Transferability of Machine Learning Model for IoT Device Identification and Vulnerability Assessment

by

Priscilla Kyei Danso

B.Sc. Computer Engineering, Kwame Nkrumah University of Science and Technology, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Ali A. Ghorbani, Ph.D., Faculty of Computer Science
	Sajjad Dadkhah, Ph.D., Faculty of Computer Science
Examining Board:	Rongxing Lu, Ph.D., Faculty of Computer Science, Chair
	Haruna Isah, Ph.D., Faculty of Computer Science
	Nanwei Wang Ph.D., Department of Mathematics & Statistics

This thesis is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

December, 2022

© Priscilla Kyei Danso, 2022

Abstract

The lack of appropriate cyber security measures deployed on IoT makes these devices prone to many security issues. Machine learning (ML) models used to monitor devices in a network and make predictions by differentiating between benign and malicious devices have made tremendous strides. However, most of the research in profiling and identification uses the same data for training and testing. Hence, a slight change in the data renders most learning algorithms to work poorly. This study uses a transferability approach based on the concept of transductive transfer learning for IoT device profiling and identification. We propose a three-component system comprising the device type identification, the vulnerability assessment, and the visualization module. The device type identification component uses the underlying concept of transductive transfer learning, where the trained model is transferred to a remote lab for testing. This type of transfer learning works by explicitly assigning labels to the testing data in the target domain using the test feature space in the target domain, with training data from the source domain. The test dataset (target domain) will employ the trained model (source domain) knowledge. Furthermore, the vulnerability of the predicted device type is assessed using three vulnerability databases: Vulners, NVD, and IBM X-Force. Lastly, the results from the vulnerability assessment are visualized.

Dedication

To my parents, who sacrificed a lot to see me accomplish my purpose - this thesis is for you. To my siblings and friends who were pillars for me during this journey, I couldn't see the end without your counsel and encouragement. We did it!

Acknowledgements

I would like to express heartfelt gratitude to Dr. Ali Ghorbani for his constant support, candid feedback, approachable demeanour, and timely guidance throughout the thesis process. He encouraged me to push beyond my limit and work to present a factual thesis.

A big thank you to Dr. Sajjad Dadkhah for your attention to detail and novelty-thinking that made this work possible. I am grateful for all your constructive input as it shaped my thesis and contribution.

To the entire Canadian Institute for Cybersecurity (CIC) team, the Friday weekly presentations impacted the way to think as a researcher and veered my thesis in the right course; thank you!

Table of Contents

Abstract ii

Dedication iii

Acknowledgments iv

Table of Contents v

List of Tables ix

List of Figures xi

Abbreviations xiii

1 Introduction 1

1.1 Introduction 1

1.2 Problem Statement 6

1.3 Thesis Contribution 8

1.4 Thesis Organization 9

2 Background and Literature review 10

2.1 Overview 10

2.2 Internet of Things (IoT) 10

2.2.1 IoT Definition 11

2.2.2 IoT Characteristics 13

2.2.3 IoT Application Domains . . . 18
2.2.4 IoT security and privacy challenges . . . 22
2.2.4.1 Security challenges . . . 22
2.2.4.2 Privacy challenges . . . 25
2.3 Device Identification as a Solution . . . 26
2.3.1 Fingerprinting for IoT profiling and identification . . . 26
2.3.2 Machine Learning for IoT profiling and identification . . . 29
2.3.3 Deep Learning for IoT profiling and identification . . . 31
2.3.4 Blockchain for IoT profiling and identification . . . 33
2.3.5 Manufacturer Usage Description (MUD) for IoT profiling and identification . . . 34
2.4 Vulnerability Assessment in IoT . . . 36
2.5 Visualization in IoT . . . 40
2.6 IoT device identification and vulnerability assessment Taxonomy . . . 42
2.7 Closing Remarks . . . 43

3 Proposed Framework **44**

3.1 Overview . . . 44
3.2 The Experimental System Flow of the Proposed device type identification Framework . . . 47
3.2.1 Overview . . . 47
3.2.2 System Flow . . . 47
3.2.2.1 Data preprocessing . . . 49
3.2.2.2 The Proposed Feature selection . . . 50
3.2.2.3 Model Training . . . 51
3.2.2.4 Transferability of Classifiers . . . 53
3.3 Vulnerability Assessment Framework Proposed . . . 54
3.3.1 Overview . . . 54

3.3.2 Vulnerability Database . . . 55
3.3.2.1 National Vulnerability Database (NVD) . . . 55
3.3.2.2 Vulners . . . 58
3.3.2.3 IBM X-Force . . . 58
3.3.3 Vulnerability assessment overview . . . 59
3.3.3.1 Vulnerability keyword search . . . 59
3.3.3.2 Vulnerability harvesting . . . 60
3.3.3.3 Vulnerability Standardization . . . 60
3.3.3.4 Vulnerability Aggregation . . . 61
3.3.4 NVD Metrics definitions . . . 61
3.4 Closing Remarks . . . 69

4 Implementation and Experimental Results 71

4.1 Overview . . . 71
4.2 Data Acquisition . . . 71
4.2.1 CIC IoT Dataset 2022 . . . 72
4.2.2 IMC 2019 Payload Dataset . . . 73
4.2.3 IoT Sentinel Dataset . . . 75
4.3 Experiments . . . 76
4.3.1 Data collection . . . 76
4.3.1.1 Data sampling . . . 76
4.3.1.2 Feature selection . . . 78
4.3.1.3 Training and Testing of the ML methods . . . 78
4.3.1.4 Performance measuring metrics . . . 79
4.3.2 Profiling and identification . . . 81
4.3.2.1 CIC IoT dataset 2022 . . . 81
4.3.2.2 IMC 2019 Payload Dataset . . . 83
4.3.2.3 IoT Sentinel dataset . . . 84

4.3.3 Vulnerability assessment 85
4.3.4 Visualization and dashboard 88
4.4 Closing Remarks 92

5 Conclusion **94**
5.1 Open Challenges 95
5.1.1 IoT device identification open challenges 95
5.1.2 Vulnerability assessment open challenges 96
5.2 Future Research Directions 97

Bibliography **114**

Vita

List of Tables

2.1 The most popular IoT user names and passwords according to the 2018 Symantec's Internet Security Threat Report. 24

2.2 Summary of device profiling and identification review with ✗depicting research work that did not implement a vulnerability assessment while ✓depicts work that included vulnerability assessment. ML, F, DL, BC, and MUD each represent the different approaches used in the profiling and identification of a device Machine Learning, Fingerprinting, Deep Learning, Blockchain and Manufacturer Usage Description respectively. 36

2.3 Summary of Vulnerability assessment review 40

3.1 NVD CVSS v2.0 and CVSS v3.0 base score ranges and severity ratings. 69

3.2 Metrics attributes for different NVD versions. 70

4.1 Devices in the CIC lab and the corresponding device type sampled for our experiment. 73

4.2 Devices in the US testbed and the corresponding device type sampled for our experiment. 74

4.3 Devices from the IoT Sentinel dataset and the corresponding device type sampled for our experiment. 75

4.4 Feature vector name in the dataset, readable name and description. . 77

4.5 The number of records for each device type in the CIC dataset [1] after data preprocessing and the the number of sampled records used for training the ML model. 78
4.6 Performance of six classifier before and after normalization and feature selection. 81
4.7 Device type predicted by the different classifiers with the correctly predicted devices highlighted in green when the trained CIC IoT dataset 2022 was transferred to the US lab and tested with the IMC 2019 Payload Dataset [2] using the Interaction experiment conducted. . . . 84
4.8 Device type predicted by the different classifiers with the correctly predicted devices highlighted in green when the trained CIC dataset was transferred to the UK lab and tested with the IoT Sentinel dataset [3]. 85

List of Figures

1.1 Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025(in billions). [4] 2

2.1 Communication Patterns of IoT. 11
2.2 Conceptual depiction of IoT based on definitions. 12
2.3 Characteristics of IoT. 14
2.4 IoT Application Domains. 20
2.5 IoT Fingerprint Approaches. 27
2.6 A typical Machine Learning Pipeline. 29
2.7 Typical Deep learning architecture. [5] 31
2.8 Generic flow of MUD for IoT device identification. [6] 34
2.9 Taxonomy of IoT device identification and vulnerability assessment. . 42

3.1 The overall system framework of the proposed system. 44
3.2 The Experimental System Flow of the Proposed device type identification framework based on the concept of transfer learning. 48
3.3 Vulnerability assessment overview. 55

4.1 20 top features using ANOVA technique explained in section 3.2.2.2 and their respective score on the CIC dataset [1]. 79
4.2 Confusion Matrix of the six classifier used for device type identification. 82
4.3 Exploratory Analysis of the NVD data feed. 86
4.4 Most common vulnerabilities or approaches 87

4.5 Sunburst chart showing hierarchical results for the respective databases and the predicted device types. 89
4.6 Scatter chart of device types with score3 on x-axis and exploitability score on y-axis. 90
4.7 Bar chart showing the average CSSV score3 of the different predicted device types in the three different databases. 91
4.8 Bar chart showing different vendor and their aggregated CSSV score3. 91
4.9 Scatter plot of the different vendors with CSSV score3 on the xaxis and the CSSV exploitability score on the yaxis. 92

List of Abbreviations

IoT	Internet of Things
DoS	Denial of Service
ML	Machine Learning
WiFi	Wireless Fidelity
DL	Deep Learning
NVD	National Vulnerability Database
DNS	Domain Name System
KNN	K-Nearest Neighbor
DT	Decision Tree
RF	Random Forest
SVM	Support Vector Machine
MLP	MultiLayer Perceptron
GBC	Gradient Boosting Classifier
MUD	Manufacturer Usage Description
ANOVA	Analysis of Variance
GDPR	General Data Protection Regulation
EU	European Union
FTC	Federal Trade Commission
CVE	Common Vulnerabilities and Exposures
NIST	National Institute of Standards and Technology
CVSS	Common Vulnerability Scoring System
CVE	Common Vulnerabilities and Exposure

Chapter 1

Introduction

1.1 Introduction

We live in the early stages of the Internet of Things (IoT) era, which connects the physical and digital worlds and profoundly influences every part of our livelihood. In line with this, IoT has made headway in several industries, including retail, healthcare, and agricultural applications [7]. By 2022, it is anticipated that there will be up to 500 IoT devices per household [8]. It is gradually becoming apparent that the Internet of Things (IoT) is bringing about a paradigm shift in the digital age [9]. IoT is a technological revolution affecting all facets of life and businesses [10]. According to estimates, there will be 75 billion IoT-connected devices in use by 2025, making IoT a part of our everyday life [11]. The exponential growth of connected devices is both astounding and ground-breaking. As per Figure 1.1, between 2021 and 2025, the global deployment of IoT devices is predicted to reach 30.9 billion units, up from 13.8 billion deployed in 2021. The number of IoT devices deployed is significantly greater than the number of non-IoT devices, and this difference is growing exponentially, making the globe more linked than ever.

Smart home appliances, industrial equipment, and automobiles are standard IoT

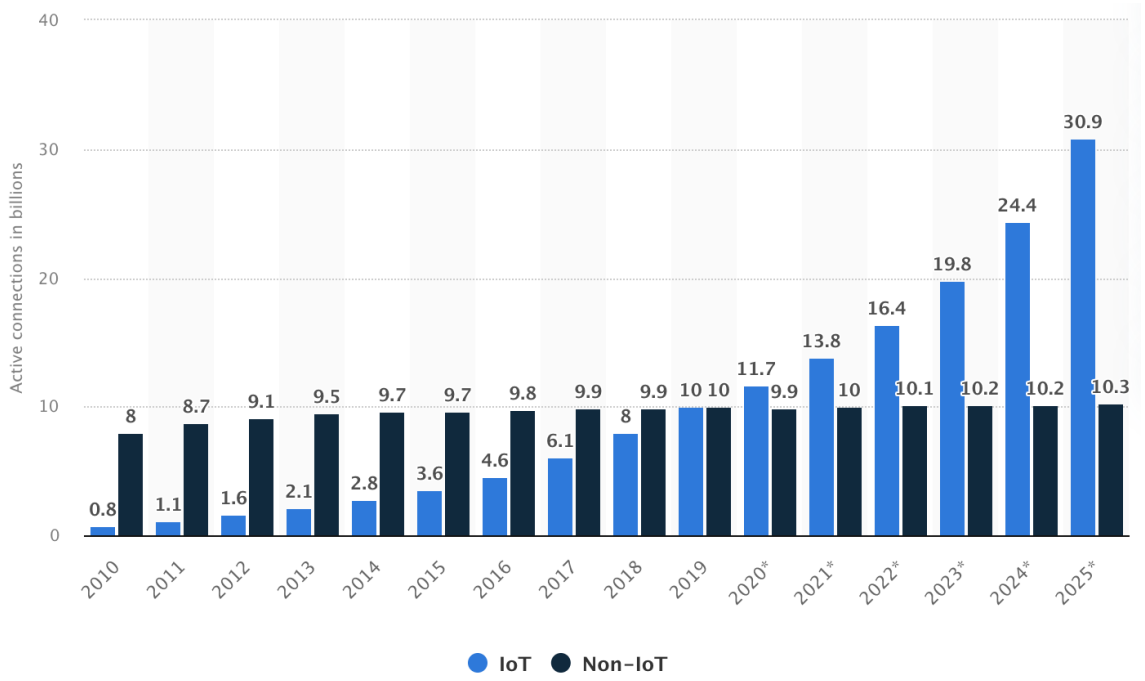


Figure 1.1: Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025(in billions). [4]

connections. Compared to IoT connections, non-IoT connections, such as those between smartphones, desktops, and laptops, are expected to total 10 billion by 2025, which is one-third less than connections between IoT devices [4]. With these staggering statistics, McKinsey Global Institute projects that by 2025, the potential economic impact might reach $11.1 trillion [12]. With such lucrative financial returns, it is not unexpected that digital titans such as Google and Samsung purchased Nest and SmartThings for $3.2 billion and $200 million, respectively, in 2014 [13].

Since many physical devices are used in IoT-based intelligent industry environments that are highly efficient and require little human labor, their security measures are of utmost priority. Major security concerns include poor physical security, vulnerable software or firmware, and an inadequate number of authentication and authorization procedures [14]. Thus, despite their many advantages in terms of convenience and usage, they also involve a wide range of security threats and vulnerabilities. The isolation of such devices with limited connectivity to other devices through a gateway

is a critical factor in attack mitigation for IoT devices [15]. Considering security risks, a superior technique for network administration is precise IoT device identification, as opposed to device isolation [16].

Understanding the IoT devices connected to a network is necessary for network management. Insider attacks might be carried out using a susceptible network-connected device. As a result, to monitor and safeguard the entire network, devices linked to the network must be uniquely identified [17]. Network administrators require a deeper understanding of the connected devices when more IoT devices are added to a network. IoT devices require more frequent monitoring to mitigate and identify intrusion and malware since they lack the computing power of regular networked devices.

IoT device security is crucial for the device and other network elements. For example, the Mirai botnet attack from October 2016 is still the most catastrophic Denial of Service (DoS) attack on IoT devices. The attack on the Dyn infrastructure rendered the Dyn DNS servers inoperable. The cyberattack recruited linked instruments from smart homes into "botnets" using the Mirai malware [18]. Passive fingerprinting, in which the network packets are collected by sniffing, can automate detecting connected devices using device fingerprinting. Administrators can remotely identify the type of device from its network activity attributable to device fingerprinting [16].

However, numerous security and privacy risks are pervasively ingrained by the enormous surge of digitally linked devices sharing large volumes of data [19]. Particularly as this IoT becomes prevalently interwoven in the day-to-day life of people, users must have confidence that these devices are safe from threats [20]. IoT devices are also vulnerable to various harmful passive and active attacks that compromise their security and privacy, which might impair their operation [21]. Recent cyberattacks, including Hajime, IoT Reaper, BrickerBot, or Mirai [22], use IoT device weaknesses to seize control of the targets and build IoT botnets, which are typically employed to

launch Denial of Service (DoS) attacks. Because of these flaws, IoT device manufacturers collaborate with standardisation organisations to create the next generation of safer, more standardised intelligent devices. However, security certification is still a problematic undertaking [23].

IoT communications must also be protected since large volumes of data can be transmitted between devices as well as between those devices and servers or customers [24]. If the CIA (Confidentiality, Integrity, or Availability) of this data is compromised, it could have catastrophic consequences. Inadequately protected data streams on these devices act as possible access for attack and vulnerability to data theft [20]. Because of power consumption, memory usage, connectivity, and battery or power availability issues, IoT devices cannot offer the same level of security as traditional computers.

Moreover, Machine learning (ML) models used to monitor devices in a network and make predictions by differentiating between normal and illegitimate devices have made tremendous strides. The assumption that the training and test data are taken from the same domain is one that is frequently made when dealing with supervised or semi-supervised machine learning problems. Ergo, most learning techniques are ineffective when the distribution changes [25]. This criterion implicitly presupposes that the training and test sets still have the same attribute representations and come from the same distribution [26].

Finding training data that fits the feature vector and data distribution characteristics of the test data can occasionally take time and effort. Consequently, creating an effective model for a target domain that has been trained from a pertinent source domain while using the unlabeled test data to make predictions is essential [27].

The motivation for this study on transferability and the underlying idea of transfer learning comes from the aforementioned problem. The use of labelled training data from a source domain to utilise information available therein to improve a model's

performance in a target domain with little to no labelled target training data has been shown to be successful for a variety of real-world applications [28].

This thesis uses transductive transfer learning for device profiling and identification using several ML algorithms. This transferability considerably improves learning algorithm performance without incurring the cost of labeling target data by taking advantage of the unlabeled test data in the target domain to adapt to the developed model in the source domain.

We propose a three-component system comprising a device profiling and identification module, a vulnerability assessment module, and a visualisation dashboard module. The device profiling and identification module used transferability, as mentioned earlier. We created a system that gathers all incoming and outgoing network packets connected devices send. In order to determine whether any inbound packets are from legitimate devices, all incoming packets are inspected, and their matching device type is profiled and identified. The presumption is that the packets a device produces can be used to establish its type [29].

Forty-eight (48) network and statistical features are extracted from the packet generated for each device. The features are trained on six machine learning algorithms to determine the feasibility of implementing the proposed system using several performance metrics. Our work evaluates the results for two instances; when both train and test data are drawn from the same distribution and when the training and test data are drawn from a different distribution. The dataset had class imbalance problem was addressed by randomly selecting an equal number of records for each class to guarantee that each distribution class had an equal probability of being chosen.

From data with additional training and test data, the trained model is transferred to another testbed in a different location with other devices but identical device types. For each of the instances mentioned earlier, the predicted device type by the classifier is collated and used input keywords to search each of the three vulnerability

databases, i.e., NVD, IBM X-Force, and Vulners. Search data from each vulnerability database is harvested into local databases. Search results from IBM X-Force and Vulners are standardized by retrieving the CVE-IDs and searching them in the NVD database. In this study, NVD is assumed as the trustworthy vulnerability search database hence the used as the "master" vulnerability source. The device type from each database is aggregated into a local database. The vulnerability assessment is performed on each device for potential severity, impact score, and exploitability, among other metrics. The assessed vulnerabilities are displayed in the visualisation dashboard.

1.2 Problem Statement

IoT has the power to gather, measure, and comprehend the world around us, enabling modernizations that enhance living satisfaction. IoT devices typically operate in an unattended setting, making them vulnerable to physical intrusion. To achieve security objectives, remedies must take the big picture into account. IoT devices are typically connected via wireless networks, where a hacker could intercept a communication channel and acquire private data [30]. IoT devices' low computational and power capabilities prevent them from supporting elaborate security mechanisms [31]. As a result, IoT devices pose a greater security risk than typical computing platforms. Due to increased linked devices, administrators and owners are concerned about management and monitoring. A crucial first step in anomaly detection is uniquely identifying each device, whether by wireless connectivity, behavioural patterns, physical characteristics, or network-based traces [6]. As a result, the first and most crucial step in protecting all other IoT devices in a network is isolating malfunctioning devices and restricting connectivity.

IoT device monitoring capabilities can intelligently offer a defense in identifying

and profiling the devices. The data exploratory techniques of ML are effective for identifying device behavioural patterns and characteristics. However, a fundamental condition for the success of ML algorithms is that the distributions of both the training data and the test data are similar [32].

The data used to train ML models differs significantly from the data utilized in real-world IoT applications. The aforementioned problem usually calls for large amounts of labeled training data from various IoT devices [33]. However, manually classifying this massive amount of data can be time-consuming and labor-intensive [34]. This makes it difficult for ML-based algorithms to detect IoT devices in various situations. Based on the underlying concept of transductive transfer learning, this thesis presents a transferability approach for IoT device profiling and identification using ML models.

An IoT device identification and vulnerability system is the primary objective of this research. The identification system uses a transferability method based on supervised ML, which employs the concept of transductive transfer learning to enhance conventional ML models by enhancing learning in a comparable target task by applying information learned from multiple source tasks [35]. The packet capture module, feature extraction module, data preprocessing module, training module, and testing and evaluation module make up the conceptually suggested device type identification architecture. A sniffer built inside the packet-capturing module is in charge of monitoring both inbound and outgoing network traffic. The feature extraction module is used to extract practical characteristics from the gathered network packet traces, which are then applied for traffic categorization purposes [36]. The training module, which is based on the idea of transfer learning, is the main module in this research. After a successful training session, the model's information is transported to a different lab for testing.

1.3 Thesis Contribution

This research makes these contributions:

- A complete system comprising device identification and profiling, vulnerability assessment, and visualization dashboard module.

  The proposed framework is a pipeline comprising data acquisition, preprocessing, ML training, search databases for vulnerability, harvest search results, standardise harvested search results, aggregate results, and visualise on the dashboard. This will be explained further in Section 3;

- A transferability approach with the concept of transductive transfer learning for IoT device identification and profiling.

  This study employed a transferability approach with the underlying concept of transductive transfer learning for a dataset with training and test data drawn from a different distribution. In this approach, no labeled data in the target domain (test data) are available, while labeled data in the source domain are available. This is further continued in subsection 3.2.2.4;

- An evaluation of the proposed transferability approach on three publicly available IoT datasets: CIC IoT dataset 2022 (for training), IMC 2019 Payload dataset (for testing), and IoT Sentinel (for testing) and six machine learning algorithms.

  Six machine learning classifiers and three publicly accessible datasets were used to evaluate the proposed methodology. In subsection 4.2, we summarize the various datasets, their network configuration, and packet capture strategy. Finally, in subsection 3.2.2.3, we briefly explain the six ML models employed in this study;

- A vulnerability assessment of the IoT device identified and profiled by the classifier.

  After identifying the device in the network, we further assess the device's vulnerability using three vulnerability databases; NVD, IBM X-Force, and Vulners. We further investigate the NVD database and analyse the different metrics used for vulnerability assignment in subsection 3.3.2. The section provides a description and background of the different databases.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

- Chapter 2: We review the concept of the Internet of Things, its inception, definitions, characteristics, and security and privacy concerns. We also review some of the identification techniques employed in IoT as well as vulnerability assessment approaches;

- Chapter 3: We describe the conceptual experimental framework proposed for this thesis;

- Chapter 4: We discuss and analyze the experimental results of our approach;

- Chapter 5: We conclude our work and suggest some future research direction.

Chapter 2

Background and Literature review

2.1 Overview

In this Chapter, we examine the background, development, applications, and security issues related to IoT devices. We propose IoT device identification as the appropriate response to the majority of these issues. As a result, we looked at previous research on IoT device identification and profiling as well as vulnerability analysis in an IoT system. We then compile our results from the reviews we conducted, developed a taxonomy to conclude our review of the related works in this domain.

2.2 Internet of Things (IoT)

The term Internet of Things stems from the words "Internet" - the huge global network of networks that communicate via Transmission Control Protocol (TCP) and Internet Protocol (IP) [37] and "Things" - any object with a unique identification, access to the "Internet", knowledge of its location and status, and the ability to add operations and intelligence while integrating the physical and digital worlds together [38].

"Things" in the Internet of Things *can be a person with a heart monitor implant, a*

farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an Internet Protocol (IP) address and can transfer data over a network [39]. These widely heterogeneous connections and objects communicate with one another via a variety of communication protocols, including human-to-human (H2H), human-to-thing (H2T), thing-to-thing (T2T), and thing-to-things (T2Ts) [40]. Figure 2.1 depicts the various communication in a typical IoT infrastructure.

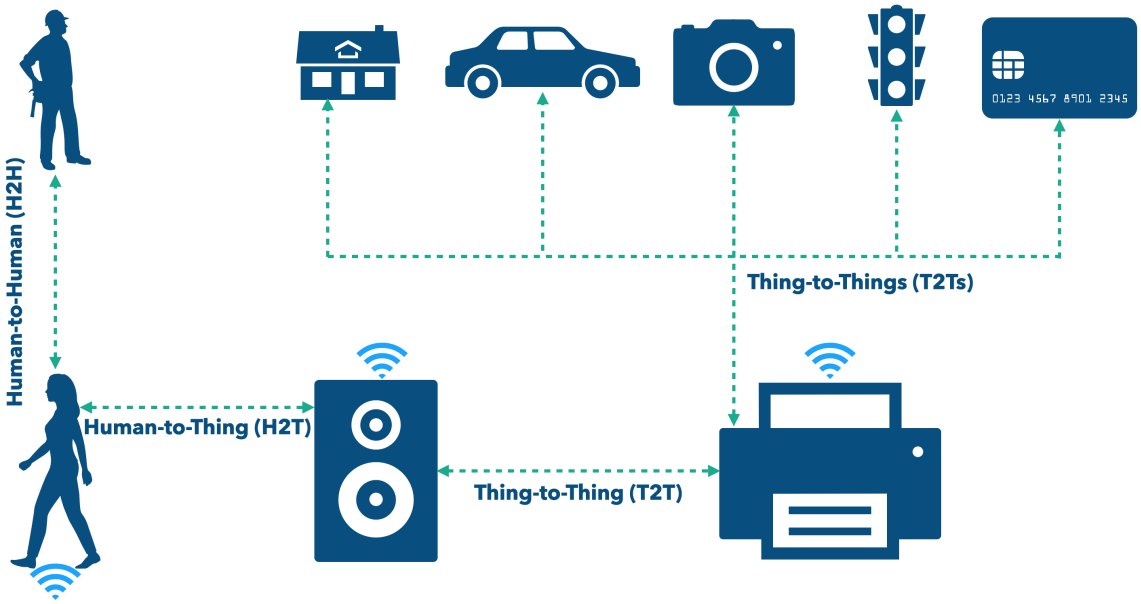


Figure 2.1: Communication Patterns of IoT.

2.2.1 IoT Definition

Over the decades, IoT has been developing and is still the newest and most popular idea in the IT industry. There is no unified definition of IoT because of the diversity of advanced technologies and architecture.

Kevin Ashton first used the Internet of Things (IoT) in 1999. He defined it as a network of connectivity between the physical world and the Internet through the utilization of RFID and ubiquitous sensing technologies that monitor and detect the

real world [41].

According to the Internet Engineering Task Force (IETF), "*IoT is the network of physical objects, or 'things' embedded with electronics, software, sensors, actuators, and connectivity to enable objects to exchange data with the manufacturer, operator, and/or other connected devices*" [42]

European Parliamentary Research Service referred to "*IoT as a distributed network connecting physical objects capable of sensing or acting on their environment and communicating with each other, other machines, or computers. The data these devices report can be collected and analyzed to reveal insights and suggest actions that will produce cost savings, increase efficiency or improve products and services*" [43].

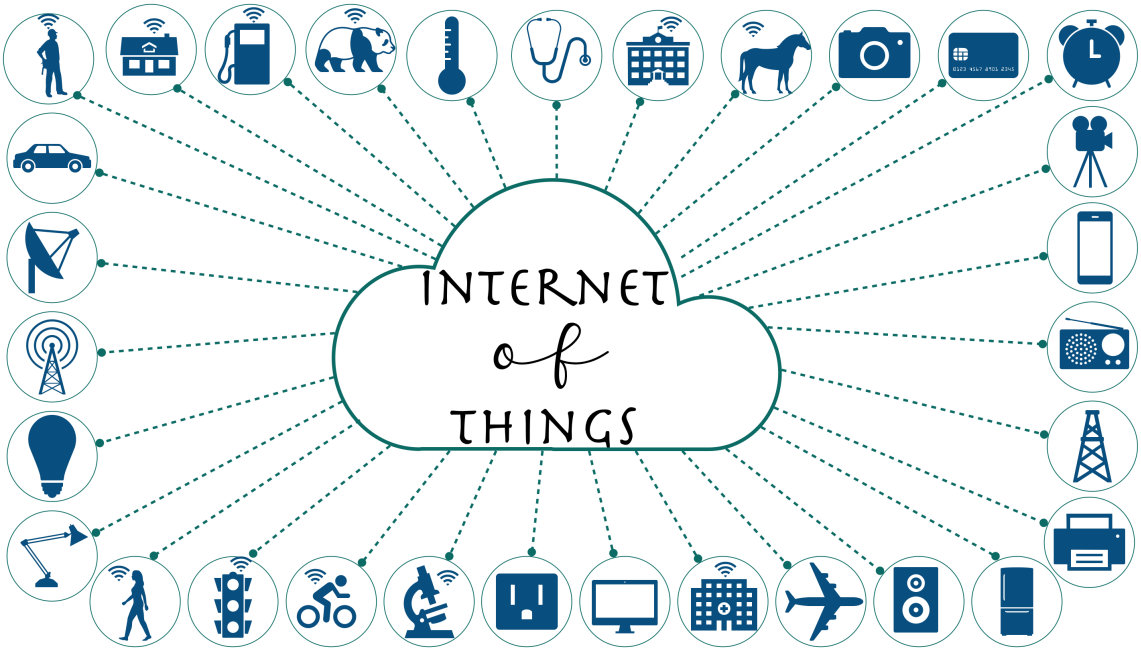


Figure 2.2: Conceptual depiction of IoT based on definitions.

The IoT European Research Cluster (IERC) definition states that IoT is "*A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual 'things' have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network*" [44].

Based on the diverse definition of IoT by the different entities, Figure 2.2 is a conceptual depiction of IoT.

2.2.2 IoT Characteristics

There is a contrast between traditional computer systems and IoT devices. They differ significantly from traditional systems in that IoT provides significant security challenges due to their intrinsic traits like heterogeneity, interconnectivity, scalability, and resource limitations. The detailed descriptions of these particular foundational traits may be found in [45, 46] and are shown in Figure 2.3. Each of the qualities is described below:

- **Communication:** A device's designation as an IoT device is independent of its sophistication. This could be something as straightforward as an alarm clock or as intricate as an Audi electric automobile, or a chicken with a biochip transponder implanted. Its connection to the internet, whether directly or indirectly, is a crucial component. Directly linked is rather self-explanatory—it might connect using a regular Ethernet cable, WiFi, or any other common internet interface. Devices that are indirectly connected may employ a technology like Bluetooth or Zigbee to connect to a gateway, which is subsequently connected to the internet, making them a little more enigmatic.

- **Computation:** The "nervous system" and computational power of the IoT is represented by processing elements (such as microcontrollers, microprocessors, SOCs, and FPGAs) and software programmes. Several hardware platforms, including Arduino, UDOO, FriendlyARM, Intel Galileo, Raspberry PI, BeagleBone, Cubieboard, WiSense, Mulle, and T-Mote Sky, have been created to execute IoT applications. Operating systems are essential since they operate throughout a device's activation period [46]. IoT services can be hosted on a

range of open-source and commercial cloud infrastructures and architectures. These infrastructures provide a way for the acquired data to be promptly analyzed and for connected devices to send their data to the cloud in order to benefit from the discovered knowledge.

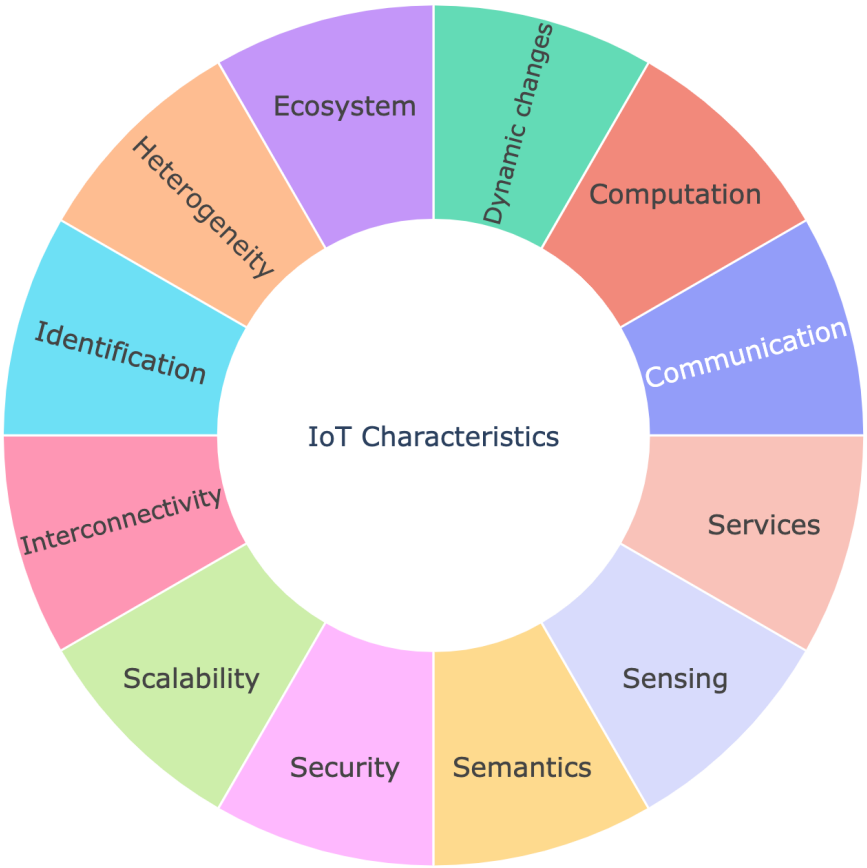


Figure 2.3: Characteristics of IoT.

- **Dynamic changes:** Device scenarios, such as location and speed, as well as the states of devices, such as sleeping and waking up, switch on and/or switched off, also alter dynamically. Additionally, the number of devices may fluctuate. The first and most important step in any IoT use case is to gather data and transform it so that business actions are taken with it. Throughout the entire process, several IoT components must dynamically change their states. For instance, a temperature sensor's input will change continuously depending on

the time of the day (morning, afternoon, evening), location, environment, and other factors.

- **Ecosystem:** Many manufacturers and companies are exploiting the framework of IoT to support the production of their own devices. The significance of the framework is evolving as the number of devices increases exponentially (the ability to support diverse communication protocols, data protocols, technologies, and devices). The devices' ability to cooperate and exchange information with one another is mostly ensured by the architecture. It is essential for preventing cross-interference between the devices as well.

- **Heterogeneity:** A Bluetooth sensor interoperable with an industrial drone deployment coupled with a WiFi-connected monitoring station, and a smart vending machine or a connected medical device are examples of a typical smart industry. There is no doubt about it: heterogeneous IoT is commonplace. The term "heterogeneity" refers not only to the communication protocols (Bluetooth, WiFi, ZigBee, or Z-Wave) but also to the various devices' infrastructure and identification protocols, hardware and software platforms and versions. Determining broad best practice guidelines is challenging due to the variability of the IoT environment. Additionally, it is impossible to create a security solution that is universally applicable because different applications, communication protocols, and hardware platforms may have different security needs.

- **Identification:** Identification is the unique characteristic that is used to uniquely identify IoT devices. To identify users and match services to their needs, the IoT needs identification. There are numerous ways to identify objects, including electronic product codes (EPC) and ubiquitous codes (uCode). It is essential to address the objects to distinguish between the object ID and its address, which may include IPv6 and IPv4. Networked objects might em-

ploy public IP addresses rather than private ones. Each object in the network has a distinct identity attributable to identification mechanisms.

- **Interconnectivity:** According to the various definitions provided by the different schools of thought, it is clear that any "thing" with internet access can eventually be absorbed into the world's technology and communication architecture [46]. Any object may monitor and engage with its environment if it is equipped with the appropriate sensors. A remote control can be used to operate a TV, alarms, and security cameras in a typical smart home. Conversely, with the use of a smartphone app, one may adjust the temperature, use voice control to draw the blinds, and turn on or off the light or music. This is how pervasive our world is becoming as a result of the development and widespread use of IoT devices.

- **Scalability:** Scalability should be taken into account during the system design phase because of the rising ubiquity of IoT. The ability of a system to expand without impacting its performance is frequently used to define scalability. This can be done by enhancing an existing system with greater hardware resources or by introducing new software layers. In other words, the system can accommodate more users and data without suffering performance degradation. The current rate of data production is 2.5 quintillion bytes per day, but the IoT will only accelerate this growth [47]. The management and storage of the data gathered, as well as its processing for application purposes, will become even more crucial given this enormous amount.

- **Semantics:** Semantic in the IoT refers to the ability to extract knowledge by different machines to provide the required services. Semantic technologies together with other sophisticated processing techniques can build on machine-interpretable modelling paradigms to offer great potential for characterizing ob-

jects, distributing and integrating information, and drawing conclusions from new knowledge [48]. For example, measurement readings from metres can be used to accurately estimate and measure power usage in smart grids; analyzing a combination of air quality, temperature, traffic, humidity, proximity, and pressure sensory data records offer meaningful insights that can be used to achieve competitive advantage. These data can be analyzed and transformed into intelligent decision-making for a better understanding of our physical world. To improve seamless integration, raw data in a variety of formats is turned into standardized, machine-readable data.

- **Sensing:** IoT refers to the collection of information from connected things within a network and its subsequent transmission to a data warehouse, database, or cloud. In order to take precise actions based on the services needed, the acquired data is examined. IoT sensors include everything from water quality sensors, which are utilized to observe water quality, to temperature sensors, which track changes in temperature and turn them into data, to biomedical sensors, which are employed in the medical field to provide early illness diagnosis and enhance patient monitoring. Optionally, IoT sensors could be actuators, wearable sensing devices, etc. IoT sensors are now essential for increasing security and safety, cost savings, and performance improvement.

- **Services:** When a thing performs a task for a user, it is referred to as a service. This is because the thing is responding to the information that the user receives from the physical world. Reaching the level of ubiquitous services is the ultimate objective of all IoT applications. The majority of the currently available applications offer collaboratively aware, identity-related, and data consolidation features [49]. These characteristics consolidate the data to enable decision-making and subsequent actions to be taken while making services accessible to anybody, anywhere, at any time. In Section 2.2.3, the specific

IoT services and domains will be covered in detail.

- **Security:** Security is one of the key components of the IoT ecosystem. IoT security should not be an afterthought, but rather integrated into the design and production process. Sensitive data is transferred from consumers to servers via data communication across the whole IoT ecosystem cycle. To prevent data exploitation and alteration, appropriate safety, security, and privacy requirements must be followed while developing an IoT system. An IoT ecosystem can inevitably collapse if any one of its components is compromised. Other concerns include those related to hacking, data governance, and standardization.

2.2.3 IoT Application Domains

IoT is focused on making users' life easier. From inclusive education to autonomous vehicles, from intelligent farming to effective healthcare, and from mechanized farming and so on. In practically every industry, IoT applications are becoming more and more prevalent. In Figure 2.4, a few of the IoT's widely used applications are displayed as stated in [45, 46, 50, 51].

- **Smart Education:** Future education will be more accessible and participatory attributable to IoT in schools. Schools can accommodate the unique educational needs of individuals who have special needs with the aid of IoT. For instance, they may provide a "read text" option that would allow visually challenged students to have their textbooks read aloud to them. With the help of IoT devices, teachers may monitor the learning progress of their pupils in real-time and students have improved access to educational materials and messaging services. Additionally, IoT enables educational institutions to monitor vital resources and enhance campus safety.

- **Smart Transport:** In smart vehicles, human control is eliminated hence these vehicles must effectively cooperate to maintain efficient traffic flow on roads and highways. More and more advanced electronic components are being installed in modern vehicles to ensure traffic control, safe driving, and driver efficiency [52]. Vehicle communication makes it possible for vehicles to exchange data and analyze it instantaneously [53].

- **Smart Home:** A standard smart house has a collection of sensors for monitoring environmental factors including light, weather, humidity, and closeness. One or more measurements can be taken by each sensor. All sensors provide data visualization and storage, allowing the user to access the data at any time and from any place. The refrigerator may add a food item to the grocery list or alert if a dessert spoils in a couple of days [54]. Surveillance cameras, intelligent locks, motion, smoke, and other sensors are all connected devices that alert you in case of an intrusion.

- **Smart City:** A Smart city is a conceptual urban development model for urban clusters that maximize individual, the social, and technological potential for improving the quality of life. [55]. Federal programs, transportation and traffic management, electricity, water, healthcare, innovative urban agriculture, and waste management are just a few of the numerous industries that will undergo modernization and improvement in a smart city [56].

- **Smart Buildings:** Building automation systems (BAS) in smart buildings are linked to the Internet. BAS enables management and monitoring of various HVAC, lighting, security, safety, entertainment, etc. Furthermore, it could improve building upkeep and power usage. For instance, a blinking washing machine or dryer can issue warnings to indicate something is wrong [46].

- **Retail:** IoT is utilized in retail to assist brands in tracking products along

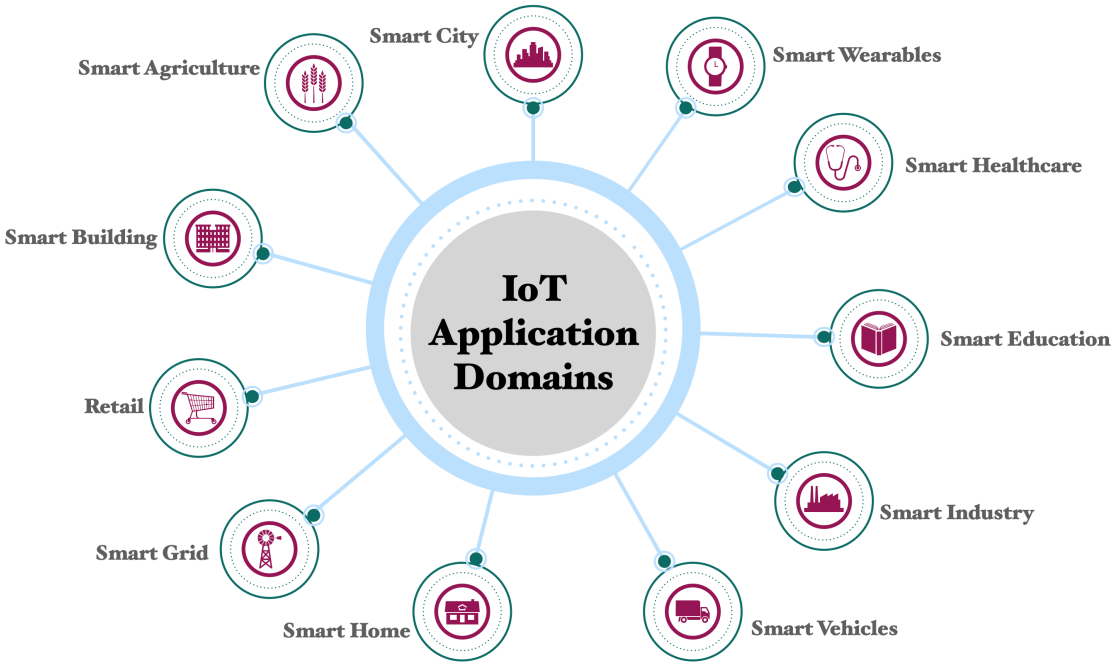


Figure 2.4: IoT Application Domains.

the supply chain. It provides retailers with the transparency they need to keep an eye on goods movement, conditions, and location while also predicting an exact delivery time. Utilizing internet of things technologies, the retail industry can achieve cost savings, operational efficiency, and innovation. In order to better inform consumer strategy, retailers can use the IoT to track customer buying behaviours and analyze their purchase histories, insights, and location data [57].

- **Smart Grid:** Power consumption and management must be examined as smart homes and cities proliferate. Using current information technology, smart grids can distribute electricity more efficiently and respond to a variety of situations and events. In principle, the Smart grid might perform the required procedures in response to grid events such as power generation, transmission, distribution, and consumption. As an illustration, in the event of thermal insulation, the SG may automatically modify the power flow and resume the distribution grid's electrical delivery service.

- **Smart Industry:** Organisations are investing in technologies that can oversee the entire manufacturing process without human intervention to streamline the production pipeline with the aid of IoT. This ultimately reduces labour costs, man-hours, and productivity time by using a higher level of security for devices and equipment. Long term, increased productivity will allow manufacturers to produce more.

- **Smart Agriculture:** Farmers have the opportunity to change the agricultural business by using sophisticated IoT farming applications to optimize several labour-intensive farm processes [58]. IoT allows farmers to track and remotely monitor their crops and machinery by using technological tools like drones, smartphones, and computers. Precision agriculture sensor monitoring networks are widely used to assess agri-related information like temperature, humidity, soil PH, soil nutrition levels, water level, etc. [59].

- **Smart wearables:** Smart wearables are becoming more and more common in daily life as they collect and analyze data, sometimes make informed decisions, and communicate the response to the user. Smart devices known as IoT-enabled wearables can be attached to or tattooed on the skin, integrated into clothing and other items, or worn as an exterior accessory. Some of the wearables that have been created for various uses include smart watches, wristbands, eyewear, headsets, earbuds, body straps, foot and hand-worn devices, and smart jewellery [60].

- **Smart Healthcare:** Healthcare industry employs IoT to remotely transfer health information that has been evaluated to processing centres so that appropriate actions can be taken while the physiological conditions of patients are being monitored by sensors [46]. For instance, a tethered inhaler and accompanying app can assist patients with respiratory disorders in tracking the

administration of their prescription and providing audible and visual notifications to prompt them to take a dose [61].

2.2.4 IoT security and privacy challenges

Security is a top priority since IoT is a development of the conventional, unencrypted Internet paradigm, where connections involving the digital and physical worlds collide. In addition to providing secure and reliable communication for both types of human-device interactions, IoT security measures must combat traditional networking threats. We outline some of the current security and privacy challenges in IoT.

2.2.4.1 Security challenges

The IoT platform can seamlessly integrate with other application domains as explained in section 2.2.3, generating security issues about the data gathered and efficient management. Observations based on industry analysis include the following:

- **Object Identification and locating in IoT:** In addition to providing security, identity identification operates as a restriction mechanism on unauthorized nodes. [62]. Prior to other security concerns, identifying an object is the most crucial issue. An effective process of item identification highlights the object's characteristics while also identifying the object in a unique way. A host can be uniquely identified on the Internet using a Domain Name System (DNS), a reliable identifying method. It is still susceptible to man-in-the-middle and DNS cache poisoning attacks, among other types of attacks [63].

- **Authentication and authorization:** Traditional authentication and authorization techniques such as public-key cryptosystems, and id/password might

not be appropriate for IoT devices and networks due to their heterogeneity and complexity [63]. In the case of public-key cryptosystems, the work of managing keys will become challenging due to the continually expanding number of devices. Hence, an attacker may use weak authentication techniques to append and impersonate rogue nodes or tamper with data integrity, invading IoT devices and network connections. In these situations, there is also a constant risk that the transferred and used authentication keys will be misplaced, destroyed, or tampered with.

- **Limited encryption capability:** IoT systems and devices with constrained computing capabilities demand effective, robust, and interoperable encryption mechanisms. The system ought to take into account the more complicated nature of data transfer within an IoT-integrated environment. Data encryption is a must-have for maintaining the confidentiality of information transmitted between nodes [62].

- **Insufficient Physical security:** Most IoT devices run on their own in unsupervised contexts [64]. With little effort, a malicious actor may easily gain physical access to such devices without authorization and then take over. As a result, the devices would then sustain physical damage from an attacker, who might also reveal the cryptographic techniques used, duplicate their firmware using a malicious node, or corrupt their control or data [65].

- **Insecure web interface:** Most IoT-based products use a web or mobile interface to communicate and receive data generated from different service providers. It is discovered that the web interface is more vulnerable to the OWASP IoT Top 10 vulnerabilities, such as weak default credentials, inadequate session management, and cross-site scripting vulnerabilities [66]. Programming errors made by engineers are inevitable during development. Soft-

ware vulnerabilities are flaws that cause security problems [63].

Table 2.1: The most popular IoT user names and passwords according to the 2018 Symantec's Internet Security Threat Report.

Rank	**User name**	**2017 Percent**	**Password**	**2017 Percent**
1	root	40	system	10.3
2	admin	17.3	sh	10.2
3	enable	10.3	123456	9.1
4	shell	10.2	admin	3.7
5	guest	1.5	1234	3.1
6	support	1.3	password	2.5
7	user	1.1	12345	2.5
8	ubnt	0.9		2.3
9	DUP root	0.6	root	2.1
10	supervisor	0.5	support	1.2

- **Continuous availability:** It is risky for IoT platforms to continuously protect against continuous and repetitive attacks like Denial of Service (DOS) attacks since they may have an influence on the overall core ecosystem of reliant systems. Making sure IoT services are available and ongoing while preventing any possible performance breakdowns and disruptions is the primary problem.

- **Periodic security upgrades:** It is common for vendors to ignore security patches for extended periods of time or to lack automated methods of updating patches. Even currently offered update methodologies lack assurances for their integrity, making them open to fraudulent changes and widespread application. For IoT devices to be secure and functional, operating systems, embedded firmware, and software should be updated regularly [65].

- **Use of default IoT parameters:** The insecure default usernames and passwords established by the manufacturer are used by about 70% of the devices and sensors. The most prevalent vulnerability in the IoT has been identified as having weak, guessable, or hardcoded passwords [67]. The majority of consumers refuse to alter the devices' factory-set login and password. Users

typically pick passwords that can be quickly and easily guessed, nevertheless, while changing their passwords. Furthermore, few manufacturers enforce the use of secure passwords. The most popular IoT user names and passwords are displayed in Table 2.1 according to the 2018 Symantec's Internet Security Threat Report [68].

2.2.4.2 Privacy challenges

IoT systems gather data that may be confidential to a corporate institution or personal to a user. The following must be managed more effectively during implementation:

- **Transparency and the ethics of data collection:** Due to the volume of information available in potentially complicated IoT ecosystems and the discreet techniques of data gathering, users are unaware of the data practises of IoT devices and their makers [69]. Hence, our private lives will inevitably change profoundly as linked devices become more prevalent. The widespread usage of equipment and networks with lax security postures contributes to some of the threats. The primary concern with the data collected is who will have access to it and how it will be used [70].

- **Huge data generation:** The IoT devices are producing an enormous amount of data. While IoT tracking entails tracking a device, the goal is to comprehend the behavioural patterns of the person using the device. The wealth of knowledge about the person, their actions, movements, and preferences are what gives that information its value, in fact. Additionally, persistent patterns of location data associated with a certain device may be able to provide insight into that device's position at specific times of day, ultimately revealing sensitive information such as workplace or home [70].

- **Privacy Regulatory and Compliance Requirements:** The utilization of numerous networks, sensors, objects, and applications along with the world-wide nature of IoT devices have dramatically expanded this difficulty. As a result, data may be gathered, analyzed, evaluated, and utilized across numerous jurisdictions with various laws and regulations.

2.3 Device Identification as a Solution

Device profiling and identification have emerged as a cutting-edge strategy that considerably reduces some, but not all, of the security concerns in IoT devices that are listed in the subsection 2.2.4.1. The fingerprinting of a device is one of the well-known methods for device identification. The majority of the time, using device network traffic or physical properties, and behavioural patterns, there are numerous techniques to fingerprint the device. Similar to user authentication, device identification validates the legitimacy of the attached device to the network [6]. Due to the heterogeneity of the IoT ecosystem, accurate device identification is necessary but also challenging.

Device identification describes a method that determines an IoT device based on its features. Cui et al. [71] described IoT device identification: "*the input is various data collected from a device, e.g. sensors' data, network data, etc.; the output is a label for the IoT device indicating the type of the device.*"

2.3.1 Fingerprinting for IoT profiling and identification

Device fingerprinting can be physical, wireless, based on network traces, or behavioural, according to Noman et al. [6]. The different fingerprinting approaches used by different research are reviewed. Figure 2.5 shows the different fingerprint approaches as suggested by [6].

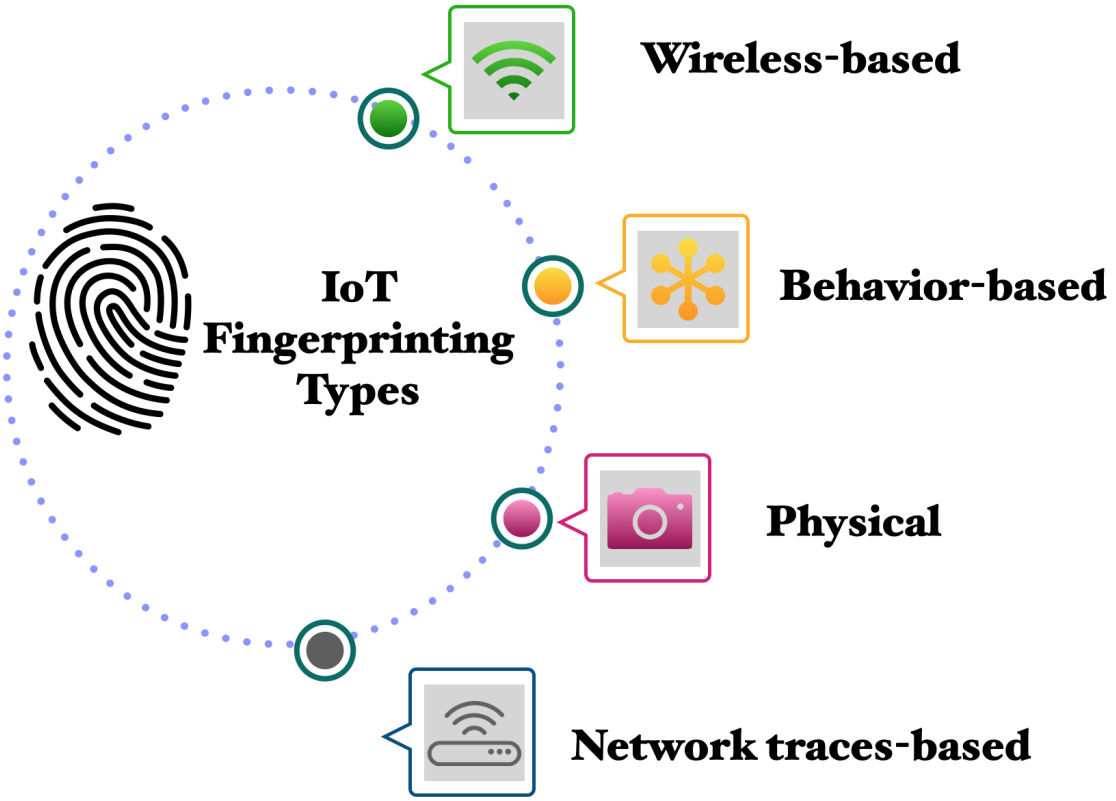


Figure 2.5: IoT Fingerprint Approaches.

In IoTSense, Bezawada et al. [72] simulated a device's activity as a set of employed protocols and a group of monitored request and response patterns. The inbound and outbound network data of the device is gathered, and then relevant features that are indicative of the device's behaviour are extracted. The features are then combined using a statistical model, which is then utilized as a guide for determining the behavioural disparities of the device. For feature classification, a variety of ML classifiers are employed, including Decision Tree, Majority Voting, and k-Nearest Neighbors (kNN). The derived device feature vector is applied to the trained machine-learning models to identify distinct device kinds. The proposed methodology demonstrated that it is possible to accurately identify various device kinds by fingerprinting.

Miettinen et al. [73] proposed IoT Sentinel which discovers and identifies new IoT device types that are added to the network, assesses the vulnerability of the device

using the device type, and performs mitigation techniques based on the vulnerability found. Observed network activity serves as the foundation of the fingerprint proposed in this study. This work leverages the distinctiveness of a new device added to the smart home network to the gateway as these devices adhere to a certain predefined vendor-specific process. The device type identification employs a dual identification method. The initial step involves binary classification to determine the appropriate device type. The second fold employs multiple classifiers that accept unknown fingerprints, match various device kinds, and break ties when multiple devices match using a distance-based metric.

Locality-sensitive IoT fingerprinting (LSIF), a unique method for identifying IoT devices, is presented by Charyyev et al. [74]. A locality-sensitive hash (LSH) function called Nilsimsa is used to construct the traffic profile of an IoT device from the flow of its network traffic. A series of incoming and outgoing packets of the specified IoT device's MAC address is considered to be the network traffic flow in this instance. Following that, a signature database is then used to hold the relevant device target variable and produced a hash set. A comparison is done on the LSH of a new device joining the network and the hash values previously stored in the database. The highest average hash similarity score between each recorded device and the device being identified is computed. As a result, the device with the highest mean similarity score is chosen as the predicted identity.

Vaidya et al. [75] proposed the IoT-ID, a device-specific identifier that may be utilized for device identification and gathers the properties of the device. Physically unclonable functions (PUFs), the foundation of IoT-ID, make use of variances in the construction process to produce a distinctive fingerprint for integrated circuits. The authors blend elements from the clock and Analog-To-Digital Converter (ADC) PUFs to create a device-specific identity. Two situations were employed for IoT device identification. First, the IoT device automatically triggers the device identi-

fication prior to sending any information to a controller or prior to executing any systemic decisions. In the second scenario, before acting on information from the device, the controller or aggregator may request the identity of a distant IoT device. In either scenario, a device's IoT-ID is created and verified to ascertain its identity.

2.3.2 Machine Learning for IoT profiling and identification

There is a plethora of research on IoT device identification using Machine learning. These learning algorithms can be supervised, unsupervised, semi-supervised or reinforcement learning [76]. Decision Trees, Random Forest, SVM, and Neural Networks are just a few examples of ML algorithms that can be integrated with IoT systems to improve their effectiveness and scalability. In several different computing fields, academics are using ML technology in their study. Figure 2.6 depicts a typical ML pipeline and process flow.

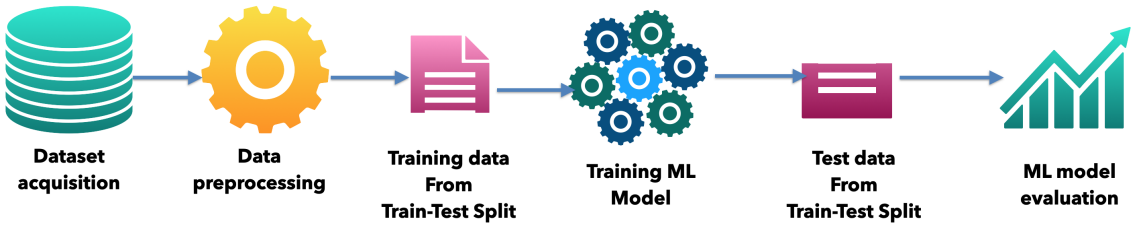


Figure 2.6: A typical Machine Learning Pipeline.

To identify unauthorized IoT devices connected to a network, Yair et al. [77] used TCP/IP traffic network data for categorization by ML. The authors assume that the dataset adequately reflected each device type on the white list. To effectively identify IoT device types from the white list, features from network traffic data were extracted using supervised machine learning, more especially Random Forest. Nine different IoT device categories totalling 17 unique IoT devices were gathered and manually labelled to train and test multi-class classifiers. 60 features were generated and eliminated several based on the following criteria: zero variance, irrelevance, or

a propensity for model overfitting; and only used the list of top characteristics to profile and identify the device type. One IoT device type was excluded from the white list to signify an unauthorized type. To verify the classifier's transportability, the model was trained on two security cameras of the same type (SimpleHome-XCS71001) in Lab A and tested with a different security camera in Lab B. (Withings WBP02WT9510). The trained classifiers obtained an average of 96% accuracy in the detection of illegitimate IoT device types.

Fürst et al. [78] proposed that information infusion aims to quickly transfer knowledge used during the training phase across tasks rather than transferring taught knowledge to another task (as in transfer learning). To accommodate asynchronously to a new situation by querying various knowledge sources, the authors used adaptive weak and strong knowledge functions that interface with a knowledge graph and external knowledge sources. Hence, weak and strong knowledge functions were proposed as different forms of knowledge functions. A 'white-box' knowledge model was created by combining these knowledge functions into a weak ensemble and a strong ensemble. The model was trained and tested using a variety of classifiers, with Random Forests (RF) showing the best results with an F1−score of 80.2%.

To accurately identify IoT devices linked to a network, Yair et al. [79] employed ML techniques (Gradient Boosting Machine, GBM, Random Forest, and XGBoost) to capture network traffic data. The authors gathered and categorized the traffic data from nine different IoT devices, as well as computers and smartphones, utilizing the HTTP packet (user agent) feature to train and evaluate the classifier. A multi-stage meta classifier was built using supervised learning; in the initial stage, the classifier can differentiate between traffic produced by IoT and non-IoT devices. The classifier correctly detects the existence of a particular IoT device within the network traffic in the second stage. 34 test instances remain unclassified, while 19 cases were incorrectly classified. Based on 7,376 test instances, the total IoT classification accuracy for the

model is 99.281%.

2.3.3 Deep Learning for IoT profiling and identification

Computing models with numerous processing layers may learn interpretations of information at various degrees of complexity attributable to deep learning [80] as seen in Figure 2.7. Since IoT applications create a great quantity for data analysis and processing, deep learning has several advantages over machine learning, the most significant of which is superior efficiency with high data size. Another advantage is that deep learning can dynamically retrieve new features for various challenges [81]. The following three varieties of deep neural networks are widely employed today: MultiLayer Perceptrons (MLP) Convolutional Neural Networks (CNN) Recurrent Neural Networks (RNN) [82].

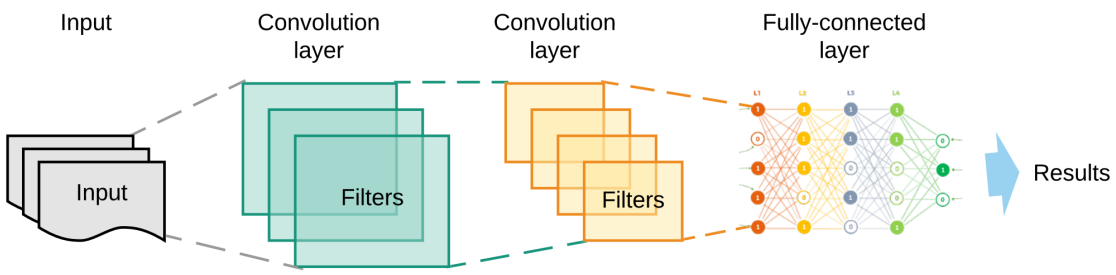


Figure 2.7: Typical Deep learning architecture. [5]

Jaidip et al. [83] detected IoT devices linked to a network using data from traffic data, precisely IoT devices not on the whitelist (unknown devices). The method the authors suggested was based on representation learning and consisted of two experiments: one for detecting legitimate IoT devices in network traffic and the second for identifying illegitimate devices that are attached to a network. A preprocessing step is to remove all headers from pcap files and convert the files into a binary file that will have the hexadecimal values that constitute the packet payload. All binary files have been altered to have 784 bytes; if there are more, the size of the file is extended

with 0x00 bytes rather than truncated. The binary files can also be transformed into grayscale graphics with a resolution of 2828 pixels, where each pixel stands for two hexadecimal numbers. The proposed method identified known devices in a network with a maximum accuracy of 99.87%. The classifier was trained on nine diverse IoT devices, with the exception of one IoT device that was regarded as an unknown IoT device, using eight classes each time. The best accuracy for identifying unidentified devices was 99.87%.

To increase IoT security, Jafari et al. [84] presented a physical layer authentication method that uses deep learning to recognize Radio Frequency (RF) devices. To ensure the adaptability of the suggested models to different wireless channel conditions, we contrast the capacity of Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) to model the RF traces acquired from ZigBee devices over a range of Signal-to-Noise Ratio (SNR) levels. This is a viable alternative for RF fingerprinting to distinguish between permitted RF devices and threat actors using identical devices due to the accessibility of wireless big data and cutting-edge deep learning capabilities. Using a window size of 32 at 50 epochs, the learning models DNN, CNN, and LSTM each achieved an accuracy of 96.3, 94.7, and 76%, respectively.

Yongxin et al. [85] suggested an improved deep learning system that uses sensed data to identify IoT devices. To boost resilience and explainability, the authors added the zero-bias layer to deep neural networks and made it possible for the proposed framework to detect unknown IoT devices. Actual data from the IoT application in aviation known as automated dependent surveillance-broadcast (ADS-B) was used to assess the performance of the suggested architecture. Among 130 airborne transponders, a remarkable 94% identification rate was recorded. Additionally, an enhancement was suggested for understandable and reliable machine learning in IoT following careful research of the behaviour of the final fully linked layer in DNNs. The

authors claim that their research can be applied to other fields like virus detection or unsupervised intrusion detection.

2.3.4 Blockchain for IoT profiling and identification

Studying the potential for device identification in a blockchain-based IoT is the contribution of the authors in this paper [86]. Devices in a smart home environment were utilized in the evaluation of the proposed methodology. A blockchain was populated using the traffic generated from the testbed configured. Blockchain transactions are recorded with identical timestamps as communications in the dataset. The overall performance of categorizing transactions as belonging to a certain device and determining the device type from the transaction transient behaviour was investigated. Based on transient behaviour discovered from the transactions, the expected device types are determined. In order to determine the kinds of participating devices, the adversary aims to compare transient signatures seen in the transaction ledger to known IoT device transient behaviour. Analysis of the proposed strategies' effects on attack success shows that they can lower attack success rates to 20–30%.

Alam et al. [87] proposed a generalized fingerprinting approach based on blockchain technology as a way to authenticate edge devices with distinctive PUF IDs (Physical Unclonable Function Identifications embedded in the device's memory during production). It recommended that blockchain technology is incorporated as a full application, API authenticator, or front end for smart contracts. They distinguish between a global and local component. While the global system verifies registered devices by anybody, anywhere without being able to pinpoint the particular manufacturer, the local implementation allows defence-in-depth authenticity. Verification happens after registration on the gateway and the blockchain ledger BL over a secured gateway. Following a thorough approval process in which hashed, values of both the blockchain ledger BL and gateway coincide, the device identification is

verified.

2.3.5 Manufacturer Usage Description (MUD) for IoT profiling and identification

In this section the Manufacturer Usage Description(MUD) for IoT device identification is reviewed. Figure 2.8 shows the generic process flow of MUD for IoT device identification.

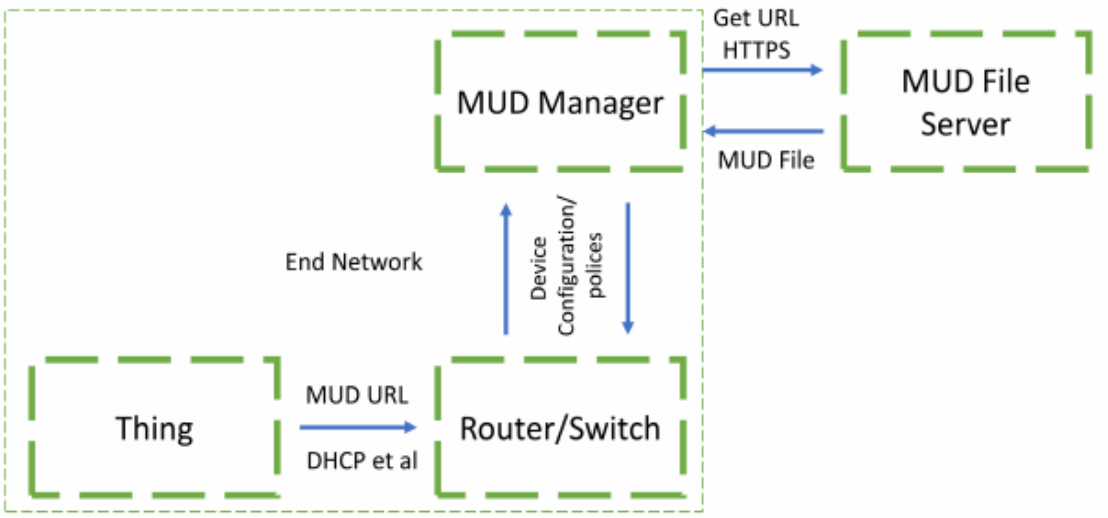


Figure 2.8: Generic flow of MUD for IoT device identification. [6]

Manufacturer Usage Description (MUD) [88] is a new standard created by the Internet Engineering Task Force (IETF). Device profiles are defined by the MUD specification. An IoT device will submit a MUD URL along with its LLDP, DHCP, or X.509 request [6] when it initially joins an access control station. The MUD manager receives the URL after it has been extracted by the Network Access Device, packetized, and sent to the Server. The MUD manager then uses HTTPS to connect to the maker's internet-hosted MUD File Server to which the URL links. The MUD file for the device in question is delivered to the MUD manager following confirmation that it was created by the device maker. This conceptual goal is converted by the

MUD Manager into a context-specific guideline and sent to the server. The server then enacts the guidelines on the network utilizing Access Control Lists (ACLs) for that IoT device's entry outlet. Then, depending on the maker's predetermined goal, accessibility to the device is granted.

A programme developed by Ayyoob et al. [89] uses the traffic trail of any IoT device as the source and immediately employs a systematic semantic framework to validate a provided MUD profile for consistency and to determine whether it complies with a particular institutional policy. The authors also offer a system for categorizing IoT devices that use network traffic flow and well-known MUD signatures to proactively classify IoT devices and monitor any changes in their network activity. The "winner" will be the device or collection of devices with the maximum similarity measure for device identification.

The authors of this paper [95] suggest a software-defined framework that enhances IoT network behavioural surveillance while also enhancing Manufacturer Usage Description (MUD)-based network intrusion detection systems. The suggested framework employs software-defined networking (SDN) and the digital twin idea to enhance security in the IoT network. In order for this to function, MUD data is profiled, and before any actions are taken in the actual network, they are assessed on the network's digital twin. The network traffic that was collected in the pcap file is used to collect a number of features. Furthermore, two profiling techniques are also used; one is dependent on network traffic flow, whereas the other is based on MUD profiles that are developed. To find the irregularities, a clustering-based outlier detection method was used on the feature set that was created. According to experimental findings, the developed framework is able to identify the majority of network-centric attacks with the maximum sensitivity (0.998) and lowest false positive rate (0.02)

Table 2.2 summarises the reviews of several researchers' works, including the different identification methodologies employed, the main goals that affected their study, and

Table 2.2: Summary of device profiling and identification review with ✗depicting research work that did not implement a vulnerability assessment while ✓depicts work that included vulnerability assessment. ML, F, DL, BC, and MUD each represent the different approaches used in the profiling and identification of a device Machine Learning, Fingerprinting, Deep Learning, Blockchain and Manufacturer Usage Description respectively.

No	Paper	Purpose	Algorithm / Tools	Type of Identification	Vulnerability assessment?
1	[90]	Detect abnormal behavior.	k-Means clustering, Support Vector Machine (SVM)	ML	✗
2	[72]	Perform device behavioral fingerprinting for device type identification.	k-nearest Neighbors, Decision tree, Gradient boosting, Majority voting.	ML	✗
3	[74]	IoT device identification using locality-sensitive hash (LSH) function.	LSH function called Nilsimsa	F	✗
4	[91]	Employ direct and indirect sources to obtain capability profiles of IoT devices.	Automated Software tool	F	✗
5	[92]	Model the behavior of the IoT devices to detect and mitigate cyber-security events.	Graph (Multi-DiGraph)	F	✓
6	[93]	Characterize and model behavioral fingerprints of IoT devices in edge networks.	Minimum description length (MDL) principle	F	✗
7	[94]	Automatic device identification and anomaly detection through network traffic analysis.	XGBoost, Rectified Linear Unit (ReLU), K-Means clustering	ML	✗
8	[75]	Device-specific identifier for IoT device identification.	Physically Unclonable functions (PUFs)	F	✗
9	[77]	Identify unauthorized IoT devices connected to a network.	Random Forests (RF)	ML	✗
10	[78]	Transfer knowledge for robust and transferable ML for IoT systems	Random Forests (RF)	ML	✗
11	[79]	Accurately identify IoT devices linked to a network	Gradient Boosting Machine (GBM), Random Forest, and XGBoost	ML	✗
12	[83]	Detect IoT devices in a network using data from traffic data	Neural Network	DL	✗
13	[84]	Recognize Radio Frequency (RF) devices	Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM)	DL	✗
14	[85]	Identify IoT devices	Deep Neural Networks	DL	✗
15	[86]	IoT device identification in a smart home environment	Blockchain	BC	✗
16	[87]	Authenticate edge devices	Blockchain ledger BL	BC	✗
17	[6]	IoT device profiling	Manufacturer Usage Description (MUD)	MUD	✗
18	[89]	Categorize IoT device traffic flow and well-known MUD signatures	Manufacturer Usage Description (MUD)	MUD	✗
19	[95]	Detect behavioral anomalies of inbound and outbound traffic in the IoT ecosystem	Software-defined networking (SDN) Manufacturer Usage Description (MUD)	MUD	✗

whether or not the authors included a vulnerability assessment component.

2.4 Vulnerability Assessment in IoT

The goal of vulnerability assessments has always been to help organizations find and fix systemic weaknesses before they are exploited. Weidman et al. defined vulnerability as "*a flaw within a system, application or service which allows an attacker to circumvent security controls and manipulate systems in ways the developer never intended*" [96]. Hence, given that attackers are progressively targeting IoT-based devices, it is damaging to ignore research into device vulnerabilities. Vulnerability as-

sessment analyzes a computer software, system, hardware, network, or application in order to find, quantify, and grade systemic flaws that may be systematically fixed [97]. Two frequently used methods that might be employed to evaluate and pinpoint the weaknesses of loT devices are active and passive procedures. This is a crucial component of vulnerability analysis. The National Vulnerability Database (NVD) hosts a collection of known vulnerabilities which can be used for passive vulnerability assessment. This can be achieved by examining various system installation components and collecting unique attributes about these components without inflicting harm on the devices. Passive, albeit potentially safer, may not accurately represent the vulnerability of the system since it predominantly uses previously maintained databases [96]. On the other hand, active vulnerability assessment that includes directly uncovering open ports may end in device destruction and a corresponding interruption in function [98]. Common passive evaluation tools include NVD, CNVD, CNNVD, Vulners, X-Force, etc. While Nessus, Vulners, Shodan, Nmap, and other technologies may be utilized in active evaluation.

Anand et al. [99] incorporated vulnerability analysis and power storage techniques. This integration is intended to offer a workable way to overcome the difficulties in smart agriculture. The collector, the analyzer, the quantifier, and the mitigator make up the vulnerability assessment component. The collector uses Nmap to acquire data about the network and IoT devices. The analyzer juxtaposes the data from the NVD database using the data it has collected as input. The quantifier assesses the vulnerabilities based on the risk that a certain IoT poses to the system using the CVSSv3 metric from the analyzer. Using the vulnerabilities and the associated score, the mitigator proposes ways to resolve the vulnerability. The mitigator also provides manufacturers and IoT users with information about vulnerabilities and their remedies. In the event that known vulnerabilities are mitigated promptly, the IoT device is less likely to become the target of a potential adversary.

Multi-layer Vulnerability Keywords Matching is a method that Lou et al. [100] developed to assess the studied vulnerabilities of IoT devices. From the IoT device, Nmap was used for reconnaissance. Identifying open ports and the services that are currently using them requires the usage of Nmap. To accurately extract vulnerability search terms, the authors use two techniques. First, information on the class of products, operating systems, and software versions affected by the vulnerability is taken from the CPE during the usual extraction step. The second stage involves retrieving additional keywords from the CVE description by analyzing its syntactic patterns.

The objective of this research by Samtani et al. is to use passive and active vulnerability assessment approaches to pinpoint the weaknesses of Internet-enabled SCADA systems in light of the possible serious repercussions of perpetuating SCADA systems. In particular, the authors gather a sizable testbed of SCADA devices from the IoT search engine Shodan and analyze their vulnerabilities using Nessus and the National Vulnerability Database (NVD). According to the study's findings, numerous SCADA systems from well-known manufacturers like Siemens and Rockwell Automation are susceptible to SSH exploit, Man-In-The-Middle (MITM), and default login threats [98]. In order to perform passive vulnerability evaluation, the authors initially saved the 1,903,365 NVD vulnerability data feeds into local media. Nessus is employed by the active vulnerability assessment part of the proposed architecture to proactively examine the SCADA devices.

Hashmat et al. [101] proposed an automated context-aware IoT vulnerability assessment rule-set architecture in their work. The presented technique automatically recognizes IoT devices and the processes they execute, collects their corresponding vulnerabilities, converts them into rules, and applies those rules to security solutions. 49 IoT devices were employed to assess and evaluate the proposed framework. The findings showed that the suggested framework dynamically created rules to protect

against every vulnerability found in the network under evaluation. This research has also suggested IoT vulnerability assessment rule-set components that should be taken into account when creating any IoT vulnerability assessment. The authors aim to bridge the gaps in existing layers of protection against growing IoT vulnerabilities using the suggested technique. This technique was effective in locating flaws in the devices. The suggested technique was successful in locating device vulnerabilities that conventional mechanisms would not have picked up on. This study's presentation of different and required IoT-specific identification rule-set components was based on the evaluation.

In order to forecast vulnerability metrics, a methodology that initially uses machine learning and natural language processing to examine vulnerability details is implemented by Xuanyu et al. [102]. The network connection and vulnerability information for each connection point is then presented using a two-tiered graphical security framework, with an attack graph in the top layer and an attack tree in the bottom layer. By identifying possible attack pathways, this security system immediately evaluates the IoT network's security. Utilizing smart building data as a proof-of-concept because it comprises a wide range of real-world IoT devices and possible weaknesses, the authors assess the practicality of the proposed method. The assessment of the suggested methodology reveals its efficiency in terms of dynamically detecting the vulnerability indicators of new security flaws with, on average, more than 90% accuracy and locating the most susceptible attack vectors in an IoT ecosystem.

Our examination of the literature on vulnerability assessment is summarised based on the goal of the research, the vulnerability database or tool used and the type of analysis - Active, Passive or both in Table 2.3.

Table 2.3: Summary of Vulnerability assessment review

No	Paper	Purpose	Database/ Tool	Active / Passive Analysis
1	[99]	Incorporated vulnerability analysis to overcome the difficulties in smart agriculture.	NVD	Passive
2	[100]	Assess the known vulnerabilities of IoT devices	Nmap, CVE	Both
3	[98]	Employed passive and active vulnerability assessment approaches to pinpoint the weaknesses of Internet-enabled SCADA systems	Shodan Nessus NVD	Both
4	[101]	Presented an automated context-aware IoT vulnerability assessment	NVD	Passive
5	[102]	Use ML and natural language processing to examine vulnerability details to predict vulnerability metrics	NVD	Passive

2.5 Visualization in IoT

We review some of the works done in displaying the results of data captured from IoT to gain meaningful insights and make business decisions.

Pieter et al. proposed a collection of propulsion systems that are a part of the public test and development platform known as the Smart Maintenance Living Lab, which is used to speed rolling-element bearing lifespan studies [103]. The system also comprises a configurable dashboard tool for fleet management and visualization, as well as a modular IoT middleware cloud platform for dependable data intake and retention. In this study, a comprehensive framework for IIoT systems with particular criteria for security, adaptability, and resilience is proposed. The resultant platform gives ML algorithms baseline information to optimize. The authors claim the proposed system can be employed in many IoT use cases, including smart cities, transportation, and healthcare, to benefit from its use.

Building information modelling (BIM) is the digital depiction of a facility's structural and functional attributes for decision-making purposes throughout its life span. There are still several limitations to the Internet of Things (IoT) and business information management (BIM). The approach to collect and visualize sensor data in BIM is described in this work [104]. It also covers the creation of a platform that enables the communication between sensors using various protocols. The authors make use of Dynamo to continuously update information that has been visualized in

BIM using sensor data as input, as well as how visualization may aid in management choices that result in energy savings. The example used in this study shows how the suggested system may be used to incorporate sensors and display sensor data within the perspective of a comfortable environment. The system may be modified to watch over two separate spaces, one using the same contextual view and the other a different one.

Precision farming can be enhanced by augmented reality (AR), particularly in fluid indoor planting. In this research, the authors present an innovative framework termed AR-IoT that incorporates IoT into an AR-based ecosystem [105]. The AR-IoT solution improves object engagement by seamlessly superimposing IoT data onto physical things. The system includes a multi-camera, an inexpensive and non-destructive imaging platform for the IoT. It is made so that a farmer may see the crops from various perspectives, which eliminates visibility issues. The technology has the ability to seamlessly combine IoT data with AR resolution. Additionally, the farmer will have direct access to IoT data from the outside world. As a result, AR-IoT would improve surveillance functions, aid farmers in accurately ensuring the quality of the crop, and lower the cost of planting operations.

The growth of Big Data by IoT devices is advantageous for Smart Cities. However, visualization methods are a crucial part of surveillance and preventative operations. Ana et al. provide an approach for handling big data that is based on visualization methods [106]. By supporting users in making decisions, the system seeks to enhance the collection of information. Provided is a case study involving service requests for a fire department. In this way, the information obtained from citizen calls will be used to apply our results. The outcomes of the proposed work will, therefore, serve to increase the efficiency of resources, and, as a consequence, the sustainable development of a Smart City, which can operate more effectively with fewer resources. These visualizations show inbound real-time data from IoT sensors so that users may

compile the essential information to make methodical and consequential choices.

2.6 IoT device identification and vulnerability assessment Taxonomy

A taxonomy that captures the background of IoT and the findings of the literature research was developed in this study. A thorough taxonomy that includes the study of IoT devices, device identification, and vulnerability assessment is provided in Figure 2.9. The taxonomy covers a range of topics vital to comprehending IoT

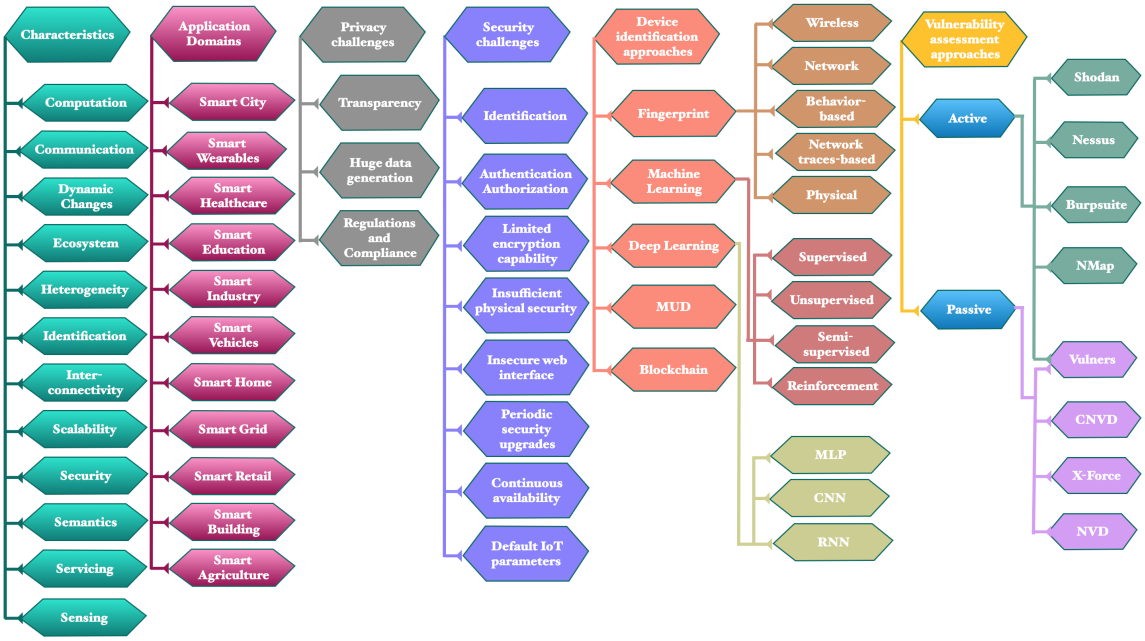


Figure 2.9: Taxonomy of IoT device identification and vulnerability assessment.

security concerns and proposes identification as the answer to the majority of security difficulties. The taxonomy consists of six sections: (1) IoT characteristics, (2) IoT application domains, (3) and (4) security and privacy challenge IoT faces respectively, (5) IoT device identification techniques and their various types, and (6) vulnerability assessment evaluation techniques and tools used to conduct analysis. The many topics covered by the taxonomy are related in a diverse range of aspects as the

sources of IoT security flaws and/or potential remedies.

2.7 Closing Remarks

In this chapter, we studied the history of the Internet of Things (IoT) device, which encapsulates several definitions from various schools of thought, characteristics, and security and privacy issues. Our research divided the five methods used to identify IoT devices into five categories: fingerprinting, ML, DL, blockchain, and MUD. We also looked at the works of other researchers in the field of IoT device vulnerability assessment and the various analytical approaches. Finally, we developed a taxonomy to consolidate our results. The proposed framework will be discussed in more detail in the next chapter.

Chapter 3

Proposed Framework

3.1 Overview

As Figure 3.1 shows, the proposed framework comprises three main components: device identification and profiling, vulnerability assessment, and the visualization/-dashboard module.

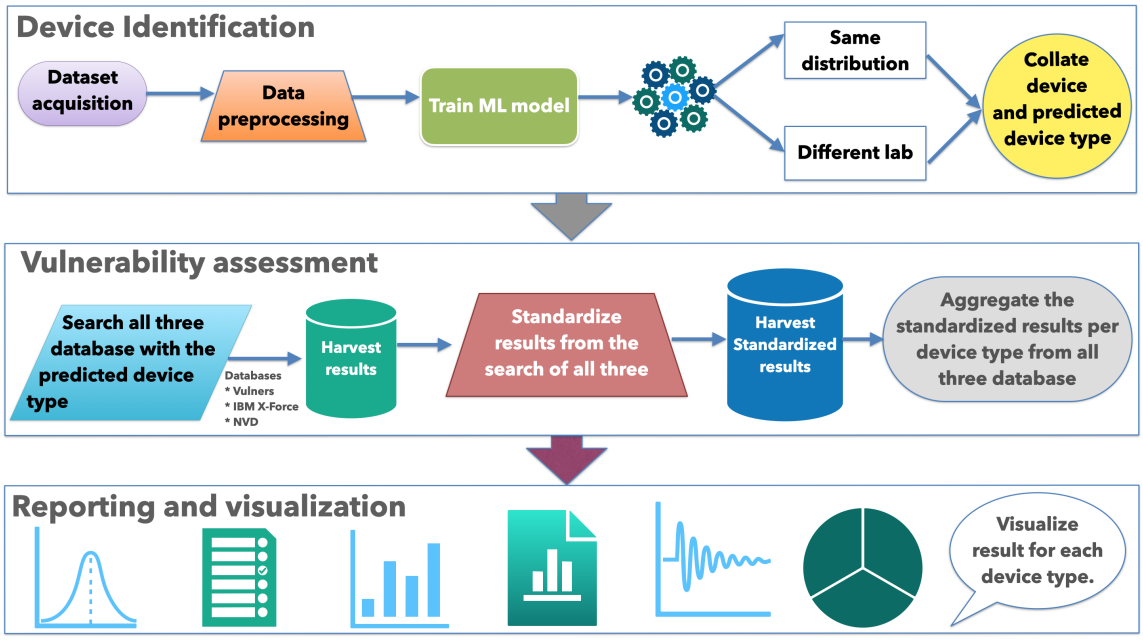


Figure 3.1: The overall system framework of the proposed system.

The main reason for the proposed framework is to have a compound system re-

sponsible for identifying the devices and device types in a network using ML and simultaneously assessing and visualizing the vulnerabilities of the predicted device type:

Algorithm 1 highlights the lists formal steps of our proposed framework, and below is a brief description.

(a) Extract features from each IoT device;

(b) Feature features extracted are used to train the ML model after data preprocessing;

(c) If a dataset is from the same distribution, train a percentage (70%) of the ML model with a percentage and test the trained model with the remaining percentage (30%);

(d) Assuming the training data (source) is a different distribution from the testing data(target), use the trained model from the source domain in the device identification of the target domain;

(e) For each instance in steps c) and d) the predicted device type by the classifier is collated;

(f) The predicted device type is used as a keyword to search all three databases (NVD, IBM X-Force, and Vulners) for vulnerability assessment;

(g) The results from the search are harvested in a local database;

(h) For uniformity, the harvested search results from IBM X-Force, and Vulners are standardized using the NVD database;

(i) The standardized results in then grouped per device type i.e., Audio, Camera, Home Automation, and Smart hub;

(j) The resultant aggregate is visualized and displayed on a dashboard.

Algorithm 1: Algorithm for the proposed framework

Input: IoT device feature set

Output: Vulnerability of the IoT devices

Data: Testing set x(same distribution) and y(different distribution)

$DL = dl1, dl2, \cdots, dl_n$ /* where each element represents the feature set per device. */

Function `MLTraining(`*deviceFeatures*`):`

  if *dataset from the same distribution* **then**

    trainedmodel =train(dl_i in *data*(DS)) /* Train ML model with 70% of the data */

    predicteddevicetypes = *trainedmodel*(x) /* Test trained ML model 30% of the same data used for training */

  else if *dataset is from different distribution* **then**

    predicteddevicetypes = *trainedmodel*(y) /* Test the trained ML model with another dataset */

  else

    None

  return *predicteddevicetypes* /* Collate device and predicted device type identification. */

Function `VulnerAssess(`*predicteddevicetype*`):`

  foreach *predicteddevicetype in predicteddevicetype* **do**

    vulsearch = searchDB(*predicteddevicetypes*)
    /* Search all three databases(NVD, IBM X-Force, and Vulners) for vulnerabilities with the predicted device type as a keyword. */

    harvestVul = harvestResults(*vulsearch*)
    /* Store all search results from the three databases(NVD, IBM X-Force, and Vulners) into a local database */

    standardizeSearch = standardSearch(*harvestVul*)
    /* Standardize the harvested IBM X-Force, and Vulners search results by using the NVD database for a uniformity */

    aggregateVul = aggregate(*standardizeSearch*)
    /* Aggregate the standardized results per device type from all three standardized database. */

  return *aggregateVul*

Function `VisualizeVul(`*aggregateVul*`):`

  foreach *vulneraility in aggregateVul* **do**

    visualVul = displayVisualization(*vulneraility*)

  return *visualVul*

3.2 The Experimental System Flow of the Proposed device type identification Framework

3.2.1 Overview

Identifying IoT devices is the initial step in locating any harmful or unidentified IoT devices in the network [107]. Therefore, the device must have a reliable and efficient way to distinguish between approved and illegal IoT devices when they join the network. Different studies [3, 6, 74, 107–109] have suggested a number of methods for locating IoT devices in a network. To train a model in one lab and transfer the trained model to another lab for testing, we present a transferability strategy in this study effort based on the idea of transfer learning.

The proposed identification system will be installed in an IoT gateway, the entry point to which the IoT devices are linked, to offer the most significant level of safety and security for IoT devices in a network. The proposed system, which will be installed in the gateway, would continually watch over all incoming and outgoing network data to enforce the devices connecting to the network are genuine.

3.2.2 System Flow

This study proposes a transferability approach based on the underlying concept of transductive transfer learning by employing the use of several supervised ML techniques to train a model in one lab (source) and transfer the knowledge of the trained model to another lab (target) to enhance the learning process. Transductive transfer learning assumes that X_{train}, X_{test} are from the same distribution, D. The transferability approach put forward in this research uses datasets different from the data used for training. As a result, we presume that data can be derived from two distinct distributions, D^{source} and D^{target}. This particular type of transfer learning

works by explicitly assigning labels Y_{test}^{target} to test data X_{test}^{target} which is typically derived from D^{target}, with training data (X_{train}^{source} , Y_{train}^{source}) drawn from D^{source} [110].

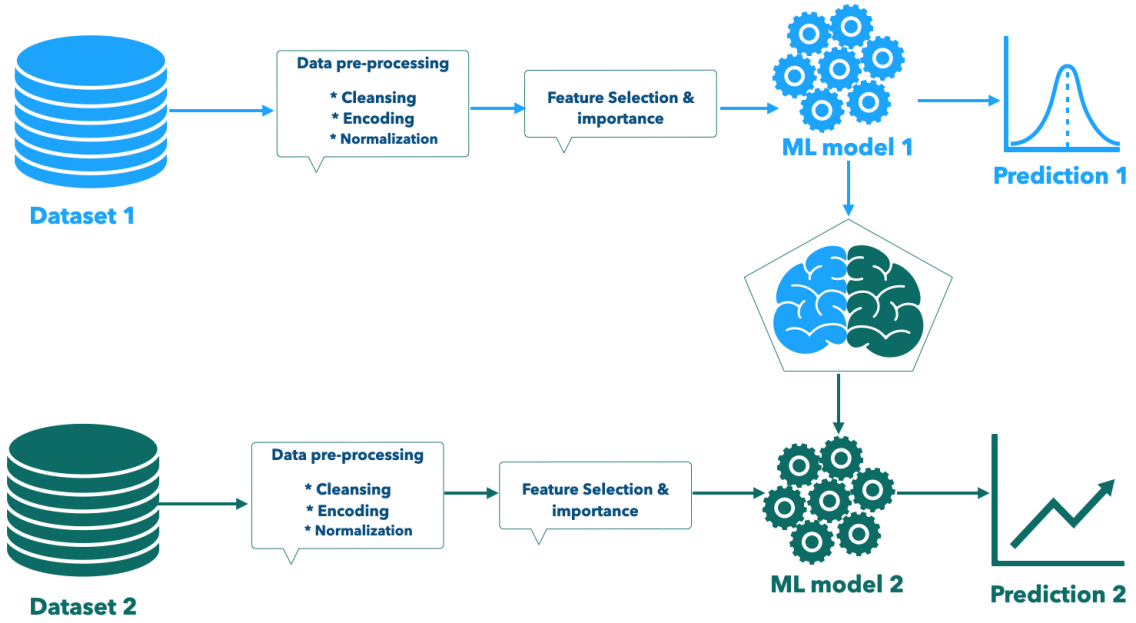


Figure 3.2: The Experimental System Flow of the Proposed device type identification framework based on the concept of transfer learning.

An overview of the proposed device type identification framework is analyzed. A detailed explanation of the feature selection and classifier modelling methods used is provided. Figure 3.2 depicts the proposed device type identification framework. The framework is divided into four modules: data preprocessing, feature selection, model training, and transferability of the classifier.

The following is a summary of each module:

1. *Data preprocessing:* During this stage, the feature set is synthesized statistically to ensure that the selected preprocessing approach is relevant to accommodate missing values. Before training, the preprocessed feature set is scaled into a regularized range using normalization. Additionally, the feature set is transformed and encoded for categorical values. To improve the overall system performance of our approach, we extracted a collection of the most statistically important features for our device identification and profiling framework.

2. *Feature selection module:* This phase is responsible for reducing the number of feature sets used for training the ML model with the goal of lowering computational cost and increasing the efficiency of the model. This happens by selecting a fraction of the important feature set and only using those relevant features to train the ML model.

3. *Model training:* This study employs a supervised ML technique to profile and identify the different types of devices in the network. Several ML models are investigated to determine the most effective method for our approach.

4. *Transferability of classifier:* This research employs a transferability approach for device type identification based on the concept of transductive transfer learning. The knowledge from the trained model is transferred to a different lab for testing and predicting the corresponding target label which in this research is the respective device types.

3.2.2.1 Data preprocessing

Data preprocessing is an important stage in any ML task. The use of unclean data to train your model will inevitably result in an incorrectly trained model that is irrelevant to the analysis [111]. All deficient data records for the measured variables in the dataset are handled at this stage. To scale our data into a regularized range, we used Z-score normalization [112], also known as standardization. Z-score normalization re-scales features in accordance with the standard normal distribution property with μ=0 and σ=1, where μ is the mean (average) and σ are the standard deviation from the mean. The formula below is used to compute the standard score or z-score of the sample.

$$z = \frac{x_i - \mu}{\sigma} \qquad [113] \qquad (3.1)$$

One of the common obstacles faced in any ML problem is the unequal number of samples in each class distribution. This scenario, known as the imbalanced problem, has an impact on the performance of many ML algorithms. A model trained with such a dataset is mostly biased towards the minority target label in the dataset due to their fewer number hence misclassifying the minority as belonging to a majority class. To prevent overfitting, an equal number of records are sampled from each class in the target variable during the training of the classifier.

3.2.2.2 The Proposed Feature selection

Finding a subset of input variables that can effectively represent all of the input data while minimizing the influence of noise or unnecessary attributes yet delivering accurate prediction results is the aim of feature selection [114]. Additionally, feature selection lowers computation demands, lessens the impact of the "curse of dimensionality", and enhances generalization performance [115]. In this study, the analysis of variance (ANOVA) feature selection method was employed. ANOVA is a data analysis technique in which one or more dependent variables are evaluated under different conditions identified that are determined by one or more measurements [116]. ANOVA presupposes that the variables have a Gaussian distribution and that there is a linear relationship between the feature and the target [117].
F-test is employed in ANOVA to determine whether there are any significant differences between the groups [118]. The F-ratio of an ANOVA will be close to 1 if there is no significant contrast between the groups and all variances are identical [119]. Equal variation across groups indicates that this attribute has no effect on response and cannot be used in model training. The SelectKBest module chooses the best

feature set for the training using the score values from the ANOVA module [120].

3.2.2.3 Model Training

In this phase, four supervised ML techniques will be used in training our model to identify the different device types. The ML models are; Decision tree (DT), Random Forest (RF), Gradient Boosting Classifier (GBC), Support Vector Machine (SVM), k-Nearest Neighbor (KNN), and MultiLayer Perceptrons (MLP) are used.

Decision tree (DT): DT classifier is a tree-based approach that describes any route starting from the root with a sample segregating series until a Boolean result is obtained at the leaf node [121]. The objective is to learn straightforward decision rules derived from the data characteristics to build a machine that predicts the value of a dependent variable [122]. DTs are consecutive models that intuitively integrate a series of straightforward tests that each match an attribute's absolute value to a criterion or its nominal value to a range of potential values [123]. Anthony J. et al. emphasize that a divide-and-conquer strategy is used in decision tree analysis for categorization. In big datasets, decision trees may be used to find characteristics and uncover trends that are crucial for discriminating and predictive modelling [124]. The capacity of DTs to deconstruct a complicated decision-making procedure into a series of smaller options, which results in a solution that is frequently clearer to understand, is its most crucial quality [125].

Random Forest (RF): The RF classifier is an ensemble classifier that builds trees using a substitution bagging method to select a slice of training data to predict outcomes [126]. RF incorporates a variety of randomized decision trees and averages their predictions. Each decision tree is built with its nodes divided into a random number of characteristics. The final classification decision based on all tree assignment probabilities is computed using arithmetic means. As a result, every decision tree makes a judgement on the target label class of new, unlabeled incoming data.

The most popular target class will then be selected. The method has demonstrated high efficiency in situations when the amount of variables is significantly more than the amount of observations [127]. In contrast to individual decision trees, Random Forest is widely recognized for its lack of sensitivity to parameterization and resistance to overfitting [128]

Gradient Boosting Classifier (GBC): Gradient Boosting (GB) constructs a group of shallow trees sequentially, with each tree learning from and improving upon the one before it. The GB approach theoretically increases prediction accuracy since it intelligently merges more trees by fixing errors in its prior base models [129]. The GB algorithm builds a new decision tree known as a weak learner at each iteration that best minimizes the loss function. The algorithm iterates until the user-specified optimum amount of iterations are achieved. The decision trees that were introduced to the model at earlier phases are not changed at each subsequent step. The model is strengthened in the areas where it underperforms by adapting decision trees to the deviation [130]. GM is incorporated with several regularization techniques to reduce the eventual overfitting [131].

Support Vector Machine (SVM): SVMs have grown in prominence over the past ten years as a result of their capacity to approach challenging, highly nonlinear tasks in a systematic, organized fashion while also minimizing the issues associated with overfitting on easier challenges [132]. SVM can accommodate extremely high-dimensional feature vectors since its effectiveness is not as much influenced by the size of classified vectors as it is by the performance of traditional classifiers. This will help with fault categorization as well as there may not need to be a strict restriction on the number of characteristics used as the foundation for problem analysis. SVM has risen to prominence and popularity in the ML field because of its superior generalization capabilities when compared to more established techniques like neural networks [133].

K-Nearest Neighbor (KNN): KNN has been widely used in classification problems

[134]. The distance function, which calculates how different or similar two instances are, is the foundation of KNN. The distance function is frequently the typical Euclidean distance between two instances [135]. In order to predict the target label of the new point using the outcomes of these training samples, nearest neighbour algorithms first find the training samples that are closest to the new point [136]. Since KNN is straightforward, efficient, and resilient, it has been frequently utilized for a long period.

MultiLayer Perceptron (MLP): MLP is a type of neural network. A nonlinear translation between an input vector and an output vector is described as a network of straightforward linked neurons, or nodes, that make up a multiLayer perceptron [137]. The nodes are interconnected by weights and output signals that are functions of the aggregate of the node's inputs, as adjusted by a straightforward nonlinear transfer, or activation function. It is said that MLPs are fully coupled. Every node in the subsequent and prior levels is linked to every other node. The linking weight scales and feeds forward the output of a node. Thus, "feed-forward neural network" suggests a processing orientation for input. The ability of MLP to generalize to new, untested input data is an advantage.

3.2.2.4 Transferability of Classifiers

In this study, the IoT traffic data was collected from three completely different labs located in distant countries, denoted as Lab A - CIC lab (source), Lab B - Mon(IoT)r lab (target), and Lab C - IoT Sentinel lab (target). The dataset for Lab B and C is to verify the proposed transferability approach for device profiling and identification. Additionally, each of these labs has a completely different network configuration and is populated with different devices. However, there were some devices that were common in the source and each of the target labs and not a single common device in all three labs. It is important to highlight that our work may be impacted by the

various privacy and security compliance legislation of these different labs situated in a completely different locations from each other, such as GDPR General Data Protection Regulation (GDPR) for the European Union (EU), and Federal Trade Commission (FTC) for the US. Lab A which is used for training is located in Canada; Lab B is located in the US and Lab C is located in Europe.

Using the input variables from each of the target sources and the trained model from the source domain, the objective at this stage is to assign labels, which are the device types, to each of these target feature sets. As a result, the trained model from the source will be used for testing rather than retraining the model in the target domain. We compared the performance of classifiers when they were trained on IoT device data acquired in one lab and then evaluated data collected from other IoT devices as part of our analyses of the transferability using a range of various IoT device types from each lab.

3.3 Vulnerability Assessment Framework Proposed

3.3.1 Overview

Our work does not end after the IoT device type identification by the ML classifier. We further analyze the possible vulnerability of the predicted device type. The vulnerability assessment implemented in this work is in four steps

1. Search all three databases (NVD, Vulners, and IBM X-Force) with the predicted device type by the classifier.

2. Harvest each individual keyword search into respective local databases.

3. Standardize results from the harvested search for Vulners, and IBM X-Force to have the same attributes as that of NVD.

4. Aggregate the standardized results per device type from all three databases.

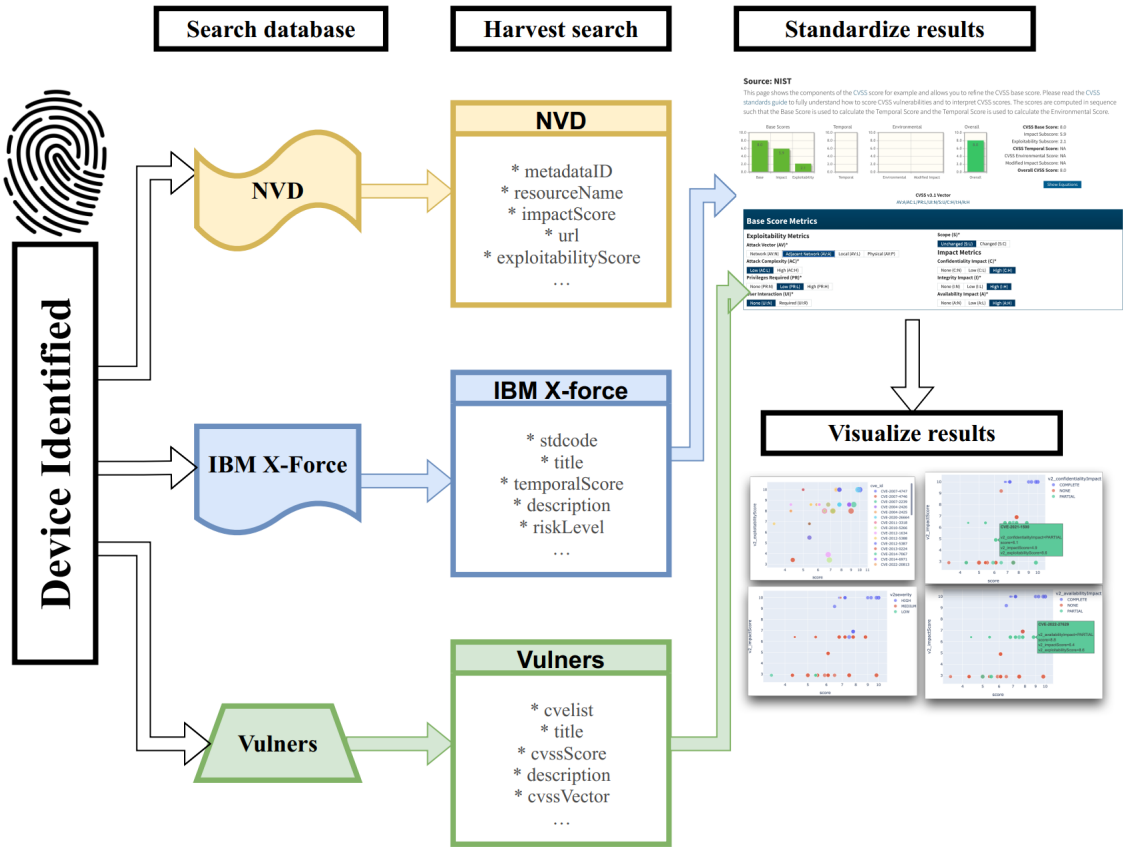


Figure 3.3: Vulnerability assessment overview.

3.3.2 Vulnerability Database

In this section, we describe the four well-respected publicly available data sources leveraged in this study.

3.3.2.1 National Vulnerability Database (NVD)

NVD is the de facto standard vulnerability information source [138]. The database is managed and regulated by the National Institute of Standards and Technology (NIST). NIST provides an ongoing analysis of Common Vulnerabilities and Exposures (CVEs) and assigns Common Vulnerability Scoring System (CVSS) base

metrics for each vulnerability, and will update a score if more information becomes available. Ongoing analysis and scoring help NVD users understand the potential severity of each issue, and help users to prioritize vulnerability management activities. NIST works directly with vendors and researchers to improve the quality of the published data and to provide the public with accurate scoring data [139].

Definitions Before we go any further into the overview of the NVD database, certain commonly used terminologies need to be explained. The definition outlined below is stated by the NIST [139].

1. Common Vulnerabilities and Exposures (CVE) - "*a weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability. Mitigation of the vulnerabilities in this context typically involves coding changes, but could also include specification changes or even specification deprecations (e.g., removal of affected protocols or functionality in their entirety)*";

2. Common Vulnerability Scoring System (CVSS) – "*a set of metrics for communicating the characteristics and impacts of IT vulnerabilities. NVD analysts provide a baseline analysis of the potential severity of a given vulnerability, based on publicly available information from researchers and vendors*";

3. Common Platform Enumeration (CPE) – "*a software enumeration that assists in communicating the hardware and software platforms that may be affected by a given vulnerability. Use of CPE enables users (and automated tools) to quickly identify vulnerabilities that may affect systems under their purview*";

4. Common Configuration Enumeration (CCE) – "*a set of unique platform-based identifiers associated with security configuration statements. CCE supports*

evaluation of configuration compliance with a given security configuration baseline"; and

5. Common Weakness Enumeration (CWE) – "*a taxonomy for identifying the common sources of software flaws (e.g., buffer overflows, failure to check input data). Where practical, NVD associates a given CVE vulnerability to the underlying CWE weakness*".

It is worth noting that the official CPE dictionary is managed by the NVD, while the MITRE organization is in charge of the CVE Program. Because NVD is based on the CVE List and is fully synchronized with it, all revisions to CVE are immediately reflected in NVD. On July 13, 2022, the NVD ceased producing fresh data for CVSSv2. Although the NVD would not further officially generate CVSSv2 for incoming CVEs, current CVSSv2 data will stay in the repository.

The NVD database is made accessible through XML and JSON feeds. The complete archival data bundle can also be downloaded commencing in 1999. Moreover, a search function has been added to the NVD website. The NVD is aiming to shift any existing data feed clients to more recent Application Programming Interfaces (APIs) as it prepares to discontinue its outdated data feeds in late 2023. NIST acknowledges there are several benefits to using APIs, which have been the tried-and-true approach to web-based automation for more than a decade. Related to archival factors and a lengthy period of data collection, the data is neither perfectly uniform nor error-free. There are about 900 entries without a CPE identification, earlier entries omit certain information for instance vulnerability score, the URL in the reference field is often obsolete, and many records have CPEs that are contradictory or do not appear in the CPE dictionary (approximately 100,000 CPEs) [140].

In this work, an assumption made is that the NVD is considered one of the most reputable data sources based on the following factors;

- It is managed by a trusted U.S. governmental agency (The National Institute of Standards and Technology, NIST).
- It is updated on the regular basis - yearly JSON data feed is once per day while the "recent" and "modified" feeds are updated every two hours. Furthermore, each feed is updated nightly, but only if the content of that feed has changed.
- The maintenance and sponsorship of the system are by reputable agencies; the U.S. Department of Homeland Security (DHS) and the Cybersecurity and Infrastructure Security Agency (CISA).

3.3.2.2 Vulners

Vulners [138] gathers data from a myriad of sources about cyber breaches including ICS-CERT advisories by vendor, cybersecurity-focused posts, also a majority coming from vulnerability databases. The current service has about 131 sources from NVD, JVNDB, US-CERT and ZDI. The platform is constantly updated with new sources and a revision of the current source. Hence the data validity and integrity can differ as there is sometimes omitted information from the original entry. Vulners offers an API as a means to have the data locally available to manage and maintain. Vulners, in contrast to the other two sources, is a premium service with restricted free non-commercial use of an API and free use of the website's search engine. Vulners only permits 100 API calls per month; if this amount is exceeded, users are asked to upgrade their service.

3.3.2.3 IBM X-Force

The IBM X-Force database is gradually working its way to becoming one of the largest databases for vulnerability assessment and threat detection. Developers and researchers at IBM spent endless hours putting this together. Making IBM incorporate this data into their services and products. X-Force is a result of information

gathered from different sources like the internet, research, and IBM Internet Security Systems software, among other related software. The IBM X-Force database has approximately 40,000 distinct vulnerabilities. The limitation is that IBM X-Force allows 5,000 records per month and maximum of 200 records displayed per every API call.

3.3.3 Vulnerability assessment overview

Following successful device type identification by our ML classifier, the predicted device type is passed as a keyword to each of the three vulnerability databases stated in subsection 3.3.2. The following step entails gathering information from the three sources and storing it in their respective local database storage. The information is then standardized in the following step. The standardization works on Vulners and IBM X-Force search results. This works by taking the CVE-ID from each search result and passing it to the NVD database to retrieve the JSON response for that specific CVE. The NVD database is assumed to be the "master source", with most or all CVEs approved and verified. The steps are depicted in Figure 3.3.

3.3.3.1 Vulnerability keyword search

The classifier predictions are passed to the API endpoints as keywords, and a JSON response is returned from the API based on the keyword predicted by the classifier. In order to obtain vulnerabilities, an API call is used to access the IBM X-Force and Vulnerability database. In contrast, the NVD data stream [1] is scraped for each individual year. To synchronize the local database and the NVD website's feed, a cron job is executed to update the data feed every time it is updated on the NVD website. The scraped feeds are in a zipped JSON format, hence a script is run every year to unzip them. Our NVD database is constructed by combining the

[1] https://nvd.nist.gov/vuln/data-feeds

unzipped data feeds for every year together. This data is searched for any potential vulnerabilities related to the predicted device type from the classifier.

3.3.3.2 Vulnerability harvesting

Three vulnerability databases are searched for each of the five predicted device types (Audio, Appliance, Camera, Home Automation, and Smart Hub). As a result, there will be 15 iterations of $3^{vd} * 5^{pd}$, where 3^{vd} stands for the three vulnerability databases and 5^{pd} for the five anticipated device kinds. We identified a small number of IoT device vulnerabilities based on a preliminary examination of the NVD data feed. As a result, rather than the actual devices themselves, our vulnerability evaluation concentrated on the device types. Some modifications were made in order to generate vulnerability data for the relevant device kinds.The predicted device types and the keyword searched are: *Audio*{Audio and Speaker}, *Camera*{Camera and Video}, *Home Automation*{Plugs, Bulb, and Lamp}, *Appliances*{Brewer, Kettle, Microwave, Vacuum, Washer, Dryer}, and *Smart Hub*{Hubs, Smarthub, and Smartthings}. This was to ensure there is a comprehensive list of vulnerabilities that encompasses the individual devices that belong to each device type category. In order to meet the requirements for storage, the obtained search results were normalized and synthesized before being saved in a PostgreSQL database [2].

3.3.3.3 Vulnerability Standardization

The attributes and representation of the search results from the various databases varied. The gathered search results were made more uniform and reliable by standardizing them. The NVD is presumed to be a more reliable source for vulnerability assessment [141], yet the other sources discovered vulnerabilities that NVD had missed and vice versa. Using the NVD database, the data obtained from Vulners

[2]https://www.postgresql.org/

and IBM X-Force was standardized. Parsing the CVEID from the IBM X-force and Vulners database to the NVD database to retrieve the results produced is one of the data processing operations involved in this stage. The standardized search results are stored in a local database.

3.3.3.4 Vulnerability Aggregation

In this step, the standardized vulnerabilities that were received from each data source are combined into a single device type. The NVD data is synchronized with the gathered standardized Vulners and IBM X-Force data. For example in order to represent the *Camera* for further research, NVD_{camera}, $Vulners_{camera}$, and $XForce_{camera}$ are combined into one vulnerability data source. The other device types are subject to this aggregation.

3.3.4 NVD Metrics definitions

There are certain metrics in the JSON response from the keyword search below that are self-explanatory. Consequently, we will choose and describe the key attributes that we employed in our research.

Listing 3.1: NVD JSON response from data feed extracted.

```
{{
  "CVE_data_type" : "CVE",
  "CVE_data_format" : "MITRE",
  "CVE_data_version" : "4.0",
  "CVE_data_numberOfCVEs" : "11524",
  "CVE_data_timestamp" : "2022-08-23T07:00Z",
  "CVE_Items" : [ {
    "cve" : {
      "data_type" : "CVE",
      "data_format" : "MITRE",
      "data_version" : "4.0",
      "CVE_data_meta" : {
        "ID" : "CVE-2022-0001",
        "ASSIGNER" : "secure@intel.com"
      },
      "problemtype" : {
        "problemtype_data" : [ {
          "description" : [ {
```

```
        "lang" : "en",
        "value" : "NVD-CWE-noinfo"
      } ]
    } ]
  },
  "references" : {
    "reference_data" : [ {
      "url" : "http://www.openwall.com/lists/oss-security/2022/03/18/2",
      "name" : "[oss-security] 20220318 Xen Security Advisory 398 v2 - Multiple speculative security issues
           ",
      "refsource" : "MLIST",
      "tags" : [ "Mailing List", "Third Party Advisory" ]
    }, {
      "url" : "https://www.oracle.com/security-alerts/cpujul2022.html",
      "name" : "N/A",
      "refsource" : "N/A",
      "tags" : [ "Patch", "Third Party Advisory" ]
    } ]
  },
  "description" : {
    "description_data" : [ {
      "lang" : "en",
      "value" : "Non-transparent sharing of branch predictor selectors between contexts in some Intel(R)
           Processors may allow an authorized user to potentially enable information disclosure via local
           access."
    } ]
  }
},
"configurations" : {
  "CVE_data_version" : "4.0",
  "nodes" : [ {
    "operator" : "OR",
    "children" : [ ],
    "cpe_match" : [ {
      "vulnerable" : true,
      "cpe23Uri" : "cpe:2.3:h:intel:celeron_j4005:-:*:*:*:*:*:*:*",
      "cpe_name" : [ ]
    },  {
      "vulnerable" : true,
      "cpe23Uri" : "cpe:2.3:h:intel:pentium_silver_n5000:-:*:*:*:*:*:*:*",
      "cpe_name" : [ ]
    } ]
  }, {
    "operator" : "OR",
    "children" : [ ],
    "cpe_match" : [ {
      "vulnerable" : true,
      "cpe23Uri" : "cpe:2.3:a:oracle:communications_cloud_native_core_binding_support_function
           :22.1.3:*:*:*:*:*:*:*",
      "cpe_name" : [ ]
    }]
  } ]
},
"impact" : {
  "baseMetricV3" : {
    "cvssV3" : {
```

```
        "version" : "3.1",
        "vectorString" : "CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N",
        "attackVector" : "LOCAL",
        "attackComplexity" : "LOW",
        "privilegesRequired" : "LOW",
        "userInteraction" : "NONE",
        "scope" : "CHANGED",
        "confidentialityImpact" : "HIGH",
        "integrityImpact" : "NONE",
        "availabilityImpact" : "NONE",
        "baseScore" : 6.5,
        "baseSeverity" : "MEDIUM"
      },
      "exploitabilityScore" : 2.0,
      "impactScore" : 4.0
    },
    "baseMetricV2" : {
      "cvssV2" : {
        "version" : "2.0",
        "vectorString" : "AV:L/AC:L/Au:N/C:P/I:N/A:N",
        "accessVector" : "LOCAL",
        "accessComplexity" : "LOW",
        "authentication" : "NONE",
        "confidentialityImpact" : "PARTIAL",
        "integrityImpact" : "NONE",
        "availabilityImpact" : "NONE",
        "baseScore" : 2.1
      },
      "severity" : "LOW",
      "exploitabilityScore" : 3.9,
      "impactScore" : 2.9,
      "acInsufInfo" : false,
      "obtainAllPrivilege" : false,
      "obtainUserPrivilege" : false,
      "obtainOtherPrivilege" : false,
      "userInteractionRequired" : false
    }
  },
  "publishedDate" : "2022-03-11T18:15Z",
  "lastModifiedDate" : "2022-08-19T12:28Z"
 },
}
```


- **CVE ID:** Is a unique identifier to acquire the details of vulnerabilities.
- **Assigner:** The entity that assigned the vulnerability. This can be MITRE or authorized organizations known as CVE Numbering Authorities (CNAs).
- **Problem Type:** Enables the use of a universal understanding of discussion for addressing, identifying, and resolving the root causes of web application

security vulnerabilities that may be present in network infrastructure, code, or design. CWE-352, for instance, stands for Cross-Site Request Forgery (CSRF).

- **References:** Additional information about a vulnerability can be found in URLs, which may not be present in the CVE description. References are given resource tags, such as press/media releases, vendor advisories, third-party advisories, technical papers, and VDB entries, among others. Users utilize these tags to easily classify the sort of information that each reference includes.

- **Description:** Functions as a synopsis of the vulnerability and might contain details about the affected product, consequences, attack method, weakness, or other pertinent technical data.

- **Configurations:** It is a holding structure for a collection of nodes, each of which has CPE Name Match Criteria. The product description is expressed as an organized, logical AND-OR expression since the extent of the vulnerability may be complex—it may, for instance, refer to a specific software version running on a specific hardware platform. There are three sorts of configurations: Basic, Running On/With, and Advanced.

  - Basic: A single node that houses a single set of matching criteria.
  - Running On/With: A collection of nodes with compatibility requirements for both susceptible and unvulnerable matches.
  - Advanced: A complicated arrangement of nodes with several enumerations.

- **Vector String:** Represent the values that were utilized to calculate the score represented as condensed text.

- **Attack Vector (AV):** Acts as an indication of the level of access required for an adversary to exploit the vulnerability. The four attack vectors into which a vulnerability can be divided are as follows.
  - Network (N): An exploitable flaw that may be accessed through an OSI layer 3 (the network layer) component tied to the network stack.
  - Adjacent Network (A): An attack is restricted to a single common physical or logical network component and cannot be carried out at the network layer.
  - Physical (P): Access necessitates that the attacker physically interacts with or alters the weak link, for as via connecting a peripheral device to a system.
  - Local (L): A flaw that may be taken advantage of without the need for a network connection and can be read, written, or executed.
- **Attack Complexity (AC):** This statistic describes the external circumstances that must occur for the vulnerability to be exploited. This most frequently relates to either necessary user engagement or certain underlying system setups. This can either be Low or High.
  - Low (L): There are no mitigating situations or specialized access requirements. An attacker can anticipate consistent success while attempting to compromise the weak link.
  - High (H): Attack success is contingent on factors outside of the attacker's control. For an exploit to be effective, the adversary must put up a certain level of preparation work.
- **Privileges Required (PR):** This measure specifies the degree of access or privileges that an adversary must possess in order to successfully exploit.

  - None (N): The attack cannot be carried out without any specific permissions or privileges.
  - Low (L): The attacker needs just the bare minimum of user privileges to take advantage of the vulnerability.
  - High (H): Successful exploit requires administrative or comparable authentication privileges.

- **User Interaction (UI):** Refers to the amount of work or participation that must be done by users beyond the perpetrator in order to exploit the vulnerability.
  - None (N): The system may be taken advantage of without any user involvement.
  - Required (R): Exploit necessitates user interaction, such as persuading a user to click a button in an email, in order to exploit the vulnerability.

- **Scope (S):** Evaluates whether a vulnerability may have a cascade effect on other systems or parts.
  - Unchanged (U): The domestic security body is the only entity that may be negatively impacted by the exploited vulnerability.
  - Changed (C): A vulnerability that has been exploited may have repercussions on other systems.

- **Confidentiality Impact (CI):** Ensure that only authorized users have full rights to the intended data and disclosure of confidential information.
  - None (N): The affected component does not experience any confidentiality losses.

  - Low (L): Exposure to some restricted data is gained, but the adversely affected component does not suffer a direct, significant loss as a result of the data leak.
  - High (H): Complete secrecy is lost, exposing to the adversary any information contained inside the damaged component.

- **Integrity Impact (I):** Is the question of determining if the secured data has been altered or manipulated in any manner.
  - None (N): No integrity is compromised inside the affected component.
  - Low (L): Data manipulation is conceivable, but it doesn't have a significant, immediate effect on the component that's being affected.
  - High (H): A full compromise of integrity or security has occurred.

- **Availability Impact (A):** The corrupted machine will be unable to complete its intended purpose, making information unavailable when it is needed.
  - None (N): Within the damaged component, availability is unaffected.
  - Low (L): The availability of resources is interrupted or performance is relatively poor.
  - High (H): There is a prolonged or permanent loss of availability, allowing the attacker to completely block accessibility to the services in the damaged component.

- **Access Vector (AV):** This indicator shows the method of exploiting the vulnerability. The vulnerability score increases with the potential distance at which an attacker can access a host.

- Network (N): The adversary does not necessitate local network access or local access because the susceptible component is tied to the network stack.
- Adjacent Network (A): The collision domains of the susceptible component must be accessible to the attacker in order to get neighbouring network access.
- Local (L): Physical access to the targeted system is a prerequisite for gaining local access.

- **Access Complexity (AC):** This statistic evaluates the difficulty of the attack needed to take advantage of the weakness after an adversary gained a foothold in the target system.
  - High (H): Advanced access circumstances can be created by using social engineering techniques, enhanced privileges, or inactive susceptible configuration.
  - Medium (M): The access requirements call for a moderate bit of social engineering, an untrustworthy level of authorization, and data collecting.
  - Low (L): Complicating factors or specialized access requirements don't exist.

- **Authentication (Au):** This metric counts how many attempts an attacker has to log in to a target to take advantage of a weakness.
  - Multiple (M): Exploit necessitates two or more authentication attempts, regardless of the same credentials usage.
  - Single (S): A system log-in is necessary for the vulnerability to be exploited.
  - None (N): Exploiting the vulnerability doesn't need authentication.

- **Severity:** CVSS is a technique that provides a qualitative indicator of severity. NVD provides qualitative severity ratings of Low, Medium, and High for CVSS v2.0 base score ranges in addition to the severity ratings for CVSS v3.0 as they are defined in the CVSS v3.0 specification in Table 3.1.

Table 3.1: NVD CVSS v2.0 and CVSS v3.0 base score ranges and severity ratings.

CVSS v2.0 Ratings		CVSS v3.0 Ratings	
Severity	**Base Score Range**	**Severity**	**Base Score Range**
		None	0.0
Low	0.0-3.9	Low	0.1-3.9
Medium	4.0-6.9	Medium	4.0-6.9
High	7.0-10.0	High	7.0-8.9
		Critical	9.0-10.0

- **Exploitability Score:** This score is generated using base exploitability measurements (AV, AC, PR, UI)
- **Impact Score:** Base Impact data are employed to calculate the Impact score.
- **Base Score:** This metric is used to describe the fundamental traits of each vulnerability.

NVD has transitioned from version 2 to version 3 and even version 3.1 lately. The various measures utilized for the scoring system are affected by the different versions. The various metrics that changed in line with version evolution are displayed in Table 3.2.

3.4 Closing Remarks

We described the different modules used in our research in this chapter. The device identification, vulnerability assessment, and visualization processes were broken down into their respective components. The many modules (data preprocessing,

Table 3.2: Metrics attributes for different NVD versions.

	Access Vector (AV)	Access Complexity (AC)	Confidentiality Impact (C)	Integrity Impact (I)	Availability Impact (A)
CSSV Version 2.0 Ratings	Network	Low	None	None	None
	Adjacent Network	Medium	Partial	Partial	Partial
	Physical	High	Complete	Complete	Complete
	Local				

	Attack Vector (AV)	Attack Complexity (AC)	Confidentiality Impact (C)	Integrity Impact (I)	Availability Impact (A)
CSSV Version 3.0 Ratings	Network	Low	None	None	None
	Adjacent Network	High	Low	Low	Low
	Local		High	High	High
	Critical				

feature selection, model training, and transferability of classifier) are equally discussed in each of the device identification components, as well as the suggested transferability concept's underpinnings. Keyword search, vulnerability, harvesting, standardization, and aggregation are the procedures involved in arranging the data obtained from the three separate vulnerability databases (NVD, IBM X-Force, and Vulners). The employed vulnerability metrics are explained.

We will share our findings from the implementation and experiments we ran using the methodology described in this chapter in the following chapter.

Chapter 4

Implementation and Experimental Results

4.1 Overview

This section evaluates the data collection, data sampling or reduction technique used in this research work as well as the feature selection technique. Additionally, we describe our approach in the training and testing of the machine learning models and the performance metrics used on the training dataset. Analyses of trends and patterns in the NVD data feed stored locally were conducted. The vendor of the predicted device type and the device predicted by the ML model are both evaluated and visualized at the conclusion.

4.2 Data Acquisition

The datasets utilized for the evaluation of the transferability approach for device type identification based on the idea of transfer learning are detailed in this section.

4.2.1 CIC IoT Dataset 2022

The CIC IoT Dataset 2022 [1] was generated by capturing the network traffic by running the following experiments:

1. *Power:* Each device was powered on individually and started capturing the network traffic separately.

2. *Idle:* During what is termed as idle time, the whole network traffic was recorded from late at night to early in the morning. The entire lab was evacuated during this time, and no human interactions ensued.

3. *Interactions:* Every conceivable functionality of IoT devices has been extracted, along with the related network activity and transmitted packets for each functionality or activity.

4. *Scenarios:* The network activity inside a smart house was simulated using a collection of devices in six distinct types of scenario experiments. These tests were conducted to observe how several devices functioned simultaneously.

5. *Active:* All day long, the entire network communications were captured. Users were permitted to use the lab to generate both active and passive traffic for devices interacted with.

6. *Attacks:* Two distinct attacks, Flood and RTSP-Brute Force were conducted on certain devices while simultaneously recording the network data generated by these attacks.

Table 4.1 shows devices in the CIC IoT dataset 2022 sampled for our research work [1].

[1] https://www.unb.ca/cic/datasets/iotdataset-2022.html

Table 4.1: Devices in the CIC lab and the corresponding device type sampled for our experiment.

No.	Device Name	Device Type
01.	Amazon Echo Dot	Audio
02.	Amazon Echo Spot	Audio
03.	Amazon Echo Studio	Audio
04.	Google Nest Mini	Audio
05.	Sonos One	Audio
06.	Amcrest Camera	Camera
07.	ArloQ Camera	Camera
08.	Borun Camera	Camera
09.	DLink Camera	Camera
10.	HeimVision Camera	Camera
11.	HomeEye Camera	Camera
12.	Luohe Camera	Camera
13.	Nest Camera	Camera
14.	Netatmo Camera	Camera
15.	SimCam	Camera
16.	Arlo Base Station Camera	Camera
17.	Amazon Plug	Home Automation
18.	Globe Lamp	Home Automation
19.	Gosund Plug	Home Automation
20.	Heimvision Lamp	Home Automation
21.	Teckin Plug	Home Automation
22.	Yutron Plug	Home Automation
23.	D-Link Water sensor	Home Automation
24.	Philips Hue	Smart Hub
25.	Ring Basestation	Smart Hub
26.	Eufy Homebase	Smart Hub
27.	Atomi coffee maker	Appliance
28.	iRobot roomba	Appliance
29.	Smart board	Appliance

4.2.2 IMC 2019 Payload Dataset

Network traffic from 81 devices recorded during manual and automated trials makes up the dataset [2]. 35 were purchased from UK stores (UK devices) and deployed in a UK testbed, whereas 46 were purchased from US stores (US devices). There are 26 identical devices with the same model name in each of the two labs. Power,

[2]https://github.com/NEU-SNS/intl-iot

Table 4.2: Devices in the US testbed and the corresponding device type sampled for our experiment.

No.	Device name	Device type
01.	Amcrest Cam	Camera
02.	Blink Cam	Camera
03.	Lefun Cam	Camera
04.	Luohe Cam	Camera
05.	Microseven Cam	Camera
06.	Wansview Cam	Camera
07.	Yi Cam	Camera
08.	Sengled hub	Smart Hub
09.	Smartthings	Smart Hub
10.	Blink Hub	Smart Hub
11.	Insteon hub	Smart Hub
12.	Wink 2 hub	Smart Hub
13.	Philips Hue	Smart Hub
14.	Xiaomi hub	Smart Hub
15.	TP-Link Bulb	Home Automation
16.	Amazon Echo Dot	Audio
17.	Amazon Echo Spot	Audio
18.	Amazon Echo Plus	Audio
19.	Google Home Mini	Audio
20.	Behmor Brewer	Appliance
21.	GE Microwave	Appliance
22.	Samsung Washer	Appliance
23.	Xiaomi Cleaner	Appliance
24.	Xiaomi Rice Cooker	Appliance

interaction, idle, and uncontrolled experiments were the four types of experiments that were carried out [2].

- *Power* experiments involve turning on the device and capturing the network traffic.
- *Interaction* experiments capture the network transfer while devices are actively in use.
- *Idle* experiments record the network traffic of an IoT device not in use.
- *Uncontrolled* experiments allowed 36 research participants the opportunity to

use IoT devices for their intended use in a studio apartment in the US lab only.

For our analysis, 24 devices in the US testbed were sampled, as indicated in Table 4.2.

4.2.3 IoT Sentinel Dataset

IoT Sentinel dataset[3] consists of a selection of daily-used IoT devices from the European market to conduct this research. The network configuration took into account both WiFi and Ethernet-connected devices that joined directly to the network as well as ZigBee and Z-Wave devices connected to the network using an intermediary hub [73]. Using an app given by the vendor, the IoT device was configured to connect directly over WiFi or Ethernet, and the network's WiFi credentials were sent to the device via this connection. After that, the device resets and uses the supplied

Table 4.3: Devices from the IoT Sentinel dataset and the corresponding device type sampled for our experiment.

No.	Device name	Device type
01.	D-Link Cam	Camera
02.	D-Link Day Cam	Camera
03.	EdimaxCam 1	Camera
04.	EdimaxCam 2	Camera
05.	EdnetCam 1	Camera
06.	EdnetCam 2	Camera
07.	D-Link Switch	Smart Hub
08.	Hue Switch	Smart Hub
09.	Ednet Gateway	Smart Hub
10.	Max Gateway	Smart Hub
11.	WeMo Insight Switch 1	Smart Hub
12.	WeMo Insight Switch 2	Smart Hub
13.	WeMo Switch 1	Smart Hub
14.	WeMo Switch 2	Smart Hub
15.	Lightify	Smart Hub
16.	HomeMatic plug Switch	Smart Hub
17.	Edimax Plug 1	Home Automation
18	Edimax Plug 2	Home Automation

[3]https://research.aalto.fi/en/datasets/iot-devices-captures

credentials to access the network. The authors concentrated on tracking the transfer or intermediary traffic produced by the IoT gateway for the ZigBee and Z-Wave devices. The normal device setup procedure was carried out 20 times for each tested device in order to collect enough fingerprints.

Table 4.3 displays the devices from the IoT Sentinel dataset which was procured in the European market and the corresponding device type sampled for our experiment.

4.3 Experiments

This section evaluates the data collection, data sampling or reduction technique used in this research work as well as the feature selection technique. Additionally, we describe our approach in the training and testing of the machine learning models and the performance metrics used on the training dataset.

4.3.1 Data collection

CIC IoT dataset 2022 [1] functions as the training data (source) while IMC 2019 Payload dataset [2] and IoT Sentinel dataset [73] are used to evaluate our proposed framework as the test data (target). 48 features were extracted from each of the pcap files of the different datasets with output as a CSV file. Table 4.7 contains the feature set that makes up the generated CSV files, as well as descriptions and definitions of each feature. The collection of features created for the source domain and the target domain are identical. The resulting CSV files are used as input for the detailed subsection 3.2.2.1 data preprocessing stage.

4.3.1.1 Data sampling

An unbalanced dataset has a detrimental effect on the precision of class predictions in most classification problems [142]. This skewness was addressed in this study by

Table 4.4: Feature vector name in the dataset, readable name and description.

	Column in dataset	Readable column name	Description
1	L4_tcp	TCP	The transmission control protocol is used to connect network devices to the internet (ConnectionOriented).
2	L4_udp	UDP	User datagram protocol is used for establishing low-latency and loss-tolerating connections (Connectionless).
3	L7_http	HTTP	Hypertext transfer protocol is used to deliver data on the world wide web (WWW)
4	L7_https	HTTPS	Secure version of HTTP
5	ip_padding	IP Padding	Padding is basically used to make sure that the IP packet header has a length that is a multiple of 32 bits
6	ip_ralert	Router Alert	Used to inform a router that the current IP packet has some peculiarities that should be studied before it is forwarded on Service
7	port_class_src	Source Port number class	Port number class, No port = 0, well known port (0,1023) = 1, registered port (1024,49151)=2,dynamic port (49152,65535) =3
8	port_class_dst	Destination Port number class	
9	pck_size	IP packet size (length)	Entire packet size in bytes including IP header embedded protocol headers and data a 16-bit field.
10	pck_rawdata	Raw Data	Data that has not been processed for use
11	ip_dst_new	Destination IP address	Destination IP address of the IoT Device
12	ethernet_frame_size	Ethernet Frame size	Type of the Ethernet frame used by the device
13	ethernet_frame_type	Ethernet Frame type	The frame size of a standard Ethernet frame is the sum of the Ethernet header, the payload, and the Frame Check Sequence (FCS) field.
14	ttl	TTL for TCP and UDP packets	Time to live (TTL) or hop limit is the lifespan or lifetime of a packet.
15	total_length	Total Packet Length	Packet size including header
16	protocol	Protocol (L3, L4,L7)	Protcols used in layerr 3, 4 and 5
17	source_port	Source Port	Source Port of the device
18	dest_port	Destination Port	Destination Port of the device
19	DNS_count	DNS count	Number of DNS request
20	NTP_count	NTP Count	Number of NTP requests
21	ARP_count	ARP Count	Number of ARP requests
22	cnt	Number of Protocols (Protocol count)	Number of protocols used by the device
23	L3_ip_dst_count	Number of destinations IP	Number of destination IP addresses accessed by the IoT device
24	most_freq_d_ip	Most Frequent Destination IP address	Destination IP address that is visited most frequently
25	most_freq_prot	Most Frequent Protocol	The protocol that is used most frequently
26	most_freq_sport	Most Frequent Source Port	source port number used more frequently
27	most_freq_dport	Most frequent destination port number	Port number that is most frequently accessed by the IoT device.
28	sum_et	Sum of packet size	Aggregate of the data
29	min_et	Minimum packet	Minimum ip packet size in in 20 packets
30	max_et	Maximum packet	Maximum ip packet size in 20 packets
31	med_et	Median packet	The median of ip packet size in 20 packets
32	average_et	Average packet	Average length of the ip packet in 20 packets
33	var	Variance packet	The measurement of the spread between numbers in 20 packets
34	q3	third quartile packet	The central point that lies between the median and the highest number of distribution
35	q1	first quartile packet	The middle number that falls between the smallest value of the data set and the median.
36	iqr	inter-quartile packet	The amount of spread in the middle 50% of 20 packets
37	sum_e	Sum of frame	Aggregate of the data
38	min_e	Minimum ethernet	Minimum ethernet size in in 20 packets
39	max_e	Maximum ethernet	Maximum ethernet size in 20 packets
40	med	Median frame	The median of ethernet size in 20 packets
41	average	Average frame	Average length of the ethernet frame in 20 packets
42	var_e	Variance frame	The measurement of the spread between numbers in 20 packets
43	q3_e	third quartile frame	The central point that lies between the median and the highest number of distribution.
44	q1_e	first quartile ethernet	The middle number that falls between the smallest value of the data set and the median.
45	iqr_e	inter-quartile frame	the amount of spread in the middle 50% of 20 packets
46	epoch_timestamp	epoch timestamp of packet	The time stamp the specific packet was recorded
47	inter_arrival_time	inter arrival time	The time between the current and previous packet
48	time_since_previously_displayed_frame	time since previously displayed frame	The time between the current packet and the first packet that was captured.

randomly selecting an equal number of records for each class to guarantee that each distribution class had an equal probability of being chosen. While the issue of class imbalance is solved here, we know that for our model to generalize properly, we need a large diverse training set. Trimming the number to 1635 for each class while the maximum is 191,812 is Table 4.5 highlights the data sampled for each class in our training dataset.

The reason there are fewer entries in the "Appliance" category is because there are fewer packets that they produce. Among the devices in this category that are used

Table 4.5: The number of records for each device type in the CIC dataset [1] after data preprocessing and the the number of sampled records used for training the ML model.

No.	Device Type	Total no. of records
1	Camera	191812
2	Audio	19580
3	SmartHub	12711
4	HomeAutomation	8593
5	Appliances	1635

No.	Device Type	Sampled no. of records
1	Camera	1635
2	Audio	1635
3	SmartHub	1635
4	HomeAutomation	1635
5	Appliances	1635

for training are a coffee maker, a vacuum, and a smart board. Due to their minimal connectivity, these devices normally only produce a small number of packets hence fewer records for training compared to other devices.

4.3.1.2 Feature selection

The findings of the feature selection used in this study, as described in subsection 3.2.2.2, are reviewed in this section. The top 20 features were selected. The topmost feature appears to be from Figure 4.1 *DNS_count* which is the Domain Name Server request made by a particular device, followed by the most frequent destination IP (*most_freq_d_ip*). The number of destination IP addresses that a device contact is an important feature because some devices have less communication, consequently, an attribute that can be utilized to distinguish one type of device from another. A camera, for example, would have very few destination IP addresses, whereas a speaker would have much more outbound communication whether in interaction, power or idle mode.

4.3.1.3 Training and Testing of the ML methods

Scikit learn[4], a Python library, is used in the implementation of the device profiling and identification component in our system. To determine which machine learning technique performs the best, six machine learning models are utilized to train and

[4]https://scikit-learn.org/stable/

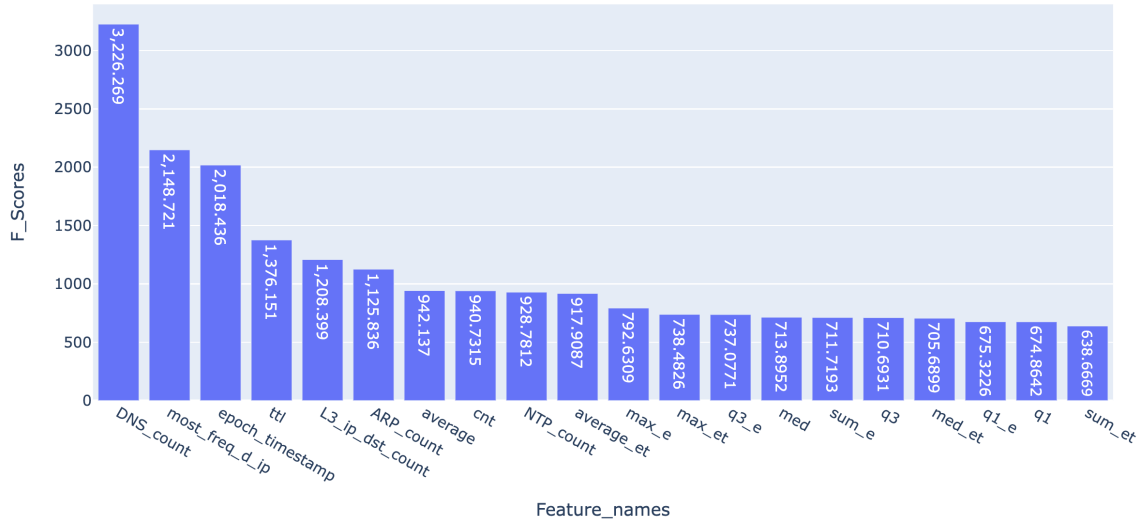


Figure 4.1: 20 top features using ANOVA technique explained in section 3.2.2.2 and their respective score on the CIC dataset [1].

test the dataset for both when the data is from the same distribution or a different distribution. When using data from the same distribution, we first cross-validate each of our six machine learning classifiers using a 7:3 split, which involves training on 70% of the data that was randomly chosen and testing on the remaining 30%.

4.3.1.4 Performance measuring metrics

In this section, we outline five benchmark performance indicators for assessing how well our classifiers performed on the dataset with the training and test data taken from the set. Macro-averaging is employed for the overall performance of precision, recall, and F1-score. For datasets with training and testing data drawn from a different distribution (lab), a different metric is used to evaluate the device identification performance. In this thesis, we adopted the testing metric used in [1] called the Inference percentage. More detail on the metrics for assessing datasets based on the same distribution is discussed in a later section.

True Positive (TP) is when a device is shown to correspond to a device type and the classifier also accurately identifies the device as belonging to that given device type.

True Negative (TN) occurs when a device does not fall into a certain device type and the test accurately predicts the device does not fall into that stated device type.
False Positive (FP) is when a device is identified by the classifier as belonging to a particular device type but is not truly a member of that device type.
False Positive (FN) occurs when a device is predicted by the classifier to not belong to the intended device type category.
Confusion matrix is a visual depiction of the model's performance in predicting the various classes [143].
Accuracy is the proportion of accurate predictions to all observations.

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

Precision is calculated as the number of correctly identified device types by the number of correctly or incorrectly predicted device types by the classifier.

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

Detection Rate shows how many actual device types the classifier classified properly.

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

F1-score is a geometric average of precision and recall.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

When a trained model is transferred to a different lab for testing, the Inference Percentage (IP) is the metric for evaluation which Sajjad et al. [1] described as follows;
Inference percentage, IP is computed for the predicted IoT device. Following classification prediction, a unique count of the target label prediction is compiled. This key-value pair, which contains the name of the particular device type and its associated count, is sorted

in decreasing order of magnitude. The name of the device type with the greatest value is the device type that is assigned to a specific feature space.

4.3.2 Profiling and identification

4.3.2.1 CIC IoT dataset 2022

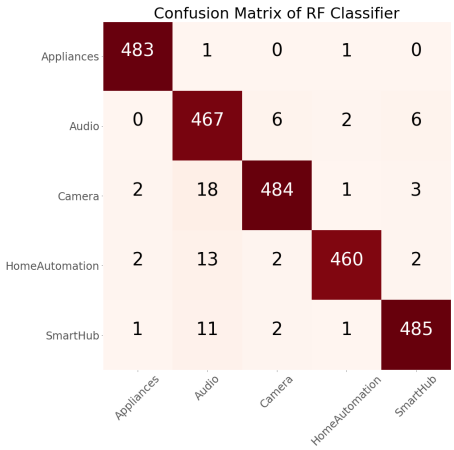
Using CIC IoT dataset 2022 [1] as our source domain and training dataset, Table 4.6 compare the performance of the RF, DT, GBC, KNN, SVM, and MLP concerning the performance metrics: Accuracy, Precision, Recall, F1-score, and processing time (in seconds) before and after normalization and selecting the top 20 features. GBC, RF, DT, and KNN performed considerably well in both before the two data preprocessing techniques despite a drop in the performance after.

Table 4.6: Performance of six classifier before and after normalization and feature selection.

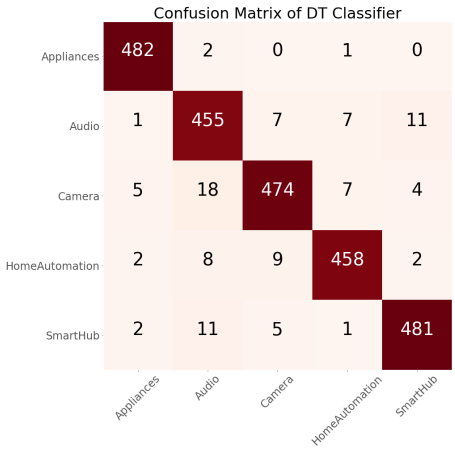
	Before Normalization and Feature selection					
Metric	**MLP**	**GBC**	**SVM**	**DT**	**RF**	**KNN**
Accuracy (%)	61	**100**	65	**99**	**100**	**96**
Recall (%)	61	**100**	65	**99**	100	**96**
Precision (%)	67	**100**	68	**99**	**100**	**96**
F1-Score (%)	57	**100**	63	**99**	**100**	**96**
Time (s)	**2.11**	111.6	3.89	0.13	1.34	**0.18**

After Normalization and Feature selection					
MLP	**GBC**	**SVM**	**DT**	**RF**	**KNN**
91	98	**85**	96	97	91
91	98	**85**	96	97	91
92	98	**87**	96	97	91
91	98	**85**	96	97	91
6.28	**31.20**	**1.66**	**0.04**	**0.8**	0.41

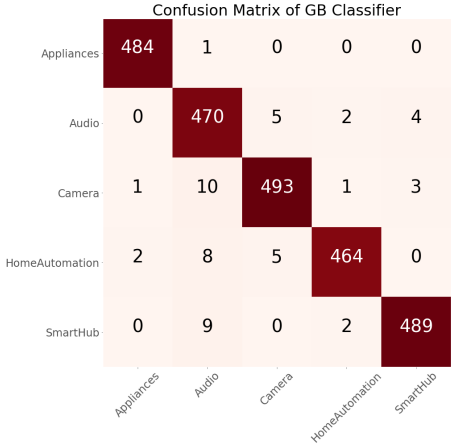
The accuracy, recall, precision, and F1-score of MLP and SVM increased after normalization and feature selection yielding 91%, 91%, 92%, 91% and 85%, 85%, 87%, 85% respectively from 91%, 91%, 92%, 91% and 85%, 85%, 87%, 85% respectively before normalization and feature selection. However, in terms of time there was a general improvement in GBC, SVM, DT, and RF after normalization and feature selection and a slight decrement in time for MLP and KNN from 2.11s and 0.18s to 6.28s and 0.41s respectively and a massive reduction in processing time for GBC from 111.6s to 31.20s. GBC is an ensemble method that is generally time-consuming due to the underlying concept of incrementally improving classifier performance after each iteration. The reason for the decreased time after the data preprocessing is also that the model was trained using fewer features, which required less computing time. Given our present findings, it is important to note that the source data has 8175 records overall due to the data sample of 1635 records for each target



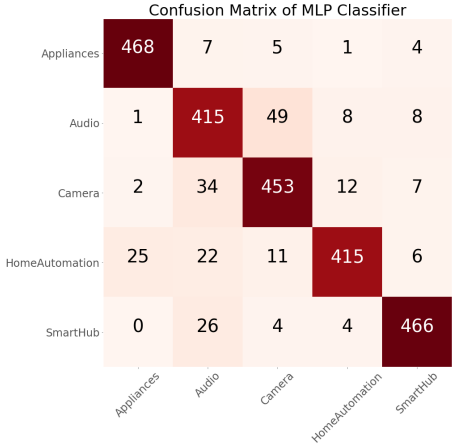
(a) Random Forest



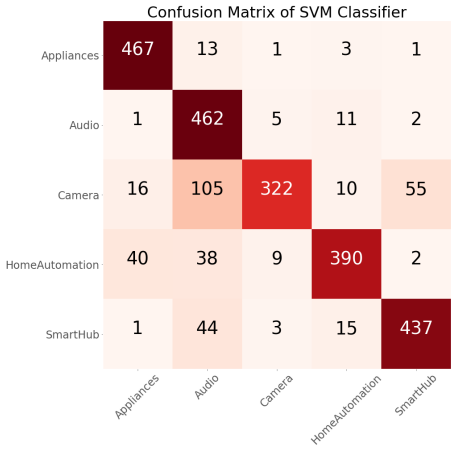
(b) Decision Tree



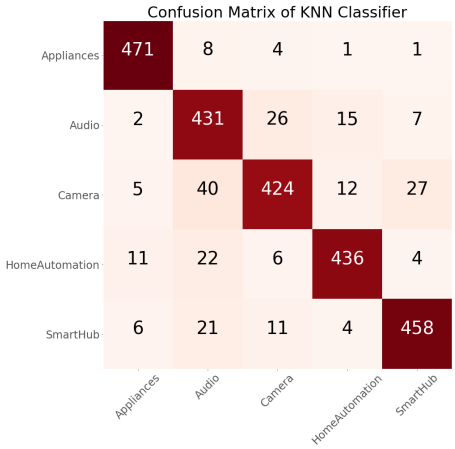
(c) Gradient Boosting



(d) Multi-Layer Perceptron



(e) Support Vector Machine



(f) K-Nearest Neighbor

Figure 4.2: Confusion Matrix of the six classifier used for device type identification.

variable to prevent overfitting. The situation is that if this number rises, it follows that the amount of time needed for training and testing will proportionally rise. Depending on the task at hand, one can then decide which metric precedes the other - the accuracy, recall, precision or time it takes for the ML to process the model.

Figure 4.2 represent the confusion matrix of each of the classifiers used in training our dataset. Random Forest correctly classified most of the devices while misclassifying 2, 8, 24, 19, 15 Appliance, Audio, Camera, HomeAutomation and SmartHub devices respectively in Figure 4.2a. Decision Tree depicted in Figure 4.2b misclassified 2, 26, 34, 19, 19 Appliance, Audio, Camera, HomeAutomation and SmartHub devices respectively. In Figure 4.2c, Gradient Boosting classifier misclassified 1, 11, 15, 15, 11 Appliance, Audio, Camera, HomeAutomation and SmartHub devices respectively. Multi-layer Perceptron misclassified 17, 66, 55, 64, 34 Appliance, Audio, Camera, HomeAutomation and SmartHub devices respectively illustrated in Figure 4.2d. In Figure 4.2e, Support Vector Machine misclassified 18, 19, 186, 89, 63 Appliance, Audio, Camera, HomeAutomation and SmartHub devices respectively. The figure in Figure 4.2f shows that K-Nearest Neighbor misclassified 14, 50, 86, 43, and 42 Appliance, Audio, Camera, HomeAutomation and SmartHub devices respectively.

The best performing classifier from our results is the Gradient Boosting classifier with fewer misclassification closely followed by Random Forest and Decision Tree.

4.3.2.2 IMC 2019 Payload Dataset

To test the transferability technique based on the concept of transductive transfer learning proposed in this study, the IMC 2019 Payload Dataset was employed. The model was trained in the source domain which is located in Canada at the Canadian Institute for Cybersecurity (CIC) lab using the CIC IoT dataset 2022 [1] and tested in the target domain which is located in the US at Northeastern University's Mon(IoT)r Lab. Table 4.7 demonstrates our results using the metric, Inference Percentage explained in subsection 4.3.1.4. RF failed to classify any other device while classifying 6 out of 8 cameras from the table. 5 out of 8 cameras and 3 out of 5 appliances were accurately classified by DT.

Table 4.7: Device type predicted by the different classifiers with the correctly predicted devices highlighted in green when the trained CIC IoT dataset 2022 was transferred to the US lab and tested with the IMC 2019 Payload Dataset [2] using the Interaction experiment conducted.

		Device type prediction Classifier					
No.	**Device name**	**RF**	**DT**	**GBC**	**KNN**	**SVM**	**MLP**
1.	Amazon Cloudcam	**Camera**	Appliance	Appliance	Home Automation	Smart Hub	Home Automation
2.	Amcrest Cam	**Camera**	Home Automation	HomeAutomation	Smart Hub	**Camera**	**Camera**
3.	Blink Cam	Home Automation	**Camera**	**Camera**	Audio	Audio	Audio
4.	Lefun Cam	**Camera**	**Camera**	Audio	Smart Hub	**Camera**	**Camera**
5.	Luohe Cam	**Camera**	**Camera**	**Camera**	Smart Hub	**Camera**	**Camera**
6.	Microseven Cam	**Camera**	**Camera**	**Camera**	Audio	Smart Hub	Smart Hub
7.	Wansview Cam	**Camera**	Appliance	**Camera**	Smart Hub	Smart Hub	Smart Hub
8.	Yi Cam	Audio	**Camera**	Audio	Smart Hub	**Camera**	**Camera**
9.	Insteon	Camera	Camera	Camera	Audio	Audio	Audio
10.	Philips Hue	Audio	Camera	Camera	Audio	Audio	Audio
11.	Sengled hub	Camera	Camera	Camera	**Smart Hub**	**Smart Hub**	Appliance
12.	Smartthings	Camera	Camera	Camera	Audio	Camera	**Smart Hub**
13.	Wink 2 hub	Camera	Home Automation	Camera	**Smart Hub**	Audio	Audio
14.	Xiaomi hub	Camera	Camera	Camera	**Smart Hub**	Camera	Camera
15.	TP-Link Plug	Camera	Camera	Camera	Audio	Smart Hub	Smart Hub
16.	Amazon Echo Dot	Camera	Camera	Camera	**Audio**	**Audio**	**Audio**
17.	Amazon Echo Spot	Camera	Camera	Camera	**Audio**	**Audio**	**Audio**
18.	Amazon Echo Plus	Camera	Appliance	Appliance	**Audio**	**Audio**	**Audio**
19.	Google Home Mini	Camera	Camera	Camera	**Audio**	Camera	Smart Hub
20.	Behmor Brewer	Smart Hub	Camera	Smart Hub	**Appliance**	Camera	**Appliance**
21.	GE Microwave	Camera	**Appliance**	**Appliance**	Audio	Smart Hub	Audio
22.	Samsung Washer	Camera	**Appliance**	**Appliance**	Audio	Audio	Audio
23.	Xiaomi Cleaner	Smart Hub	**Appliance**	Appliance	Smart Hub	Smart Hub	Home Automation
24.	Xiaomi Rice Cooker	Camera	Camera	Camera	Smart Hub	Smart Hub	**Appliance**

GBC accurately predicted 2 out of 5 appliances and 4 out of 8 cameras. 3 out of 6 Smart Hubs, 4 out of 4 Audio, and 1 out of 5 Appliance products were predicted by KNN. 4 out of 8 cameras, 1 out of 5 smart hubs, and 3 out of 4 audio devices were accurately classified by SVM. The most diverse classifier appears to be MLP, which correctly predicts 10 out of 24 gadgets. It makes predictions for 4 out of 8 cameras, 1 out of 6 smart hubs, 3 out of 4 audio devices, and 2 out of 5 appliances. None of the six classifiers correctly predicted the Insteon hub, Philips Hue, and TP-Link plug belonging to Smart Hub, Smart Hub, and Home Automation.

4.3.2.3 IoT Sentinel dataset

For the verifiability of our proposed transferability approach, we test our work on the IoT Sentinel dataset. The devices in this dataset were purchased in the European market, hence it is assumed the lab is located in Europe. Table 4.8 displays the findings for the devices drawn from the dataset utilized in this study.

The dataset was limited in terms of the diversity in the devices compared to that used for

Table 4.8: Device type predicted by the different classifiers with the correctly predicted devices highlighted in green when the trained CIC dataset was transferred to the UK lab and tested with the IoT Sentinel dataset [3].

		Device type prediction Classifier					
No.	**Device name**	**RF**	**DT**	**GBC**	**KNN**	**SVM**	**MLP**
1.	D-Link Cam	Camera	Camera	Camera	Smart Hub	Smart Hub	Camera
2.	D-Link Day Cam	Camera	Appliance	Appliance	Smart Hub	Camera	Camera
3.	D-Link Switch	Camera	Camera	Appliance	Audio	Smart Hub	Audio
4.	Edimax cam1	Camera	Camera	Camera	Smart Hub	Smart Hub	Camera
5.	Edimax cam2	Camera	Appliance	Appliance	Smart Hub	Smart Hub	Camera
6.	Edimax plug1	Camera	Camera	Camera	Smart Hub	Smart Hub	Smart Hub
7.	Edimax plug2	Camera	Appliance	Appliance	Audio	Smart Hub	Smart Hub
8.	Ednet cam1	Camera	Appliance	Appliance	Smart Hub	Smart Hub	HomeAutomation
9.	Ednet cam2	Camera	Camera	Appliance	Smart Hub	Camera	Camera
10.	Ednet gateway	Camera	Appliance	Appliance	HomeAutomation	Smart Hub	Smart Hub
11.	HomeMaticPlug switch	Camera	Camera	Camera	Audio	Smart Hub	Home Automation
12.	Hue switch	Camera	Camera	Appliance	Audio	Smart Hub	Smart Hub
13.	Lightify	Camera	Appliance	Appliance	Smart Hub	Smart Hub	Home Automation
14.	Max gateway	Camera	Camera	Audio	Audio	Smart Hub	Home Automation
16.	Smarter coffee	Camera	Camera	Smart Hub	Smart Hub	Smart Hub	Camera
15.	WeMoInsight Switch1	Camera	Appliance	Camera	Audio	Camera	Camera
16.	WeMoInsight Switch2	Camera	Appliance	Camera	Smart Hub	Camera	Camera
17.	Wemo switch	Camera	Camera	Camera	Audio	Smart Hub	Camera
18.	Wemo switch2	Camera	Camera	Camera	Audio	Smart Hub	Camera

training. From the table SVM appears to have outperformed the other classifiers, correctly classifying 12 devices as belonging to their respective categories, 2 out of 6 cameras, 8 out of 10 smart hubs, and failed to classify any home automation device. RF, DT, and GBC classified all 6, 3 and 2 cameras respectively but failed to classify the other devices.

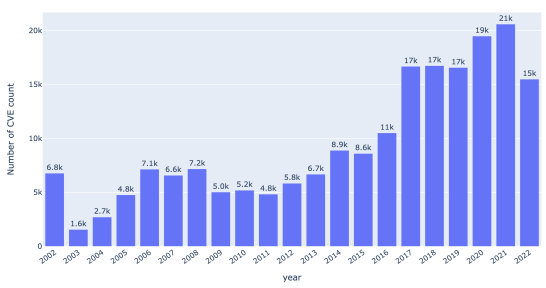
Second to SVM is MLP which correctly classified 7 devices; 5 out of 6 cameras, 2 out of 10 smart hubs, and failed to classify any home automation device. The dataset had devices that had two instances. For instance, KNN correctly classified WeMoInsight Switch2 as a Smart Hub but failed to classify the second instance.

4.3.3 Vulnerability assessment

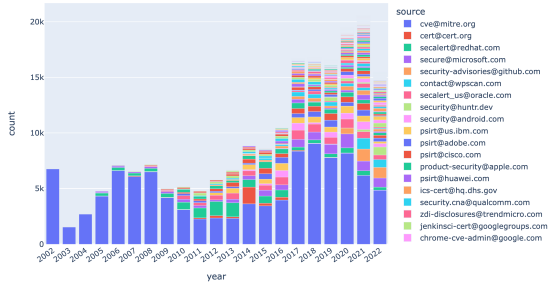
The NVD data stream that has been collected and saved in a local database is examined for patterns and trends prior to evaluating the vulnerabilities of the expected device kinds.

The total number of CVEs that have been reported throughout time is represented in Figure 4.3a.

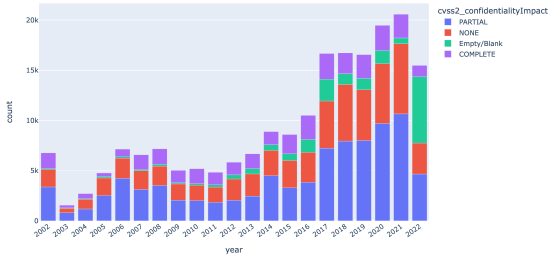
This helps to comprehend both the highest and lowest points of the CVE recording. According to Figure 4.3a, the maximum number of CVEs appears to have been recorded



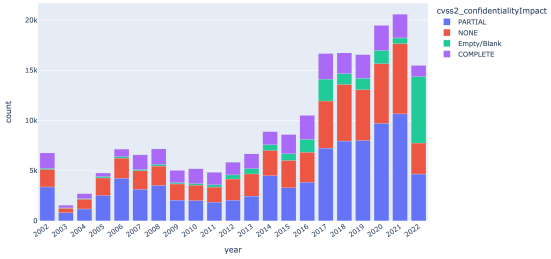
(a) Number of CVE per year



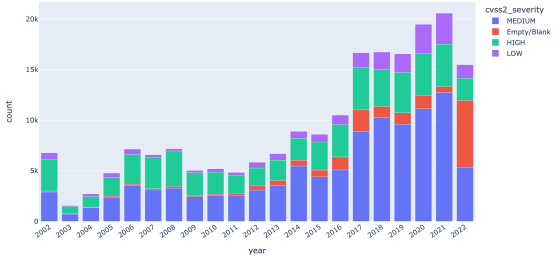
(b) CVE Reporting Sources



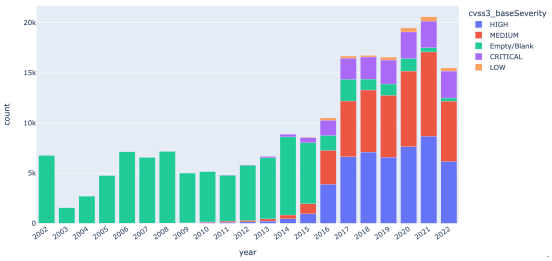
(c) Confidential Impact V2



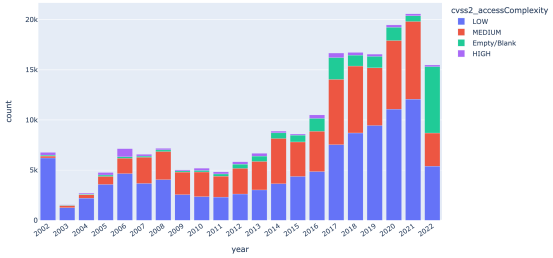
(d) Confidential Impact V2



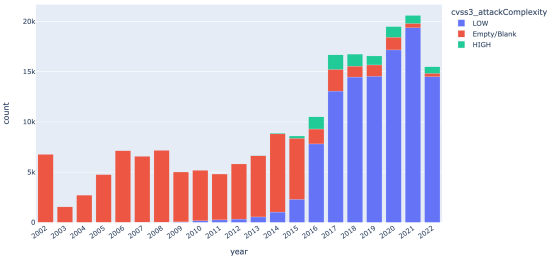
(e) Severity Distribution V2



(f) Severity Distribution V3



(g) Access Complexity Distribution V2



(h) Attack Complexity Distribution V3

Figure 4.3: Exploratory Analysis of the NVD data feed.

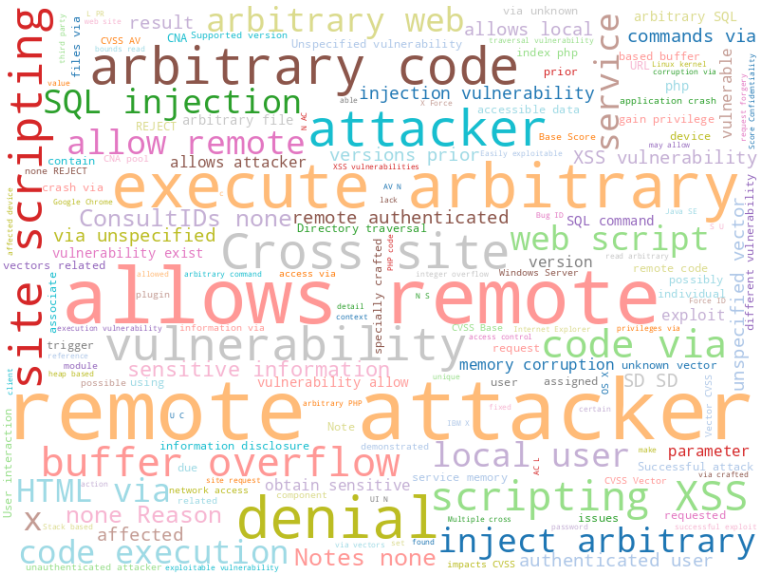


Figure 4.4: Most common vulnerabilities or approaches

in 2021. The many reporting platforms or organizations that produce known cybersecurity vulnerabilities are represented in Figure 4.3b. From this figure, it appears that CVE, which is run by MITRE, is the main organization that reports and records the bulk of these vulnerabilities. This is because NVD and the MITRE CVE list are completely synced. Therefore, vulnerabilities listed in CVE are also included in NVD.

When a device, software, or hardware, is exploited it affects the confidentiality (information disclosure), integrity (information modification), and availability (resource availability) of the object. NVD has undergone a series of versioning; 2.0, 3.0, and now version 3.1. This was already stated in the preceding chapter in part 3.3.4. We examine the effects of information disclosure on a vulnerability over time in Figure 4.3d using the NVD scoring system 2.0. This figure demonstrates how the majority of exploits throughout the years have exposed "Complete" information, which has led to the disclosure of sensitive, private, and personal data during these attacks. There are certain exploits, however, that just reveals "No" or "Partial" information. Figure **??** illustrates the information disclosure comparable to version 2 using the current version 3 scoring methodology. This chart illustrates that from the year 2002 to 2015, there were a majority of "Empty/Blank" spaces. This is a result of the new scoring system being used in the years that followed.

Similar to version 2, several exploits in version 3 had a “High” impact on information exposure, while others had a “Low” or “No” impact on confidentiality. This visualization is a simple graph which shows the distribution of vulnerabilities by severity over time. The choice of LOW, MEDIUM and HIGH is based upon the CVSS V2 Base score. Figure 4.3e uses the version 2 grading system to show the degree of attacks that were exploited and how severe they were. According to the figure, the attacks are generally of “Medium” severity, with 2022 showing a significant portion of “Empty/Blank” entries. Version 3 depicted in Figure 4.3f includes the additional metric “Critical” among the possibilities for the severity. Again, version 3 was in use recently, therefore there are many “Empty/Blank” entries in the years prior. Most vulnerabilities in version 3 that have been reported have a severity of “High” or “Medium”, with a small number of “Critical” situations also being reported.

The version 2 scoring system in Figure 4.3g demonstrates how simple and technically sophisticated to exploit a vulnerability. According to the figure, the majority of these exploits had “Low” attack complexity, which has highly negative implications because there aren’t adequate security measures in place. There are also a few “Empty/Blank” entries and some “Medium” and “High” access complexity entries. Figure 4.3h makes it clear that several of these vulnerabilities didn’t warrant highly skilled or technically inclined adversaries. The majority of attacks in this version are “Low” in complexity, a handful are “High” while earlier years are “Empty/Blank”.

The most frequently used phrases from the descriptions of each CVE vulnerability or the techniques used by attackers to conduct an exploit are shown in Figure **??**. The words “remote attacker”, “allows remote”, “execute arbitrary code”, and “site scripting”, to name a few, are some of the most apparent choices.

4.3.4 Visualization and dashboard

The framework that is proposed includes a dashboard that shows a depiction of the vulnerabilities found for each type of predicted device. To improve the presentation of the

evaluation results, we create a web-based visualization dashboard interface. Plotly [5], a Python library, is the programming language used to integrate all graphs and display them on the portal. The respective device type data had diverse attributes that are compelling to plot. However, we picked the attributes that best articulate the recent vulnerabilities and the scope of this research specifically exploitability score, impact score, and score3.

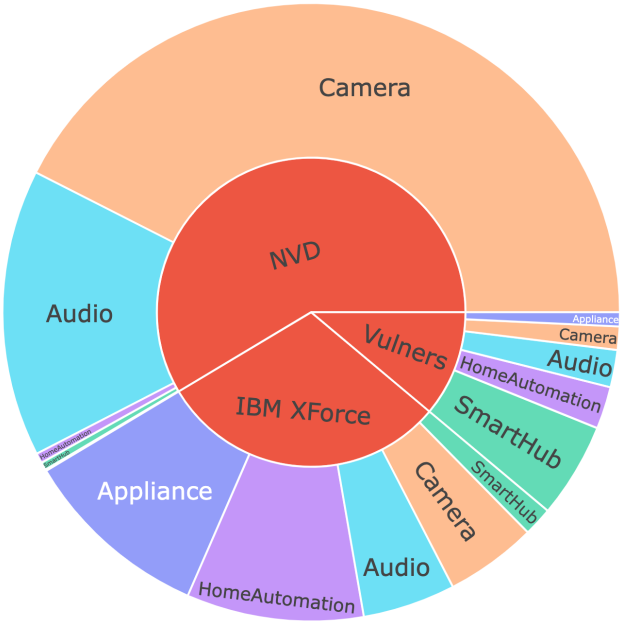


Figure 4.5: Sunburst chart showing hierarchical results for the respective databases and the predicted device types.

Practicality regarding the chart display and the visual representation was used to demonstrate the vulnerability. The charts were supposed to visually represent the searched results that had been analyzed. The test data used to perform the visualization is the IMC 2019 Payload Dataset [2]. When creating the graphs for this study, two possibilities were taken into account:

1. *Device type evaluation:* Visualising the average vulnerability for each device type.

   Figure 4.5 is a sunburst graph which is essentially a hierarchical multi-level chart that uses the different databases as the root and the predicted device types as the children. The value of the predicted device type for each database is computed as the average

[5]https://plotly.com/python/plotly-express/

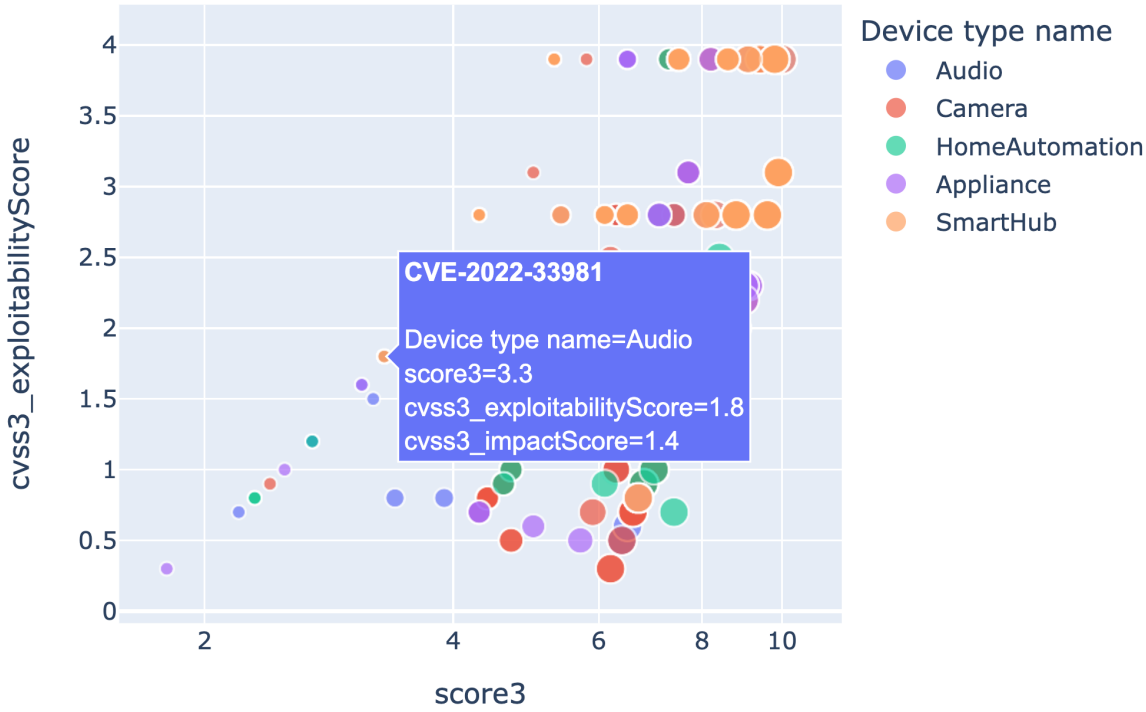


Figure 4.6: Scatter chart of device types with score3 on x-axis and exploitability score on y-axis.

of the CSSV score3 for each device type. Hence Camera in the NVD database has a larger portion than the other inferring Camera has a high CSSV score3 average. Figure 4.6 displays a scatter plot of the various Common Vulnerabilities Exposures (CVEs), the type of device, the CSSV version 3 score, the exploitability score, and the impact score. Hovering over the scatter plot is an Audio device with the CVE_ID CVE-2022-33981, the CSSV version 3 score of 3.3, the exploitability score of 1.8, and the impact score of 1.4. Figure 4.7 shows a bar chart which combines numerous data points per each device type per each database.

2. *Vendor evaluation:* Displaying the vulnerability rating for each vendor per device. Figure 4.8 is a bar chart which computes the aggregate CSSV score3 score for each vendor. The current hover shows Insteon as a vendor with a CSSV score3 of 9.86. Using the CSSV score3 on the x-axis and the CSSV exploitability score on the y-axis, Figure 4.9 depicts a scatter plot of the various vendors with CSSV score3 on the xaxis and the CSSV exploitability score on the yaxis. Amazon is listed as a

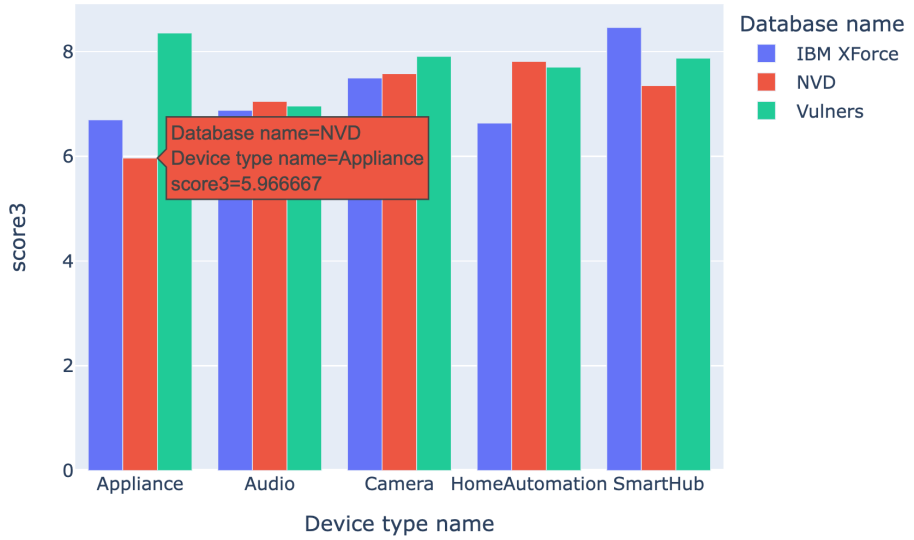


Figure 4.7: Bar chart showing the average CSSV score3 of the different predicted device types in the three different databases.

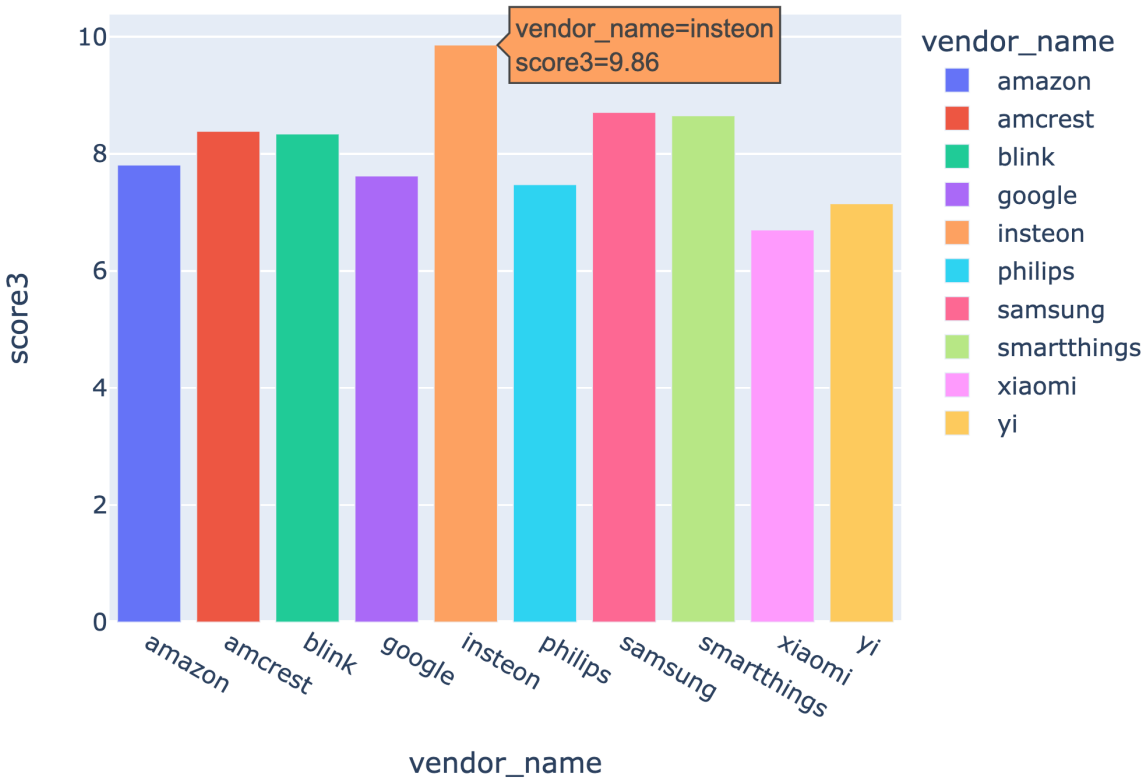


Figure 4.8: Bar chart showing different vendor and their aggregated CSSV score3.

vendor with an average CSSV score3 of 7.8, an average exploitability score of 3.1, and an average impact score of 4.7 in the current hover.

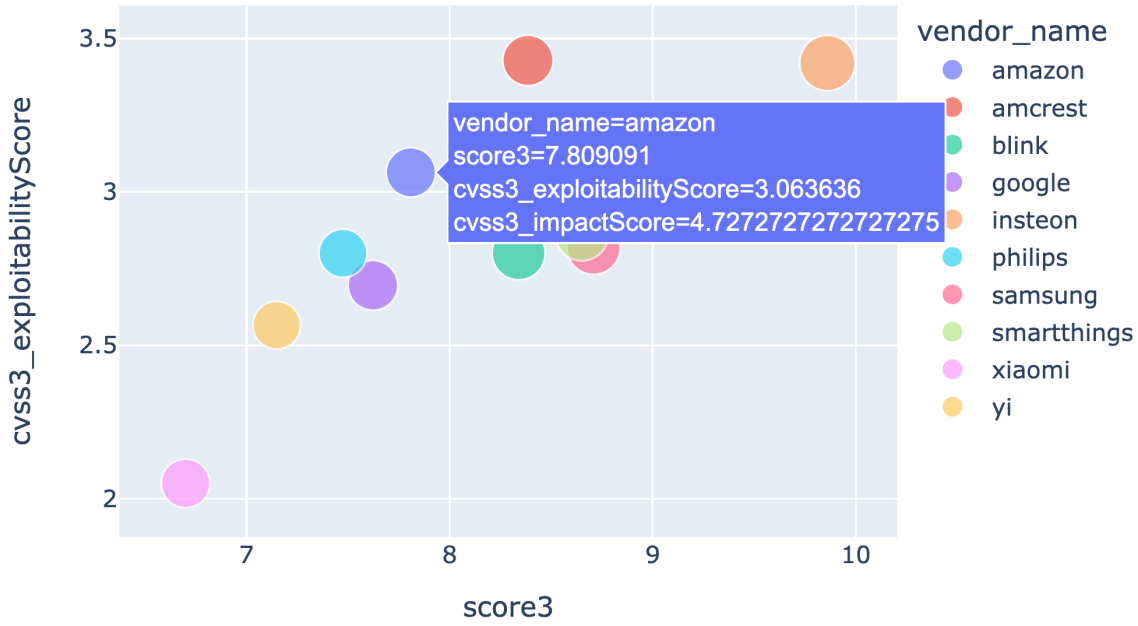


Figure 4.9: Scatter plot of the different vendors with CSSV score3 on the xaxis and the CSSV exploitability score on the yaxis.

4.4 Closing Remarks

In this chapter, we analyze and discuss our findings for both the situations in which training and test data were taken from the same distribution and the situations in which they were taken from separate distributions. In order to evaluate the model, it was then transported from the CIC IoT dataset (source) to the ICM Payload and IoT Sentinel dataset (target). We show and discuss the findings from our training of six ML models on the CIC dataset, both before and after data normalization and feature selection using ANOVA and SelectKBest (20 features for training). In order to validate the suggested transferability technique based on the underlying concept of transductive transfer learning employed in this thesis, the trained model from the source was moved to two separate target domains for testing to predict the label (corresponding device type for each device). There is no need for training in the target domain; the feature vectors in both datasets must match. The vulnerability databases used in this study for evaluating IoT devices include NVD, IBM X-Force, and Vulners. Using the data feed gathered and stored in a local database, a detailed analysis was done with a focus on the NVD database to discover patterns and trends. Last

but not least, device type and vendor analysis and visualization were performed using the projected device types as input.

The final chapter of this thesis will explore its limitations and offer ideas for future research possibilities.

Chapter 5

Conclusion

IoT devices are becoming increasingly integrated into our daily routines and lives. All businesses, including finance, healthcare, retail, education, and agriculture, can benefit from IoT's work. Such significant introduction and aggregation come with several security vulnerabilities. The management and monitoring of these devices will become the next major issue as the number of these devices increases. As a result, it became crucial to uniquely identify each device connected to the network, giving rise to device profiling and identification. According to one study, device identification is the initial stage in anomaly detection. A step in the correct path toward identifying a susceptible or exploited device is to establish behavioural patterns, network-based qualities, and physical characteristics that can uniquely identify a device. Device variability of this sort needs rapid and effective detection techniques. Due to the many devices from different manufacturers with their various communication and transmission methods and other functional differences, this might not be easy. Additionally, using methods that place a computational burden on these devices will affect how well they function, owing to the resource limitations of these devices. We discussed the various IoT device identification methods and vulnerability assessment techniques used. A simple taxonomy that scholars and businesses can use was developed as a result of the study for this thesis. Such a comprehensive study will save researchers time because it will serve as a foundation upon which to develop.

This study proposed a transferability approach based on transductive transfer learning

for IoT device profiling and identification. Forty-eight features were extracted from each device's pcap files while selecting the best 20 features using a statistical feature selection technique called ANOVA. The 20 best features were used to train the different ML models. For verifiability, the two publicly available datasets were used to test our proposed approach. CIC IoT dataset 2022 [1] was used as our training dataset (source domain), while our test datasets (target domain) are IMC 2019 Payload dataset [2] and IoT Sentinel dataset [73]. Experimenting with multiple ML models on the CIC IoT dataset 2022, where the training and test data is drawn from the same distribution, the Gradient boosting classifier achieved the best results in terms of Accuracy, Precision, Recall, and F1-score. Experimenting with multiple ML models where the CIC IoT dataset 2022 is the source and IMC 2019 Payload Dataset is the target domain Multilayer Perceptron outperformed the other classifiers. Experimenting with multiple ML models where the CIC IoT dataset 2022 is the source and the IoT Sentinel dataset is the target domain Support Vector classifier outperformed the other classifiers. The predicted device type from the classifier was searched through three vulnerability databases (NVD, Vulners, and IBM X-Force), and the results were displayed on a visualization dashboard. Cameras in the NVD database have the highest CSSV score, while the Insteon vendor has the highest vulnerability. From training and testing the dataset with multiple classifiers when the training and test data is from a different distribution, the experiments indicate that some classifiers work best for a particular device type category.

5.1 Open Challenges

In this section, we discuss challenges such as IoT device profiling and identification problems, and IoT vulnerability assessment difficulties.

5.1.1 IoT device identification open challenges

IoT device identification are susceptible to a myriad of challenges as follows.

- **Missing data:** Both network connection problems and data measurement inaccu-

racies can affect IoT devices. Supervised models cannot effectively manage large missing data sequences caused by connection concerns. It is difficult for a model to determine if the missing data stream belongs to one device or another. When an ML model is trained, the such missing value must be patched in, which presents several difficulties.

- **Data Diversity:** The high diversity of data types from many data sources is a significant problem for IoT device identification and profiling. There is a vast range of IoT hardware, including mobile phones, PCs/laptops, tablets, to IoT-ranged devices. Data variability has made it difficult to analyze, store, and maintain its integrity.

- **Availability of Profiling and Identification Related Datasets:** For the ML algorithms to be trained properly, a high-quality training dataset is necessary. Given that these training datasets serve as the foundation for gaining model knowledge, the training data ought to include details that nearly entirely represent the attribute and characteristics of an actual device. Real-time data streaming data quality is still a concern since IoT devices produce massive amounts of data.

5.1.2 Vulnerability assessment open challenges

Vulnerability assessment in IoT is a challenging task due to the following issues.

- **Lack of dedicated IoT vulnerability dataset:** Vulnerabilities impacting the IoT devices, which span many domains including hardware, software, and networking, cannot be properly managed by vulnerability databases. The de facto vulnerability database had limited vulnerabilities recorded for IoT devices. Vulners database had poorly recorded and structured data and IBM X-Force has the limitation of the number of devices to display at a time.

- **Realtime Vulnerability assessment:** The analysis conducted on the devices is passive which will not be applicable in a real-world setting where attacks are real-time. There is therefore the need for the implementation of an active vulnerability

system to continuously monitor and analyze the status of every device in the network while uniquely verifying the identity of the device at hand.

- **Technique Selection:** It is necessary to convert and organize data for analysis since they are not always provided in an analysis-ready state. High dimensionality, sophistication in association identification, volatile patterns, diverse distribution, and adaptability are just a few of the difficulties that large IoT data must overcome. In this situation, choosing an approach depends on how it will be used.

5.2 Future Research Directions

There are several existing borderline cases that need to be addressed and answers found in the IoT domain are to impact lives in the upcoming years. As an overview, the following list includes some of the future directions for the IoT profiling and vulnerability assessment domain:

- **Lifelong Learning:** The goal of lifelong machine learning (ML) is to build a model that can continually carry out the retraining procedure for discovering new, developing patterns associated with each behaviour. The short-term learning of risks and assaults aimed at IoT systems may not be sufficient for long-term security in a real-world IoT context. The model should be capable of adapting to novel contexts and picking up new knowledge over time.

- **Edge Computing:** The migration of service delivery from the cloud to the network edge, known as edge computing, offers a potential remedy for the IoT age. For IoT security, ML implementation at the edge may reduce latency, enable almost real-time device identification, and boost energy conservation. An efficient architecture for data processing with less network traffic burden may be provided by such an arrangement. This is an additional measure to safeguard their privacy and security.

- **Blockchain integration:** By combining IoT with blockchain technologies, every communication and information exchange between devices will become autonomous.

Blockchain is a new technology that secures network transactions using encryption. Transaction authentication using a decentralized database design relies on the consent of several stakeholders in the system as opposed to a single authority, as is typical in centralized systems. Therefore, compared to centralized systems, blockchain technologies can make transactions safer and more transparent. IoT systems are by definition dispersed. In order to secure IoT devices, the distributed digital ledger technology known as blockchain can be quite useful.

- **MUD integration:** In MUD, manufacturers rather than administrators are in charge of defining device behavioural profiles. Yet, the MUD design enables the manufacturer-defined network access policies to be automated. Network rules like Access Control Lists are included in MUD files (ACLs). It specifies the limitations on the types of devices with whom it may interact, such as talking to other products from the same manufacturer, interacting with a certain service, such as DNS, or granting and disallowing access to a particular port. By specifying the targeted device behaviour and confining the device to the primary goal, the integration of MUD in IoT seeks to decrease the attack surface area on the device.

Bibliography

[1] Dadkhah Sajjad, Mahdikhani Hassan, Kyei Danso Priscilla, Zohourian Alireza, Anh Truong Kevin, and A. Ghorbani Ali. Towards the development of a realistic multidimensional iot profiling dataset. 2022.

[2] Jingjing Ren, Daniel J. Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 267–279, New York, NY, USA, 2019. Association for Computing Machinery.

[3] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, Tommaso Frassetto, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel demo: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2511–2514, 2017.

[4] Statista Research Department. Internet of things (iot) and non-iot active device connections worldwide from 2010 to 2025 (in billions) [graph]. 2020.

[5] Yongxin Liu, Jian Wang, Jianqiang Li, Shuteng Niu, and Houbing Song. Machine learning for the detection and identification of internet of things devices: A survey. *IEEE Internet of Things Journal*, 9(1):298–320, 2022.

[6] Noman Mazhar, Rosli Salleh, Muhammad Zeeshan, and M. Muzaffar Hameed. Role of device identification and manufacturer usage description in iot security: A survey. *IEEE Access*, 9:41757–41786, 2021.

[7] R. Porkodi and V. Bhuvaneswari. The internet of things (iot) applications and communication enabling technology standards: An overview. In *2014 International Conference on Intelligent Computing Applications*, pages 324–329, 2014.

[8] Gartner says a typical family home could contain more than 500 smart devices by 2022, Sep 2014.

[9] Somayya Madakam, Vihar Lake, Vihar Lake, Vihar Lake, et al. Internet of things (iot): A literature review. *Journal of Computer and Communications*, 3(05):164, 2015.

[10] Adai Mohammed Almomani and Mohd Nordin Abdul Rahman. A literature review of the adoption of internet of things: Directions for future work.

[11] Statista Research Departmentn. Internet of things (iot) connected devices installed base worldwide from 2015 to 2025. 2016.

[12] J. Manyika, Michael Chui, Peter Bisson, Jonathan R. Woetzel, Richard Dobbs, Jacques Bughin, and Danor Aharon. The internet of things: mapping the value beyond the hype. 2015.

[13] Felix Wortmann and Kristina Flüchter. Internet of things. *Business & Information Systems Engineering*, 57(3):221–224, Jun 2015.

[14] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. *Computer*, 50(2):76–79, 2017.

[15] Narges Yousefnezhad, Avleen Malhi, and Kary Främling. Automated iot device identification based on full packet information using real-time network traffic. *Sensors*, 21(8), 2021.

[16] Ahmet Aksoy and Mehmet Hadi Gunes. Automated iot device identification using network traffic. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, 2019.

[17] Bingdong Li, Jeff Springer, George Bebis, and Mehmet Hadi Gunes. A survey of network flow applications. *Journal of Network and Computer Applications*, 36(2):567–581, 2013.

[18] Nadia Chaabouni, Mohamed Mosbah, Akka Zemmari, Cyrille Sauvignac, and Parvez Faruki. Network intrusion detection for iot security based on learning techniques. *IEEE Communications Surveys Tutorials*, 21(3):2671–2701, 2019.

[19] Vinita Malik and Sukhdip Singh. Security risk management in iot environment. *Journal of Discrete Mathematical Sciences and Cryptography*, 22(4):697–709, 2019.

[20] Karen Rose, Scott D. Eldridge, and Lyman Chapin. The internet of things : An overview understanding the issues and challenges of a more connected world. 2015.

[21] Narges Yousefnezhad, Avleen Malhi, and Kary Främling. Security in product lifecycle of iot devices: A survey. *Journal of Network and Computer Applications*, 171:102779, 2020.

[22] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[23] Sara N. Matheu-García, José L. Hernández-Ramos, Antonio F. Skarmeta, and Gianmarco Baldini. Risk-based automated assessment and testing for the cybersecurity certification and labelling of iot devices. *Computer Standards Interfaces*, 62:64–83, 2019.

[24] José Roldán, Juan Boubeta-Puig, José Luis Martínez, and Guadalupe Ortiz. Integrating complex event processing and machine learning: An intelligent architecture for detecting iot security attacks. *Expert Systems with Applications*, 149:113251, 2020.

[25] Yuan Shi, Zhenzhong Lan, Wei Liu, and Wei Bi. Extending semi-supervised learning methods for inductive transfer learning. In *2009 Ninth IEEE International Conference on Data Mining*, pages 483–492, 2009.

[26] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

[27] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, May 2016.

[28] Oscar Day and Taghi M. Khoshgoftaar. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4(1):29, Sep 2017.

[29] Chin-Wei Tien, Tse-Yung Huang, Ping-Chun Chen, and Jenq-Haur Wang. Using autoencoders for anomaly detection and transfer learning in iot. *Computers*, 10(7), 2021.

[30] Mohammed Ali Al-Garadi, Amr Mohamed, Abdulla Khalid Al-Ali, Xiaojiang Du, Ihsan Ali, and Mohsen Guizani. A survey of machine and deep learning methods for internet of things (iot) security. *IEEE Communications Surveys Tutorials*, 22(3):1646–1685, 2020.

[31] Mohamed Abomhara and Geir M Køien. Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security and Mobility*, pages 65–88, 2015.

[32] Long Wen, Liang Gao, and Xinyu Li. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1):136–144, 2019.

[33] Ly Vu, Quang Uy Nguyen, Diep N. Nguyen, Dinh Thai Hoang, and Eryk Dutkiewicz. Deep transfer learning for iot attack detection. *IEEE Access*, 8:107335–107344, 2020.

[34] Anna L. Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, 2016.

[35] Lisa Torrey and Jude Shavlik. Transfer learning. *IGI Global*, 2010.

[36] Abhishek Verma and Virender Ranga. Elnids: Ensemble learning based network intrusion detection system for rpl based internet of things. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–6, 2019.

[37] P. Gralla and S. Ishida. *How the Internet Works*. How it works series. Que, 1998.

[38] Louis Coetzee and Johan Eksteen. The internet of things - promise for the future? an introduction. In *2011 IST-Africa Conference Proceedings*, pages 1–9, 2011.

[39] Nomusa Nomhle Dlamini and Kevin Johnston. The use, benefits and challenges of using the internet of things (iot) in retail businesses: A literature review. In *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pages 430–436, 2016.

[40] Oscar Garcia-Morchon, Sandeep Kumar, and Mohit Sethi. Internet of Things (IoT) Security: State of the Art and Challenges. RFC 8576, April 2019.

[41] B.B. Gupta and Megha Quamara. An overview of internet of things (iot): Architectural aspects, challenges, and protocols. *Concurrency and Computation: Practice and Experience*, 32(21):e4946, 2020. e4946 CPE-18-0159.R1.

[42] Roberto Minerva, Abyi Biru, and Domenico Rotondi. Towards a definition of the internet of things (iot). *IEEE Internet Initiative*, 1(1):1–86, 2015.

[43] Ron Davies. The internet of things: opportunities and challenges. 2015.

[44] P.P. Ray. A survey on internet of things architectures. *Journal of King Saud University - Computer and Information Sciences*, 30(3):291–319, 2018.

[45] O. Vermesan and P. Friess. *Internet of Things Applications - From Research and Innovation to Market Deployment*. River Publishers, 2022.

[46] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376, 2015.

[47] Konstantinos Domdouzis, Peter Lake, and Paul Crowther. Big data. In *Concise Guide to Databases*, pages 141–163. Springer, 2021.

[48] Payam Barnaghi, Wei Wang, Cory Henson, and Kerry Taylor. Semantics for the internet of things: Early progress and back to the future. *Int. J. Semant. Web Inf. Syst.*, 8(1):1–21, jan 2012.

[49] Matthew Gigli, Simon GM Koo, et al. Internet of things: services and applications categorization. *Adv. Internet Things*, 1(2):27–31, 2011.

[50] Nivedita Mishra and Sharnil Pandya. Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review. *IEEE Access*, 9:59353–59377, 2021.

[51] A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. van Kranenburg, S. Lange, and S. Meissner. *Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model*. Springer Berlin Heidelberg, 2013.

[52] J.P. Hubaux, S. Capkun, and Jun Luo. The security and privacy of smart vehicles. *IEEE Security Privacy*, 2(3):49–55, 2004.

[53] Ayesha Sadiq, Muhammad Umar Javed, Rabiya Khalid, Ahmad Almogren, Muhammad Shafiq, and Nadeem Javaid. Blockchain based data and energy trading in internet of electric vehicles. *IEEE Access*, 9:7000–7020, 2021.

[54] Avi Itzkovitch. The internet of things and the mythical smart fridge, Sep 2013.

[55] Margarita Angelidou. Smart city policies: A spatial approach. *Cities*, 41:S3–S11, 2014. Current Research on Cities.

[56] Patrizia Lombardi, Silvia Giordano, Hend Farouh, and Wael Yousef. Modelling the smart city performance. *Innovation: The European Journal of Social Science Research*, 25(2):137–149, 2012.

[57] Felipe Caro and Ramin Sadr. The internet of things (iot) in retail: Bridging supply and demand. *Business Horizons*, 62(1):47–54, 2019.

[58] Nikesh Gondchawar, RS Kawitkar, et al. Iot based smart agriculture. *International Journal of advanced research in Computer and Communication Engineering*, 5(6):838–842, 2016.

[59] Mahammad Shareef Mekala and P. Viswanathan. A survey: Smart agriculture iot with cloud computing. In *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, pages 1–7, 2017.

[60] F. John Dian, Reza Vahidnia, and Alireza Rahmati. Wearables and the internet of things (iot), applications, opportunities, and challenges: A survey. *IEEE Access*, 8:69200–69211, 2020.

[61] Amy H. Y. Chan, Roy A. Pleasants, Rajiv Dhand, Stephen L. Tilley, Stephen A. Schworer, Richard W. Costello, and Rajan Merchant. Digital inhalers for asthma or chronic obstructive pulmonary disease: A scientific perspective. *Pulmonary Therapy*, 7(2):345–376, Dec 2021.

[62] Hui Suo, Jiafu Wan, Caifeng Zou, and Jianqi Liu. Security in the internet of things: A review. In *2012 International Conference on Computer Science and Electronics Engineering*, volume 3, pages 648–651, 2012.

[63] Zhi-Kai Zhang, Michael Cheng Yi Cho, Chia-Wei Wang, Chia-Wei Hsu, Chong-Kuan Chen, and Shiuhpyng Shieh. Iot security: Ongoing challenges and research opportunities. In *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, pages 230–234, 2014.

[64] Ismail Butun, Patrik Österberg, and Houbing Song. Security of the internet of things: Vulnerabilities, attacks, and countermeasures. *IEEE Communications Surveys Tutorials*, 22(1):616–644, 2020.

[65] Nataliia Neshenko, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum, and Nasir Ghani. Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. *IEEE Communications Surveys Tutorials*, 21(3):2702–2733, 2019.

[66] Pietro Ferrara, Amit Kr Mandal, Agostino Cortesi, and Fausto Spoto. Static analysis for discovering iot vulnerabilities. *International Journal on Software Tools for Technology Transfer*, 23(1):71–88, Feb 2021.

[67] T. Perković, A. Dagelić, M. Bugarić, and M. Čagalj. On wpa2-enterprise privacy in high education and science. *Security and Communication Networks*, 2020:3731529, Sep 2020.

[68] Chris Harvey. Symantec 2018 internet security threat report — data3, 2022.

[69] Heather Richter Lipford, Madiha Tabassum, Paritosh Bahirat, Yaxing Yao, and Bart P. Knijnenburg. *Privacy and the Internet of Things*, pages 233–264. Springer International Publishing, Cham, 2022.

[70] Policy, Research Group, et al. The internet of things: An introduction to privacy issues with a focus on the retail and home environments. *Office of the Privacy Commissioner of Canada, Feb*, 2016.

[71] Laizhong Cui, Shu Yang, Fei Chen, Zhong Ming, Nan Lu, and Jing Qin. A survey on application of machine learning for internet of things. *International Journal of Machine Learning and Cybernetics*, 9(8):1399–1417, Aug 2018.

[72] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. Iotsense: Behavioral fingerprinting of iot devices, 2018.

[73] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2177–2184, 2017.

[74] Batyr Charyyev and Mehmet Hadi Gunes. Locality-sensitive iot network traffic fingerprinting for device identification. *IEEE Internet of Things Journal*, 8(3):1272–1281, 2021.

[75] Girish Vaidya, Akshay Nambi, T.V. Prabhakar, Vasanth Kumar T, and Suhas Sudhakara. Iot-id: A novel device-specific identifier based on unique hardware fingerprints. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 189–202, 2020.

[76] Y. Zhang. *New Advances in Machine Learning*. IntechOpen, 2010.

[77] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martín Ochoa, Nils Ole Tippenhauer, Juan David Guarnizo, and Yuval Elovici. Detection of unauthorized iot devices using machine learning techniques. *CoRR*, abs/1709.04647, 2017.

[78] Jonathan Fürst, Mauricio Fadel Argerich, Bin Cheng, and Ernö Kovacs. Towards knowledge infusion for robust and transferable machine learning in iot. *Open Journal of Internet Of Things (OJIOT)*, 6(1):24–34, 2020.

[79] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. Profiliot: A machine learning approach for iot device identification based on network traffic analysis. In *Proceedings of the Symposium on Applied Computing*, SAC '17, page 506–509, New York, NY, USA, 2017. Association for Computing Machinery.

[80] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[81] He Li, Kaoru Ota, and Mianxiong Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. *IEEE Network*, 32(1):96–101, 2018.

[82] Wei Lyu, Zhaoyang Zhang, Chunxu Jiao, Kangjian Qin, and Huazi Zhang. Performance evaluation of channel decoding with deep neural networks. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018.

[83] Jaidip Kotak and Yuval Elovici. Iot device identification using deep learning. *CoRR*, abs/2002.11686, 2020.

[84] Hossein Jafari, Oluwaseyi Omotere, Damilola Adesina, Hsiang-Huang Wu, and Lijun Qian. Iot devices fingerprinting using deep learning. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 1–9, 2018.

[85] Yongxin Liu, Jian Wang, Jianqiang Li, Houbing Song, Thomas Yang, Shuteng Niu, and Zhong Ming. Zero-bias deep learning for accurate identification of internet-of-things (iot) devices. *IEEE Internet of Things Journal*, 8(4):2627–2634, 2021.

[86] Ali Dorri, Clemence Roulin, Shantanu Pal, Sarah Baalbaki, Raja Jurdak, and Salil Kanhere. Device identification in blockchain-based internet of things. *arXiv preprint arXiv:2202.09603*, 2022.

[87] Shireen Rafat Alam, Saurabh Jain, and Rajesh Doriya. Security threats and solutions to iot using blockchain: A review. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 268–273, 2021.

[88] Eliot Lear, Ralph Droms, and Dan Romascanu. Manufacturer Usage Description Specification. RFC 8520, March 2019.

[89] Ayyoob Hamza, Dinesha Ranathunga, Hassan Habibi Gharakheili, Theophilus Benson, Matthew Roughan, and Vijay Sivaraman. Verifying and monitoring iots network behavior using MUD profiles. *CoRR*, abs/1902.02484, 2019.

[90] Soo-Yeon Lee, Sa-rang Wi, Eunil Seo, Jun-Kwon Jung, and Tai-Myoung Chung. Profiot: Abnormal behavior profiling (abp) of iot devices based on a machine learning approach. In *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 1–6, 2017.

[91] Andrei Bytes, Sridhar Adepu, and Jianying Zhou. Towards semantic sensitive feature profiling of iot devices. *IEEE Internet of Things Journal*, 6(5):8056–8064, 2019.

[92] G. Spathoulas, S. Evangelatos, M. Anagnostopoulos, G. Mema, and S. Katsikas. Detection of abnormal behavior in smart-home environments. In *2019 4th South-*

East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), pages 1–6, 2019.

[93] Kuai Xu, Yinxin Wan, Guoliang Xue, and Feng Wang. Multidimensional behavioral profiling of internet-of-things in edge networks. In *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, pages 1–10, 2019.

[94] Chin-Wei Tien, Tse-Yung Huang, Ping Chun Chen, and Jenq-Haur Wang. Automatic device identification and anomaly detection with machine learning techniques in smart factories. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3539–3544, 2020.

[95] Prabhakar Krishnan, Kurunandan Jain, Rajkumar Buyya, Pandi Vijayakumar, Anand Nayyar, Muhammad Bilal, and Houbing Song. Mud-based behavioral profiling security framework for software-defined iot networks. *IEEE Internet of Things Journal*, pages 1–1, 2021.

[96] G. Weidman. *Penetration Testing: A Hands-On Introduction to Hacking*. No Starch Press, 2014.

[97] Yi Hu, Dustin Sulek, Anthony Carella, Joshua Cox, Allyson Frame, Karen Cipriano, and Hong-xia Wang. Employing miniaturized computers for distributed vulnerability assessment. In *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 57–61, 2016.

[98] Sagar Samtani, Shuo Yu, Hongyi Zhu, Mark Patton, and Hsinchun Chen. Identifying scada vulnerabilities using passive and active vulnerability assessment techniques. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 25–30, 2016.

[99] Pooja Anand, Yashwant Singh, Arvind Selwal, Mamoun Alazab, Sudeep Tanwar, and Neeraj Kumar. Iot vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges. *IEEE Access*, 8:168825–168853, 2020.

[100] Jie Luo and Jian Wang. Vulnerability assessment of iot devices through multi-layer keyword matching. In *2021 International Conference on Computer, Internet of Things and Control Engineering (CITCE)*, pages 138–143, 2021.

[101] Fabiha Hashmat, Syed Ghazanfar Abbas, Sadaf Hina, Ghalib A. Shah, Taimur Bakhshi, and Waseem Abbas. An automated context-aware iot vulnerability assessment rule-set generator. *Computer Communications*, 186:133–152, 2022.

[102] Xuanyu Duan, Mengmeng Ge, Triet Huynh Minh Le, Faheem Ullah, Shang Gao, Xuequan Lu, and M. Ali Babar. Automated security assessment for the internet of things. In *2021 IEEE 26th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 47–56, 2021.

[103] Pieter Moens, Vincent Bracke, Colin Soete, Sander Vanden Hautte, Diego Nieves Avendano, Ted Ooijevaar, Steven Devos, Bruno Volckaert, and Sofie Van Hoecke. Scalable fleet monitoring and visualization for smart machine maintenance and industrial iot applications. *Sensors*, 20(15), 2020.

[104] Kai-Ming Chang, Ren-Jye Dzeng, and Yi-Ju Wu. An automated iot visualization bim platform for decision support in facilities management. *Applied Sciences*, 8(7), 2018.

[105] Pilaiwan Phupattanasilp and Sheau-Ru Tong. Augmented reality in the integrative internet of things (ar-iot): Application for precision farming. *Sustainability*, 11(9), 2019.

[106] Ana Lavalle, Miguel A. Teruel, Alejandro Maté, and Juan Trujillo. Improving sustainability of smart cities through visualization techniques for big data from iot devices. *Sustainability*, 12(14), 2020.

[107] Roman Kolcun, Diana Andreea Popescu, Vadim Safronov, Poonam Yadav, Anna Maria Mandalari, Richard Mortier, and Hamed Haddadi. Revisiting iot device identification. *arXiv preprint arXiv:2107.07818*, 2021.

[108] Ruizhong Du, Jingze Wang, Shuang Li, and Weiwei Liu. A lightweight flow feature-based iot device identification scheme. *Sec. and Commun. Netw.*, 2022, jan 2022.

[109] Jay Thom, Nathan Thom, Shamik Sengupta, and Emily Hand. Smart recon: Network traffic fingerprinting for iot device identification. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0072–0079, 2022.

[110] Andrew Arnold, Ramesh Nallapati, and William W. Cohen. A comparative study of methods for transductive transfer learning. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 77–82, 2007.

[111] Jinlin Zhu, Zhiqiang Ge, Zhihuan Song, and Furong Gao. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. *Annual Reviews in Control*, 46:107–133, 2018.

[112] Jianchang Mao and A.K. Jain. A self-organizing network for hyperellipsoidal clustering (hec). *IEEE Transactions on Neural Networks*, 7(1):16–29, 1996.

[113] S. Gopal Krishna Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *CoRR*, abs/1503.06462, 2015.

[114] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[115] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers Electrical Engineering*, 40(1):16–28, 2014. 40th-year commemorative issue.

[116] Mansour Sheikhan, Mahdi Bejani, and Davood Gharavian. Modular neural-svm scheme for speech emotion recognition using anova feature selection method. *Neural Computing and Applications*, 23(1):215–227, Jul 2013.

[117] ML2021dsb. Chi square and anova — feature selection for ml. 2020.

[118] Dariswan Janweri Perangin-Angin and Fitra A. Bachtiar. Classification of stress in office work activities using extreme learning machine algorithm and one-way anova f-test feature selection. In *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 503–508, 2021.

[119] Angus M. Brown. A new software for carrying out one-way anova post hoc tests. *Computer Methods and Programs in Biomedicine*, 79(1):89–95, 2005.

[120] Jason Brownlee. How to choose a feature selection method for machine learning. *Machine Learning Mastery*, 10, 2019.

[121] Feng-Jen Yang. An extended idea about decision trees. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 349–354, 2019.

[122] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[123] Sotiris B Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283, 2013.

[124] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004.

[125] S.R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[126] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[127] Gérard Biau and Erwan Scornet. A random forest guided tour. *TEST*, 25(2):197–227, Jun 2016.

[128] Ned Horning et al. Random forests: An algorithm for image classification and generation of continuous fields data sets. In *Proceedings of the International Conference on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences, Osaka, Japan*, volume 911, pages 1–6, 2010.

[129] Yanru Zhang and Ali Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324, 2015. Big Data in Transportation and Traffic Engineering.

[130] Samir Touzani, Jessica Granderson, and Samuel Fernandes. Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy and Buildings*, 158:1533–1543, 2018.

[131] G. Biau, B. Cadre, and L. Rouvière. Accelerated gradient boosting. *Machine Learning*, 108(6):971–992, Jun 2019.

[132] Sophia Kaplantzis, Alistair Shilton, Nallasamy Mani, and Y Ahmet Sekercioglu. Detecting selective forwarding attacks in wireless sensor networks using support vector machines. In *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 335–340. IEEE, 2007.

[133] Achmad Widodo and Bo-Suk Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6):2560–2574, 2007.

[134] Liangxiao Jiang, Zhihua Cai, Dianhong Wang, and Siwei Jiang. Survey of improving k-nearest-neighbor for classification. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, volume 1, pages 679–683, 2007.

[135] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[136] scikit-learn developers. 1.6. nearest neighbors.

[137] M.W Gardner and S.R Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14):2627–2636, 1998.

[138] Marcin Rytel, Anna Felkner, and Marek Janiszewski. Towards a safer internet of things—a survey of iot vulnerability data sources. *Sensors*, 20(21):5969, 2020.

[139] Harold Booth, Doug Rike, and Gregory Witte. The national vulnerability database (nvd): Overview, 2013-12-18 2013.

[140] Grzegorz J. Blinowski and Pawel Piotrowski. CVE based classification of vulnerable iot systems. *CoRR*, abs/2006.16640, 2020.

[141] Afsah Anwar, Ahmed Abusnaina, Songqing Chen, Frank Li, and David Mohaisen. Cleaning the NVD: comprehensive quality assessment, improvements, and analyses. *CoRR*, abs/2006.15074, 2020.

[142] Shivani Tyagi and Sangeeta Mittal. Sampling approaches for imbalanced data classification problem in machine learning. In *Proceedings of ICRIC 2019*, pages 209–221. Springer, 2020.

[143] Konstantinos Veropoulos, Colin Campbell, Nello Cristianini, et al. Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on AI*, volume 55, page 60. Stockholm, 1999.

Vita

Candidate's full name:
Priscilla Kyei Danso

University attended (with dates and degrees obtained):
Master of Computer Science under the supervision of Dr. Ali. A. Ghorbani and Sajjad Dadkhah, University of New Brunswick (UNB), Fredericton, New Brunswick, Canada, Expected May 2023.

Bachelor of Science Computer Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, June 2016
Publications:

- S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong and A. A. Ghorbani, "Towards the Development of a Realistic Multidimensional IoT Profiling Dataset," 2022 19th Annual International Conference on Privacy, Security & Trust (PST), 2022, pp. 1-11, doi:10.1109/PST55820.2022.9851966
- Ensemble-based Intrusion Detection for Internet of Things devices (Accepted by the **IEEE HONET 2022** IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT & AI)
- Transferability of Machine learning algorithm for IoT device profiling and identification (Under revision upon comments from **IEEE Internet of Things Journal**)
- Human-Centric Machine Learning: The Role of Users in the Development of IoT Device Identification and Vulnerability Assessment (Accepted at **HCI International Conference**)

Conference Presentations:

Ensemble-based Intrusion Detection for Internet of Things devices (Accepted by the **IEEE HONET 2022** IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT & AI), December 2022