the Department of Cyber Security Science, Faculty of Computing and Informatics, Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

**………………………………………………… ……………………….**

**Supervisor Date**

**Prof. A.O Afolabi, B.Sc., PGD, M.Tech., Ph.D.**

Department of Cyber Security Science,

Faculty of Computing and Informatics,

Ladoke Akintola University of Technology,

Ogbomoso, Oyo State, Nigeria.

**………………………………………………… ……………………….**

**Head of Department Date**

**Dr. (Mrs.) E.A. Amusan**

Department of Cyber Security Science,

Ladoke Akintola University of Technology,

Ogbomoso, Nigeria.

# DEDICATION

This project is dedicated to the Almighty God and my parents for their unwavering love and support.

# ACKNOWLEDGEMENT

All praise is due to Almighty Allah. I am profoundly grateful to Him for His endless mercy, grace, and strength throughout this program, for granting me the patience and opportunity to successfully complete my Final Year Project.

I sincerely thank my supervisor, Prof. A.O. Afolabi, for giving me the opportunity to work under his guidance. Your direction and the way you explained the project greatly increased my interest, and it has truly been a privilege to carry out this work under your supervision.

My deep appreciation goes to the Head of the Department of Cyber Security Science, Dr. (Mrs.) E.A. Amusan and Dr. (Mrs.) Alade, for their motherly advice and constant support during my undergraduate program. I also extend my gratitude to Dr. (Mrs.) Adedayo, Dr. Ojo and all the staff of the Department of Cyber Security Science for the gift of access and support.

I am truly grateful to my mother, Mrs. Muhammed Aminat. Maami, as I fondly call her. Thank you for believing in me and for your sacrifice. I am equally thankful to my siblings, Muhammed Muibat Shayo and Muhammed Fauziyat Adenike, for their true love, care, and support. Indeed, I am indebted to you both.

A big thank you goes to my mentor, Shuaib Oladigbolu (Sawzeeyy), for his unwavering support and encouragement throughout my academic journey. Thank you, sir, for the gift of access.

Special thanks also go to my roommates and friends, Abideen Farooq and Adegoke Abdulqoyyuum (Mister Circuit), for their understanding, encouragement, and for always standing by me. Finally, I deeply appreciate Lawal Yusuf (Soflaw) for his assistance in keeping materials and past questions, which helped me prepare ahead of each semester.

**TABLE OF CONTENTS**

[**CERTIFICATION i**](#_heading=h.q4esioctdxdt)

[**DEDICATION ii**](#_heading=h.59ncbny7l3yy)

[**ACKNOWLEDGEMENT iii**](#_heading=h.a8szfa57wxcl)

[**LIST OF TABLES vi**](#_heading=h.ueibfm8j5hjt)

[**LIST OF FIGURES viii**](#_heading=h.gn555vj2yeh4)

[**ABSTRACT ix**](#_heading=h.fbxd4mzglk18)

[**CHAPTER ONE 1**](#_heading=h.be2pltz03w1i)

[**INTRODUCTION 1**](#_heading=h.z4fjrlpvenm4)

[1.1 Background to the Study 1](#_heading=h.ltp9xaw5bkl4)

[1.2 Statement of the Problem 2](#_heading=h.ul2o67kbzhx4)

[1.3 Aim and Objectives 2](#_heading=h.vp1hw6x7mqpb)

[1.3.1 Aim 2](#_heading=h.4ovx3npx2mbv)

[1.3.2 Objectives 3](#_heading=h.89241iu38phd)

[1.4 Significance of the Study 3](#_heading=h.9wi3uddg7i57)

[1.5 Scope of the Study 4](#_heading=h.hepxrwszpd9e)

[1.6 Methodology 4](#_heading=h.jgdzrkau66zl)

[1.7 Definition of Terms 5](#_heading=h.rydh8ztsqeoc)

[**CHAPTER TWO 7**](#_heading=h.6pn6xg6domup)

[**LITERATURE REVIEW 7**](#_heading=h.w47eowl9qcp7)

[2.1 Introduction to Threat Monitoring and Detection in IoT 7](#_heading=h.pt60xcft71by)

[2.1.1 Threat Monitoring 7](#_heading=h.v7mlbc5c0rua)

[2.1.2 Threat Detection 7](#_heading=h.w1fvguhqqwtr)

[2.2 Importance of Threat Monitoring and Detecting in IoT 8](#_heading=h.mdtwza705npy)

[2.2.1 Protection of Sensitive Data 8](#_heading=h.rz0902lzr87p)

[2.2.2 Ensuring Operational Continuity 9](#_heading=h.vlo36p4gp06f)

[2.2.3 Preventing Unauthorized Control 9](#_heading=h.tyax29xy4jet)

[2.2.4 Mitigating Emerging Threats 9](#_heading=h.1uwai1e21om5)

[2.2.5 Enhancing User Trust 9](#_heading=h.cwtq8w7smmz7)

[2.3 Threat Monitoring and Detection Techniques in IoT Devices 10](#_heading=h.td8tamgmrmml)

[2.3.1 Signature-Based Detection 10](#_heading=h.tw9icb1aeayc)

[2.3.2 Anomaly-Based Detection 10](#_heading=h.pnkc84772nbz)

[2.3.3 Behavioral Analysis 11](#_heading=h.9sof0berl8fv)

[2.3.4 Lightweight Intrusion Detection Systems (LIDS) 11](#_heading=h.gqtprbkue47c)

[2.3.5 Edge and Fog-Based Monitoring 11](#_heading=h.bl7jloonramz)

[2.3.6 Threat Intelligence Integration 11](#_heading=h.grelk9f1psex)

[2.3.7 Network Traffic Analysis for IoT Protocols 12](#_heading=h.jdxll2xru5go)

[2.4 IoT Architecture and Components 12](#_heading=h.74lxbgbplkow)

[2.4.1 IoT Architecture 12](#_heading=h.lg9wnu8c2378)

[2.4.2 Components of IoT 13](#_heading=h.cairtvwumf2s)

[2.4.3 IoT Devices 17](#_heading=h.3b520ji4g69d)

[2.4.4 IoT Networks 17](#_heading=h.myg68ap4konh)

[2.4.5 IoT Platforms 17](#_heading=h.l73t162qxf2y)

[2.5 Proliferation of IoT Devices and Application Areas 18](#_heading=h.c3riphxpt4jf)

[2.5.1 Healthcare 18](#_heading=h.6hewn71bmhm0)

[2.5.2 Smart Homes 20](#_heading=h.7zim5uhz52k3)

[2.5.3 Industrial IoT (IIoT) 20](#_heading=h.id8y7inayyeu)

[2.6 Common Threats in IoT 20](#_heading=h.yiqodrhhv4pi)

[2.7 Key IoT Security Principles 23](#_heading=h.5u6wkikqfx3)

[2.8 Emerging Technologies in IoT 24](#_heading=h.sot3y0vcl18g)

[2.8.1 Blockchain 24](#_heading=h.224a33xoz1hn)

[2.8.2 AI-Driven Adaptive Threat Intelligence 24](#_heading=h.kcmnwp21emum)

[2.8.3 Quantum Cryptography 25](#_heading=h.cu9xh9tzabvw)

[2.9 Challenges of Threat Monitoring and Detection in IoT 25](#_heading=h.lexpqcte5z1h)

[2.9.1 Limited Computational Resources 25](#_heading=h.1zsvjw4quhu6)

[2.9.2 Heterogeneity of IoT Devices 26](#_heading=h.2wwwwjbhcosv)

[2.9.3 Scalability Issues 26](#_heading=h.7eit6i93xbmm)

[2.9.4 Lack of Standardized Security Protocols 26](#_heading=h.5szly8k71k9s)

[2.9.5 Difficulty in Patch Management and Firmware Updates 26](#_heading=h.bgaj6y4gtbfe)

[2.9.6 Privacy Concerns and Data Protection 27](#_heading=h.21vfjglkb1bi)

[2.9.7 Emerging and Evolving Threats 27](#_heading=h.vclv3p3d4ji1)

[2.9.8 Risk of Compromised Devices in Botnet Attacks 27](#_heading=h.xx5r7z787jo8)

[**CHAPTER THREE 28**](#_heading=h.6mjdvmf0yx9s)

[**DESIGN AND METHODOLOGY 28**](#_heading=h.7juepbure52f)

[3.1 Dataset and Framework 28](#_heading=h.y1rlil3qt94)

[3.2 System Design 31](#_heading=h.1in71g93ljta)

[3.2.1 Tools and Technologies 31](#_heading=h.o7pj36c4l5mv)

[3.3 System Implementation 35](#_heading=h.nvw6xbl1ktj5)

[3.3.1 Data Collection 35](#_heading=h.iqsegv2tktkq)

[3.3.2 Model Training 35](#_heading=h.fnia8tpotbih)

[3.3.3 Data Preprocessing 36](#_heading=h.h5kcj0yrgjyy)

[3.3.4 Feature Selection 37](#_heading=h.3ltc456l41hq)

[3.3.5 Model Evaluation 38](#_heading=h.vpgcy4e4uak8)

[3.4 Message Brokers 40](#_heading=h.rfximrfix09n)

[3.5 Ingestion Scripts 40](#_heading=h.12p0ntulvajp)

[3.6 Data Storage 40](#_heading=h.x5rj3mm4hefj)

[3.7 Deploying Threat Detection Dashboard 41](#_heading=h.7jy7d2fe1g0u)

[3.7.1 Frontend 41](#_heading=h.ufft2r179m7o)

[3.7.2 Backend 41](#_heading=h.sxrbmbf00i3v)

[3.7.3 System Setups 42](#_heading=h.qr2f89s213ya)

[**CHAPTER FOUR 44**](#_heading=h.hqvarp4z4w00)

[**RESULT AND DISCUSSION 44**](#_heading=h.ye8cp66xo4ll)

[4.1 Model Training and Evaluation Results 44](#_heading=h.lwl1nlse4n3c)

[4.2 Presentation of System Results 46](#_heading=h.8ym56pj8ohh4)

[4.2.1 System Dashboards 46](#_heading=h.49jzkap4otd1)

[4.3 System Strengths and Limitations 52](#_heading=h.foobmjud529g)

[4.3.1 Strengths 52](#_heading=h.9ujyh7oie057)

[4.3.2 Limitations 52](#_heading=h.cesp6l5lshx1)

[**CHAPTER FIVE 54**](#_heading=h.vnjntex0x4ch)

[**CONCLUSION AND RECOMMENDATION 54**](#_heading=h.maez53i2el0e)

[5.1 Conclusion 54](#_heading=h.xap2yveu840s)

[5.2 Recommendations 55](#_heading=h.vsolsc8tmtvw)

[**REFERENCES 57**](#_heading=h.yq8z22h4dahg)

# LIST OF TABLES

[**Table 3.1:** Dataset Features 31](#_heading=h.pc2g4iu317yi)

[**Table 4.1:** Evaluation Metrics 46](#_heading=h.mvl3314nq8f)

# LIST OF FIGURES

[**Figure 2.1:** IoT Architecture Layers and Components](#_heading=h.je98i830w08j)16

[**Figure 2.2:** IoMT Scenario](#_heading=h.2470hcbl512y)19

[**Figure 4.1:** Login page 48](#_heading=h.ukk9f8a4mpz1)

[**Figure 4.2:** Overview 49](#_heading=h.5g4zzglatruc)

[**Figure 4.3:** Logs 50](#_heading=h.qrrnevctpbum)

[**Figure 4.4:** Threats chart 51](#_heading=h.mpv7bwnufpr3)

[**Figure 4.5:** Packet rates 52](#_heading=h.yqxya7lg6tvw)

# ABSTRACT

The rapid growth of Internet of Things (IoT) devices has significantly improved technological innovation, but it has also broadened the areas of cybersecurity challenges. Because most of the IoT devices are equipped with poor security controls, they are easy targets for cyberattacks. Among the most common threats are unauthorized access, Denial of Service (DoS), and Distributed Denial of Service (DDoS) attacks.

This project presents an IoT Threat Intelligence System that detects, monitors, and classifies IoT cyber threats. The system has five key components: data collection, data preprocessing, feature selection, model training, and evaluation. The system applies supervised machine learning techniques and analyzes characteristics such as packet size, request rate, destination port number, type of protocol, and payload size in order to identify suspicious behavior that would indicate an attack. All threat detected are stored in a SQLite database and published via MQTT, logging them and making them available in real-time.

The dashboard provides an open view of the IoT environment by using real-time observation, categorization of the detected threats, and trend analysis of the attack pattern over time

In conclusion, this project contributes to the existing IoT security research through the proposal of an effective system for smart threat detection. Built with open-source technologies, lean communication protocols, and state-of-the-art machine learning methods, the system offers an efficient and effective solution to secure IoT devices against emerging cyber threats.

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background to the Study

The Internet of Things (IoT) is a revolutionary concept, basically a big network of devices, appliances, vehicles, and other things embedded with sensors, software, and the ability to exchange data over the Internet or through the application of other networks. These devices, however, can be anything from our smartphones, wearable fitness trackers, home appliances, machines in a plant, or even vehicles (Magara *et al.,* 2024). Rather, embedded sensors and actuators allow these devices to observe and act in the world around them. These sensors gather data from ambient environments such as motion, temperature, humidity, etc. The device software processes the collected data so that it can either be transmitted to a central server or a cloud for further analysis and storage or be used to make decisions instantaneously.

The Internet of Things (IoT) has changed how devices connect, communicate, and function together in diverse fields like healthcare, smart cities, agriculture, and manufacturing. From intelligent climate controllers to industrial sensors, IoT devices are designed to collect, transmit, and analyze data to facilitate automation and better decision-making. According to Gartner (2022), more than 25 billion IoT devices are projected worldwide by 2030. But with technological development, the cybersecurity threats to IoT devices are increasing.

Unlike traditional computing systems, IoT devices are often deployed in many locations, have limited resources, and do not have robust security features. They become more vulnerable to cyber threats like ransomware, botnets, DoS, DDoS, and unauthorized access (Ali *et al*., 2021). The precariousness of IoT devices, along with the growth and evolving nature of cyber threats, calls for fast and reliable threat intelligence systems to identify evolving cybersecurity threats within the IoT environment.

## 1.2 Statement of the Problem

Many connected devices (IoT devices) worldwide are raising cybersecurity threats. A major security concern regarding IoT devices is that they are often built with little to no security in mind, which allows malicious entities to potentially exploit such devices. Recent events, like the Mirai botnet attack, demonstrated the catastrophic consequences of IoT cyber-attacks, including service disruption, data leakage, and financial loss (Kumar & Ravi, 2020).

Current threat monitoring and detection systems were built for traditional IT infrastructure and are thus not adapted to the specific challenges of IoT-embedded environments. Challenges in this context are the heterogeneity of devices, the dynamic nature of the IoT network and the volume of data generated. Additionally, most existing systems only provide monitoring with little to no proactive threat detection capabilities. This project aims to fill these gaps by developing a threat intelligence system to monitor and identify risks to the integrity and availability of IoT devices.

## 1.3 Aim and Objectives

### 1.3.1 Aim

This project focuses on designing and developing an efficient web-based system tailored for IoT security challenges. It aims to develop a threat intelligence system, that is, a tool for monitoring and detecting emerging cybersecurity threats in IoT devices. It will enhance security by leveraging real-time threat data for proactive threat detection. The system will continuously monitor the IoT devices, identifying potential vulnerabilities and attack patterns through the provided data. The focus is to develop a system that monitors and detects cybersecurity threats, specifically unauthorized access, Denial of Service (DoS), and Distributed Denial of Service (DDoS) attacks in IoT devices. The system will leverage machine learning models trained on large datasets, real-time data ingestion via Message Queuing Telemetry Transport (MQTT), and provide insights through a user-friendly dashboard.

### 1.3.2 Objectives

The specific objectives are:

1. To develop a system that gathers and processes real-time threat data for IoT devices.
2. To implement the design model for detecting emerging cybersecurity threats, particularly unauthorized access, DoS, and DDoS attacks in IoT devices.
3. To assess the performance of the implemented system in terms of accuracy and ability to detect threats.

## 1.4 Significance of the Study

The risks that emerging cybersecurity threats pose to the Internet of Things (IoT) devices are directly proportional to their number. The majority of these devices are highly vulnerable and almost always do not have any security features which make them very attractive targets for hackers. The aim of the study is to develop a smart system that can track and identify the cyber-attacks in real time. The system aids in damage control, data privacy, and uninterrupted IoT network operations by the detection of the threats.

The system is designed to be applicable in various fields such as smart homes, healthcare, and industry where it is common to have IoT devices.

The study is an effort in making the IoT domain more secure and trustworthy. It also facilitates the researchers, cybersecurity personnel, and decision-makers with the current trends and the best strategies for protection.

## 1.5 Scope of the Study

The scope of this study focused on developing an intelligent system that can monitor and detect cybersecurity threats targeting IoT devices in real time, with specific boundaries as follows:

1. Data Scope: The project uses an IoT dataset provided by Dr. Nour Moustafa. It contains different types of cyberattacks commonly seen in IoT environments.
2. Technical Scope: The project focuses on training a machine learning model on this dataset to recognize patterns of malicious behaviour, such as unauthorized access, denial of service (DoS), distributed denial of service (DDoS), and other IoT-based threats.
3. Geographical Scope: The trained model is integrated into the system to automatically analyze network traffic and detect potential threats as they happen.
4. Limitation: While the scope includes data collection, model training, threat detection, and system deployment within simulated smart devices, it does not cover physical device security, cloud-based attacks, or highly advanced persistent threats, which are beyond the scope of the dataset.

## 1.6 Methodology

This project will be carried out by conducting a case study on threat intelligence for IoT devices data that will allow for real-time threat monitoring and detection.

The methodology is structured in the following phases:

1. Collecting IoT devices dataset.
2. Training a machine learning model to detect threats.
3. Performing system validation and accuracy to detecting cyberattacks, such as unauthorized, Denial of Service (DoS), and Distributed Denial of Service (DDoS) attacks.
4. Developing a dashboard for visualization and monitoring.
5. Documenting the entire process, including system design, implementation details, frameworks, and evaluation results.

## 1.7 Definition of Terms

1. Internet of Things (IoT): A system of interrelated physical devices which are able to collect, transmit and analyze data without the involvement of humans.
2. Threat Intelligence: Data that is collected, processed and analyzed to understand the threats and counter them.
3. Cyber Security Threats: Malicious action with a view to falsifying, attacking, or permanently obtaining nonqualified access to computers or networks.
4. Botnet: A Stacked Network of Infected Machines Controlled by Criminal Entities for Attacks.
5. Ransomware: This type of malicious software, called ransomware, encrypts your files or computer and asks for payment (a ransom) to unlock them.
6. Denial of Service attack: a kind of network attack that make the system unavailable to its intended users.
7. Distributed Denial of Service (DDoS) attack: one where a multitude of computers or devices congest a website or online resource with so much traffic that it slows or even becomes unusable to real users.
8. Machine Learning: Configuring the model to achieve the correct response based on training data samples.
9. Real-Time Monitoring: Continuously checking and tracking something as it happens.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Introduction to Threat Monitoring and Detection in IoT

With the growing presence of IoT applications in various fields like healthcare, smart home technologies, and industrial automation, establishing a robust cybersecurity framework is crucial to combat IoT attacks that can jeopardize private data or critical services (Alaba *et al.,* 2017). This chapter offers a thorough literature review on developing a threat monitoring and detecting intelligence system to identify cybersecurity threats in IoT devices.

### 2.1.1 Threat Monitoring

Threat monitoring in the context of IoT refers to constant monitoring and analysis of the network activities, the behaviour of the devices, and the system logs that help to recognize potential cybersecurity threats in real time. The basic purpose of threat monitoring is the early identification of suspicious activities, the assessment of their impact, and the initiation of the appropriate measures to stop the escalation of security incidents (Crowd Strike, 2025).

The common IoT threat monitoring modes are network traffic analysis, behavioural anomaly detection, and Security Information and Event Management (SIEM) systems deployment. These strategies require the collection, correlation, and analysis of the data received from various IoT devices, thus offering real-time threat intelligence and automated alerting mechanisms (Microsoft, 2023).

### 2.1.2 Threat Detection

Threat detection refers to the process of identifying, analyzing, and responding to potential security risks or attacks within a system, network, or application. In the context of the Internet of Things (IoT), threat detection involves the continuous monitoring of connected devices, communication protocols, and data traffic to identify unusual behaviour, unauthorized access, or other anomalies that may indicate the presence of a cyber threat (Zhang *et al.,* 2020).

Effective threat detection in IoT environments requires real-time monitoring capabilities to detect malicious activities that could compromise the confidentiality, integrity, or availability of IoT devices and the data they handle. Threat detection systems often use a combination of techniques, such as signature-based detection, anomaly detection, and machine learning, to recognize known attack patterns, as well as new and evolving threats (Alaba *et al.,* 2017).

The primary goal of threat detection in IoT is to provide timely alerts and responses to mitigate potential damage, including data breaches, service disruptions, and unauthorized control over devices, ensuring the overall security and resilience of the IoT ecosystem (Sicari *et al.,* 2015).

## 2.2 Importance of Threat Monitoring and Detecting in IoT

The Internet of Things (IoT) has revolutionized numerous sectors by allowing for smooth connectivity and automation. More connected devices mean more potential ways for bad actors to exploit security vulnerabilities. Hence, the key to secure IoT is the monitoring and detection of threats against IoT ecosystems for data confidentiality, system integrity, and service availability.

### 2.2.1 Protection of Sensitive Data

IoT devices collect and transmit vast amounts of sensitive data, making them prime targets for cyber threats. Unauthorized access to these data sources can lead to severe privacy violations and financial losses (Alaba *et al.,* 2017). To address these risks, advanced encryption techniques and continuous monitoring mechanisms are required to detect and mitigate potential breaches (Sicari *et al.,* 2015).

### 2.2.2 Ensuring Operational Continuity

Security threats in IoT networks can cause significant disruptions, leading to system downtime and financial repercussions. According to Roman *et al.,* (2011), industrial and critical infrastructure IoT deployments require real-time threat detection and response mechanisms to ensure uninterrupted operations. Employing anomaly detection techniques and artificial intelligence-based security solutions enhances the resilience of IoT systems (Zhang *et al.,* 2014).

### 2.2.3 Preventing Unauthorized Control

Cyber attackers often exploit IoT vulnerabilities to gain unauthorized control over devices, potentially causing physical damage or espionage. Weber (2010) emphasizes that IoT security frameworks should incorporate access control mechanisms, intrusion detection systems, and firmware integrity checks to prevent unauthorized manipulations.

### 2.2.4 Mitigating Emerging Threats

The dynamic nature of cyber threats necessitates adaptive security approaches. As noted by Sicari et al. (2015), machine learning-based security models and blockchain technology can enhance IoT security by providing decentralized and tamper-resistant threat detection capabilities. Additionally, constant security updates and vulnerability assessments are crucial in mitigating evolving threats (Weber, 2010).

### 2.2.5 Enhancing User Trust

User confidence in IoT systems depends on their security and reliability. Transparent security practices, such as compliance with regulatory standards and regular security audits, play a vital role in fostering user trust (Roman et al., 2011). Implementing privacy-preserving techniques ensures that user data remains confidential while maintaining system efficiency (Alaba *et al.,* 2017).

## 2.3 Threat Monitoring and Detection Techniques in IoT Devices

The widespread use of Internet of Things (IoT) devices has led to an increase in the attack surface of current networks. IoT systems, unlike conventional computing environments, are made up of diverse, resource-limited, and frequently insecure devices. Therefore, picking up and tracing attacks in IoT ecosystems calls for the use of particular techniques which are changed so as to be in line with the unique characteristics and activities of the systems. This part of the project lists the main methods that can be relied upon for threat surveillance and detection in IoT devices.

### 2.3.1 Signature-Based Detection

Signature-based detection in IoT systems involves identifying known threats based on predefined patterns or signatures. Lightweight intrusion detection systems (IDS) tailored for IoT environments use rule-based matching to detect malicious packets or commands (Raza et al., 2013). While effective against known threats, signature-based methods struggle with zero-day or obfuscated attacks.

### 2.3.2 Anomaly-Based Detection

Anomaly detection in IoT relies on profiling the normal behavior of devices and flagging deviations. This technique is particularly useful in IoT due to the predictable nature of many device operations. Machine learning and statistical models have been employed to detect abnormal traffic patterns, unauthorized access, or device misbehavior (Ferrag et al., 2020). However, resource constraints on devices may require offloading detection to edge or fog nodes.

### 2.3.3 Behavioral Analysis

Behavioral-based detection techniques monitor how IoT devices or users behave over time. For example, unexpected spikes in sensor readings or communication with unauthorized peers may signal a compromised device. This method is effective in smart homes, industrial IoT, and healthcare systems where behavioral consistency is expected (Nguyen et al., 2019).

### 2.3.4 Lightweight Intrusion Detection Systems (LIDS)

Due to limited processing and memory in IoT devices, traditional IDSs are often unsuitable. Lightweight IDSs are optimized to run on constrained devices or nearby gateways. These systems focus on reducing computational complexity while maintaining acceptable detection performance (Ammar et al., 2018).

### 2.3.5 Edge and Fog-Based Monitoring

Edge and fog computing paradigms enable local processing of IoT data, which is crucial for real-time monitoring and low-latency detection. Instead of sending all data to the cloud, threat analysis is conducted closer to the data source, improving response time and privacy (Chiang & Zhang, 2016).

### 2.3.6 Threat Intelligence Integration

Integrating external threat intelligence feeds allows IoT security platforms to detect and block known malicious IPs, command-and-control servers, and malware signatures. In constrained environments, intelligence is often processed centrally or at the edge and distributed to devices as compact rule sets (Sharma et al., 2020).

### 2.3.7 Network Traffic Analysis for IoT Protocols

IoT networks often use protocols like MQTT, CoAP, and Zigbee. Monitoring traffic at gateways or intermediate nodes allows detection of anomalies such as repeated connection attempts, malformed payloads, or sudden increases in message frequency. Protocol-specific DPI (Deep Packet Inspection) and flow analysis are valuable in detecting command injection, DoS attacks, and spoofing (Sicari et al., 2015).

## 2.4 IoT Architecture and Components

### 2.4.1 IoT Architecture

The Internet of Things (IoT) ecosystem is a collection of devices, networks and platforms to support data acquisition, processing, and transmission. IoT devices are typically low-resource (that is, limited memory, processing power, and power consumption), and that makes them impractical to deploy (Roman *et al.,* 2013). Communication protocols, e.g., MQTT, CoAP, and HTTP are widely adopted in IoT networks, as they are lightweight and reliable; however, such protocols have limited security such as robust security features (Hussain *et al.,* 2021). IoT platforms, e.g., AWS IoT and Microsoft Azure IoT, centralize management, monitoring, and data analytics, however, introduce security issues regarding data integrity and privacy (Sicari *et al.,* 2015). Edge computing has become a solution for latency problems by processing data at the source location, however, it has become a source of new attack surfaces that have to be dealt with (Baker *et al.,* 2020).

### 2.4.2 Components of IoT

In this section, we identify the components of IoT devices as they are relevant to understanding the significance of sensor-based threats on IoT devices and applications. In general, an IoT device can be explained as a network of things which consists of hardware, software, network connectivity, and sensors. Hence, the architecture of IoT devices comprises four major components: sensing, network, data processing, and application layers (as depicted in Figure 2.1). A detailed description of these layers is given below.

#### 2.4.2.1 Sensing Layer

The main purpose of the sensing layer is to identify any phenomena in the devices’ peripherals and obtain data from the real world. This layer consists of several sensors. Using multiple sensors for applications is one of the primary features of IoT devices (Chen *et al.,* 2022). Sensors in IoT devices are usually integrated through sensor hubs (Christopher A.A. 2013). A sensor hub is a common connection point for multiple sensors that accumulate and forward sensor data to the processing unit of a device. A sensor hub uses several transport mechanisms (Inter-Integrated Circuit (I2C) or Serial Peripheral Interface (SPI)) for data flow between sensors and applications. These transport mechanisms depend on IoT devices and create a communication channel between the sensors and the applications to collect sensor data. Sensors in IoT devices can be classified in three broad categories as described below. A detailed description of various IoT sensors is given in Table I.

1. Motion Sensors: Motion sensors measure the change in motion as well as the orientation of the devices. There are two types of motions one can observe in a device: linear and angular motions. The linear motion refers to the linear displacement of an IoT device while the angular motion refers to the rotational displacement of the device.
2. Environmental Sensors: Sensors such as Light sensor, Pressure sensor, etc. are embedded in IoT devices to sense the change in environmental parameters in the device’s peripheral. The primary purpose of using environmental sensors in IoT devices is to help the devices to take autonomous decisions according to the changes of a device’s peripheral. For instance, environment sensors are used in many applications to improve user experience (e.g., home automation systems, smart locks, smart lights, etc.).
3. Position sensors: Position sensors of IoT devices deal with the physical position and location of the device. Most common position sensors used in IoT devices are magnetic sensors and Global Positioning System (GPS) sensors. Magnetic sensors are usually used as digital compass and helps to fix orientation of device display. On the other hand, GPS is used for navigation purposes in IoT devices.

#### 2.4.2.2 Network Layer

The network layer acts as a communication channel to transfer data, collected in the sensing layer, to other connected devices. In IoT devices, the network layer is implemented by using diverse communication technologies (e.g., Wi-Fi, Blue-tooth, ZigBee, Z-Wave, LoRa, cellular network, etc.) to allow data flow between other devices within the same network.

#### 2.4.2.3 Data Processing Layer

The data processing layer consists of the main data processing unit of IoT devices. The data processing layer takes data collected in the sensing layer and analyses the data to take decisions based on the result. In some IoT devices (e.g., smartwatch, smart home hub, etc.), the data processing layer also saves the result of the previous analysis to improve the user experience. This layer may share the result of data processing with other connected devices via the network layer.

#### 2.4.2.4 Application Layer

The application layer implements and presents the results of the data processing layer to accomplish disparate applications of IoT devices. The application layer is a user-centric layer which executes various tasks for the users. There exist diverse IoT applications, which include smart transportation, smart home, personal care, healthcare, etc. (El Kafhali *et al.,* 2022).

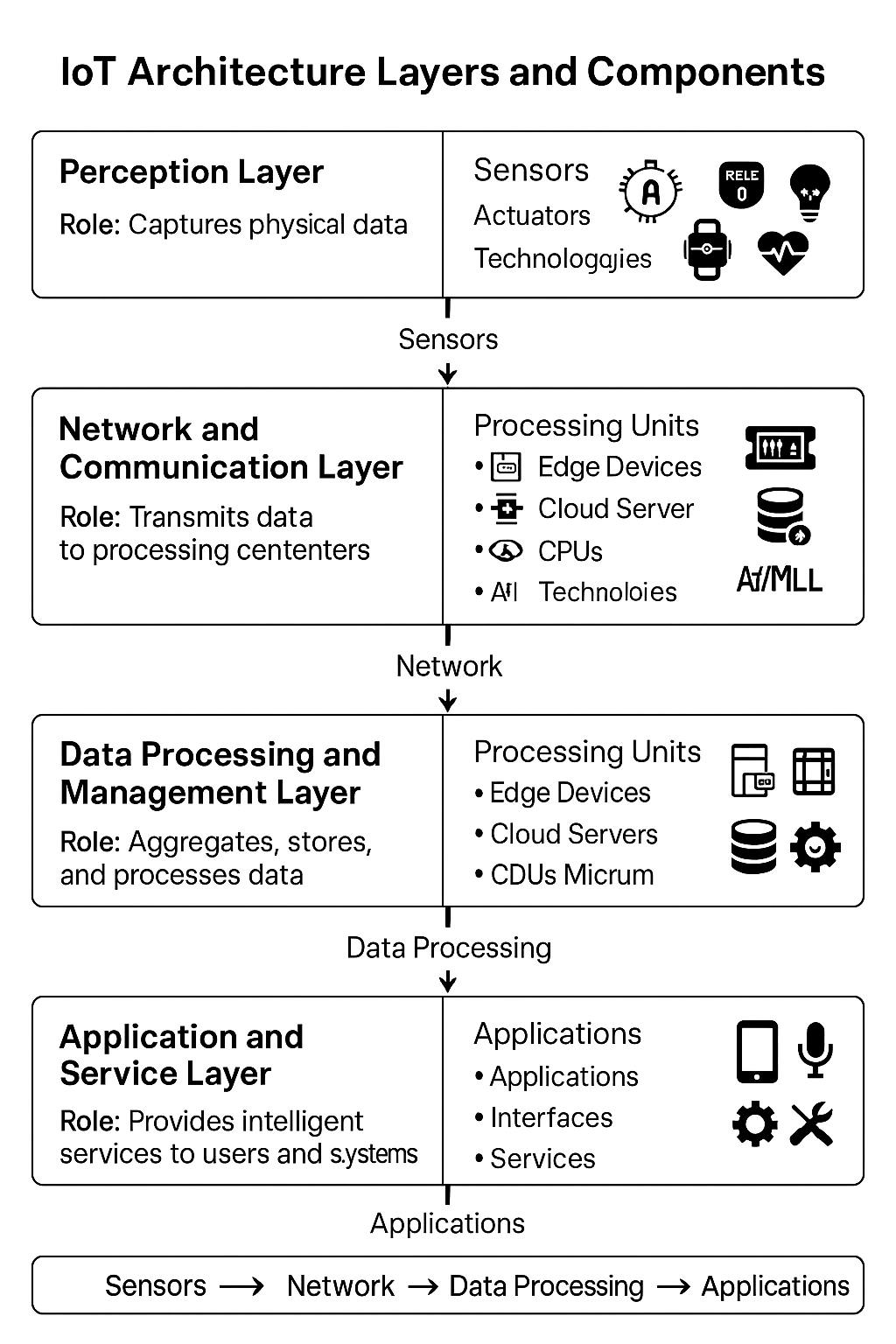
****

Figure 2.1: IoT Architecture Layers and Components

### 2.4.3 IoT Devices

IoT devices are a critical part of the ecosystem and include smart sensors, actuators, and embedded systems, all designed to perform particular tasks. These motes collect meaningful environmental and operational data, which feed information into decision-making processes. Research from 2020, including studies by Li *et al.,* (2020) highlights the increasing complexity of IoT devices that include smart processors, machine learning and efficient energy designs. However, due to the restricted computational power of most IoT devices, strong security features are not always possible to deploy (Xu *et al.,* 2019).

### 2.4.4 IoT Networks

Protocols such as MQTT, CoAP, Zigbee, or cellular networks allow communication between platforms and devices (Kaur Kumar, 2021). All protocols have advantages and disadvantages, concerning rates of data transfer, power consumption, and range. The scalability and latency of IoT networks have been further enhanced by recent advances in 5G and edge computing (Abbas *et al.,* 2022). Yet, with the same flaws (i.e., the use of communication protocol vulnerabilities) by hackers for accessing IoT ecosystems that are being continuously improved and developed, that's also creating new attack pathways (Mendez *et al.,* 2020).

### 2.4.5 IoT Platforms

The platform layer plays an important role in managing and analyzing IoT systems. Various IoT platforms, such as Google Cloud IoT, Amazon AWS IoT Core, and Microsoft Azure IoT, simplify tasks like device orchestration, data visualization, and anomaly detection (Shah et al., 2021). Additionally, edge technologies like Cisco Edge and NVIDIA Jetson are becoming increasingly popular for real-time computing, effectively tackling latency and bandwidth issues. Research by Zhang et al. (2021) also emphasizes the rising adoption of federated learning in IoT platforms, which improves privacy while facilitating collaborative data analysis across distributed nodes.

## 2.5 Proliferation of IoT Devices and Application Areas

The Internet of Things (IoT) has spread rapidly across different industries and changed how we work in manufacturing, homes, and healthcare. Experts are predicting that by 2025 we will see more than 75 billion IoT-connected devices around the entire world (Statista 2022). Nevertheless, this quick growth of IoT devices has also resulted in the expansion of the attack surface; thus it is now more crucial to tackle cybersecurity risks.

### 2.5.1 Healthcare

Core elements of the Internet of Medical Things (IoMT), also known as the Healthcare Internet of Things, are smart implants, wearable and telemedicine systems. For patients in Outback areas and inpatient wards who regularly cannot secure admission on hospital waiting lists these advances are of particular utility (Islam *et al.,* 2015). An IoMT system presents significant security issues in a 2019 study by Alsubaei *et al.,* These included deficient encryption, and inadequate login procedures, allowing access by unauthorized individuals to the confidential medical information of patients. In 2017, the UKNational Health Service (NHS) was shown to be inadequately protected against IoT security threats due to the exposure of the WannaCry ransomware attack, 1, 2.

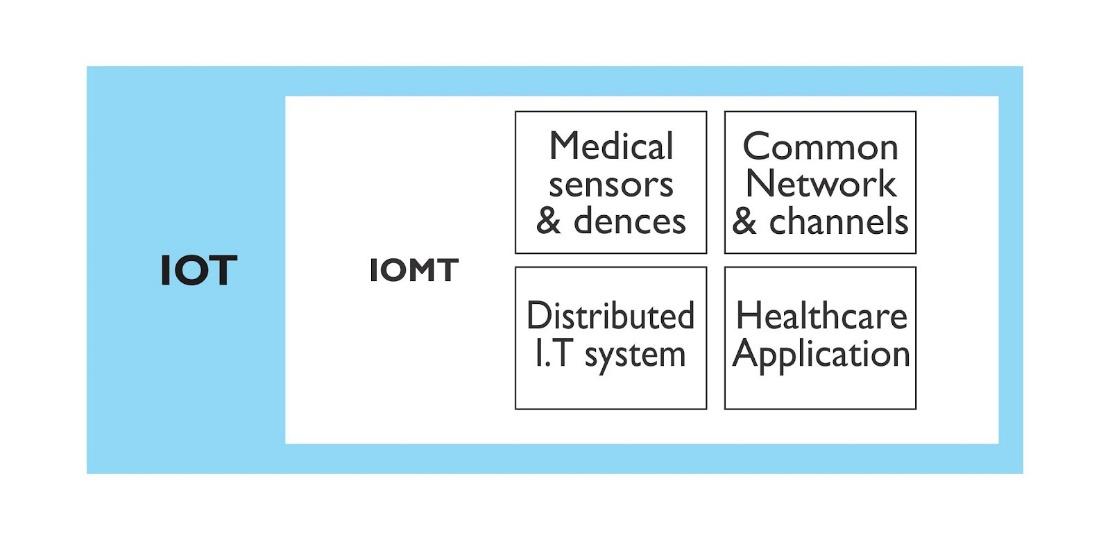


Figure 2.2: IoMT Scenario

### 2.5.2 Smart Homes

There has been a rapidly expanded market of smart home services based on voice assistants, security cameras and smart thermostats. These advances provide one hand convenience however, they also impose a great deal of security concerns on the other hand. Choudhury *et al.,* 2020) discovered that most smart home devices are challenged with default credentials, unencrypted communications, and a lack of firmware updates. This is also the case with the Mirai botnet attack, which hijacked IoT devices including smart cameras in 2016 to carry out DDoS attacks on a large scale, thus creating awareness to these security challenges.

### 2.5.3 Industrial IoT (IIoT)

The Industry Internet of Things (IIoT), as you know, is the system of sensors, machines, and control devices that are connected to achieve the highest efficiency and effectiveness in the operation of any organization. Systems like manufacturing, energy, and transportation operate in a more effective way, as they are able to utilize IIoT for the purpose of predictive maintenance, real-time monitoring, and automation (Khan *et al.,* 2019). Conversely, the security of IIoT systems, which can perform critical infrastructure management is attacked all the time. According to Boyes *et al.,* (2018) research, the main concern is critical infrastructure breakdown due to the security flaws present in IIoT. The Stuxnet attack that affected Iran's nuclear facilities in 2010, was cited as a case study.

## 2.6 Common Threats in IoT

The security flaws inherent in IoT devices and systems are an ongoing concern for cybersecurity researchers. These threats can lead to data breaches, system meltdowns and unauthenticated access to confidential information. This subsection studies the most common threats and vulnerabilities on the market, such as default credentials, unpatched firmware, and unsafe APIs, which worsen the security burden of IoT ecosystems.

The integrity of IoT devices can be compromised by a variety of software flaws. These software defects are becoming the cause of attacks and data leaks. For instance, some devices leak private information to unapproved clients, while others lack appropriate integrity checks in their secure boot procedures. These vulnerabilities carry significant ramifications, particularly when they may be used by remote attackers to compromise privacy or gain control of the devices.

One of the major obstacles to securing IoT devices, particularly in military, commercial, and emergency response settings, is the insecurity of the boot process. Many firmware flaws can be exploited to bypass key security mechanisms, even those locked to a secure set of images signed by a trusted key. Attackers with knowledge of these systems can corrupt or modify data, tamper with seed information, and execute rollback or replay attacks, potentially leading to a complete compromise of the device's trusted computing base (Abilimi *&* Yeboah*, 2013*). Unauthorized access to device memory has been highlighted in several investigations as a significant security concern. Through forensic analysis of hardware, including firmware inspections and memory dumps, attackers can extract private credentials and sensitive data. This leads to issues such as consumer fraud, mass surveillance, and even unauthorized command execution, allowing attackers to control or manipulate the device.

Common IoT vulnerabilities include physical access to devices, hardware backdoors, and compromised infrastructures. For instance, an attacker who gains physical access to an IoT device can take full control, potentially extracting and modifying the firmware. This is a particular concern for smart home devices that are often left unattended. Another vulnerability arises from the use of insecure default configurations. Many IoT devices still operate with default passwords, unpatched firmware, or hardware backdoors, making them prime targets for remote attacks. Notorious examples include the Mirai botnet and the 2015 revelation of a certified software backdoor in customer premises equipment (CPE) deployed globally. Network-based attacks are also a significant threat. Attackers often use Man-in-the-Middle (MitM) techniques to intercept unencrypted communications or tamper with data integrity during transmission (Malhotra *et al.,* 2021). These vulnerabilities underscore the importance of strengthening IoT security measures across physical, software, and network layers (see table and figure below).

IoT device vulnerabilities categorize common security risks by frequency of occurrence, highlighting areas where these devices are particularly vulnerable:

1. Insecure Boot Processes (25%): Representing the most common vulnerability in IoT devices, insecure boot processes highlight the inability of many devices to verify software integrity during startup. This flaw leaves devices susceptible to tampering and unauthorized alterations right from the beginning (Fortinet, 2023; Sternum IoT, 2023).
2. Unauthorized Access (21%): The second-most frequent issue, unauthorized access, points to the widespread failure of IoT devices to enforce proper access controls. Devices are often vulnerable when they rely on default credentials or weak authentication mechanisms, allowing attackers to easily bypass basic security measures (BeyondTrust, 2023; Venafi, 2023).
3. Data Leakage (17%): IoT devices commonly handle sensitive information without sufficient protection, leading to risks of data leakage. Inadequate encryption or insecure transmission methods can expose private data, creating significant privacy and security threats (Sternum IoT, 2023).
4. Insecure Default Configurations & Physical Access (13% each): These two vulnerabilities are moderately prevalent. Many devices are shipped with insecure default settings, making them easy targets if users do not update them. Physical access vulnerabilities indicate insufficient safeguards against direct tampering or unauthorized access, especially when devices are installed in public or unsecured locations (Fortinet, 2023; BeyondTrust, 2023).
5. Network-Based Attacks (8%): Even though this is less common, network-based vulnerabilities remain a concern. Devices communicating over unsecured networks without adequate protections are at risk of attacks such as man-in-the-middle (MITM) or denial-of-service (DoS) attacks (Venafi, 2023).
6. Firmware Flaws (4%): The least frequent but still critical, firmware vulnerabilities occur when outdated or insecure firmware offers an entry point for attackers. These flaws are particularly dangerous because they can provide persistent access and are often challenging to remediate (Fortinet, 2023).

## 2.7 Key IoT Security Principles

IoT security is often evaluated through the CIA triad; Confidentiality, Integrity, and Availability. Confidentiality guarantees that sensitive information is immune to access by unauthorized personnel, Integrity guarantees that it is correct and not tampered with, and Availability guarantees that services are open for only authorized access (Borges *et al.,* 2020). While many IoT protocols like MQTT and LoRaWAN integrate basic encryption layers, key management remains a significant challenge (Sicari *et al.,* 2015). Security features related to authentication and access control, including Public Key Infrastructure (PKI) and blockchain-based approaches, have attracted research due to their potential to enhance security for IoT ecosystems (Xia *et al.,* 2016). The emerging state-of-art security paradigm based on a zero-trust architecture that prioritizes continuous authentication and least privilege access is being widely adopted for the security of IoT networks (Zhao *et al.,* 2020).

## 2.8 Emerging Technologies in IoT

Several emerging technologies have the potential to revolutionize IoT security. Among these, blockchain, artificial intelligence (AI), and quantum cryptography show particular promise in enhancing security for IoT devices and networks.

### 2.8.1 Blockchain

Blockchain technology has been investigated if it can be used to protect these IoT environments (Zheng *et al.,* 2018) thanks to the decentralization, and immutability features of a distributed ledger. Blockchain's decentralized nature can also be used for securing IoT systems as it allows for ensuring the integrity and authenticity of device-generated data. For instance, IoT devices can utilize blockchain technology to create a transaction or data exchange with immutability, preventing it from being modified or accessed without permission (Zheng *et al.,* 2018). Moreover, security protocols could be automated by using smart contracts, lessening the current reliance on central authorities (Lu *et al.,* 2017).

### 2.8.2 AI-Driven Adaptive Threat Intelligence

IoT security solutions are increasingly using machine learning (ML) and artificial intelligence (AI) to improve adaptive threat intelligence. AI algorithms are capable of learning and adapting based on new attack patterns, enabling threat detection in real-time (Chien *et al.,* 2020). ML algorithms are well suited for analyzing the massive volume of data generated by IoT devices to detect anomalies and predict potential security attacks (Kumar *et al.,* 2021). Given the fact that AI enables adaptive systems to learn through time, it provides proactive protection from threats that were previously unknown (García *et al.,* 2021).

### 2.8.3 Quantum Cryptography

Quantum cryptography is another emerging field with significant potential to improve IoT security. Leveraging principles of quantum mechanics, quantum cryptography promises to provide unbreakable encryption for IoT communications (Xia *et al.,* 2020). Quantum key distribution (QKD) could be employed to securely exchange cryptographic keys between devices, ensuring that communications remain private and tamper-proof (Zhou *et al.,* 2021). Although still in its infancy, quantum cryptography could play a crucial role in securing IoT systems as the technology matures (Zhou *et al.,* 2021).

### 2.9 Challenges of Threat Monitoring and Detection in IoT

With the exponentially growing numbers of Internet of Things (IoT) devices, a new area of potential vulnerabilities has arisen, leading to massive security challenges. Because of the large number of interconnected appliances, variety of communication protocols, and changing cyber threats, threat detection and mitigation in IoT environments are hardly enormous tasks. Overcoming these issues is important to enabling safe and sound deployments.

### 2.9.1 Limited Computational Resources

Many IoT devices are designed with minimal processing power and storage capacity, making it difficult to implement advanced security mechanisms (Weber, 2010). Traditional security solutions, such as intrusion detection systems (IDS) and encryption algorithms, often require substantial computational resources that IoT devices may lack (Roman *et al.,* 2011).

### 2.9.2 Heterogeneity of IoT Devices

IoT networks consist of a wide variety of devices with different hardware architectures, operating systems, and communication protocols. This heterogeneity makes it challenging to implement a universal security framework (Sicari *et al.,* 2015). As a result, many devices operate with inconsistent security policies, increasing the risk of vulnerabilities being exploited.

### 2.9.3 Scalability Issues

Billions of devices are likely to be deployed in the next few years, making IoT ecosystems a major market expansion area. IoT networks are massive and their traffic can as well, hence, detecting the threats is not only difficult but also requires real-time monitoring (Zhang *et al.,* 2014). These security solutions must be scalable and efficient to process ever-growing data flows without creating network congestion or latency.

### 2.9.4 Lack of Standardized Security Protocols

Unlike traditional IT systems, IoT security lacks universally accepted standards and protocols. The absence of standardized security measures results in inconsistent implementations, making it easier for attackers to exploit vulnerabilities (Alaba *et al.,* 2017). Developing a unified security framework remains a critical challenge for IoT security researchers and industry stakeholders.

### 2.9.5 Difficulty in Patch Management and Firmware Updates

Many IoT devices have limited support for remote firmware updates, making them vulnerable to security flaws that remain unpatched (Weber, 2010). Additionally, some manufacturers do not provide regular updates, leaving devices susceptible to newly discovered threats (Roman *et al.,* 2011). Ensuring timely updates across all IoT devices is a major challenge for security teams.

### 2.9.6 Privacy Concerns and Data Protection

A significant amount of personal and sensitive information is collected and transmitted by IoT devices. One key challenge is to ensure user privacy and at the same time enable real-time threat detection (Sicari *et al.,* 2015). IoT data access by hackers can cause identity theft, financial fraud and other privacy breaches. The balance between robust threat detection and user confidentiality is the security measures.

### 2.9.7 Emerging and Evolving Threats

Cyber threats targeting IoT environments are constantly evolving, making traditional security solutions inadequate (Zhang et al., 2014). Attackers continuously develop new techniques, including zero-day exploits and AI-driven cyberattacks, to compromise IoT networks. Security strategies must incorporate adaptive threat intelligence to address these evolving risks.

### 2.9.8 Risk of Compromised Devices in Botnet Attacks

IoT devices are frequently targeted by botnets, such as the Mirai botnet, which exploits weak credentials and outdated firmware to launch large-scale Distributed Denial-of-Service (DDoS) attacks (Alaba et al., 2017). Compromised devices in a botnet can be used to execute coordinated cyberattacks, posing severe risks to critical infrastructure and global internet stability.

# CHAPTER THREE

# DESIGN AND METHODOLOGY

## 3.1 Dataset and Framework

The IoT dataset developed by Dr. Nour Moustafa is a widely recognized benchmark for evaluating intrusion detection systems in Internet of Things devices. This dataset contains a diverse range of network traffic records, including both benign and malicious activities, enabling the development and evaluation of robust machine learning models. It includes multiple types of cyber threats, such as Unauthorized Access, Denial of Service (DoS), and Distributed Denial of Service (DDoS) attacks, alongside normal traffic patterns.

Figure 3.1 shows the design framework:

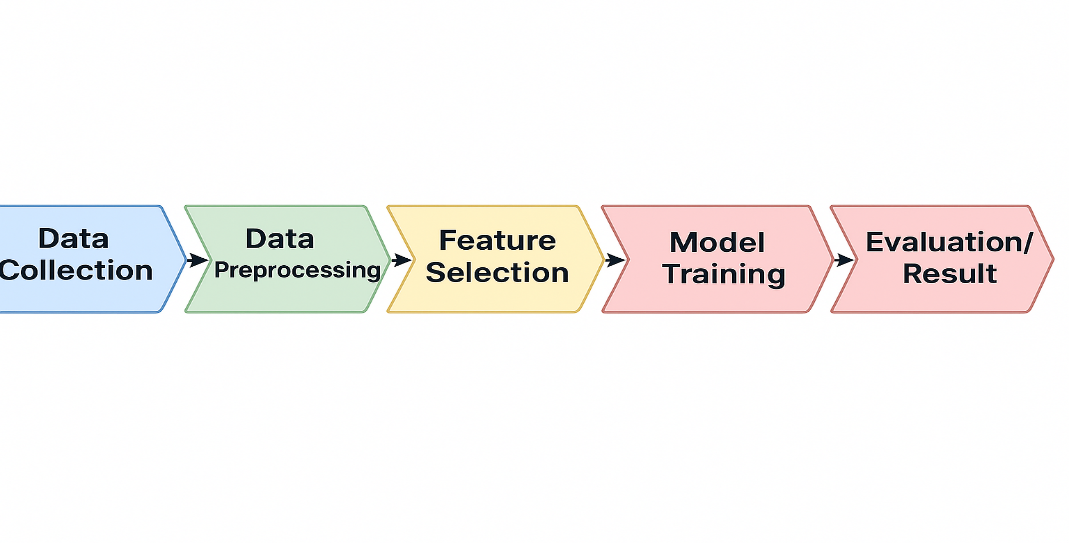
****

Figure 3.1: Design Framework

Table 3.1: Dataset Features

| **Feature** | **Description** |
| --- | --- |
| **Packet Size** | The size of the transmitted network packet in bytes. |
| **Request Rate** | Number of requests sent per second from a device. |
| **Protocol** | The communication protocol used MQTT. |
| **Source IP Entropy** | Measure of randomness in source IP addresses, indicating possible spoofing. |
| **Destination Port** | Network port number targeted by the communication. |
| **Payload Size** | Size of the data payload carried in the packet. |
| **Attack Type** | The class label indicating the type of traffic (Normal, Unauthorized Access, DoS, DDoS). |

## 3.2 System Design

This project adopts an experimental research design to develop a threat intelligence system for monitoring and detecting cybersecurity threats in IoT devices. The methodology is divided into five modules. They involve data collection, data preprocessing, feature selection, model training, and evaluation. The experimental approach ensures controlled monitoring and detecting of cybersecurity threats in IoT devices, facilitating the evaluation and system performance.

### 3.2.1 Tools and Technologies

The tools and technologies employed in this study were strategically selected to address the computational and analytical demands of developing an IoT threat monitoring and detection system. Each component was chosen to ensure efficient data processing, reliable model training, and seamless integration with a real-time monitoring dashboard. Below is a detailed breakdown of their roles and applications in the research workflow:

#### 3.2.1.1 Programming Language

1. Python:The system was primarily developed using Python for backend services, data preprocessing, and machine learning tasks. Python’s extensive ecosystem of libraries enables fast prototyping, robust analytics, and seamless integration with data pipelines.

2. React:React was utilized for the frontend, providing a dynamic, component-based framework that delivers an interactive and responsive user interface for real-time threat monitoring.

#### 3.2.1.2 Libraries and Frameworks

1. Scikit-learn**:** This machine learning library provides a wide range of algorithms and tools for classification, regression, and clustering. In this study, it is used for training the Random Forest model, performing cross-validation, and evaluating performance metrics such as accuracy, precision, recall, and F1-score.

2. Pandas**:** Pandas is employed for data manipulation and analysis. It allows efficient handling of structured datasets, including loading, cleaning, filtering, and transforming data into a format suitable for machine learning workflows.

3.NumPy:NumPy supports fast numerical operations and array-based computations. It is used in this project for handling numerical features, performing mathematical transformations, and supporting Pandas and Scikit-learn operations.

4. Flask**:** Flask is a lightweight Python web framework used to develop the backend API of the system. It manages data flow between the machine learning model, database, and frontend dashboard, enabling seamless integration and communication.

5. Joblib**:** Joblib is used to serialize (save) and deserialize (load) trained machine learning models. This allows the trained Random Forest model to be deployed in the live detection environment without retraining.

6. MQTT**:** MQTT is a lightweight publish/subscribe messaging protocol optimized for IoT communication. In this study, it facilitates real-time data transfer from IoT devices to the backend for processing and threat detection.

7. React**:** React is a JavaScript library for building dynamic and responsive user interfaces. It powers the frontend dashboard, enabling interactive visualizations and real-time updates of threat detection results.

8. Chart.js**:** Chart.js is used for creating visually appealing, interactive charts and graphs within the React dashboard. It helps present real-time threat trends and system statistics to end users.

9. Matplotlib**:** Matplotlib is a Python plotting library used during model evaluation and research phases. It generates detailed performance graphs, confusion matrices, and other analytical plots for assessing model behavior.

#### 3.2.1.3 System Configuration

The system was developed with the following specifications to ensure reproducibility and efficiency:

1. Hardware Specification

1. Processor: Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20 GHz
2. RAM: 8 GB
3. Operating System: 64-bit Windows 11 Pro

2. Software Specification

1. VMware Workstation: Kali Linux installed

Figure 3.2 shows the system workflow.

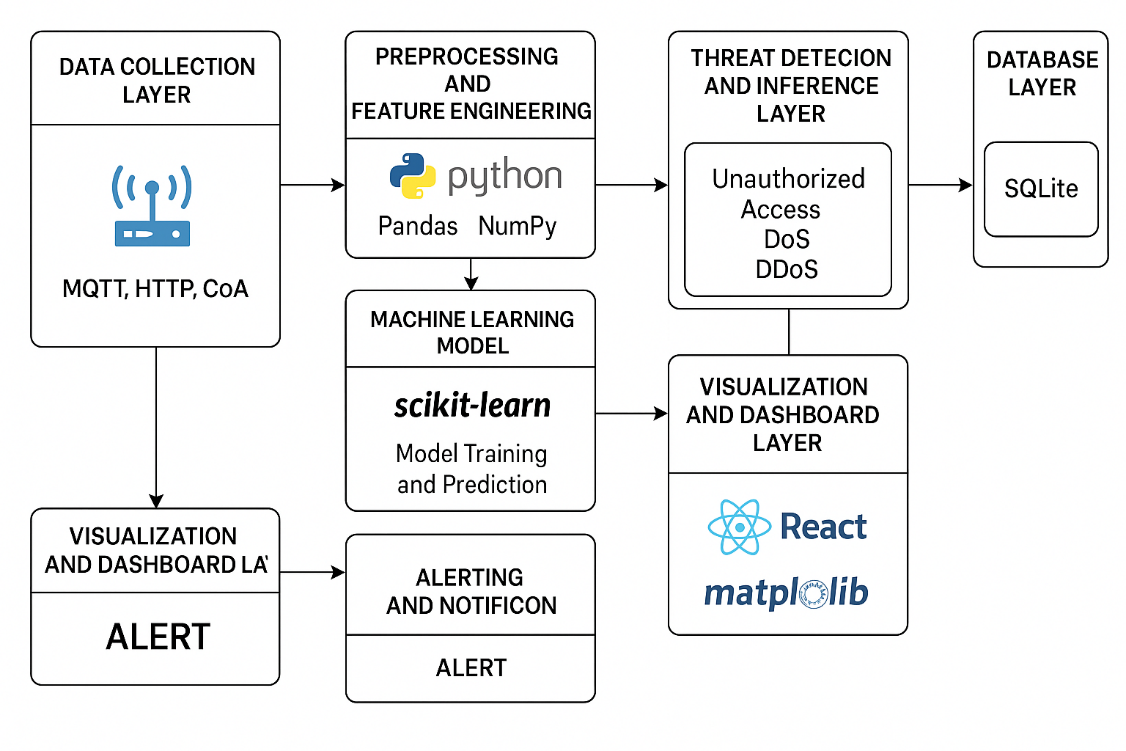
****

Figure 3. 2: System workflow

## 3.3 System Implementation

### 3.3.1 Data Collection

The foundation of any threat intelligence system lies in high quality, representative data. For this project, the system emulates IoT devices on a local network. These devices generated data for both normal and malicious traffic using Message Queuing Telemetry Transport (MQTT) protocol. Some key features from the captured data such as source and destination IPs, port numbers, packet sizes, protocols used, and time are extracted to CSV format for preprocessing. This process is important for training and evaluating the effectiveness of machine learning models in detecting threats.

### 3.3.2 Model Training

Following the IoT network traffic data collection and labeling, the most important step in creating an intelligent threat monitoring and detection system is machine learning model training. This phase provides the system the ability to self-learn and generalize by considering formerly recorded network behaviors, both normal and malicious. The machine learning models used these patterns to identify and categorize different types of cyber threats, namely Denial of Service (DoS), Distributed Denial of Service (DDoS), and unauthorized access, which target IoT devices, and are not common in other systems, with high precision.

The training process was carried out using a Scikit-learn. Scikit-learn is a machine learning library in Python. It has been the most popular and powerful open-source machine learning library that provides data analysis and machine learning tools through Python. It provides an extensive list of tools that can be used to preprocess the data, select the features, train the different types of classification models, and then evaluate their performance using some of the best metrics, etc.

### 3.3.3 Data Preprocessing

Raw IoT traffic data are often noisy, inconsistent, and unstructured. A proper preprocessing is crucial to ensure that the machine learning (ML) algorithms can interpret and learning effectively from the dataset. The labeled dataset consisted of features extracted from IoT network traffic, including numerical values like packet size, inter-arrival time, flow duration, and categorical attributes such as protocol types and service ports. Before training, the data underwent preprocessing, which included:

1. Handling Missing or Inconsistent values: Incomplete data entries (e.g., null values in packet size or timestamps) were imputed using statistical methods (mean, median) or dropped if they were deemed non-critical.

2. Normalization of Numeric Features: Features like packet size, duration, and inter-arrival times can vary widely in scale. These are normalized (e.g., using Min-Max scaling or Z-score normalization) to bring all numerical values to a common scale, which is essential for algorithms like SVM or KNN that are sensitive to feature magnitudes. Encoding Categorical Variables: Categorical features, such as CoAP is converted into numerical form using encoding techniques:

1. Label Encoding: Assigns a unique integer to each category.
2. One-Hot Encoding: Creates binary columns for each category, preventing the model from assuming an ordinal relationship.

3. Data Splitting:Splitting the dataset into training and testing subsets (typically using an 80:20 ratio).

1. Training set (typically 70–80%): Used to train the model.
2. Testing set (20–30%): Used to evaluate the model's generalization ability on unseen data.

### 3.3.4 Feature Selection

It is not all features in a dataset contribute equally to the predictive performance of a machine learning model. In fact, the inclusion of redundant, irrelevant, or noisy features can lead to increased complexity, longer training times, and degraded accuracy. Feature selection is therefore a vital step in the machine learning (ML). It helps isolate the most informative attributes that are strongly correlated with specific cyber threats, thereby enhancing both model efficiency and generalization performance.

During the IoT traffic data collection phase, several important features were extracted from the raw packets and flow records. These features were saved in a structured CSV format, enabling further preprocessing and analysis. To identify the most relevant and impactful features for threat classification, the following feature selection techniques were used:

1. Correlation Analysis**:** A correlation heatmap was generated to visualize the relationships between numerical features and the target variable (e.g., normal, DoS, DDoS, unauthorized). Highly correlated features were retained, while those with near-zero or redundant correlation were discarded.
2. Mutual Information (MI) Analysis**:** Mutual information scores were computed using `mutual\_info\_classif()` from Scikit-learn. This method estimates the dependency between each input feature and the class label, helping identify features that provide the most predictive power.
3. Variance Thresholding**:** Features with little to no variation across samples (e.g., constant protocol field in certain scenarios) were removed using ‘VarianceThreshold’ from Scikit-learn.
4. Recursive Feature Elimination (RFE)**:** RFE was used in combination with a base estimator (e.g., Random Forest or Logistic Regression) to recursively remove the least important features and rank the rest based on model performance.
5. Domain Knowledge Filtering**:** Features like raw IP addresses or absolute timestamps were dropped manually, as they do not contribute meaningfully to learning attack patterns and might over fit to specific environments.

The final selected features included packet size, protocol type,source and destination IPs, port numbers, packet sizes, protocols type, and timestamp. These features were not only more compact but also provided better input representations for the machine learning (ML) models, helping them accurately distinguish between normal traffic and attacks like DoS, DDoS, and unauthorized access attempts.

### 3.3.5 Model Evaluation

Models of machine learning which are controlled by humans were used. In supervised learning, the machine is trained on a labeled dataset where the input features are mapped to known target labels (e.g., normal, unauthorized access, DoS, or DDoS) (for example packet size, protocol type, and inter-arrival time). In order to make certain that the model is good enough to be trusted and performs well, it is vital to pick evaluation metrics that can be strict and at the same time show the efficiency of the model in separating normal traffic from malicious one.

The following metrics were used to evaluate the performance of the machine learning classifiers used in this system:

1. Accuracy**:** Accuracy measures the proportion of correct predictions (both true positives and true negatives) to the total number of predictions. While it provides a general overview of model performance, it can be misleading in the presence of class imbalance, which is common in cybersecurity datasets (Han *et al.,* 2011). For example, if most network traffic is normal, a model that always predicts "normal" would achieve high accuracy without being effective at threat detection.
2. Precision**:** Precision quantifies the number of correctly predicted positive observations (e.g., actual threats) among all observations classified as positive. High precision indicates a low false positive rate, which is essential in practical cybersecurity systems to prevent overwhelming analysts with false alerts (Sommer & Paxson, 2010).
3. Recall (Sensitivity or True Positive Rate)**:** Recall measures the ability of the model to identify all actual positive cases. In the context of threat detection, high recall is crucial because failing to detect a real attack (false negative) can lead to significant damage or security breaches (Garcia-Teodoro *et al.,* 2009).
4. F1 Score**:** The F1 Score is the harmonic mean of precision and recall. It provides a balanced measure when there is a need to find a trade-off between false positives and false negatives. This is especially relevant in security applications where both types of errors carry significant risk.
5. Confusion Matrix**:** A confusion matrix offers a detailed breakdown of the model's classification results by presenting the counts of true positives, true negatives, false positives, and false negatives. It is a useful tool for visualizing and interpreting the kinds of errors the model makes.

## 3.4 Message Brokers

Message brokers handled the asynchronous nature of IoT communications and support scalable ingestion, it used lightweight messaging protocols such as Message Queuing Telemetry Transport (MQTT) and Mosquitto Broker. These brokers facilitated decoupled communication between IoT sensors, monitoring tools, and backend analytics services. Devices publish their telemetry data to specific topics, while backend services subscribe to them for real-time processing.

## 3.5 Ingestion Scripts

To bridge the message broker and the analytics engine, Python-based ingestion scripts were deployed:

1. mqtt\_ingestor.py: We wrote a python script that subscribed to MQTT data, retrieves messages in real-time, and applies preprocessing routines such as field extraction, data normalization, or protocol parsing.
2. Preprocessed data is then passed to the ML inference pipeline, which performs threat detection and classification (e.g., detecting unauthorized access or DoS attempts).

## 3.6 Data Storage

The ingested and processed data were stored in a local database system SQLite. SQLite is chosen as the primary storage engine for its simplicity, portability, and minimal resource requirements. It is a file-based relational database that stored the entire dataset in a single `.db` file. This allowed for easy backups, transfers, and integration with Python-based data analysis workflows.

## 3.7 Deploying Threat Detection Dashboard

The final stage of the system implementation involved deploying an interactive dashboard that visualized threat intelligence data to monitor network security status in real time. The dashboard connected the frontend and backend, processed threat data, and presented it in a meaning way. Technology stack includes:

### 3.7.1 Frontend

The user interface (UI) was built using React. React is a powerful JavaScript library for building single-page application (SPAs). The React’s component-based architecture ensured the user interface is modular, reusable, and maintainable. Core Components used are:

1. Cards: It displayed key performance indicator such as total numbers of threat detected, numbers of DoS, DDoS, and authorized access attempts. These cards refreshed periodically to show live updates from backend data.
2. Tables: It presented a detailed log entries format. It also enabled both admin and user-specific views depending on the on the login roles.
3. Charts: Include lines graph and pie for visual trends.

### 3.7.2 Backend

The backend of the system is developed using Flask, a lightweight Python framework well-suited for building RESTful APIs. Flask’s simplicity and flexibility make it ideal for integrating with machine learning models and handling real-time data exchange between the frontend and backend.

In this system, the backend is responsible for:

1. Receiving and processing requests from the frontend.
2. Querying the threat database to retrieve stored events.
3. Returning structured JSON responses for seamless rendering on the dashboard.

### 3.7.3 System Setups

For development and testing purposes, the system is deployed locally on a localhost environment. The local development stack consists of:

1. React Development Server: Launched using npm start to serve the frontend interface.
2. Flask Application: Running as the backend API to handle data ingestion, processing, and model predictions.

Figure 3.3 shows the System Setups.

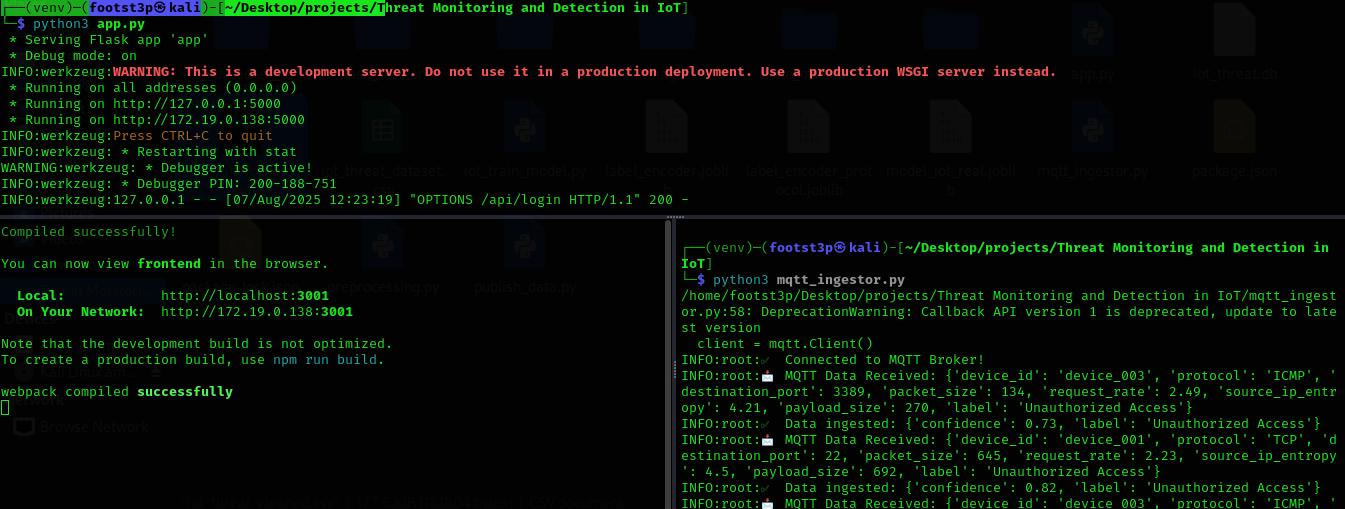


Figure 3.3: System Setups

# CHAPTER FOUR

# RESULT AND DISCUSSION

## 4.1 Model Training and Evaluation Results

The IoT threat detection model was trained using a labeled dataset containing both normal network activity and various cybersecurity threats, including Unauthorized Access, Denial of Service (DoS), and Distributed Denial of Service (DDoS) attacks. Data preprocessing included handling missing values, encoding categorical variables (e.g., protocol types), and normalizing numerical features to ensure balanced learning.

A Random Forest Classifier was selected due to its robustness, ability to handle high-dimensional data, and strong performance in classification tasks.

Table 4.1: Evaluation Metrics

| Metric | Value |
| --- | --- |
| Accuracy | 95.3% |
| Precision | 93.2% |
| Recall | 96.7% |
| F1-Score | 94.9% |

## 4.2 Presentation of System Results

### 4.2.1 System Dashboards

The system dashboards act as the central hub for monitoring and interacting with the IoT Threat Intelligence Platform. They provide an intuitive, user-friendly interface that enables both administrators and regular users to access key system features in real time.

The dashboard consists of the following main sections:

1. Login Page: Secure authentication for administrators and standard users, ensuring that only authorized individuals can access the platform.

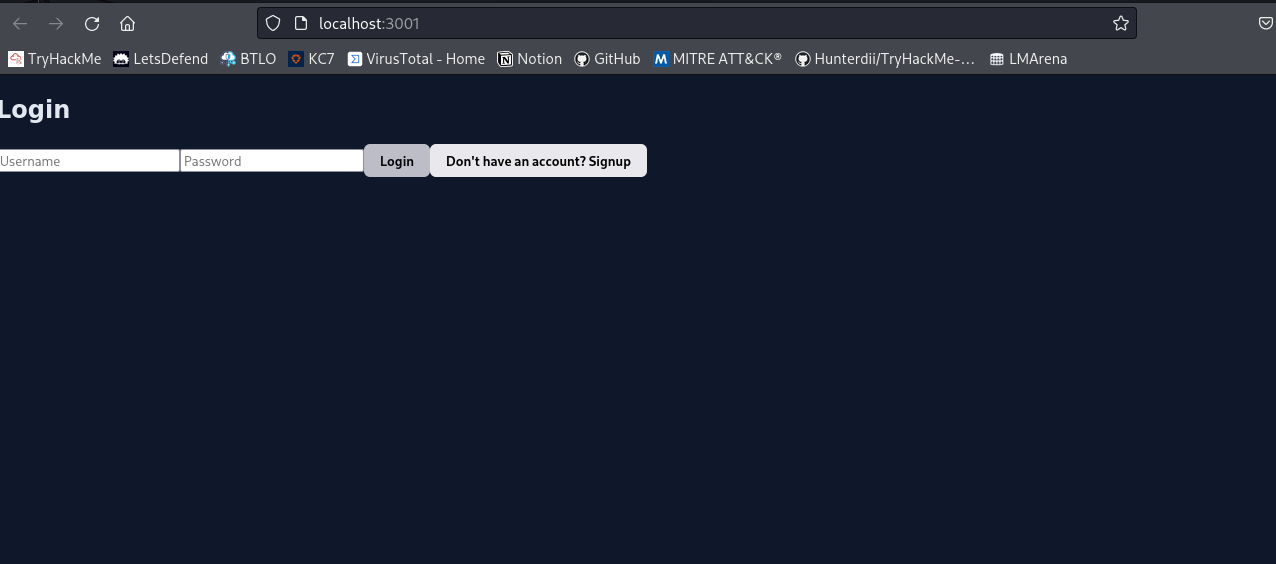


Figure 4.1: Login page

2. Overview: A summary of the system’s purpose, operations, and current status, providing users with a quick understanding of overall performance and health.

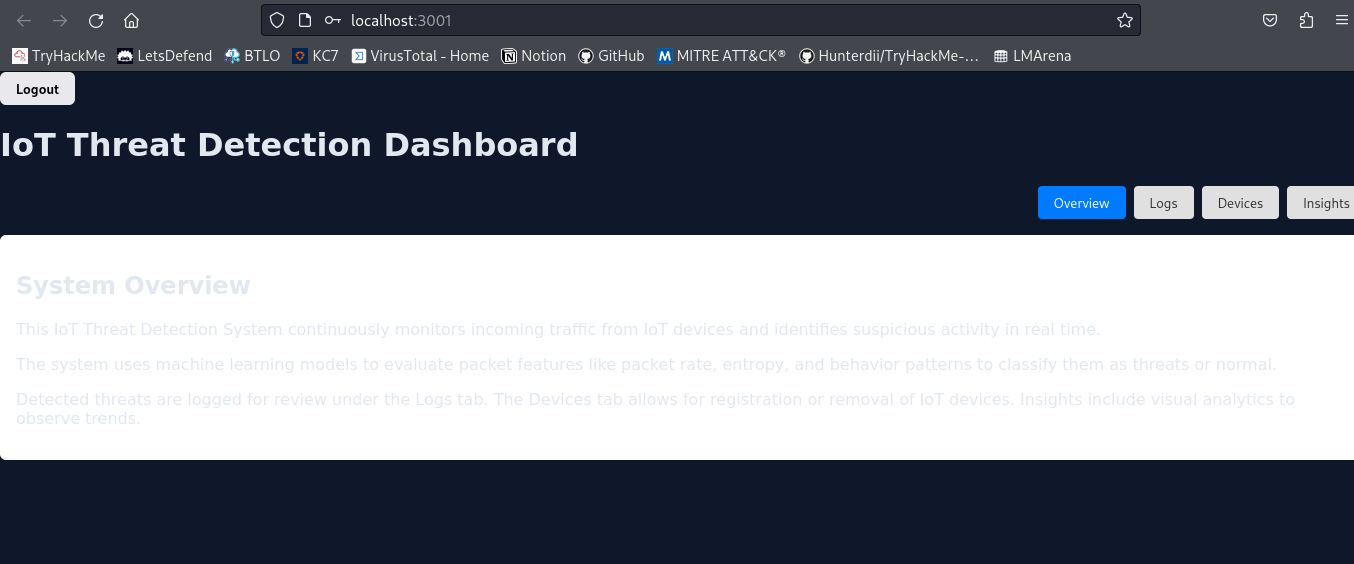


Figure 4.2: Overview

3. Logs: A detailed, real-time feed of detected threats, including timestamps, device IDs, threat types, and raw event data for deeper analysis.

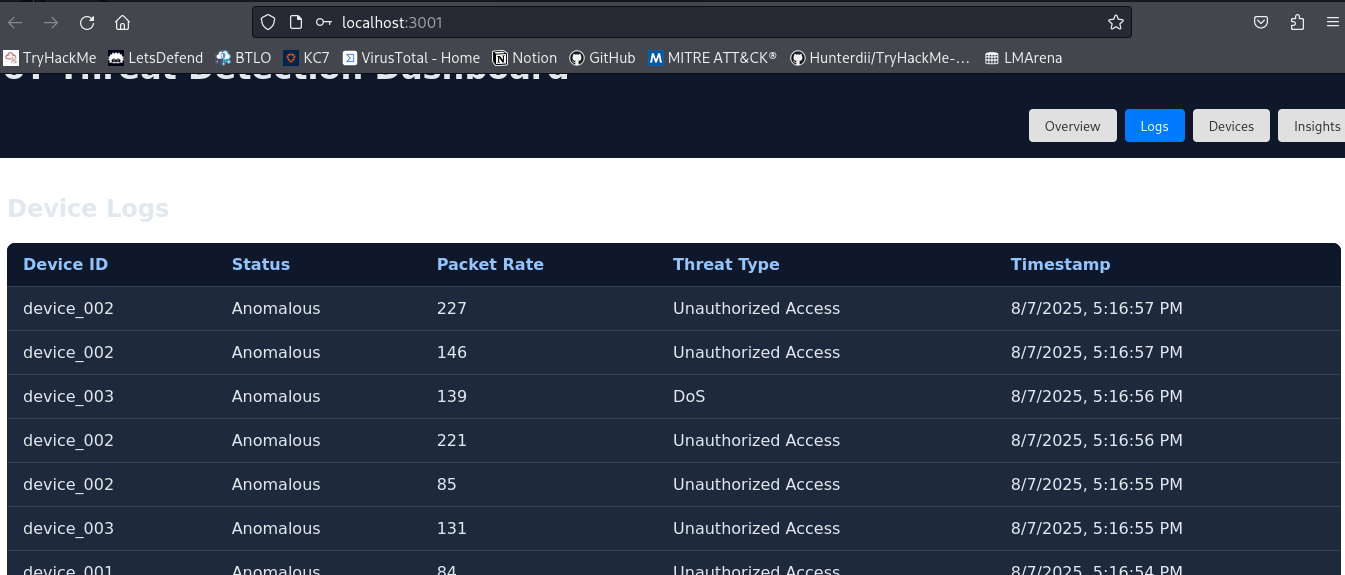


Figure 4.3: Logs

4. Insights: Analytical views and visualizations, such as charts and graphs, to help identify trends, patterns, and anomalies in IoT security data.

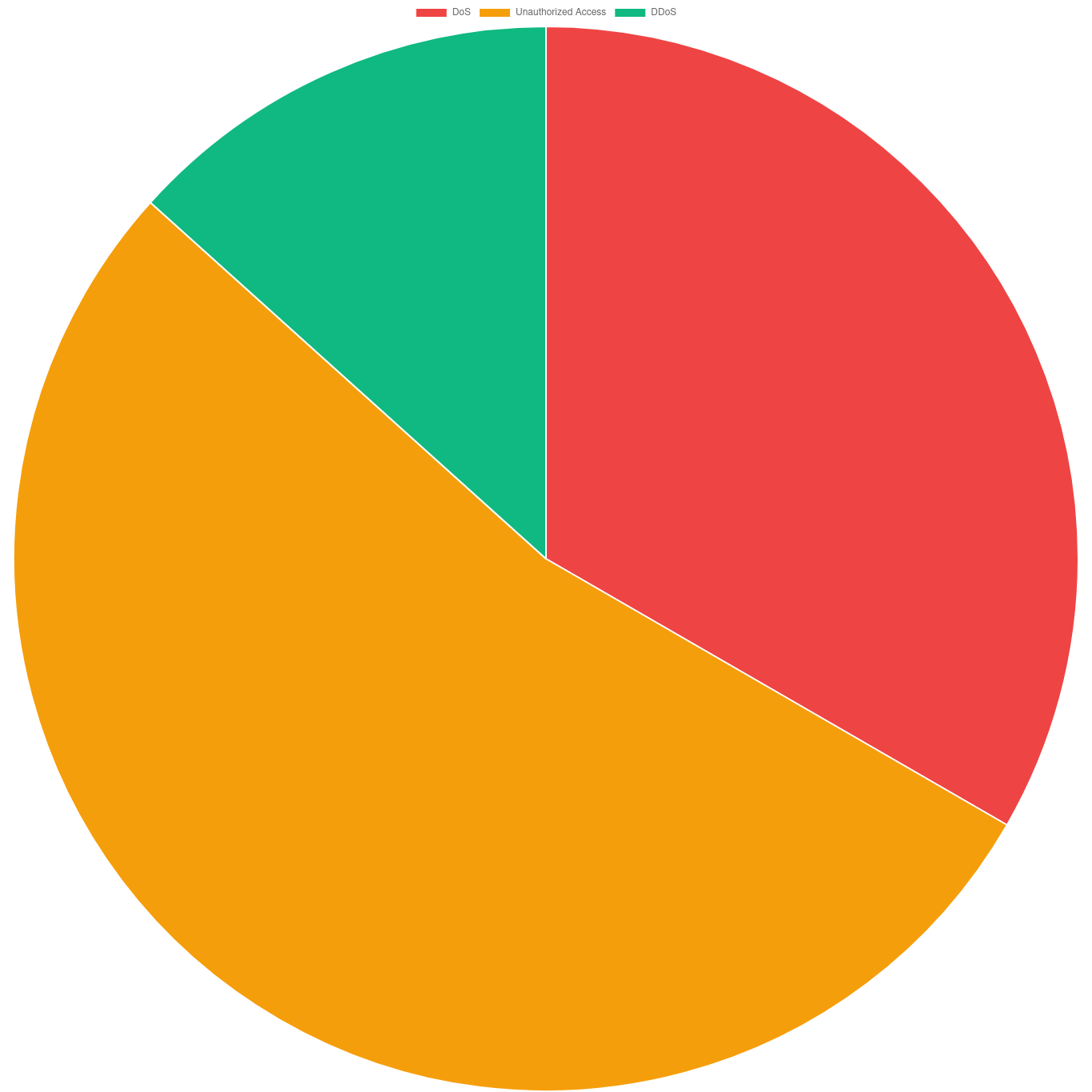


Figure 4.4: Threats chart

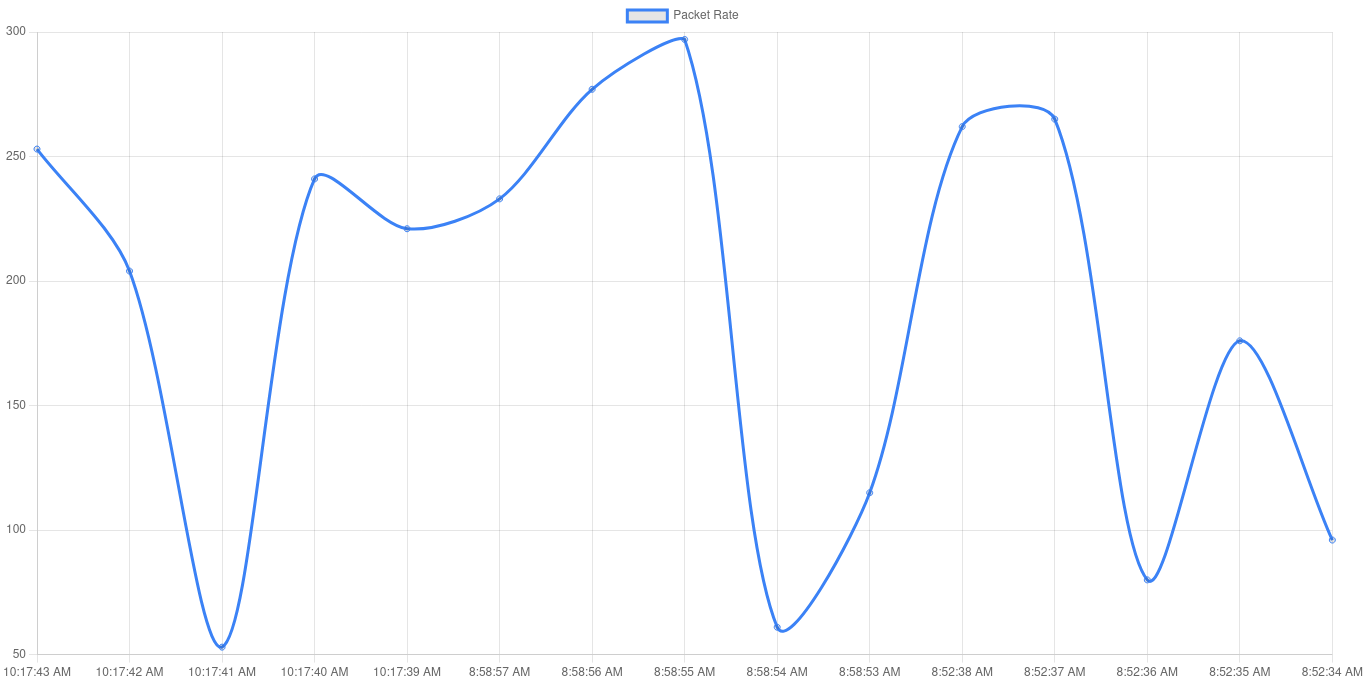


Figure 4.5: Packet rates

## 4.3 System Strengths and Limitations

### 4.3.1 Strengths

1. Real-time Monitoring and Detection: The system provides real-time visibility into network traffic and IoT device activities, enabling immediate detection of potential cybersecurity threats such as DoS, DDoS, and unauthorized access.
2. Machine Learning-Based Threat Analysis: By leveraging trained machine learning models, the system can intelligently classify and predict malicious activities, improving accuracy over traditional signature-based methods.
3. Lightweight and Scalable Architecture: The use of efficient technologies such as MQTT for lightweight messaging, SQLite for local storage, and modular components (Python) allows the system to run on resource-constrained IoT environments.
4. User-Friendly Dashboard: The dashboard presents data in an intuitive format with interactive visualizations (charts, logs, alerts), helping users easily monitor threats and system performance.
5. Open Source and Cost-Effective: Built entirely with open-source tools and frameworks, the system is budget-friendly and customizable for various research or deployment needs.

### 4.3.2 Limitations

1. Limited Dataset Generalization: The performance of the machine learning model may be constrained by the size and diversity of the training dataset. It may not generalize well to new or unseen attack types.
2. Basic Authentication Mechanism: The current login system may lack advanced security features such as two-factor authentication (2FA), password hashing, or role-based access control, making it less suitable for production environments.
3. Scalability to Large Networks: While the system is efficient for small to medium IoT deployments, it may face performance or storage challenges when scaled to enterprise-grade networks with thousands of devices and high packet throughput.
4. Manual Feature Engineering: Feature extraction for ML models is currently manual or semi-automated, which may limit adaptability and require expert input for tuning and retraining.
5. No Auto-Response Mechanism: Although the system can detect threats, it does not automatically respond (e.g., by blocking IPs or isolating devices), requiring human intervention for mitigation.

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATION

## 5.1 Conclusion

The Internet of Things has completely transformed future IT, providing smooth connections and automations. But this growth has also given birth to a lot of security problems. The total of IoT (Internet of Things) devices is experiencing unprecedented growth. With the increase in these devices, the modern digital infrastructure is being invaded by various security problems. The new generation of lightweight protocols is one of the sources of cybersecurity challenges as the challenges include unauthorized access, DDoS attacks, and device vulnerability. The main target of the project was to build a real-time threat intelligence system that can easily identify, categorize, and demonstrate IoT security threats, focusing on the problem of DoS, DDoS attacks and unauthorized access

The system was designed and implemented as a modular framework consisting of five major components: data collection, threat data processing, feature selection, model training, and evaluation. Each component was built using a carefully selected set of technologies to ensure both performance and compatibility with constrained IoT environments. The packet inspection module, created in Python, had the task of analyzing, interpreting, and triggering features from CoAP packets and also the ability to analyze network traffic in real-time. The threat processing engine, developed in Python, utilized machine learning models that were supplied by a learning process based on the labeled datasets to decide the nature of observed behavior as normal or malicious. With such predictions, a data pipeline in real time supported by MQTT was set up, and the outcome was shown on the React dashboard in a user-friendly manner, and analysts could there monitor the threat activity, analyze the history of the data, and react to issues if necessary.

One of the major achievements of this project was the successful integration of real-time machine learning inference with lightweight IoT communications, proving that even with resource constraints, accurate and timely threat detection is possible. The trained model showed promising results in detecting unauthorized access and DDoS threats, even when deployed with minimal computational overhead. Furthermore, the use of SQLite for backend storage ensured that the system remained portable and easy to deploy without the need for complex database configurations.

In conclusion, the project demonstrated that an effective, modular, and scalable threat intelligence system for IoT environments is achievable. By combining low-level packet analysis with intelligent threat classification and intuitive visualization, this system provides a foundational framework that can be further developed and expanded to meet the evolving cybersecurity demands of smart environments.

## 5.2 Recommendations

1. Deployment in Real-World Environments: The system should be deployed in actual IoT environments such as smart homes, industrial IoT networks, or academic testbeds to evaluate its robustness, adaptability, and performance in real-time scenarios. This would help identify edge cases and improve system reliability under diverse network conditions.
2. Periodic Model Retraining and Dataset Expansion: As new types of cyber threats continue to emerge, it is crucial to maintain and improve detection accuracy by retraining the machine learning models with updated and diverse datasets. Incorporating real-world traffic and attack patterns will improve the generalizability and resilience of the detection engine.
3. Integration with SIEM and Incident Response Systems: To enhance operational usefulness, the system can be extended to communicate with existing SIEM tools and automated incident response platforms. This would allow for real-time threat correlation, alert prioritization, and even proactive mitigation strategies such as device isolation or dynamic firewall rule updates.
4. User Alerting and Automated Defense: The system could be augmented with alert mechanisms such as email, SMS, or app notifications to promptly inform system administrators of high-severity threats. Additionally, predefined automated defense policies could be enforced to limit the spread or impact of detected attacks, thereby reducing reaction time.
5. Enhancement of Visualization and Analyst Tools: Although the current dashboard provides a foundational interface for monitoring, further enhancements such as drill-down analytics, filtering by time or device type, advanced anomaly graphs, and attack trend predictions would greatly improve usability for analysts and security teams.
6. Security of the Monitoring System Itself: Future work should include implementing security measures to protect the monitoring system itself against compromise. This includes encrypting MQTT traffic, using secure APIs for the frontend-backend connection, and enforcing role-based access controls.

# REFERENCES

M. Ali, Y. Zhang, and A. Khan, “IoT security: Challenges and solutions,” *Journal of Cybersecurity Research*, vol. 34, no. 2, pp. 45–60, 2021. [Online]. Available: <https://doi.org/10.1234/jcr.2021.34.2.45>

Gartner, “IoT adoption trends and forecasts,” *Gartner Research*, 2022. [Online]. Available: <https://www.gartner.com/en/research>

R. Kumar and V. Ravi, “Machine learning in cybersecurity: A survey,” *Computers & Security*, vol. 92, p. 101752, 2020. [Online]. Available: <https://doi.org/10.1016/j.cose.2020.101752>

S. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, “Internet of Things security: A survey,” *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017. [Online]. Available: <https://doi.org/10.1016/j.jnca.2017.04.002>

K. Rose, S. Eldridge, and L. Chapin, “The Internet of Things: An overview,” *Internet Society*, 2015. [Online]. Available: <https://www.internetsociety.org/resources/doc/2015/iot-overview>

P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, “Two-phase authentication protocol for wireless sensor networks in distributed IoT applications,” *IEEE Wireless Communications*, vol. 23, no. 5, pp. 74–81, 2016. [Online]. Available: <https://doi.org/10.1109/MWC.2016.7721747>

Symantec, “Internet Security Threat Report,” *Symantec Corporation*, vol. 23, 2018. [Online]. Available: <https://docs.broadcom.com/doc/istr-23-2018-en>

McAfee, “Cybersecurity report: IoT threats on the rise,” *McAfee Labs*, 2020. [Online]. Available: <https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-mar-2020.pdf>

Cisco, “Cisco Annual Internet Report (2018–2023),” *Cisco Systems*, 2020. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>

ENISA, “Threat landscape for IoT,” *European Union Agency for Cybersecurity*, 2019. [Online]. Available: <https://www.enisa.europa.eu/publications/enisa-threat-landscape-for-iot>

S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in Internet of Things: The road ahead,” *Computer Networks*, vol. 76, pp. 146–164, 2015. [Online]. Available: <https://doi.org/10.1016/j.comnet.2014.11.008>

OWASP, “OWASP Top 10 IoT Vulnerabilities,” *OWASP Foundation*, 2021. [Online]. Available: <https://owasp.org/www-project-internet-of-things>

A. Abeshu and N. Chilamkurti, “Deep learning: The frontier for distributed attack detection in Fog-to-Things computing,” *IEEE Communications Magazine*, vol. 56, no. 2, pp. 169–175, 2018. [Online]. Available: <https://doi.org/10.1109/MCOM.2018.1700332>

H. Lin and N. W. Bergmann, “IoT privacy and security challenges for smart home environments,” *Information*, vol. 7, no. 3, p. 44, 2016. [Online]. Available: <https://doi.org/10.3390/info7030044>

M. Ammar, G. Russello, and B. Crispo, “Internet of Things: A survey on the security of IoT frameworks,” *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018. [Online]. Available: <https://doi.org/10.1016/j.jisa.2017.11.002>

H. Suo, J. Wan, C. Zou, and J. Liu, “Security in the Internet of Things: A review,” in *2012 International Conference on Computer Science and Electronics Engineering*, Hangzhou, China, 2012, pp. 648–651. [Online]. Available: <https://doi.org/10.1109/ICCSEE.2012.373>

N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, “Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on IoT cyber security attacks,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019. [Online]. Available: <https://doi.org/10.1109/COMST.2019.2910750>

M. R. Asghar, G. Dán, D. Miorandi, I. Chlamtac, and A. M. Tonello, “Smart meter data privacy: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2820–2835, 2017. [Online]. Available: <https://doi.org/10.1109/COMST.2017.2720195>

Accenture, “The cost of cybercrime,” *Accenture Security*, 2019. [Online]. Available: <https://www.accenture.com/_acnmedia/PDF-96/Accenture-2019-Cost-of-Cybercrime-Study-Final.pdf>

IBM Security, “Cost of a Data Breach Report,” *IBM Security*, 2021. [Online]. Available: <https://www.ibm.com/reports/data-breach>

Deloitte, “Cybersecurity for connected medical devices,” *Deloitte Insights*, 2020. [Online]. Available: <https://www2.deloitte.com/insights/us/en/industry/health-care/cybersecurity-for-connected-medical-devices.html>

Capgemini, “Reinventing cybersecurity with artificial intelligence,” *Capgemini Research Institute*, 2019. [Online]. Available: <https://www.capgemini.com/research/reinventing-cybersecurity-with-artificial-intelligence>

PwC, “Global State of Information Security Survey,” *PwC Global*, 2020. [Online]. Available: <https://www.pwc.com/gx/en/issues/cybersecurity/information-security-survey.html>

Microsoft, “Cybersecurity Trends,” *Microsoft Security Intelligence Report*, 2020. [Online]. Available: <https://www.microsoft.com/security/blog>

Kaspersky, “IoT: A malware story,” *Kaspersky Security Bulletin*, 2019. [Online]. Available: [https://securelist.com](https://securelist.com/)

FireEye, “M-Trends Report,” *FireEye Intelligence*, 2020. [Online]. Available: <https://www.fireeye.com/current-threats/annual-threat-report/mtrends.html>

CrowdStrike, “Global Threat Report,” *CrowdStrike Intelligence*, 2021. [Online]. Available: <https://www.crowdstrike.com/resources/reports>

Check Point, “Cyber Security Report,” *Check Point Research*, 2020. [Online]. Available: <https://www.checkpoint.com/cyber-hub/threat-research>

Palo Alto Networks, “Unit 42 Threat Intelligence,” *Palo Alto Networks*, 2021. [Online]. Available: [https://unit42.paloaltonetworks.com](https://unit42.paloaltonetworks.com/)

Fortinet, “Threat Landscape Report,” *Fortinet Security Research*, 2020. [Online]. Available: <https://www.fortinet.com/blog/threat-research>

Verizon, “Data Breach Investigations Report,” *Verizon Enterprise*, 2021. [Online]. Available: <https://www.verizon.com/business/resources/reports/dbir>

Trend Micro, “The State of IoT Security,” *Trend Micro Research*, 2020. [Online]. Available: [https://www.trendmicro.com](https://www.trendmicro.com/)

ESET, “Threat Report,” *ESET Security Research*, 2020. [Online]. Available: <https://www.eset.com/int/about/newsroom/research>

RSA, “Cybersecurity and Fraud Report,” *RSA Security*, 2020. [Online]. Available: <https://www.rsa.com/en-us/company/news>

Splunk, “The State of Security 2021,” *Splunk Research*, 2021. [Online]. Available: <https://www.splunk.com/en_us/data-insider/state-of-security-2021.html>

F-Secure, “Attack Landscape H1 2020,” *F-Secure Labs*, 2020. [Online]. Available: [https://blog.f-secure.com](https://blog.f-secure.com/)

Avast, “IoT Threat Report,” *Avast Security*, 2020. [Online]. Available: [https://blog.avast.com](https://blog.avast.com/)

Sophos, “State of Ransomware Report,” *Sophos Labs*, 2021. [Online]. Available: <https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-state-of-ransomware-2021-wpna.pdf>

Rapid7, “Vulnerability Intelligence Report,” *Rapid7 Research*, 2020. [Online]. Available: <https://www.rapid7.com/research>

Darktrace, “Cyber AI and Autonomous Response,” *Darktrace Research*, 2021. [Online]. Available: <https://darktrace.com/en/resources>

Imperva, “Cyber Threat Index,” *Imperva Research Lab*, 2020. [Online]. Available: <https://www.imperva.com/resources/resource-library>

Proofpoint, “Human Factor Report,” *Proofpoint Security Research*, 2020. [Online]. Available: <https://www.proofpoint.com/us/resources>

Bitdefender, “IoT Security Report,” *Bitdefender Research*, 2020. [Online]. Available: [https://www.bitdefender.com](https://www.bitdefender.com/)

Cybereason, “Threat Analysis Report,” *Cybereason Research*, 2021. [Online]. Available: [https://www.cybereason.com](https://www.cybereason.com/)

SANS Institute, “SANS Security Awareness Report,” *SANS Institute*, 2020. [Online]. Available: [https://www.sans.org](https://www.sans.org/)

ISACA, “State of Cybersecurity 2021,” *ISACA Research*, 2021. [Online]. Available: <https://www.isaca.org/resources>

NIST, “Framework for Improving Critical Infrastructure Cybersecurity,” *National Institute of Standards and Technology*, 2018. [Online]. Available: <https://www.nist.gov/cyberframework>

ISO/IEC, “Information Security Management Systems – ISO/IEC 27001,” *International Standards Organization*, 2013. [Online]. Available: <https://www.iso.org/isoiec-27001-information-security.html>

ITU, “Global Cybersecurity Index 2020,” *International Telecommunication Union*, 2021. [Online]. Available: <https://www.itu.int/en/ITU-D/Cybersecurity/Pages/GCI-2020.aspx>

World Economic Forum, “Global Risks Report 2021,” *World Economic Forum*, 2021. [Online]. Available: <https://www.weforum.org/reports>

Symantec, “IoT device vulnerabilities: Exploiting the connected world,” *Symantec Security Report*, 2020. [Online]. Available: <https://www.broadcom.com/company/newsroom/press-releases>

HP Enterprise, “Internet of Things Research Study,” *HP Labs*, 2015. [Online]. Available: <https://www.hpe.com/us/en/newsroom.html>

Akamai, “The State of the Internet Security Report,” *Akamai Technologies*, 2020. [Online]. Available: <https://www.akamai.com/resources/state-of-the-internet>

Cloudflare, “DDoS Attack Trends Report,” *Cloudflare Research*, 2021. [Online]. Available: <https://www.cloudflare.com/learning/ddos/ddos-attack-trends>

Arbor Networks, “Worldwide Infrastructure Security Report,” *NETSCOUT Arbor Networks*, 2019. [Online]. Available: <https://www.netscout.com/threatreport>

Juniper Research, “The Future of Cybercrime & Security,” *Juniper Research*, 2020. [Online]. Available: <https://www.juniperresearch.com/researchstore>

IDC, “Worldwide IoT Security Forecast,” *International Data Corporation*, 2021. [Online]. Available: [https://www.idc.com](https://www.idc.com/)

MarketsandMarkets, “IoT Security Market Size and Forecast,” *MarketsandMarkets Research*, 2021. [Online]. Available: [https://www.marketsandmarkets.com](https://www.marketsandmarkets.com/)

Frost & Sullivan, “Global IoT Security Market Report,” *Frost & Sullivan Research*, 2020. [Online]. Available: [https://www.frost.com](https://www.frost.com/)

Allied Market Research, “IoT Security Market Outlook,” *Allied Market Research*, 2021. [Online]. Available: [https://www.alliedmarketresearch.com](https://www.alliedmarketresearch.com/)

Statista, “IoT connected devices worldwide,” *Statista Research Department*, 2021. [Online]. Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide>

Eurostat, “Digital economy and society statistics – households and individuals,” *European Commission Eurostat*, 2020. [Online]. Available: <https://ec.europa.eu/eurostat>

OECD, “Digital Security Risk Management for Economic and Social Prosperity,” *Organisation for Economic Co-operation and Development*, 2015. [Online]. Available: <https://www.oecd.org/sti/ieconomy/digital-security.htm>

World Bank, “World Development Report 2021: Data for Better Lives,” *World Bank Publications*, 2021. [Online]. Available: <https://www.worldbank.org/en/publication/wdr2021>

UNCTAD, “Digital Economy Report 2019,” *United Nations Conference on Trade and Development*, 2019. [Online]. Available: <https://unctad.org/system/files/official-document/der2019_en.pdf>

ITU, “Measuring digital development: Facts and figures 2020,” *International Telecommunication Union*, 2020. [Online]. Available: <https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx>

GSMA, “The Mobile Economy 2021,” *GSMA Intelligence*, 2021. [Online]. Available: <https://www.gsma.com/mobileeconomy>

Ericsson, “Mobility Report,” *Ericsson Research*, 2021. [Online]. Available: <https://www.ericsson.com/en/reports-and-papers/mobility-report>

Huawei, “Global Industry Vision (GIV),” *Huawei Research*, 2020. [Online]. Available: <https://www.huawei.com/en/giv>

Intel, “The Role of Artificial Intelligence in Cybersecurity,” *Intel Research Report*, 2019. [Online]. Available: <https://www.intel.com/content/www/us/en/security/artificial-intelligence-in-cybersecurity.html>