Towards the Development of a Realistic Multidimensional IoT Profiling Dataset

Sajjad Dadkhah
Canadian Institute for Cybersecurity
University of New Brunswick
Fredericton, Canada
sdadkhah@unb.ca

Hassan Mahdikhani
Canadian Institute for Cybersecurity
University of New Brunswick
Fredericton, Canada
hmahdikh@unb.ca

Priscilla Kyei Danso
Canadian Institute for Cybersecurity
University of New Brunswick
Fredericton, Canada
priscilla.danso@unb.ca

Alireza Zohourian
Canadian Institute for Cybersecurity
University of New Brunswick
Fredericton, Canada
alireza.zohourian@unb.ca

Kevin Anh Truong
Canadian Institute for Cybersecurity
University of New Brunswick
Fredericton, Canada
kevin.tru@unb.ca

Ali A. Ghorbani
Canadian Institute for Cybersecurity
University of New Brunswick
Fredericton, Canada
ghorbani@unb.ca

2022 19th Annual International Conference on Privacy, Security & Trust (PST) | 978-1-6654-7398-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/PST55820.2022.9851966

***Abstract*—The Internet of Things (IoT) is an emerging technology that enables the development of low-cost and energy-efficient IoT devices across various solutions from smart cities to healthcare domains. With such a complex and heterogeneous instance of IoT devices and their applications, numerous challenges arise in both device management and security concerns. Thus, it is essential to develop intelligent IoT identification/profiling and intrusion detection components that are tailored to IoT applications. Such systems require a realistic and multidimensional reference IoT dataset for training and evaluation. Device identification/profiling ensures the authenticity of the devices attached to the IoT network and environment which can be achieved by fingerprinting a device. Since fingerprinting is mostly examined by device network flows and device local attributes, we have proposed this study to intelligently recognize machine-to-machine communication and identify each device properly. In this paper, we analyzed the behaviour of 60 IoT devices during experiments conducted in our lab setup at the Canadian Institute for Cybersecurity (CIC). Our IoT devices include WiFi, ZigBee, and Z-Wave devices. We collected data from each device in four stages: powered on, idle, active, and interactions. Besides these stages, different scenario experiments were conducted using a microcosm of devices to simulate the network activity of a smart home. Additionally, we have generated two attack datasets, namely flood denial-of-service attack and RTSP brute-force attack. Lastly, we implement an extensive case study on the transferability of the RF classifier and train our model with the dataset from our lab, transfer the model to the dataset from a different lab and test the trained model on their dataset. This paper's dataset materials are available on the CIC dataset page under the CIC IoT dataset 2022[1].**

I. Introduction

The Internet of Things (IoT) represents a vision in which the Internet extends into the real world, embracing every day objects where physical items are no longer disconnected from the virtual world but can be controlled remotely and can act as physical access points to Internet services [17]. IoT devices have provided significant improvements to the quality of life for consumers. The industry is experiencing rapid growth, with a projected 30.9 billion units by 2025, a sharp jump from the 13.8 billion units that are expected in 2021 [23]. This massive development of IoT devices has put a burden on the resource-constrained network, creating a testbed of low-powered and less secure devices for attackers [11]. The ecosystem of IoT devices is exceptionally diverse in functionality, from smart boards for education [20], wearables for monitoring health, and even smart fridges that remind you when food items will expire [5]. However, with great benefits comes significant security concerns [13]. Consumers of IoT devices typically deploy many smart devices [3] into their network, with ranging functionality and manufacturer. With such a heterogeneous environment, there is bound to be a device susceptible to some level of attack. The danger exposed by these Internet-connected Things not only affects the security of IoT systems but also compromises a single component, and communication channels in IoT-based systems can affect the part, or complete Internet network [7]. Unfortunately, these security concerns often go unnoticed [6] by the average consumer, leaving them vulnerable to potential bad actors. Active attackers can use the vulnerable device as a launchpad to invoke larger-scale attacks or more sinister actors can monitor your daily life through a vulnerable camera. For example, eavesdroppers can monitor their network traffic for a device sending personally identifiable information in plain-text traffic.

Malicious actors [21] are not the only concern for consumers. Even the manufacturers of these devices may be collecting information regarding the user, either for profiling or analytics. As an example, Alexa passively listens to your

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

conversations throughout the day, continuously monitoring for its wake word [2]. While this data does not go to cloud storage, this continuous listening is a potential gateway for eavesdroppers to peer into a user's daily life. Additionally, when Alexa does detect the wake word, Alexa begins a stream to the cloud, collecting user voice and their request. Analytic teams [16] may also listen to your conversations with Alexa that are stored in the cloud, with the stated purpose being to improve Alexa's speech recognition.

Therefore, there is a dire need to understand how these devices communicate with external parties, mainly user information. We have extensively gathered network traffic produced by IoT devices of varying categories and their network traffic made by various functionalities.

In addition to collecting data in a controlled environment, we also capture device traffic throughout the day. We allow fellow researchers during this period to enter the lab at any time they wish. They can either passively generate network activity from our devices as they do other things or actively interact with the devices.

Our main contributions of this paper are as follows;

- Configure various IoT devices from WiFi to Zigbee and Z-Wave and analyze the behaviour exhibited.
- Conduct manual and semi-automated experiments of various categories on these devices.
- Analyze the network traffic when the devices are powered on for the first two minutes and powered off for the next three minutes.
- Capture and analyze different device behaviour under several different scenarios using machine learning algorithms.
- Perform possible attacks and capture the corresponding data packets.

Lastly, in the spirit of a collaborative research effort, we make our dataset and all related documents available for download on the CIC dataset page under the CIC IoT dataset 2022.

II. The Goal and Scope

This paper focuses on capturing the network traffic produced by IoT devices in our lab and analyzing the device behaviour exhibited when in different states. Additionally, the network traffic was captured when attacks were performed on certain devices. Lastly, we conducted a case study on the idea of transferability - training data from one lab transferred to another lab for testing.

We ended up with discussions around some attacks and tools used to accomplish them as well as a detailed analysis of attacks against RTSP-enabled (Real Time Streaming Protocol) cameras.

A. Scope

What type of user information can IoT devices expose to the outside world?

Due to the resource-constrained nature of IoT devices, they have very limited capabilities [18]. We categorize the types of information into three based on the network traffic captured:

- *Device data:* This contains the information related to the device, including name, unique identifier, device state, logs, etc.
- *Sensor data:* This includes information generated by the sensors of an IoT device, e.g., contact, temperature, motion detection, etc.
- *Activity data:* This is all traffic generated from user interaction with the IoT. Either by using a companion application to turn off a light bulb or a voice assistant to issue a command to turn on a device or by physically interacting with the device.

To what entity was the information relayed?

For any relay of information by an IoT device to the outside world, this data is shared with various parties. We begin by identifying and categorizing the potential entities that these may contact.

- *Primary entities:* These are mostly the manufacturers of the IoT device or a group of companies responsible for maintaining the device and its functionality.
- *Support entities:* Any company or group of companies supporting the *Primary entities* with an aspect of their operation, such as cloud providers.
- *Non-Related entities:* Any entity that is not a primary or supports device functionality. Some examples of this are advertising or analytics companies, passive eavesdroppers like Internet Service Providers (ISPs), or malicious actors.

What are the risk and privacy concerns that need to be addressed?

Significant privacy concerns exist for devices that frequently contact non-related entities (NRE). Regular consumers of IoT devices do not delve into their device traffic, so they will not be aware that some of these entities have access to their data. Furthermore, the information regarding the communications with these NREs is often not explicitly conveyed to or quickly accessible by the user. The following are some examples of data that an NRE can collect:

- network traffic that contains personal information.
- network traffic that contains audio, video, or image recordings.
- motion, temperature, or other sensor data.

B. Research Questions

In this section, we address our key research questions:

1) How do devices behave in different times and scenarios?
2) Does a device expose information or behave unexpectedly?
3) To what extent can we transfer our trained model to a different setup to examine its generalization?
4) To what extent security measures have been implemented in the devices?
5) How are IoT devices with different protocols operating in the network in terms of IoT profiling and identification?

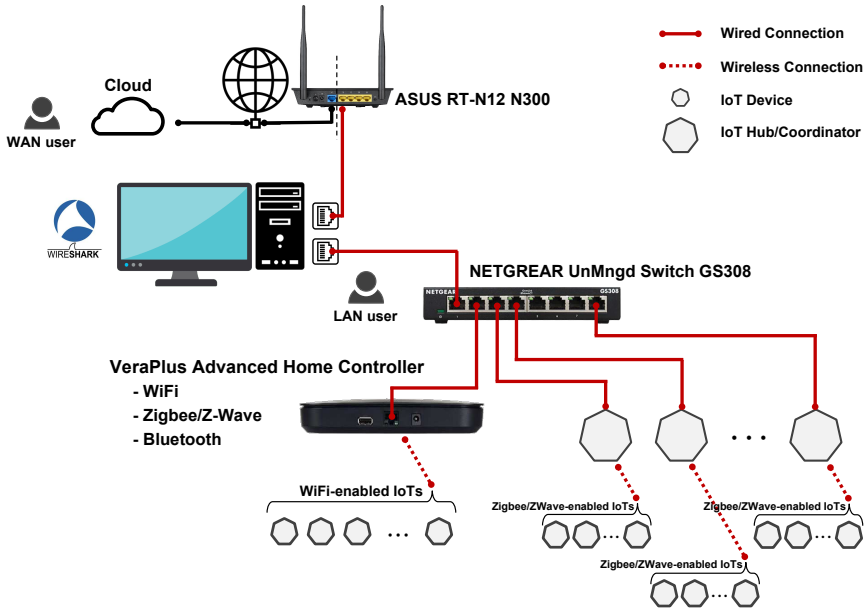


Fig. 1: Network configuration for capturing the IoT devices traffic

III. Data Collection Process

This section discusses our lab setup and configuration, the 60 IoT devices used, and different types of experiments conducted. These experiments were conducted at the Canadian Institute for Cybersecurity for three months.

A. IoT Lab Setup and Configuration

This section describes our lab's network configuration, the data collection process, and the devices' deployment. Additionally, we detail the six different categories of experimental methods. In total, we have conducted a series of reproducible experiments over a period of three months.

1) Network Configuration

Network Interface Cards (NIC). One NIC is connected to the network gateway, and the other is connected to an unmanaged network switch. IoT devices that require an Ethernet connection are connected to this switch. These devices are usually communication centers for other devices, examples being the Philips Hue Bridge and Eufy HomeBase. Finally, a smart automation hub, Vera Plus, is connected to the unmanaged switch, creating our wireless network. Vera Plus is compatible with Wi-Fi, ZigBee, Z-Wave, and Bluetooth BLE. However, it has been used as an Internet gateway and provides WiFi coverage for the WiFi-enabled IoT devices. Other smart hubs, including Philips Hue Bridge, SmartThings Hub, and Fibaro Home Center Lite, have been utilized to rule the Zigbee and Z-Wave IoT devices and sensors.

2) Data Collection

We automatically capture the network traffic via the gateway using Wireshark [7] and dumpcap [9]. Wireshark is used for our manual experiments, while dumpcap is used in semi-automated tests. As for Zigbee devices, we captured the network traffic using a Nortek HUSBZB-1 stick on an Ubuntu machine with wireshark. In the case of Z-Wave, we captured the netowrk traffic using an ACC-UZB3-E-STA stick on a Windows 10 machine with Zniffer software.

B. IoT devices

Our experiments and analyses were based on the 60 IoT devices configured in our lab, most from Canadian stores. We deployed a diverse range of IoT devices to understand how devices of different types produce traffic in isolation and how they may interact with other devices in the network.

C. Additional Tools

Our experiments involved manual and semi-automated tasks. For interaction tests utilizing voice assistants, we employed the Google Cloud Text-To-Speech synthesizer on its default settings. For interaction experiments via the companion apps, we used an iPad mini.

D. Experiments

We collected data for all the devices connected to the network. Depending on the experiment performed, the traffic was captured for the entire network (Active and Idle), a part of the network (Scenarios) or a single device, filtered by a distinctive ID such as a MAC address (Power, Interactions and Attacks).

Six different experiments were conducted on the devices:

1) Power

Before beginning any power experiments, all devices were unplugged from the power source, and the network was rebooted.

In this experiment, all of the devices in our environment were powered on individually, and the network traffic was captured in isolation. The device's MAC address is set in Wireshark as the capture filter, not to include data produced

Category	Device Name (#)	Connection Type	Brand
Camera	HeimVision Smart WiFi Camera	WiFi	HeimVision
	Netatmo Camera	WiFi	Netatmo
	Arlo Q Camera	WiFi	Arlo
	DCS8000LHA1 D-Link Mini Camera	WiFi	D-Link
	Borun/Sichuan-AI Camera	WiFi	BORUN
	AMCREST WiFi Camera	WiFi	AMCREST
	Shenzhen_73:f3:36 Home Eye Camera	WiFi	Shenzhen
	Luohe Cam Dog	WiFi	LUOHE
	SIMCAM 1S (AMPAKTec)	WiFi	SIMCAM
	Nest Indoor Camera	WiFi	Nest
Smart Lamp/Bulb	SmartThings Smart Bulb (5)	Zigbee	SmartThings
	Philips Hue White (2)	Zigbee	Philips
	HeimVision SmartLife Radio/Lamp	WiFi	HeimVision
	Globe Lamp ESP_B1680C	WiFi	The Globe Electric
Speakers	Amazon Echo Studio	WiFi	Amazon
	Amazon Echo Dot (2)	WiFi	Amazon
	Google Nest Mini	WiFi	Google
	Amazon Echo Spot	WiFi	Amazon
	Sonos One Speaker	WiFi	Sonos
Doorbell and Door/Window Sensor	Eufy HomeBase II	WiFi	Eufy
	Fibaro Door/Window Sensor (3)	Z-Wave	Fibaro
Base Station	Ring Base Station	WiFi	Ring
	Arlo Base Station	Wired	Arlo
Motion Sensor	AeoTec Motion Sensor	Zigbee	AeoTec
	Fibaro Motion Sensor (5)	Z-Wave	Fibaro
Actuator	SmartThings Button	Zigbee	SmartThings
	AeoTec Button	Zigbee	AeoTec
Multi-Sensor	AeoTec Multipurpose Sensor	Zigbee	AeoTec
Weather Station/Sensor	Netatmo Weather Station	WiFi	Netatmo
	AeoTec Water Leak Sensor	Zigbee	AeoTec
	Fibaro Flood Sensor (2)	Z-Wave	AeoTec
Vacuum	iRobot Roomba	WiFi	iRobot
Smart Hub	Vera Plus Hub	WiFi	Vera Control
	Philips Hue Bridge (2)	Zigbee	Philips
	SmartThings Hub Plug	Zigbee	Amazon
	Fibaro Home Center Lite	Z-Wave	Fibaro
Smart Plug	Yutron Plug (2)	WiFi	Yutron
	Amazon Plug	WiFi	Amazon
	Fibaro Wall Plug (2)	Z-Wave	Fibaro
	Teckin Plug (2)	WiFi	Teckin
Smart Coffee Maker	Atomi Coffee Maker	WiFi	Atomi
Smart TV	LGInnote_4e:00:82	WiFi	LGInnote
Smart Board	SMARTTec_f6:e3:cb	WiFi	SMART Technologies

TABLE I: List of IoT devices and their corresponding categories, Connection type and Brand. The numbers in parentheses are the quantity of devices or sensors we have.

by the router or other non-IoT devices that may have been connected to the network. We used a capture duration of two minutes, which is synchronized with the time the device is plugged into the power source. Once the time is up, the device is unplugged from the power source, and the capture continues for an additional three minutes to collect leftover packets. This leftover packet monitoring is repeated until there is no arrival of a new packet in the last three minutes. We performed the same process for all the WiFi, Zigbee, and Z-Wave devices. However, Zigbee and Z-Wave required more steps than Wi-Fi. Since Zigbee and Z-Wave are low-rate, their devices are not directly connected to the cloud, removing the power source will eliminate all traffic to and from the end device, except for a few periodic packets from the coordinator to mains-powered end devices, trying to find it. In the case of battery-powered devices, no residual packets will be sent, mostly not to wake them up periodically and consume their battery. Another exclusive experiment that was done on Zigbee/Z-Wave was capturing their association/inclusion process. Zigbee/Z-Wave devices will communicate with the coordinator/controller for authentication and key exchange. Therefore, we also captured the corresponding network traffic for these devices.

2) Idle

With our 60 devices, collecting idle data in complete isolation would be time-consuming and rigorous, as we had planned to capture idle time for 30 eight-hour periods. We assumed that devices in an idle state should not be producing communications with other devices in the network anyways. This is to check if the device is performing an unobservable activity that is unwanted. To mass-collect idle data, we prepared a batch file.

The batch file contains the script to collect each device's network traffic using dumpcap, with each instance being filtered by their respective MAC addresses. This script includes

a time delay to ensure the lab is empty and that the devices have enough time to reset their idle baseline. The time delay was five minutes, and the capture duration was set for eight hours. These captures often occurred at night, but there was also an occurrence on a holiday.

Additionally, we have captured 30 days' worth of idle traffic in a non-individualized format. With our prior assumption that devices in the idle state should not interact with one another, this data should be similar to the individualized capture. The only difference is that packets from various IoT devices may interleave with each other in the capture file.

3) Interactions

Interaction experiments differ slightly from the way power experiments were conducted. These experiments vary slightly in the capture duration, depending on how long it takes to complete certain activities. However, they are all consistent in that after the activity is completed, capture continues for an additional 15 seconds to capture any delayed packets that remain from the activity. Possible interactions via the companion app were discovered by either looking through the documentation provided by the manufacturer or experimenting with the app.

Four types of Interactions were considered:

- Physical (Using buttons or other manual commands on device or using native voice commands),
- LAN App (Using companion app on a mobile device, while being on same network as the IoT device),
- WAN App (Using companion app on a mobile device, while being on different network from the IoT Device), and
- Voice (Using a voice assistant, like Google assistant or Amazon Alexa, to command the IoT Device).

4) Scenarios

Scenario experiments were conducted using a microcosm of devices to simulate the network activity of a smart home. Additionally, this process provides data on how devices may interact with each other as activities occur simultaneously. There are six different types of experiments conducted. This testing used a batch file similar to the Idle experiments, although the duration for leaving and capturing is set to one minute.

- Coming Home: Start with Ring armed in Away Mode and devices off to simulate an inactive home.
  - Enter the lab. This will trigger the Ring alert.
  - Trigger Netatmo Camera and ArloQ motion detection.
  - Disarm Ring by entering passcode into the keypad.
  - Tell Google Nest Mini to turn on the Philips Hue Lights.
  - Tell Google Nest Mini to turn on the Globe Lamp.
  - Tell Alexa to start the Atomi Coffee Maker.
  - Tell Alexa to play music from the Sonos Speaker.
  - Start the Roomba cleaning procedure using LAN App connection.
- Leaving Home: Start with all devices on to simulate an active home.
  - Tell Alexa to turn off the Atomi Coffee Maker.
  - Tell Alexa to stop playing music.
  - Tell Google Nest Mini to turn off the Globe Lamp.
  - Tell Google Nest Mini to turn off the Philips Hue Lights
  - Send Roomba home using LAN App Connection
  - Leave the lab, triggering Netatmo Camera and ArloQ motion detection.
- Home Intrusion (Day): Simulate an attack on the home with sufficient lighting. Begin with Ring armed in Away mode.
  - Enter the lab. This will trigger Ring alert.
  - Trigger Netatmo Camera, ArloQ, and HeimVision Camera motion detection.
  - Trigger D-Link Water Sensor.
  - Allow Ring alarm to sound off.
  - Exit the lab, once again triggering Netatmo and ArloQ motion detection.
- Home Intrusion (Night): Simulate an attack on the home with no lighting. This testing is to test Night Vision detection from cameras with that support and sound detection for cameras without it. Begin with Ring armed in Away mode.
  - Enter the lab. This will trigger Ring alert.
  - Trigger Netatmo Camera motion detection. Trigger Arlo Q and HeimVision Camera sound detection.
  - Trigger D-Link Water Sensor.
  - Allow Ring alarm to sound off.
  - Exit the lab, once again triggering Netatmo and Arlo Q motion detection.
- Owner Error: This test is to simulate a false intruder (i.e. the owner misinputs the Ring passcode)
  - Enter the lab. This will trigger Ring alert.
  - Trigger Netatmo Camera and Arlo Q motion detection.
  - Misinput the Ring passcode 3 times within a 30 second window.
  - Allow Ring alarm to sound off.
  - Enter the correct passcode.
- IoT vs Non-IoT: This test is to analyze the integration of IoT and Non-IoT devices.
  - Tell Alexa to play music from the Sonos Speaker.
  - Have LG TV play a YouTube video.
  - Browse social media from the phone.
  - Browse Amazon from the computer.
  - Tell Google to turn on Philips Hue Lights.

E. Active

In contrast to our 30-day Idle Capture, we have also captured 30 days of capture in an Active State. We define the Active State as the period in the day when devices produce network traffic from interactions, as people sporadically enter the lab and either actively or passively interact with the

devices. Active interactions may include turning on Smart Lights, using Voice Assistants, or using the Smart TV. Examples of passive interaction are triggering Ring contact sensors or triggering various camera's motion or sound detection as they do other activities in the lab.

F. Attacks

In this experiment, we performed two different attacks on some of our devices, namely Flood attacks and RTSP Brute Force attacks. The former was done using Low Orbit Ion Cannon (LOIC) to carry out DoS attacks using HTTP, UDP, and TCP protocols. The latter was done, using Hydra and Nmap, to find the cameras' RTSP URLs by brute-forcing. The discovered URLs were then used to watch the video feed of the camera in real-time.

IV. Anomalous Devices

Some devices produced anomalous behaviour as we conducted our experiments. They may be of note as potential security concerns or unexplained behaviour.

- *Amazon Echo Spot:* During the Power Experiment procedure, we noticed a considerable amount of leftover packets arriving at the device. They were TCP re-transmission packets.
- *Gosund Smart Plug(s):* After a considerable time after a disconnection in the Power Experiment procedure, the device would receive a transmission from its cloud server. Additionally, one of our plugs suddenly changed transmission protocols when being turned on or off by the companion app. We had noticed prior to that the device was utilizing TLS and TCP protocols, but suddenly started using POP protocol. Resetting the device resolved this issue.
- *Netatmo Weather Station:* This device does not seem to produce any traffic despite interacting with it on the app, both in LAN and WAN connection settings.
- *Luohe CamDog:* During the Interaction Experiment procedure, we noticed that LAN app connections did not produce any traffic.

V. IoT Vulnerability Attacks

Additionally, we conducted a few attacks on some of our devices and captured the corresponding network traffic. The following details the successful attack attempts, organized in different attack tools.

A. Low Orbit Ion Cannon (LOIC)

The Low Orbit Ion Cannon is a targeted DoS attack targeting URLs or IP addresses. Additionally, the tool allows the attacker to specify the port to be attacked and the protocol type (TCP/UDP/HTTP) for the packet flood. For our experiments, we use a constant port value of 80. For each protocol type, we do three captures. We employ the companion app for the respective device under a LAN App connection.

- Cameras: *HeimVision Smart Wi-Fi Camera*
  This camera sees complete denial of service when flooded with packets of any protocol.
- Smart Lamp: *Globe Lamp*
  Under a TCP Flood attack, the device does not see any service issues, and the TCP flood is consistently blocked after roughly 40 seconds. Under an HTTP flood, there is no service interruption, despite the continual flow of HTTP packets. Lastly, a UDP flood creates significant service delays, and occasionally device becomes unresponsive.
- Base Station: *Ring Base Station*
  TCP flood packets are blocked within 5 seconds of starting. There were no observed functional issues. Under an HTTP flood, there is no functional error, but the packets continue to flow. Lastly, under a UDP flood, there are observed functional issues. Application has been observed to have significant delays and occasionally become non-responsive.
- Smart Hub: *Philips Hue Bridge*
  Sees significant functional errors as a result of the TCP flood. Turning On/Off the light under this attack may see delays, unresponsiveness, or a change in status on the app without a change in the bulbs. Furthermore, the companion app itself occasionally enters a re-connection state. Surprisingly, in comparison to the other devices tested, the UDP flood did not create any service issues.
- Smart Plug: *Gosund Smart Plug*
  Under a TCP Flood attack, the device does not experience any service issues, and the TCP flood is consistently blocked after 5 seconds. The HTTP flood also did not create any service issues for the device, but HTTP packets continually flowed compared to the TCP flood. While under a UDP flood, the device was completely unresponsive. Periodically, the device would send ICMP packets, briefly restoring functionality until the UDP flood continues.
- Smart Coffee Maker: *Atomi Coffee Maker*
  Under a TCP flood attack, the device does not experience any service issues. Additionally, the flood is blocked after 30 seconds. HTTP flood does not create any service issues despite continuous packet flow. Lastly, the UDP flood makes the companion app unresponsive to Start/Stop commands. Like the smart plug, the device would send ICMP packets, briefly restoring device activity before the flood continues.

B. nmap/Hydra RTSP URL Brute Force Attack

Nmap utilizes a script called "drtsp-url-brute" to conduct this attack. Hydra also utilizes this script when performing a similar attack. This attack takes advantage of listening 554 ports, typically associated with IP cameras. As a result, the data collected for this attack will not be categorical and will focus on vulnerable IP cameras.

- *Luohe CamDog:*
  After a successful RTSP Brute Force attack using nmap, the resulting RTSP URLs were inserted into a video

player and the non-protected live footage was played through an unauthorized device.

- *HeimVision Camera:*
  The same thing as Luohe CamDog happened with this camera.
- *Amcrest Camera:*
  After a successful RTSP Brute Force attack using nmap, the resulting RTSP URLs were inserted into a video player. However, a default password protected the device. This default password is simple to find, as it is listed on the Amcrest support page.
- *SimCam Camera:*
  Similarly to the Amcrest Camera, this camera provided the RTSP URLs when the nmap script is called and was protected by a password. However, the SimCam password is not a simple default password, and is only accesible through the app.

VI. Case Study

IoT network traffic was collected from our lab located at the Canadian Institute for Cybersecurity for the training process in this study. For the testing process, we used a different dataset from another lab in another country to test our case study of transferability. The network in each lab had a different configuration, setup, and IoT devices. However, despite the different devices in each lab, both labs had common IoT device types, namely Audio, Camera, and Home automation; the complete list of devices is provided in Table IV. Within both datasets, devices with identical makes and names can be found. However, there is a possibility the device models are not identical, as well as the possibility that they operate differently due to regional variations. Several different device types are used in the lab to train the classifier. Using a wide variety of IoT devices, which were partially common between the two labs, enabled us to assess the transferability of findings; in other words, comparing the classifiers' performance that was trained with data from IoT devices in our lab and tested with data from IoT devices operated by other people in a geographically remote lab [19].

We retrained our classifier five times to see how varying the predictions would be for each experiment due to the 80-20 split conducted during training. In each experiment, we trained the classifier on data from our lab; then, we tested the classifier on each device(for instance, Amazon Echo dot) from US lab [22] and examined its ability to detect the device type as an Audio device.

A detailed description of our proposed methodology is as follows:

- *Data Acquisition:* Our framework will require a training and test dataset. Hence, we will use our dataset for training our model and use the other lab's dataset [22] for testing our trained model.
- *Feature Extraction:* Relevant features are extracted from the data acquisition stage. In other words, the same features were extracted from both datasets. Each extracted field refers to a feature [12]. We select the most efficient collection of features for classifying each device type with the highest predictability and efficiency based on features that yield low false positive and high accuracy. The features include IP packet information, as well as statistical features of these packets.
- *Model Training:* To be able to identify and infer a device type, we train our machine learning algorithm using the features extracted during the feature extraction phase. IoT devices behave differently; even devices that belong to the same category. Hence devices generate different patterns. The different patterns exhibited by the various IoT devices help properly train our Machine Learning classifier.
- *Evaluation* At this stage, we analyze and evaluate our proposed transferability and how the Random Forest (RF) classifier performed in identifying devices and their respective device type.

Device type	Number of records
Audio	10,000
Camera	10,000
Home Automation	10,000
Total number of records for training	**30,000**

TABLE II: Device type and their corresponding number of records used for training in our experiment

A. Data Acquisition

At this stage, we collected data of a certain number of records for each device type. To avoid overfitting and imbalanced dataset, we made sure each device type had the same number of records as seen in Table II

We used the interaction experiment conducted in our case study. Three device categories are used in this case study: Audio, Camera, and Home automation.

In this section, the benchmark dataset used for the evaluation of our proposed transferability approach using various machine learning methods is described.

1) Consumer IoT Dataset

The dataset consists of network traffic(packet headers) from 34,586 controlled experiments and 112 hours of idle IoT traffic. Four different experiments were conducted; power, controlled, interaction, idle and uncontrolled experiments. *Power* experiments consist of powering on the device and collecting network traffic. *Interaction* experiments consist of actively interacting with IoT devices. *Idle* experiments capture the traffic of an IoT device when it is not actively used. Uncontrolled experiments was performed in the US only where 36 user study participants are allowed to use the IoT devices for their intended purpose in a studio apartment setting.

The dataset was generated from 81 IoT devices with IP connectivity: 46 procured from US stores (US devices) and deployed in our US testbed, 35 procured from UK stores (UK devices) and deployed in our UK testbed. There are 26 common devices across the two labs with the same model name in both labs. The authors categorized the devices into

L4_tcp	total_length	L3_ip_dst_count	sum_et
L4_udp	protocol	ethernet_frame_size	min_et
L7_http	source_port	most_freq_d_ip	max_et
L7_https	dest_port	most_freq_prot	med_et
port_class_src	DNS_count	most_freq_sport	average_et
port_class_dst	NTP_count	most_freq_dport	skew_et
pck_size	ARP_count	epoch_timestamp	kurt_et
ip_dst_new	var	inter_arrival_time	sum_e
cnt	q3	time_since_previously_displayed_frame	min_e
ttl	q1	q1_e	max_e
med	iqr	iqr_e	average
skew_e	kurt_e	var_e	q3_e

TABLE III: 48 Features extracted for profiling and classification

the following : cameras , smart hubs, home automation, TVs, audio, and appliances [22].

B. Feature Extraction

As we have earlier stated in our methodology, we trained our model with the dataset from our lab and tested the trained model with the dataset from the other lab. Due to the approach of our work, we need to derive the best features for differentiating each device type. Despite some common devices, the model, year of manufacture, and other physical characteristics are unknown about the other lab's devices, which can influence our model. Hence, we cautiously extracted features that uniquely profile the devices and the device type they belong to.

We extracted a total of 48 features from the network data captured for each device. Table III has the list of features used in our experiments.

All features were extracted using a Python packet manipulation tool called dpkt [1]. 48 features were extracted from each device's network capture. The final format of this capture is a CSV containing all features and the corresponding label of the device type for that file.

Classifier	Accuracy %	Training time(s)	Test time(s)
Guassian NB	62.9	2.354	0.311
Decision Tree	98.5	20.703	0.104
LDA	87.5	11.073	0.134
Ada Boost	98.7	387.078	5.424
Ridge	54.0	53.334	0.067
Perceptron	50.4	5.109	0.504
Passive-Aggressive	40.0	4.334	0.062
XGBoost	98.6	158.553	0.340
KNN	95.6	1.375	2328.246
RF	98.5	283.353	3.626
LinearSVC	50.9	6.113	0.074
SGD	34.6	19.539	0.060

TABLE IV: Different classifiers trained on our dataset with the corresponding accuracy, training time and test time

C. Model Training

We trained our dataset with different algorithms to find the most suitable classifier that works best on our dataset based on the accuracy, training and test time. Selecting, training, and tuning the machine learning model is critical as it impacts the performance of the device identification [8].

Based on the results, we decided to go with RF because we needed a classifier that would generalize, and have a higher accuracy and less prediction time on the testing dataset. The two main reasons for selecting RF [14] are as follows:

- RF classifier is computationally efficient and does not overfit [4].
- Random behavior of RF in selection of data and features can improve the performance of classification hence the results in Fig. 2b.

RF is a supervised algorithm that builds multiple decision trees, each of which is an individual classifier that increases the overall accuracy. Each decision tree only takes a random subset of features as input. The general classification is determined by a majority vote, meaning that more than half should be incorrect for the overall classification to be wrong. RF performs weak when there are many irrelevant variables. Although this lowers the chance of selecting a relevant variable, what makes this approach advantageous is that it makes it possible to see how each feature affects the final classification and determine how important each feature is [10].

D. Evaluation

To choose the best ML classification technique for our case study, we tested several machine learning classification techniques and compared them with validated accuracy. We used Jupyter Notebook[2] for our python implementations and Scikit-learn[3] library to implement, train and test our dataset. In this study, we performed an 80-20 split. 80% of our data was used to train our model, and 20% was used to test and evaluate our trained model.

1) Data Acquisition

We used the publicly available consumer IoT devices dataset [22] as our test data for this experiment. The dataset had both devices from the US and UK markets. However, in this case study, we used the US dataset[4]. Table V shows an overview of the devices sampled from the US dataset used in our study.

2) Performance Metrics

Evaluating classifiers' performance is a critical step in the construction and selection of classification models. The metrics such as accuracy, precision, recall, and f1-score are calculated from the RF classifier [15]. We define the parameters used in evaluating our metrics.

True Positives (TP) are the values that are positive and are predicted as positive by the model.

False Negatives (FN) are the values that are positive but are predicted to be negative by the model.

False Positives (FP) are the values that are negative but are predicted as positive by the model.

[2]https://jupyter.org/

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

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

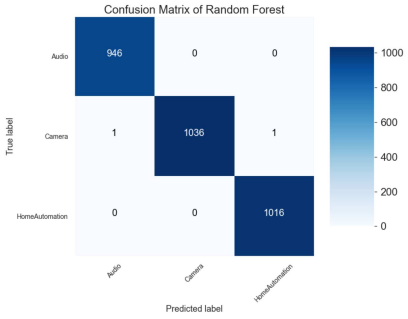
Training Dataset (CIC Lab)	
Device Type	**Device No. and Name**
Audio	01. Amazon Echo Dot
	02. Amazon Echo Spot
	03. Amazon Echo Studio
	04. Google Nest Mini
	05. Sonos One
Camera	06. Amcrest Camera
	07. ArloQ Camera
	08. Borun Camera
	09. DLink Camera
	10. HeimVision Camera
	11. HomeEye Camera
	12. Luohe Camera
	13. Nest Camera
	14. Netatmo Camera
	15. SimCam
Home Automation	16. Arlo Base Station
	17. Amazon Plug
	18. Atomi Coffeemaker
	19. Eufy Homebase
	20. Globe Lamp
	21. Gosund Plug
	22. Heimvision Lamp
	23. Philips Hue
	24. Ring Basestation
	25. Roomba vacuum
	26. Smartboard
	27. Teckin Plug
	28. Yutron Plug

(a) Training Dataset (CIC Lab)

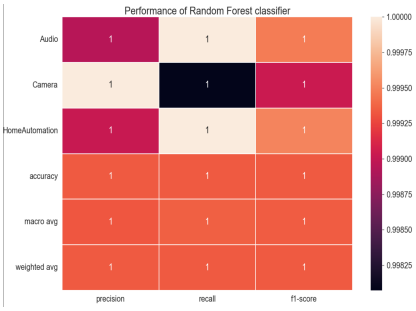
Testing Dataset (US Lab)	
Device Type	**Device No. and Name**
Audio	01. Amazon Echo Plus
	02. Amazon Echo Dot
	03. Amazon Echo Spot
Camera	04. Amcrest Camera
	05. Cloud Camera
	06. Lefun Camera
	07. Luohe Camera
	08. WansView Camera
	09. Yi Camera
Home Automation	10. T-Wemo Plug
	11. TP-Link Bulb
	12. Bulb 1
	13. Blink Hub
	14. Insteon Hub
	15. Lightify Hub
	16. TP-Link Hub
	17. Brewer
	18. iKettle
	19. Xiaomi Cleaner
	20. Xiaomi Rice Cooker

(b) Testing Dataset (US Lab)

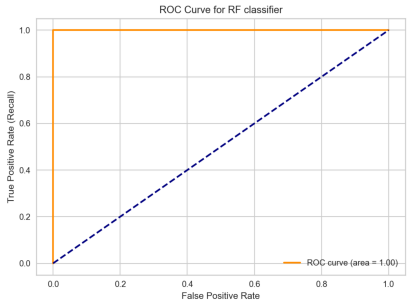
TABLE V: Different devices per category from the CIC lab used for training and devices sampled from the US lab used for testing



(a) Confusion matrix of RF



(b) Classification Performance by RF



(c) Receiver Operating Characteristic (ROC) curve

Fig. 2: Performance of RF on CIC dataset

No.	Device namec	Device Type	Device Type Inferred by the RF classifier					% IP by RF classifier				
			Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5
1	Amazon Echo Plus	Audio	Audio	Audio	Audio	Audio	Audio	60.5	78.7	82.87	87.07	71.12
2	**Amazon Echo Dot**	**Audio**	Audio	Audio	**Audio**	**Audio**	**Audio**	**67.22**	**78.11**	**56.18**	**88.28**	**83.53**
3	**Amazon Echo Spot**	**Audio**	Audio	Audio	**Audio**	**Audio**	**Audio**	**66.88**	**81.57**	**88.36**	**91.15**	**75.44**
4	**Amcrest Camera**	**Camera**	**Camera**	**Camera**	**Camera**	**Camera**	**Camera**	**99.63**	**99.61**	**99.40**	**99.74**	**99.41**
5	Cloud cam	Camera	Camera	Camera	Camera	Camera	Camera	94.46	95.74	89.47	96.31	93.88
6	Lefun Camera	Camera	Camera	Camera	Camera	Camera	Camera	92.14	97.64	93.39	96.00	95.65
7	**Luohe Camera**	**Camera**	**Camera**	**Camera**	**Camera**	**Camera**	**Camera**	**99.59**	**99.77**	**99.32**	**99.47**	**99.72**
8	Wansview Camera	Camera	Camera	Camera	Camera	Camera	Camera	99.06	99.15	99.06	99.43	98.56
9	Yi Camera	Camera	Camera	Camera	Camera	Camera	Camera	99.44	99.83	99.52	99.61	99.73
10	T-wemo plug	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	86.01	75.68	86.66	64.76	67.75
11	Bulb1	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	100	100	99.98	100	100
12	Tplink Bulb	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	99.72	99.53	99.27	99.45	99.39
13	Blink Hub	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	97.04	96.17	97.57	93.91	98.78
14	LightifyHub	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	99.66	99.45	99.72	99.48	99.66
16	Insteon Hub	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	99.29	99.07	99.60	98.76	97.15
17	Tp link hub	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	80.74	78.88	82.02	69.75	75.01
18	Brewer	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	99.27	99.76	99.39	97.60	99.39
19	iKettle	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	100	99.95	100	99.82	99.91
20	Xiaomi cleaner	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	HomeAutomation	83.68	83.16	81.84	85.53	70.26

TABLE VI: Experimental result of 5 trials for device type inference by RF and the % of inference of 5 experiments on the US dataset.The highlighted are devices common in both labs

True Negatives (TN) are the values that are negative and are predicted to be negative by the model.

Accuracy, precision, recall, and f1-score are metrics used for evaluating how well our model performed on the dataset from our lab. We introduced another metric called the *Inference Percentage* which was used to identify the device type based on the prediction of the classifier when transferred to the US lab [22]. The metrics and their corresponding definition are shown below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$
$$F1-score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Inference percentage, IP: This is calculated for each prediction made per device tested. After the classifier prediction, each device type and the corresponding predicted true positive and false negative values are retrieved and ordered in descending order of magnitude. This device type with the highest value in the distribution of the test dataset is inferred as the type of the said device been tested on. For instance, 2991 of the 3000 records used for testing were classified as Cameras(True Positive) values, while nine were mis-classified as Home Automation(False Negative). Therefore, in Table VI, we only record the highest inference percentage of the corresponding device type that classified predicted the device to belong for all five trials conducted. However, there are instances where the device type inferred is wrong. One of the reasons for misclassification can be the device model despite the device being from the same vendor because this information from the other lab is unknown.

3) Performance Evaluation

We analyze different techniques based on the accuracy, training time and testing to select the best algorithm to use in our study. From Table IV, 12 algorithms were reviewed with highest accuracy greater than 98% been Decision Tree(DT), RF, XGBoost, and AdaBoost. The training time for XGBoost, and AdaBoost are high despite less test time for XGBoost. Boosting algorithms are generally computationally expensive despite reducing training errors. Decision Tree despite having good accuracy, less training and test time can be highly unstable with a small change affecting the overall structure of the tree. Hence, RF the ideal approach used in this study with reason stated in Section VI-C

Figure 2 shows our model's evaluation performance in classifying different device types in the dataset from our lab. Figure 2a, 2b, and 2c respectively show the confusion matrix, the classification performance, and receiver operating characteristic (ROC) curve obtained from RF classifier for device type classification. From Figure 2a we can observe that 2 cameras were mis-classified as Audio and Home automation devices. The Performance of the classifier on the individual devices in Fig. 2b shows Audio, Camera and HomeAutomation had a precision of 99.99%, 100%, and 99.90, recall of 100%, 99.83%, and 99.99% and F1−score of 99.95%, 99.99%, and 99.95% respectively for each for each device type. Lastly, our ROC curve Fig. 2c graph shows how well the RF model performed at all the classification thresholds.

4) Experimental Results

In this section, we highlight the results of our classifier to uniquely infer the device type of each US IoT device using the trained model from our lab. We trained our model on 28 IoT devices from our lab; 11 cameras, five speakers, and 11 home automation devices. We then tested our trained model on 20 devices, four of which were common in both labs, and 16 devices were devices we had the categories in our lab but different devices altogether. From the experimental results Table VI, our classifier, after being tested on the other lab's dataset, was able to correctly infer all the devices to belong to their respective device types. The highest inference was Yi camera which was 4 out of the five trials conducted, the classifier accurately inferred the device type as a camera. Moreover, the Amazon echo dot is common in both labs. It received an average inference of about 74.66%. The reason for this could be different firmware, device model, etc. Lastly, it can be seen from Table VI that camera had better inference results than the other two categories. The most prevalent protocol used in cameras is the connectionless User Datagram Protocol(UDP), and our feature set took into this feature, which is distinct to cameras, hence the higher inference than the other two device types.

VII. Conclusion and Future Works

We conducted a series of experiments on 60 IoT devices and sensors with WiFi, Zigbee, and Z-Wave connectivity. Utilizing the data generated from the experiments, we carried out a case study on transferability - where the machine learning model is trained against our captured dataset, then transferred to an external dataset for device type identification testing. To the best of our knowledge, our case study is the first in this direction, where testing is peformed on devices that have not been in the training set, as well as from different geographical locations. We subsequently extracted 48 features from the network packets. For the purpose of our case study, we tested the RF classifier for training the dataset from our lab and testing the dataset from the US lab [22]. From our experimental results, our model was able to correctly infer all the device types each device belongs to, with the camera having higher inference results than Audio and Home automation.

In the future, we intend to test our case study on other data sets and the other devices not sampled from the [22] dataset. We hope to extend our work, on devices with Zigbee and Z-wave protocol regarding IoT profiling/identification and intrusion detection components. Furthermore, we plan to expand the classes (Audio, Camera, and Home Automation) used in this experiment to a more granular form (Audio, Camera, Smart hub, Appliance, Bulb, and Plug). Lastly, we aim to create a comprehensive dataset of IoT attacks to help experiment with anomaly detection of benign and malicious traffic.

References

[1] Python packet creation library, https://code.google.com/p/dpkt/, 2022.

[2] The Ambient. Is alexa always listening? is your smart speaker spying on you?, https://www.the-ambient.com/guides/does-amazon-alexa-echo-speaker-listen-conversations-2785, 2022.

[3] Eirini Anthi, Lowri Williams, Małgorzata Słowińska, George Theodorakopoulos, and Pete Burnap. A supervised intrusion detection system for smart home iot devices. *IEEE Internet of Things Journal*, 6(5):9042–9053, 2019.

[4] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016.

[5] Manuele Bonaccorsi, Stefano Betti, Giovanni Rateni, Dario Esposito, Alessia Brischetto, Marco Marseglia, Paolo Dario, and Filippo Cavallo. 'highchest': An augmented freezer designed for smart food management and promotion of eco-efficient behaviour. *Sensors*, 17(6), 2017.

[6] Tamara Bonaci, Linda Bushnell, and Radha Poovendran. Node capture attacks in wireless sensor networks: A system theoretic approach. In *49th IEEE Conference on Decision and Control (CDC)*, pages 6765–6772, 2010.

[7] 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.

[8] 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.

[9] Rohan Doshi, Noah Apthorpe, and Nick Feamster. Machine learning ddos detection for consumer internet of things devices. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 29–35, 2018.

[10] Owen Dwyer, Angelos K. Marnerides, Vasileios Giotsas, and Troy Mursch. Profiling iot-based botnet traffic using dns. *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.

[11] Masoud Erfani, Farzaneh Shoeleh, Sajjad Dadkhah, Barjinder Kaur, Pulei Xiong, Shahrear Iqbal, Suprio Ray, and Ali A. Ghorbani. A feature exploration approach for iot attack type classification. In *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 582–588, 2021.

[12] Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, and Antonio Robles-Kelly. Deep learning-based intrusion detection for iot networks. In *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 256–25609, 2019.

[13] Qi Jing, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. Security of the internet of things: perspectives and challenges. *Wireless Networks*, 20:2481–2501, 2014.

[14] XuKui Li, Wei Chen, Qianru Zhang, and Lifa Wu. Building auto-encoder intrusion detection system based on random forest feature selection. *Computers & Security*, 95:101851, 2020.

[15] Yangguang Liu, Yangming Zhou, Shiting Wen, and Chaogang Tang. A strategy on selecting performance metrics for classifier evaluation. *International Journal of Mobile Computing and Multimedia Communications*, 6:20–35, 10 2014.

[16] Imran Makhdoom, Mehran Abolhasan, Justin Lipman, Ren Ping Liu, and Wei Ni. Anatomy of threats to the internet of things. *IEEE Communications Surveys Tutorials*, 21(2):1636–1675, 2019.

[17] Friedemann Mattern and Christian Floerkemeier. *From the Internet of Computers to the Internet of Things*, pages 242–259. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[18] 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.

[19] 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.

[20] Andrew Meola. Applications of iot technology in the education sector for smarter schooling, 2022.

[21] Eid Rehman, Arif Malik, Tehmina Karamat, Aaqif Abbasi, Seifedine Kadry, Muhammad Khan, and Seungmin Rho. Intrusion detection based on machine learning in internet of things, attacks, counter measures. *The Journal of Supercomputing*, pages 1–29, 01 2022.

[22] 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.

[23] Lionel Sujay Vailshery. Iot and non-iot connections worldwide 2010-2025, 2021.