Applying IoTDevID to a New Dataset: the CIC-IoT-2022 Case Study

Kahraman Kostas, Mike Just, and Michael A. Lones

***Abstract*—In this paper, we have examined under various headings the aspects to be considered in device identification studies using machine learning methods, common mistakes that may occur and how to avoid them. Our paper briefly touched upon the following topics: identification methods and their pros and cons, available data types and properties, common mistakes made during feature extraction and their solutions, what to consider about the use of machine learning methods, and how to choose appropriate evaluation methods.**

***Index Terms*—IoT security, IoT fingerprinting, machine learning, device identification**

I. Introduction

Especially in the last two decades, we have heard the concept of the Internet of Things a lot. IoT can be defined as any kind of physical device with processing capability that can be connected to the internet or other devices [1]. IoT, which acts as a link between the digital world and the real world, are becoming increasingly present in our lives every day. Not only in our digital life on the computer/phone but also as a part of our home, work or street, they permeate every point of our physical lives. Today, the number of IoT has exceeded 10 billion, which is expected to reach 27 billion by 2025 [2].

This rapid progress brings along many problems. In a rapidly growing market, a variety of devices have been developed by many companies for many purposes in a short time. Due to the nature of IoT devices, these devices have very different hardware and software characteristics. For example, a smart kettle and a smart door lock are vastly different from each other, even though they are both IoT. The heterogeneity of these devices, combined with vulnerabilities from manufacturers and the often unfamiliar interfaces of the devices, make them potentially dangerous. According to statistics, an IoT device connected to the internet is attacked within 5 minutes and becomes the target of a specialised attack within 24 hours [3].

In order to cope with these attacks, it is essential to keep the devices up to date, identify the vulnerabilities they carry and find solutions for them. These devices may need to be updated, restricted or isolated from other devices depending on the vulnerabilities they carry. In any measure to be taken, the first step will be to identify the device. Necessary measures can be taken for the identified device by scanning for vulnerabilities from a source such as the CVE [4] database. However, the heterogeneous structure of IoT devices makes the device identification process very difficult. In this regard, many researchers are applying machine learning-based identification for more efficient solutions.

We created IoTDevID [5] to address the device identification problem. IoTDevID works at the individual packet level to identify IoT devices, whether IP or non-IP (such as ZWave, ZigBee, or Bluetooth). In doing so, it provides a high detection rate thanks to its incorporated aggregation algorithm, overcoming the disadvantage of low performance of using individual packets.

IoTDevID uses only publicly available data and its scripts is also publicly available [1]. It is therefore a transparent, reliable and repeatable study.

In the multi-layer feature selection process, device and session-based identifying features that cause over fitting are discarded, and the most appropriate feature set is created by using the genetic algorithm. In this context, IoTDevID provides generalisable and robust models. In this study, we will validate our previously created IoTDevID by applying it to a new dataset, the CIC-IoT-22 dataset. In this way, we will test the robustness and generalisability of our method with another dataset.

II. Related Work

In this section, we will review some studies in the literature on device identification using machine learning. Device identification aims to classify devices by using feature sets (fingerprints) obtained from network data as input. These features are usually derived from individual packet headers or payload, but some studies have also used flow features. There are three different approaches to the classification process [6]:

Unique Identification: By accepting each of the devices as unique, a separate class is created for each device [7].

Type Identification: Identification is performed according to the device type. If there are multiple devices of the same brand and model, they are seen as a single class [8].

Class Identification: Different devices that are not the same but have similar features, are gathered under a single class such as camera, speaker, or smart lamp [9]. Fig. 1 shows the labelling of CIC-IoT-22 dataset with three viewpoints. In the example containing 20 devices in total, 20 different labels are formed in the unique method, 13 different labels in the type method, and 3 different labels (Smart Lamp/Bulb, Speakers, and Smart Plug respectively) in the class method.

In parallel to these three approaches, we can analyse the literature as follows. In order to perform a unique classification, a dataset must have more than one device of the same make

K. Kostas, M. Just, and M. A. Lones are with the Department of Computer Science, Heriot-Watt University, Edinburgh EH14 4AS, UK, e-mail: kk97, m.just, m.lones@hw.ac.uk

Kahraman Kostas supported by Republic of Turkey - Ministry of National Education

[1]Materials available at: github.com/kahramankostas/IoTDevID-CIC.

Labels			
Class	Type	Unique	Devices
1	1	1	SmartThings Bulb 1
		2	SmartThings Bulb 2
		3	SmartThings Bulb 3
	2	4	Philips Hue White 1
		5	Philips Hue White 2
	3	6	HeimVision Lamp
	4	7	Globe Lamp
2	5	8	Amazon Echo Studio
	6	9	Amazon Echo Dot 1
		10	Amazon Echo Dot 2
	7	11	Google Nest Mini
	8	12	Amazon Echo Spot
	9	13	Sonos One Speaker
3	10	14	Yutron Plug 1
		15	Yutron Plug 2
	11	16	Amazon Plug
	12	17	Fibaro Wall Plug 1
		18	Fibaro Wall Plug 2
	13	19	Teckin Plug 1
		20	Teckin Plug 2

Fig. 1: The change of device labels of the CIC dataset according to 3 identification methods. In the unique method, each device is considered unique and labelled with this name, regardless of other devices. In the Type method, devices with the same brand and model are considered as a single device and share the same label. In the Type method, three devices that were originally labelled separately as SmartThings Bulb 1-2-3 are collected under the SmartThings Bulb label. In the Class method, devices with the same function are grouped under a single tag. For example, in the class column, 1,2,3 represent the categories Smart Lamp/Bulb, speaker, and smart plug respectively..

and model. Hamad et al. [7] used the Aalto dataset which is suitable for this task in their work aiming at a unique classification. However, the unique classification cannot be achieved by using the individual packet features. Therefore, this study used 67 features consisting of network statistics derived from 20-21 consecutive individual packet features. However, these are specific to the network in which they are produced. If the same device or model is moved to another network, these network statistics will change and the model will no longer function. In this respect, flow-based features are more likely to be used in anomaly detection rather than device identification. Conversely, the use of individual packets is also inadequate for anomaly detection [10].

There are more studies using the Type classification. Among these, IoTSentinel [8], a pioneering study, is particularly noteworthy. In this study using Aalto data, 23 individual features extracted from packet headers are used. These features, which are taken from 12 packets without repetition, are combined under MAC address guidance to create a larger fingerprint. This larger fingerprint is used as machine learning input. This study uses the RF method to classify 17 out of 27 devices with an accuracy of over 95% and, for 10 devices, the accuracy remains around 50%. However, the work is very strictly dependent on the MAC address. Therefore, it cannot solve the transfer problem, where a MAC address represents more than one device.

Another work that uses the Aalto dataset is [11]. In this study, using a genetic algorithm, 33 of the 212 features obtained from the packet headers were selected and 95% accuracy was obtained using these features. This study can be criticised for using overly specific features (such as port numbers, TCP sequence, TCP acknowledgment, and IP ID) that may leak information about inter-packet interactions, and for using a partial dataset (23 out of 27 devices were used).

Another interesting work is IoTSense [12]. This work creates a new feature set by improving IoTsentinel features. In this feature set, a 20-element feature set is created by adding 17 features from IoTsentinel and 3 yuk-based features. By combining 5 of these subsets, larger feature sets are obtained to feed the ML. Although no details are given in the study, we guess that MAC address is used to combine these 5 subsets. Although this study shows an accuracy of 99%, the fact that the dataset used is not shared and that 4 out of 14 devices were discarded during the experiment makes the result unreliable. In addition, if the MAC address is used, it will suffer from the transfer problem.

The work done by Sivanathan et al. [13] is very important in terms of providing another device identification dataset to the literature. Using NB and RF methods, the UNSW dataset containing 28 devices could be classified with 99% accuracy. However, it can be criticised that in this study identifying features were used such as flow-based features, cipher suite, DNS queries, and port numbers. In addition, some devices in the dataset were not included in the evaluation step. Finally, we will evaluate the study by Dadkhah et al. [9]. In this study, which introduced a new CIC-IoT-22 dataset to the literature, a class-based classification was used. The devices in the dataset are classified using 3 categories: Audio, Camera, and Home Automation. 12 machine learning methods were used in this study and 98% accuracy was achieved. It is also interesting to note that during the testing phase, data from a different lab and different devices were used. Even though we used the data from this study, our results are not comparable, since we used type-based evaluation and they used class-based evaluation. More information about the dataset used is given in the data section.

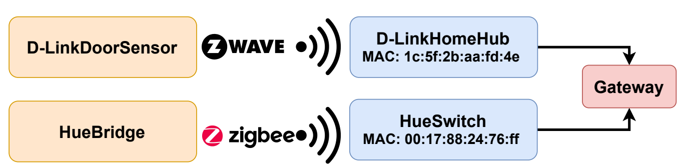


Fig. 2: Visualisation of the transfer problem in the Aalto dataset. The network data is collected at the gateway. Between the HueBridge HueSwitch there is only Zigbee as a communication medium, between the HueSwitch and gateway there is only Ethernet. Data from the HueBridge is decapsulated at the HueSwitch and re-encapsulated for transmission to the gateway. As a result of this process, both the HueBridge and HueSwitch share the same MAC address as the data is encapsulated on the same device (the same happens between D-LinkDoorSensor, D-LinkHomeHub and gateway). Therefore, studies that use MAC addresses to concatenate packets cannot overcome this problem.

To briefly summarise the "IoTDevID" work, we used the Aalto dataset for method development, analysis and feature extraction, and the UNSW dataset to demonstrate that the method is robust and generalisable. For features, we used features extracted from individual packages. After an extensive evaluation of machine learning methods, we decided on DT, which is quite acceptable in terms of performance, and also quite fast, making it very suitable for real life applications. We applied a multi-stage feature selection process, in the first step we discarded descriptive features, then we eliminated unimportant features by voting based on 6 different feature scores, and finally we generated the ideal feature set using a genetic algorithm. We achieved significant improvements in our results by using the aggregation algorithm. Compared to previous studies, we find that IoTDevID is significantly more successful. In this study, we will validate the work by applying IoTDevID to the CIC dataset.

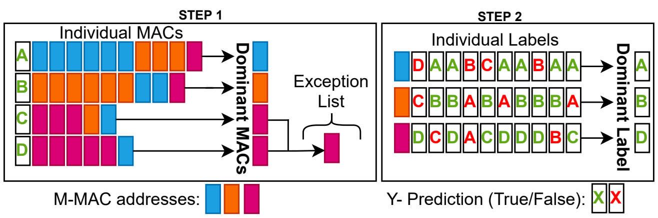


Fig. 3: The number of packets produced by the devices in the Aalto dataset.

III. MATERIALS AND METHODS

dataset

One of the biggest problems in device identification studies is the lack of adequate datasets. The simulations used in many networking studies cannot be used in device identification studies, and the fact that the dataset can only be created with real device data is the most prominent of the difficulties. To create a proper device identification dataset, many types of IoT devices are needed, and the supply of these devices is a serious financial burden. In addition, normal data collection requires a long time, labour, and specialised space. In our baseline study, IoTDevID [**?**], we used two datasets, the Altoo University [8], [14] and UNSW datasets [13], which were produced for device identification studies. In this study, we used the Altoo dataset to develop our method and the second dataset to validate our results. In 2022, a new device identification dataset, CIC-IoT-22 [9], was made public.This dataset is very interesting as it contains many types and numbers of devices, contains the state of the devices under different conditions, and contains attack data in addition to benign data. This dataset is very useful to demonstrate the usefulness, robustness, and generalisability of our method.

In this dataset, data was collected in 6 different situations. These situations can be summarised as follows. In the **Power** state, each device is isolated from other devices and rebooted and the network packets related to this device are collected. In the **Interactions** state, the device is interacted with by buttons, applications or voice commands and the network packets generated during this process are collected. In **Scenarios**, the network data of these devices are collected in situations such as entering the house, leaving the house, unauthorised entry to the house at night and day or user error. In case of **attack** state, data is collected by applying Flood attacks and RTSP Brute Force attacks to the devices. the **Idle** state consists of recording every 8-hour period for 30 days in the evening hours when the devices are working but not actively used. The **active** state contains the data of the devices being used during the day for 30 days. This data is generated by people entering the lab and using the devices.

Some important points about the dataset: In this study, the most important sections for us are IDLE and Active. In these two sections, enough data has been collected from almost all devices. Although it is stated on the paper that 60 devices were used in this process, according to our own experiments and the information provided in the dataset, these sections contain 40 devices. These 40 devices consist only of lan WIFI devices, they do not include Zigbee and z-wave devices. Zigbee and Z-Wave devices have data isolated from other devices, including power interaction and hede hodo, but these data are both very limited and do not contain normal usage data.

FE

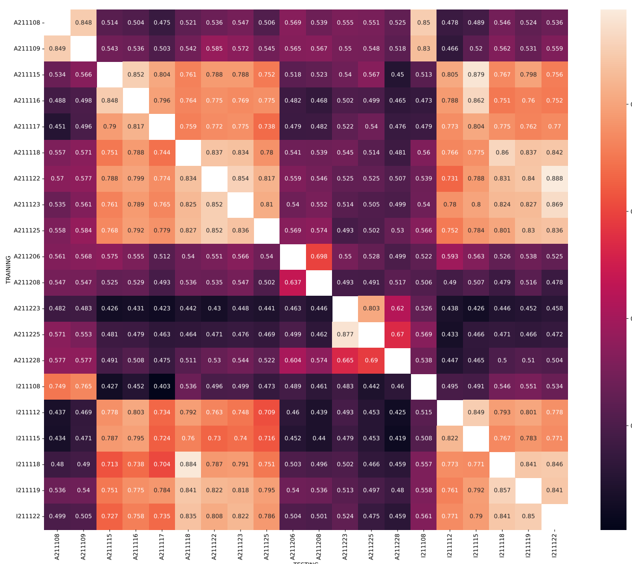
Python, Scapy and WireShark were used for feature extraction. Only individual package-based features are used for feature extraction. Many of these features are derived from packet headers, but there are also payload-based features such as payload entropy and payload bytes. Although the feature exruction system created about 100 features in total (features and their descriptions can be found in Table III.), very few of these features, only the sub-features selected during the feature selection phase of the IoTDevID study, were used in the experiment.

Labelling: Labelling was performed using the list of device names/MAC addresses couples in the dataset. In each fingerprint extracted, the source MAC address part was replaced with the given name and the MAC addresses not given in this list (5 MAC addresses that we believe belong to the hub, switch or the computer where the data is collected) were ignored.

Each of the pcap files we use for feature extraction contains network traffic recorded on a day, and is named with the date it was recorded. For example, data recorded on 24.11.2021 is labelled A211124 if Active and I211124 if Idle. In this context, 30 IDLE and 24 active sessions were recorded. as a preliminary study,, we aimed to test the performance of all these sessions by comparing them with each other. In order to compare the sessions with each other, they should contain similar devices. Unfortunately, data was not collected from every device in every session, and in some sessions some devices did not generate any data at all. Table IV.) shows how much data was generated by each device in each session in terms of network packets. Therefore, we only compare sessions that contain the same devices with each other. For this comparison, we create a session ID. In this ID, each device is represented by a binary digit. If the session has that device, it is indicated with 1, if not, it is indicated with 0. For example, if sessin1 contains devices A, and C but not device B, then the ID number is 101(ABC). Sessin1 can be compared to other sessions with the same ID number without any problem. In this context, we have created a 40-digit ID for each session according to totalling 40 devices.

The results we obtained by using devices with the same ID as training and test data are given in Fig. 4. We used the F1 score to present these results for roughly two reasons. Firstly, unlike accuracy, f1 score gives reliable results on unbalanced data sets. Secondly, the F1 score does not only give overall results, but also allows us to analyse the results by class. When the results are analysed in this context, it is seen that the F1 score varies between 40%-88% in pairwise session comparisons. Another point we would like to draw attention to here is that this process is a multiple classification process with approximately 40% classes. In this context, even 40 F1 is a much better result than chance/random success.

When the figures are analysed, it is seen that the results coinciding with specific dates in Fig. 4a (211108, 211109, 211206, 211208, 211223, 211225, 211228) are unsuccessful, on the other hand, when Fig. 4b is analysed, it is seen that the results in certain consecutive date ranges are more successful.



(a)

(b)

Fig. 4: F1 scores of sessions pairs assigned as training and test sets.

These results reflect overall success. In total, we made more than 600 session pair comparisons. In these comparisons, the first session of the pair was used as training and the second as testing. If we divide these sessions into active and idle, four different possibilities are possible Active vs Active (AA), Active vs Idle (AI), Idle vs Active (IA), and Idle vs Idle (II). In this context, the distribution of session comparisons is given in Fig. 5.

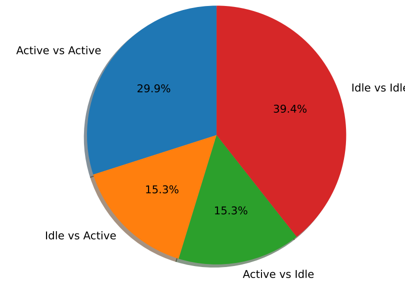


Fig. 5: The number of packets produced by the devices in the Aalto dataset.

We believe that focusing on class/device-based results will give more information. By analysing the device-based results for each session, we want to focus on the problematic devices. In this context, a device that is unsuccessful in any of the sessions, with a class-based F1 score of less than 0.50, is added to our list if it repeats this behaviour more than 12 times in all comparisons (12 corresponds to 2% of all session comparisons). Fig. 7. shows this list. The list shows the number of times the device class has failed and the distribution of these failures according to the session benchmark types.

Examining Fig. 7, we can see that, with some minor exceptions, the overall distribution of the pie chart remains the same. This shows that there is no significant difference between idle and active. On the other hand, if we focus on some devices with low performance, we can easily understand why they are included in the list. The devices with the highest number of failures are those with more than one example in the experimental set, such as Amazon Alexa Echo Dot, Gosund Plug, Gosund Socket, Teckin Plug, Yutron Plug. Since these devices are different examples of the same device (same brand and model), they should be grouped under one label (e.g. Teckin Plug 1 and Teckin Plug 2 -> Teckin Plug). We believe that the success level of most of the other devices can be improved by increasing the sample diversity. In this context, we aimed to increase the sample diversity by taking samples from multiple sessions and, as a consequence, to increase the model success. In this context, we separated all sessions into two parts, training and test sessions, in such a way that the sessions are isolated from each other. By combining the sessions that we determined as training and test, we obtained a very large dataset. While creating this dataset, we isolated idle and active data in their own types. We obtained 2 pairs of datasets, idle training and testing consisting only of idle data, and active training and testing consisting only of active data. However, we took the data of D-Link Water Sensor, a device not included in the active sessions, from the idle sessions. Another change was related to LG Smart TV device. The data for this device is only present in three of the 54 sessions. However, the data for this device is so unbalanced that the device data collected from only 3 out of 54 sessions account for about 9% of the total number of packets in all 40 devices. We removed this device from the dataset both because it did not have data from enough sessions and because its excessive number of packets distorted the distribution of the dataset.

In order to obtain a dataset that reflects the diversity of the sessions but is not too large, we reduced the number of packets in these 4 datasets to 10% of the total number of packets per dataset. Since we used random samples during this process, the packet rates obtained from the devices remained constant so that we did not damage the natural distribution of the dataset.

The comparison of the 4 different cases in terms of F1 score is given in the graph.

TABLE I: Add caption

	Data	Accuracy	F1Score	Train-t	Test-t	Al-time
Individual	AA	0.890±0.001	0.842±0.004	1.748	0.204	0
	AI	0.918±0.001	0.905±0.005	1.812	0.287	0
	IA	0.823±0.046	0.818±0.015	1.699	0.223	0
	II	0.821±0.004	0.814±0.007	1.721	0.291	0
Aggregated	AA	0.943±0.001	0.925±0.007	1.962	0.235	9.119
	AI	0.999±0.000	0.999±0.000	1.864	0.299	11.519
	IA	0.850±0.058	0.898±0.017	1.584	0.206	8.46
	II	0.904±0.004	0.912±0.006	1.630	0.313	11.267

TABLE II: Add caption

	Individual				Aggregated			
	AA	AI	IA	II	AA	AI	IA	II
Amcrest WiFi-Cam.	0.968	0.979	0.951	0.959	0.992	1.000	1.000	1.000
Amazon AE Dot	0.933	0.938	0.950	0.947	0.997	1.000	0.998	1.000
Amazon AE Spot	0.555	0.838	0.844	0.837	0.559	1.000	0.999	0.999
Amazon AE Studio	0.821	0.874	0.837	0.736	0.981	1.000	0.999	0.979
Amazon Plug	0.995	0.999	0.997	0.999	1.000	1.000	1.000	1.000
Arlo Base Station	0.984	0.819	0.624	0.864	1.000	0.998	0.454	1.000
Arlo Q Camera	0.987	0.970	0.969	0.952	1.000	1.000	1.000	1.000
Atomi Coff-Maker	0.847	0.892	0.890	0.501	0.999	1.000	1.000	0.638
Borun Camera	0.982	0.981	0.972	0.978	0.999	0.999	0.999	0.999
D-Link Mini Cam.	0.982	0.989	0.342	0.906	1.000	1.000	0.756	1.000
D-Link Water Sen.	0.930	0.936	0.936	0.934	1.000	0.989	1.000	0.989
Eufy HomeBase 2	0.815	0.812	0.782	0.806	1.000	0.998	1.000	0.998
Globe Lamp	0.654	0.823	0.916	0.431	0.900	0.997	1.000	0.246
Google Nest Mini	0.975	0.971	0.952	0.879	1.000	1.000	0.996	0.982
Gosund Plug	0.839	0.899	0.917	0.706	0.968	0.995	0.999	0.724
Gosund Socket	0.868	0.893	0.895	0.532	0.992	0.995	0.999	0.406
HeimVision S Cam.	0.986	0.998	0.934	0.984	1.000	1.000	1.000	1.000
HeimVision Lamp	0.715	0.838	0.857	0.518	0.965	0.999	1.000	0.759
Home Eye Camera	0.930	0.911	0.927	0.910	1.000	1.000	1.000	1.000
Luohe Cam Dog	0.762	0.760	0.759	0.757	1.000	0.995	1.000	0.994
Nest Indoor Cam.	0.998	0.997	0.999	0.910	0.999	0.999	1.000	0.997
Netatmo Camera	0.969	0.986	0.398	0.934	1.000	1.000	0.381	1.000
Netatmo Weather	0.826	0.827	0.868	0.845	1.000	1.000	1.000	1.000
Philips Hue Bridge	0.994	0.990	0.978	0.986	1.000	1.000	1.000	1.000
Ring Base Station	0.305	0.913	0.293	0.697	0.336	1.000	0.250	0.995
SIMCAM 1S	0.996	0.998	0.980	0.998	1.000	1.000	1.000	1.000
Smart Board	0.363	0.721	0.292	0.675	0.105	0.996	0.074	0.996
Sonos One Speaker	0.729	0.844	0.727	0.891	0.991	0.999	0.951	1.000
Teckin Plug	0.675	0.826	0.868	0.578	0.932	1.000	1.000	0.712
Yutron Plug	0.764	0.855	0.867	0.632	0.956	1.000	1.000	0.850
iRobot Roomba	0.955	0.962	0.843	0.946	1.000	1.000	0.993	1.000
Mean	0.842	0.905	0.818	0.814	0.925	0.999	0.898	0.912

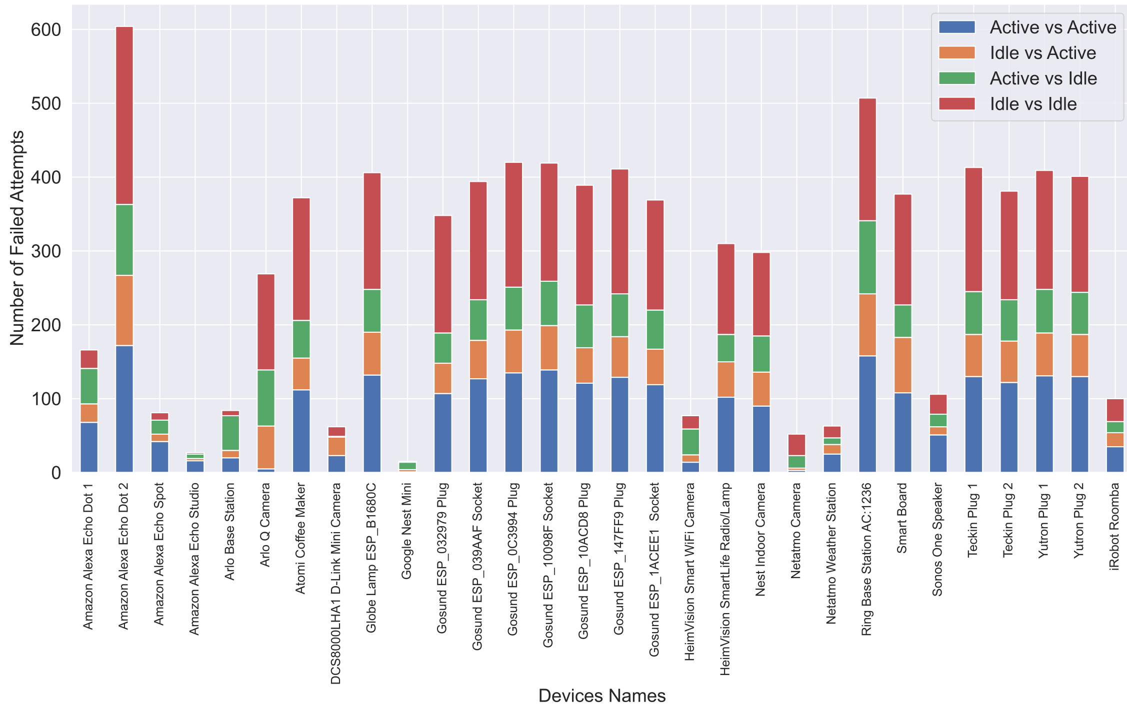


Fig. 6: The number of packets produced by the devices in the Aalto dataset.

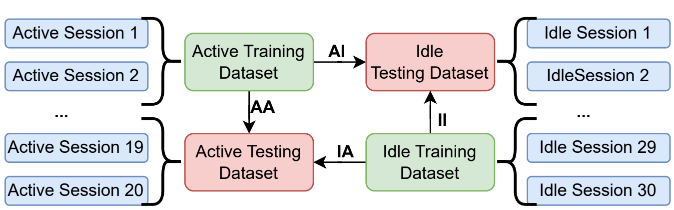


Fig. 7: The number of packets produced by the devices in the Aalto dataset.

An IoT device is any item that has a unique identity that can connect to other devices and perform control commands [**?**]. It is a bridge that connects our cyber world and our physical world [15]. Using IoT, we can manage our physical world from our cyber world.

This rapid progress has enabled many of us to include IoT devices in our lives. However, as a result of this rapid progress, many IoT devices were launched by many different manufacturers in a very short time. This rapid progress brings to mind the issue of whether devices are safe or how to ensure their safety.

It is unlikely to apply the solutions applied to classical computers to IoT devices. Because most of these devices have very limited possibilities in terms of battery, processor and storage. Also, due to the heterogeneous nature of IoT, there is no standard among these devices. Many use a unique operating system and hardware. In addition, IoT devices do not have the standard interfaces as conventional computers. This situation limits user-device interaction. In today's world, where there are a wide variety of devices, users can't take the security measures that every device needs. IoT device identification is a method that aims to find the device identity, such as brand and model, by analyzing the device behaviour.

Thanks to this method, devices on the network can be detected, and the necessary security measures can be taken for these devices. For example, the software of the needed ones can be updated, the behaviour of the devices and the addresses they will connect to can be restricted, or these devices can be isolated from the rest of the network.

In this study, we have divided the device identification process into four steps (see Figure 4), and included possible mistakes to be made in each step and how to deal with these mistakes.

Parallel to these steps, the structure of the paper is as follows. In Section IV, device identification methods are discussed. Section V examines data types and considerations when choosing the appropriate dataset. Section VI highlights the feature extraction step and its key points. Section VII discusses how to decide on the appropriate machine learning method. In Section VIII , evaluation methods and their pros and cons are explained.

IV. Familiarizing with Methods

In this section, different device identification methods are introduced and their advantages and disadvantages are discussed.

A. Define the limits of the scope of the IoT concept you will use in your study.

The Internet of Things (IoT) can be defined as all physical objects that can share data with other devices, have embedded sensors, software and various technologies [16]. However, this definition is too general. For your own work, you must decide what to consider as IoT and define the boundaries of your definitions. For example, if your data will contain, will you collect classic computers or mobile devices such as tablets and phones or routers and switches under this definition or will you consider them as non-IoT devices? Will you accept Wireless Sensor Network (WSN) devices as IoT or will you create a separate subgroup for them? When classifying IoT devices, will you focus on a certain sub-area (home type, smart cities, agriculture, army, etc.).

B. Consider different methods of device identification.

Deciding what kind of definition you will make is very important as it will affect other steps of your work. You can identify according to devices with 3 different approaches [6]:
Unique Identification: By accepting each of the devices as unique, a separate class is created for each device [7].
Type Identification: Identification is performed according to the device type. If there are multiple devices of the same make and model, they are seen as a single class [8].
Class Identification: Different devices that are not the same but have similar features, e.g., produced by the same brand for similar tasks, are gathered under a single class [11], [17].

Labels			Devices	Labels			Devices
Class	Type	Unique		Class	Type	Unique	
1	1	1	Aria	8	15	17	HomeMaticPlug
2	2	2	D-LinkCam	9	16	18	HueBridge
	3	3	D-LinkDayCam		17	19	HueSwitch
3	4	4	D-LinkDoorSensor	10	18	20	iKettle2
	5	5	D-LinkHomeHub		19	21	SmarterCoffee
	6	6	D-LinkSensor	11	20	22	Lightify
	7	7	D-LinkSiren	21	21	23	MAXGateway
	8	8	D-LinkSwitch	12	22	24	TP-LinkPlugHS100
	9	9	D-LinkWaterSensor		23	25	TP-LinkPlugHS110
4	10	10	EdimaxCam 1	13	24	26	WeMoInsightSwitch 1
		11	EdimaxCam 2			27	WeMoInsightSwitch 2
5	11	12	EdimaxPlug1101W		25	29	WeMoSwitch 1
	12	13	EdimaxPlug2101W			30	WeMoSwitch 2
6	13	14	EdnetCam 1	14	26	28	WeMoLink
		15	EdnetCam 2	15	27	31	Withings
7	14	16	EdnetGateway				

Fig. 8: Labelling the Aalto dataset according to 3 identification approaches.

Fig. 8 shows the labelling of the Aalto University IoT Devices Captures dataset [14] with three viewpoints. In the dataset containing 33 devices in total, 33 different labels are formed in the unique method, 27 different labels in the type method, and 15 different labels in the class method. Which of these three perspectives you choose is also effective in determining the network traffic characteristics you will use in your design.

C. Remember that using flow-based or packet-based features affects the generalizability of your models.

In your design, you can use packet-based, flow-based features or both. However, if you use Unique Device Identification, the features you will only get from the packets will not be enough. you will also need to use the flow-based features. On the other hand, using flow features will reduce the generalizability of your model, resulting in your model being specific only to the design you are using.

D. Analyze the strengths and weaknesses of these methods and choose according to your needs.

Type Identification and Class Identification methods allow you to obtain a model with higher generalizability by using only packet-based features. Using the Type Identification method does not promise great success in distinguishing similar devices, because similar devices show similar behavior patterns. Against this disadvantage, Class Identification can be used, which gathers similar devices under the same label. This approach is based on the assumption that similar devices can be grouped under the same group because they have similar hardware and software. However, in this method, expert involvement is required to decide on the similarities and groups of the devices. Before deciding on the method you will use in your design, it will be appropriate to examine the advantages and disadvantages of these methods, and to decide according to your purpose.

V. Familiarizing with Data

In this section, how the data you will use can be obtained, the characteristics and qualities that the data should have are examined.

A. Decide how you will obtain the data.

One of the important steps in device identification studies is how to obtain the data. In this context, you can use simulation or real devices. Although using simulation is fast, cheap, and easy, simulation programs currently lack the ability to simulate the various types of IoT devices because of the extremely heterogeneous nature of IoT devices. Therefore, the use of simulation is mostly seen in studies where many homogeneous devices such as WSN are used.

B. Do not forget about privacy if you are using real data.

In the use of real IoT devices, there are options to create an experiment set or usage of real data. When using test sets it is important that the devices can mimic as much as possible usage patterns. Although using real data may seem advantageous in many ways, the danger of causing privacy disclosures should be considered. If you are collecting data from a real environment, you should consider the ethical aspect, obtain the necessary relevant permissions and censor any information that may disclose confidentiality such as user identity or IP addresses.

C. Use data from previous studies.

Another data acquisition method is the use of other previously collected publicly available data. The use of such data is very advantageous as it does not require extra cost and provides the results to be comparable with the literature. You can scan the literature to find the appropriate data set, especially survey studies on device identification and fingerprinting can be very useful. You can also use websites specialized for datasets, e.g., Dataset Search, Kaggle, and UCI ML Repository

D. Unsure the data is suitable and has the necessary qualifications.

In addition, no matter how you obtain your data, the data you will use for device identification must have the following characteristics:

- It should contain benign data. In device identification, you need to extract the behavior patterns of the devices. For this, you need the normal/benign data of the devices. However, if the benign and malicious data packets are separated, you can also use the normal parts of the data sets prepared for anomaly detection (e.g., UNSW IoT benign and attack traces [18]). Another point that can be overlooked is that the tools used in vulnerability testing are very similar to attack tools. You can classify the data obtained by performing vulnerability tests on them as attack data, not normal data.
- The dataset should contain enough data. It is essential for a robust study that the dataset contains as many and varied devices as possible. In this way, the operation of your model is observed more soundly and the probability of success by chance is reduced.
- Devices must be labelled. In the dataset you will use, you should know which device the packets belong to (which device was produced by).

E. Ensure that the correct labels are assigned to the data.

It is quite common to use MAC and IP addresses to identify devices, but care should be taken when using these addresses. Especially if there are Non-IP devices in the datasets, this information can be misleading. The Aalto dataset is a good example for this. The non-IP HueSwitch device is connected to the gateway where data collection is made, via the HueBridge device. The HueSwitch transmits the network packets to the HueBridge device using ZigBee, which then HueBridge re-encapsulates these network packets and forwards them to the gateway via Ethernet. HueSwitch packets arriving at gateway carry the MAC addresses of the HueBridge device, not their own MAC addresses. Therefore, HueBridge and HueSwitch devices are represented by a single MAC address in the Aalto dataset. So, MAC and IP addresses cannot be used for labelling. That's why the creator of the dataset labelled the devices separately.

F. Remember that IoT devices are highly heterogeneous and this affects data distribution.

IoT devices are tools that contain a wide variety of software and hardware, produced for many different purposes. This is why data from devices can be quite diverse. So, IoT devices have an unbalanced data distribution due to their highly heterogeneous nature. One device in a network can generate a large amount of data, while another device can generate a very small amount of data. Fig. 9 shows the number of packets produced by the devices in the Aalto dataset.

G. Use data augmentation to deal with scarce data.

This unbalanced distribution of data can negatively affect some machine learning methods. You can deal with this problem by using data augmentation techniques. The most important point during this process is to isolate the test and training datasets from each other **before** data augmentation. In this way, the augmented (post-generated) data in the training dataset is prevented from leaking into the test data.

H. Protect test data from all influence

Another common mistake is to apply augmentation of test data. It is best to keep the test data in its original state. Even the simple process of resampling, such as copying packets, is harmful because it will break the distribution of test data. The distribution of test dataset should also show the true world distribution. Thus, you can evaluate the outputs of your model more realistically and check whether there is a real contribution on data augmentation.

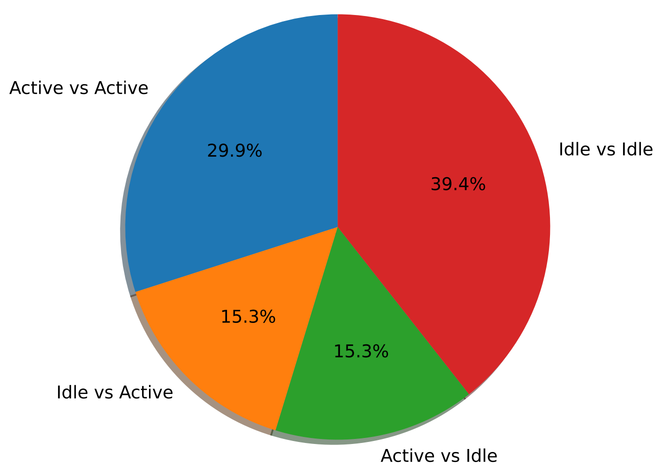


Fig. 9: The number of packets produced by the devices in the Aalto dataset.

VI. Familiarizing with Features Extraction

In this section, the feature extraction process and important points to be considered during this process are highlighted.

A. Save time by choosing the right tools for preliminary analysis.

One disadvantage of working with network data is dealing with huge amounts of data. Analysing this data and extracting features can be very time consuming. Therefore, it is very important to choose the right tools for these operations. Recently, python-based libraries such as dpkt and scapy have been used extensively in the analysis of pcap data and feature extraction. Note that although they are easy to use and supported by a lot of documentation, they are extremely slow. Especially in very large data, using C-based programs such as Wireshark for your pioneering analysis will save you time. Even, with tshark, which you can automate with bash code, you can do many operations much faster.

B. Avoid features that uniquely identify your device.

Get to know the features you will use during the feature extraction process. Avoid including passive features such as MAC and IP addresses in your feature set. These properties are identifying features. They uniquely identify devices but do not provide any information about device behaviour. Even if you get high results using these features in your experiments, these results cannot be generalized because they suffer from overfitting.

C. Avoid features that uniquely identify a session.

Some session-based features are highly identifying, despite not static. However, these features uniquely identify the session, not the device. For example, various features such as port numbers, IP ID, TCP sequence numbers are determined at the beginning of the session, and are used until the session ends. Because these properties are randomly determined, they are not predictable or generalizable. You can avoid using these features in your study. If you are going to use it, you can do it in a healthy way. For example, isolate the test and train datasets from each other to ensure that they consist of different sessions. Thus, you can prevent these features, which are identifying for each session, from leaking from the training data to the test data.

D. Beware of features that are implicitly identifying.

Although time-related features such as timestamps do not appear to be stand-alone identifying, they can be identifiers as they will uniquely show a device's operating timespan. Header checksum features are another example of this situation. Checksum features are a summary of the header in which they are used. So IP checksum contains information about IP addresses, TCP checksum contains sequence, acknowledgment numbers, UDP checksum contains port numbers. If you find it inconvenient to use any attribute in the header, you should also avoid using the header's checksum feature.

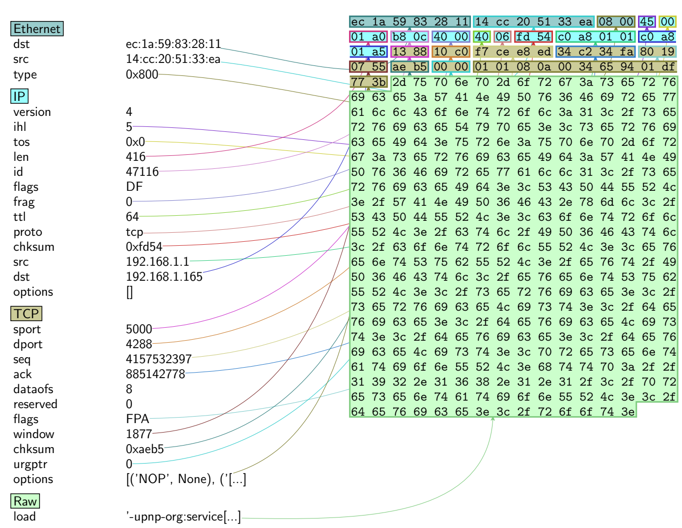


Fig. 10: The fields contained in a network packet and their byte equivalents..

E. Be careful when extracting features from raw data.

An alternative method in the feature extraction stage is the use of raw bytes. This method is based on obtaining feature sets through various size reduction methods by converting network data into raw bytes. A network packet converted to raw bytes is given in Fig. 10. This method is advantageous in terms of both obtaining original features and reducing expert involvement in the feature extraction stage. There is no harm in using this method in the packet payload, but if you include the packet headers, you may leak many features that you would not normally want to use in the feature set. So, when applying this method, it should be noted that raw data contains identifying features. Therefore, before performing feature extraction, censoring the raw data corresponding to the identifying features ensures that more robust features are extracted, and prevents errors that would lead to overfitting.

VII. Familiarizing with Machine Learning

In this section, it is focused on what should be considered when using machine learning methods and how the methods can be selected.

A. Consider other options when creating a multiclass model.

Device identification is a multi-class problem. However, there are some disadvantages of using multi-class models in solving this problem. The multi-class approach makes it difficult to extend the model. Because every time a new device is added to your system, you will have to retrain the model. In addition, scalability issues may arise as the number of devices increases. One2all approach can be preferred to deal with this problem. In this approach, a separate binary model is created for each device, and a multi-class result is obtained from the outputs of these models. In case a new device is included in this system, there is no need to recreate the whole models, only the model of this new device is added.

B. Do not underestimate the classical ML approaches

Do not underestimate the classical/older approaches when choosing the ML method you will use. Although deep learning methods have been popular recently, it has been observed that classical methods such as decision trees are much more successful than deep learning methods, especially in evaluating tabular data that lack temporal relationships, such as device identification [19].

C. Consider the interpretability of the models

Interpretability of algorithms can also be a matter of preference. Methods such as liner/logistic regression, DT, kNN have a high level of interpretability, while SVM, Ensemble methods and deep learning have a low level of interpretability (see Fig.11). High interpretability can be very useful when analysing and evaluating features [20].

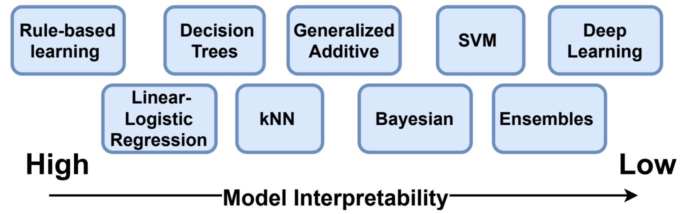


Fig. 11: Interpretability levels of some machine learning methods.

D. Note that there is no free lunch.

There is no perfect machine learning method that works best on every data type [21]. Try to find the appropriate machine learning method for your data. With so many methods available, finding the best method can be costly. You can try different strategies for this. For example, taking a look at the methods used by similar previous studies can be very helpful, especially survey studies. Another method would be to try one of each type of machine learning (for example, tree-based, kernel-based, ANN-based etc.) and experiment more deeply with the type that performs best.

E. Consider the success - inference time trade-off.

Do not accept detection success as the only criterion. In network technology, transactions are done at the micro-milliseconds level. The fast sorting capability of your model, which will be used in the real world and will work in a critical area such as security, is as important as the correct classification capability. Although some algorithms such as SVM and kNN achieve high performance levels, their inference time can be very high. Therefore, it is highly recommended to include the inference time criterion in your evaluation and to avoid algorithms with very high inference times.

VIII. Familiarizing with Evaluation

In this section, performance evaluation methods are examined and their pros and cons are discussed.

A. Consider data distribution when choosing the evaluation method.

Various evaluation criteria are used to see how successful the study is. Accuracy is the most popular of these criteria. Since it is used for evaluation in many studies, using accuracy will add comparability to your study. However, in datasets that suffer from unbalanced distribution, such as IoT datasets, using accuracy as the only method can be quite misleading. Devices with very high or very low results with too many samples may give unrealistic results by pulling the result of all data up or down. In addition, the performance of devices with too few samples may be overlooked. Another disadvantage is that accuracy is a holistic method. So, it cannot give results per device/class. This is why many studies use the recall, for the presentation of device-based results.

B. Note that some evaluation concepts have many names.

Nevertheless, naming the recall value per class is also very problematic in the literature. Many names are used to express this criterion, such as identification rate, recognition rate, detection rate, accuracy rate, and individual device classification performance. In order not to add to this confusion, you can use a simpler and more general nomenclature such as overall recall or per-class recall.

C. Remember that some methods can be misleading.

However, using recall alone can be misleading as it does not account for False Positives. As a solution to this, you can use precision with recall to keep the balance between them. Another solution is to use the F1 score, which is the harmonic mean of the recall and precision. This metric alone shows the recall-precision balance, and you can observe both overall and class-based results with it.

IX. Conclusion

In this study, the methods used in device identification with machine learning and common mistakes are discussed. In this context, by addressing the positive and negative aspects of the methods used, identification methods, appropriate data types, the feature extraction process, selection and use of machine learning techniques, and evaluation methods are examined in order to provide practical solutions to common mistakes.

References

[1] J. Fruhlinger, "What is iot?" 2023, accessed: 2023-03-29,Available at https://www.networkworld.com/article/3207535/what-is-iot-the-internet-of-things-explained.html.

[2] M. Hasan, "State of iot 2022," 2022, accessed: 2023-03-29,Available at https://iot-analytics.com/number-connected-iot-devices/.

[3] H. Modi, "Dawn of the terrorbit era," Feb 2019, accessed: 2022-03-29,Available at https://www.netscout.com/blog/dawn-terrorbit-era. [Online]. Available: https://www.netscout.com/blog/dawn-terrorbit-era

[4] T. M. Corporation, "Cve program mission," Nov 2022, accessed: 2022-03-29,Available at https://www.cve.org/.

[5] K. Kostas, M. Just, and M. A. Lones, "IoTDevID: A behavior-based device identification method for the IoT," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 741–23 749, 2022.

[6] P. Yadav, A. Feraudo, B. Arief, S. F. Shahandashti, and V. G. Vassilakis, "Position paper: A systematic framework for categorising iot device fingerprinting mechanisms," in *Proceedings of the 2nd International Workshop on Challenges in AI and ML for IoT*, 2020, pp. 62–68.

[7] S. A. Hamad, W. E. Zhang, Q. Z. Sheng, and S. Nepal, "Iot device identification via network-flow based fingerprinting and learning," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2019, pp. 103–111.

[8] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 2177–2184.

[9] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, and A. A. Ghorbani, "Towards the development of a realistic multi-dimensional iot profiling dataset," in *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*, 2022, pp. 1–11.

[10] K. Kostas, M. Just, and M. A. Lones, "Towards more generalisable models for behaviour-based IoT attack detection," 2023.

[11] A. Aksoy and M. H. Gunes, "Automated IoT device identification using network traffic," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[12] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral fingerprinting of IoT devices," in *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*, 2018, pp. 41–50.

[13] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.

[14] S. Marchal, "IoT devices captures, aalto university," 2017, accessed: 2021-08-25. [Online]. Available: https://research.aalto.fi/en/datasets/iot-devices-captures

[15] C. Patel and N. Doshi, *Internet of Things Security: Challenges, Advances, and Analytics*. CRC Press, 2018.

[16] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in iot security: Current solutions and future challenges," *IEEE Com. Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[17] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "Dïot: A federated self-learning anomaly detection system for iot," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 756–767.

[18] A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "IoT benign and attack traces," 2019, accessed: 2021-09-25. [Online]. Available: https://iotanalytics.unsw.edu.au/attack-data

[19] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[20] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[21] D. H. Wolpert, "The supervised learning no-free-lunch theorems," *Soft computing and industry*, pp. 25–42, 2002.

APPENDIX

TABLE III: The list of individual packet-based features used in device identification and feature descriptions

No	Feature	Description
1	ts	Time Stamp
2	Ether_dst	Destination Media Access Control (MAC) Address
3	Ether_src	Source MAC Address
4	IP_src	Source Internet Protocol (IP) Address
5	IP_dst	Destination IP Address
6	WS_src	WireShark Source Address
7	WS_dst	WireShark Destination Address
8	pck_size	Packet (Frame) Size
9	Ether_type	Ethernet Type
10	LLC_dsap	Logical Link Control - Destination Service Access Point
11	LLC_ssap	Logical Link Control - Source Service Access Point
12	LLC_ctrl	Logical Link Control - Control
13	EAPOL_version	Extensible Authentication Protocol (EAPOL) version
14	EAPOL_type	Extensible Authentication Protocol (EAPOL) type
15	EAPOL_len	Extensible Authentication Protocol (EAPOL) Length
16	IP_version	IP version
17	IP_ihl	IP Internet Header Length
18	IP_tos	IP type of service
19	IP_len	IP Length
20	IP_flags	IP Flags
21	IP_Z	IP Zero
22	IP_MF	IP More Fragments
23	IP_id	IP identifier
24	IP_chksum	IP Checksum
25	IP_DF	IP Don't Fragment
26	IP_frag	IP fragmentation
27	IP_ttl	IP Time To Live
28	IP_proto	IP Protocols
29	IP_options	IP Options
30	ICMP_type	Internet Control Message Protocol (ICMP) Type
31	ICMP_code	ICMP Code
32	ICMP_chksum	ICMP Checksum
33	ICMP_id	ICMP identifier
34	ICMP_seq	ICMP Sequence Number
35	ICMP_ts_ori	ICMP ConditionalField
36	ICMP_ts_rx	ICMP ConditionalField
37	ICMP_ts_tx	ICMP ConditionalField
38	ICMP_ptr	ICMP ConditionalField
39	ICMP_reserved	ICMP ConditionalField
40	ICMP_length	ICMP length
41	ICMP_nexthopmtu	ICMP Next Hop Maximum Transmission Unit (MTU)
42	ICMP_unused	ICMP ConditionalField
43	TCP_seq	TCP Sequence Number
44	TCP_ack	TCP Acknowledgment Number
45	TCP_dataofs	TCP data ofset
46	TCP_reserved	TCP Reserved
47	TCP_flags	TCP Flags
48	TCP_FIN	FINished Flag
49	TCP_SYN	Sync Flag
50	TCP_RST	Reset Flag

TABLE III: The list of individual packet-based features used in device identification and feature descriptions

No	Feature	Description
51	TCP_PSH	Push Flag
52	TCP_ACK	Acknowledgment Flag
53	TCP_URG	Urgent Flag
54	TCP_ECE	ECE Flag
55	TCP_CWR	CWR Flag
56	TCP_window	TCP Window Size
57	TCP_chksum	TCP Checksum
58	TCP_urgptr	TCP Urgent Pointer
59	TCP_options	TCP Options
60	UDP_len	User datagram protocol (UDP) Length
61	UDP_chksum	UDP Checksum
62	DHCP_options	Dynamic Host Configuration Protocol (DHCP) Options
63	BOOTP_op	Bootstrap Protocol (BOOTP) Options
64	BOOTP_htype	BOOTP Hardware Len
65	BOOTP_hlen	BOOTP Hardware Length
66	BOOTP_hops	BOOTP Hardware Options
67	BOOTP_xid	BOOTP Transaction Identifier
68	BOOTP_secs	BOOTP Seconds
69	BOOTP_flags	BOOTP Flags
70	BOOTP_sname	BOOTP Server Name
71	BOOTP_file	BOOTP Boot Filename
72	BOOTP_options	BOOTP Options
73	DNS_length	Domain Name System (DNS) Length
74	DNS_id	DNS Identifier
75	DNS_qr	DNS Query-Response
76	DNS_opcode	DNS Operation Code
77	DNS_aa	DNS Authoritative Answer
78	DNS_tc	DNS TrunCation
79	DNS_rd	DNS Recursion Desired
80	DNS_ra	DNS Recursion Available
81	DNS_z	DNS Reserved for future use
82	DNS_ad	DNS Authentic Data
83	DNS_cd	DNS Checking Disabled
84	DNS_rcode	DNS Response Code
85	DNS_qdcount	DNS The unsigned fields query count
86	DNS_ancount	DNS Answer Count
87	DNS_nscount	DNS Authority Count
88	DNS_arcount	DNS Additional Information Count
89	sport_class	Source Port Class (IoTDevID classing)
90	dport_class	Destination Port Class (IoTDevID classing)
91	sport23	Source Port Class (keep wellknown ports between 0-1023)
92	dport23	Destination Port Class (keep wellknown ports between 0-1023)
93	sport_bare	Source Port Number
94	dport_bare	Destination Port Number
95	payload_bytes	Payload size in Bytes
96	entropy	Payload Entropy
97	Protocol	WireShark Protocol
98	sport	Source Port Number
99	dport	Destination Port Number
100	Label	Packet Level Label

TABLE IV: Sessions, the total number of packets generated by the devices and devices in the session.

ML	Date	Type	AMCREST WiFi Camera	Amazon Alexa Echo Dot 1	Amazon Alexa Echo Dot 2	Amazon Alexa Echo Spot	Amazon Alexa Echo Studio	Amazon Plug	Arlo Base Station	Arlo Q Camera	Atomi Coffee Maker	Borun/Sichuan-AI Camera	D-Link Water Sensor
Train	20211102	ACTIVE	6.68	6.91	6.84	14.94	13.20	1.47	12.63	21.08	6.94	24.04	0.00
Train	20211103	ACTIVE	5.54	7.10	6.98	15.19	13.16	1.48	4.93	18.81	6.93	24.42	0.00
Train	20211105	ACTIVE	5.52	7.00	6.93	13.72	13.12	1.48	4.95	21.54	6.94	26.20	0.00
Train	20211108	ACTIVE	5.63	10.73	10.92	14.99	13.06	1.47	45.08	78.96	6.94	26.05	0.00
Train	20211109	ACTIVE	6.09	11.44	10.94	14.72	13.17	1.47	45.97	47.26	6.93	24.37	0.00
Train	20211110	ACTIVE	6.74	44.07	48.50	19.88	14.95	2.39	22.25	37.38	6.93	24.20	0.00
Train	20211112	ACTIVE	5.65	14.08	40.82	15.70	19.46	1.47	4.89	37.48	6.92	24.09	0.00
Train	20211115	ACTIVE	5.66	12.14	11.83	14.93	14.21	1.82	110	55.18	7.14	17.26	0.00
Train	20211116	ACTIVE	5.43	11.33	10.96	13.68	13.82	1.47	11.09	98.64	6.93	22.57	0.00
Train	20211117	ACTIVE	5.36	23.47	22.58	15.33	58.61	1.47	4.88	44.92	6.93	24.42	0.00
Test	20211118	ACTIVE	5.97	11.78	11.35	13.33	14.21	1.47	4.81	4.97	6.94	24.18	0.00
Test	20211119	ACTIVE	5.54	11.83	11.15	13.41	14.22	1.47	4.91	3.64	6.92	24.12	0.00
Test	20211122	ACTIVE	5.62	9.54	9.35	14.91	13.49	1.47	15.10	58.72	6.94	24.12	0.00
Test	20211123	ACTIVE	6.28	11.53	10.46	15.34	13.58	1.41	7.82	58.50	7.10	24.76	0.00
Test	20211124	ACTIVE	5.74	11.22	10.28	14.61	13.34	1.47	4.90	17.54	6.93	24.88	0.08
Test	20211125	ACTIVE	5.76	11.37	10.25	16.00	14.08	1.55	5.15	34.63	6.94	24.23	0.00
Test	20211126	ACTIVE	7.74	16.24	14.83	20.22	17.21	0.52	326	148.99	9.20	32.01	0.00
Test	20211206	ACTIVE	6.50	11.06	10.68	15.59	15.76	1.56	24.18	33.81	6.94	32.40	0.00
Test	20211207	ACTIVE	5.66	10.97	10.55	15.57	14.44	1.52	438	41.61	6.93	27.73	0.00
Test	20211208	ACTIVE	6.16	10.66	10.59	15.34	13.63	1.52	13.70	29.37	6.94	25.18	0.00
Train	20211223	ACTIVE	6.26	11.73	11.49	13.33	13.33	1.46	4.93	3.79	6.92	24.79	0.00
Train	20211225	ACTIVE	5.68	16.13	11.76	14.88	14.39	1.46	13.16	23.09	6.93	25.21	0.00
Train	20211228	ACTIVE	5.45	11.56	11.93	20.39	15.07	1.46	8.51	23.50	6.92	24.60	0.00
Train	20220103	ACTIVE	5.59	11.72	9.36	14.45	13.35	1.50	4.91	2.75	6.93	25.43	0.00
Test	20211102	IDLE	7.92	8.94	8.90	19.26	121.47	2.00	6.85	3.82	9.54	32.99	0.00
Test	20211103	IDLE	7.85	9.09	9.11	18.78	19.42	2.05	6.83	3.82	9.52	33.06	0.00
Test	20211104	IDLE	8.60	9.32	9.36	18.90	19.41	2.09	6.89	3.94	9.51	34.96	0.00
Test	20211108	IDLE	7.97	15.31	15.73	17.53	19.50	2.05	6.81	3.81	9.55	35.11	0.00
Test	20211110	IDLE	8.43	15.27	15.38	19.42	17.88	1.98	6.84	51.55	9.54	34.98	0.00
Test	20211112	IDLE	8.53	15.11	14.88	17.22	17.35	2.00	6.80	47.37	9.54	34.99	0.00
Test	20211115	IDLE	7.44	16.37	15.87	17.82	18.01	1.99	10.25	3.95	9.55	22.25	0.00
Test	20211116	IDLE	7.50	16.55	15.58	17.92	18.09	1.99	6.81	3.84	9.54	34.99	0.00
Test	20211117	IDLE	7.56	16.46	16.16	17.94	18.52	2.01	6.86	3.84	9.54	33.14	0.00
Train	20211118	IDLE	7.56	16.78	16.06	18.00	18.49	2.00	6.78	3.80	9.53	33.10	0.00
Train	20211119	IDLE	8.25	16.65	16.41	18.14	18.41	2.00	6.90	5.31	9.54	34.98	0.00
Train	20211122	IDLE	7.68	14.84	12.67	20.86	17.41	2.05	6.81	3.80	9.54	33.25	0.00
Train	20211124	IDLE	7.75	15.58	14.15	18.74	17.16	2.01	6.83	4.34	9.53	33.27	0.07
Train	20211125	IDLE	7.95	15.88	14.38	18.76	17.59	2.04	6.82	3.81	9.54	33.25	0.04
Train	20211126	IDLE	8.95	16.37	15.58	21.80	19.96	0.00	6.85	4.38	9.55	34.99	0.07
Train	20211129	IDLE	8.49	15.30	14.39	19.23	19.33	1.97	7.04	3.97	9.61	33.24	0.04
Train	20211130	IDLE	7.79	15.77	14.29	19.18	17.51	2.00	10.01	3.80	9.53	33.67	0.07
Train	20211201	IDLE	8.90	15.49	14.60	18.90	17.48	2.01	6.85	3.80	9.53	35.77	0.04
Train	20211202	IDLE	7.77	14.71	14.15	19.08	17.49	1.97	6.90	3.80	9.53	35.70	0.11
Train	20211203	IDLE	8.44	14.90	14.75	18.24	17.32	1.99	6.86	3.82	9.53	35.76	0.07
Train	20211207	IDLE	8.05	15.23	14.13	18.63	19.22	2.01	6.91	3.81	9.53	35.77	0.04
Train	20211208	IDLE	7.99	15.18	14.57	18.94	19.50	2.00	6.78	3.82	9.53	35.73	0.07
Train	20211209	IDLE	7.63	14.66	14.43	18.91	19.18	1.97	6.89	3.82	9.53	35.43	0.07
Train	20211210	IDLE	8.12	15.42	14.77	19.13	19.35	2.28	6.90	3.79	9.52	35.36	0.04
Test	20211213	IDLE	8.47	15.19	15.03	19.43	19.54	2.05	6.74	3.82	9.53	35.38	0.07
Test	20211215	IDLE	8.09	14.14	13.82	18.83	17.68	2.00	7.36	3.81	9.53	35.47	0.07
Test	20211216	IDLE	7.65	12.23	12.27	18.83	17.25	2.00	6.91	3.80	9.52	35.58	0.11
Test	20211220	IDLE	7.78	15.73	14.76	18.35	19.86	2.00	6.80	3.83	9.52	35.51	0.07
Test	20211222	IDLE	7.63	16.77	15.48	18.56	20.42	2.00	6.78	3.79	9.52	35.56	0.07
Test	20211223	IDLE	8.32	16.74	15.62	18.37	20.51	1.98	6.84	3.80	9.52	35.50	0.07

ML	Date	Type	D-Link Mini Camera	Eufy HomeBase 2	Globe Lamp ESP_B1680C	Google Nest Mini	Gosund ESP_032979 Plug	Gosund ESP_039AAF Socket	Gosund ESP_0C3994 Plug	Gosund ESP_10098F Socket	Gosund ESP_10ACD8 Plug	Gosund ESP_147FF9 Plug	Gosund ESP_1ACEE1 Socket	HeimVision Smart WiFi Camera	HeimVision SmartLife R-Lamp	Home Eye Camera	LG Smart TV
Train	20211102	ACTIVE	1.08	8.08	6.75	20.59	6.65	6.62	6.63	0.00	6.66	6.63	6.64	14.30	7.88	22.48	0
Train	20211103	ACTIVE	1.10	8.07	6.75	24.46	6.75	6.74	6.75	0.00	6.75	6.74	6.75	14.29	7.88	25.28	600
Train	20211105	ACTIVE	1.11	8.07	6.74	21.27	6.74	6.74	6.75	0.00	6.75	6.75	6.76	13.82	7.88	29.66	0
Train	20211108	ACTIVE	1.11	8.08	6.74	21.11	6.70	6.33	6.36	3.29	6.37	6.34	6.35	13.78	7.90	37.03	0
Train	20211109	ACTIVE	1.09	7.97	6.75	21.08	6.75	6.74	6.75	6.75	6.75	6.75	6.76	13.76	7.88	25.97	0
Train	20211110	ACTIVE	1.09	7.98	6.74	17.82	6.73	6.73	6.74	6.75	6.75	0.00	6.75	13.79	7.86	33.05	0
Train	20211112	ACTIVE	1.09	8.06	6.73	25.21	6.73	6.73	6.74	6.74	6.75	0.00	6.73	13.74	7.86	39.65	0
Train	20211115	ACTIVE	1.17	8.15	6.88	24.21	6.87	6.89	6.89	6.90	6.89	6.88	6.90	13.61	8.07	23.00	0
Train	20211116	ACTIVE	1.11	8.01	6.74	21.04	6.73	6.75	6.75	6.77	6.74	6.73	6.75	13.93	7.91	27.98	0
Train	20211117	ACTIVE	1.12	7.95	6.73	14.10	6.73	6.73	6.75	6.75	6.74	6.73	6.73	13.66	7.88	27.78	0
Test	20211118	ACTIVE	1.11	8.12	6.73	20.55	6.74	6.74	6.75	6.75	6.74	6.74	6.74	13.66	7.88	21.26	0
Test	20211119	ACTIVE	1.11	8.07	6.73	20.25	6.73	6.73	6.75	6.76	6.74	6.73	6.73	0.00	7.87	21.08	0
Test	20211122	ACTIVE	1.12	8.15	6.73	20.72	6.71	6.71	6.72	6.74	6.72	6.72	6.71	13.64	7.87	22.20	0
Test	20211123	ACTIVE	1.15	8.12	6.89	20.72	6.88	6.87	6.90	6.91	6.89	6.88	6.88	13.90	8.07	26.72	0
Test	20211124	ACTIVE	1.10	7.96	6.73	20.23	6.73	6.73	6.74	6.75	6.74	6.73	6.73	0.09	7.88	21.73	0
Test	20211125	ACTIVE	1.09	8.07	6.73	20.68	6.73	6.73	6.76	6.76	6.76	6.74	6.73	29.56	7.88	21.17	0
Test	20211126	ACTIVE	1.65	11.30	8.93	26.73	8.93	8.93	8.94	8.92	8.95	8.97	8.94	18.23	10.45	42.15	30
Test	20211206	ACTIVE	1.09	8.79	6.73	18.50	6.75	6.73	6.74	6.74	6.78	6.73	6.74	13.81	7.88	18.93	0
Test	20211207	ACTIVE	1.09	6.53	6.73	14.97	2.72	6.27	2.67	2.69	2.67	2.66	2.67	15.55	7.89	29.16	1805
Test	20211208	ACTIVE	1.12	8.03	6.74	14.31	1.04	0.98	0.98	1.05	0.98	0.97	0.98	5.44	7.91	39.17	0
Train	20211223	ACTIVE	1.08	7.99	6.72	19.71	6.75	6.72	6.72	6.72	6.72	6.72	6.72	13.57	7.90	17.00	0
Train	20211225	ACTIVE	1.08	8.01	6.73	19.97	6.74	6.72	6.72	6.72	6.72	6.73	6.72	13.52	7.89	19.59	0
Train	20211228	ACTIVE	1.07	8.45	6.72	13.55	6.75	6.72	6.72	6.72	6.72	6.72	6.72	13.52	7.89	19.59	0
Train	20220103	ACTIVE	1.08	8.07	6.72	14.93	6.73	6.72	6.74	6.72	6.73	6.75	6.73	14.93	7.86	16.75	0
Test	20211102	IDLE	1.46	11.04	9.26	29.43	9.26	9.25	9.26	0.00	9.27	9.25	9.26	20.23	10.84	28.76	0
Test	20211103	IDLE	1.46	11.12	9.25	30.55	9.26	9.27	9.28	0.00	9.27	9.26	9.27	20.05	10.83	28.81	0
Test	20211104	IDLE	1.47	11.23	9.26	22.58	9.27	9.26	9.26	0.00	9.28	9.26	9.28	19.78	10.83	28.76	0
Test	20211108	IDLE	1.52	10.97	9.27	29.83	9.26	9.25	9.27	9.27	9.26	9.26	9.29	19.07	10.85	28.78	0
Test	20211110	IDLE	1.48	11.27	9.25	31.72	9.25	9.25	9.27	9.27	9.27	0.00	9.25	18.83	10.82	29.00	0
Test	20211112	IDLE	1.47	11.01	9.25	40.40	9.25	9.25	9.27	9.27	9.28	9.28	9.25	18.82	10.82	28.81	0
Test	20211115	IDLE	1.48	11.08	9.26	28.80	9.26	9.25	9.29	9.30	9.29	9.28	9.25	18.84	10.82	28.50	0
Test	20211116	IDLE	1.50	11.12	9.25	32.30	9.25	9.25	9.28	9.27	9.27	9.26	9.25	18.83	10.81	28.44	0
Test	20211117	IDLE	1.51	11.02	9.25	20.19	9.25	9.25	9.27	9.27	9.27	9.27	9.25	18.87	10.83	28.82	0
Train	20211118	IDLE	1.47	11.01	9.25	28.82	9.25	9.25	9.28	9.28	9.28	9.26	9.25	13.83	10.81	28.82	0
Train	20211119	IDLE	1.46	11.37	9.25	28.66	9.25	9.25	9.26	9.29	9.28	9.28	9.25	4.64	10.82	29.03	0
Train	20211122	IDLE	1.48	10.96	9.25	28.37	9.25	9.25	9.28	9.30	9.29	9.27	9.25	18.81	10.81	28.94	0
Train	20211124	IDLE	1.49	11.17	9.24	28.26	9.25	9.25	9.29	9.28	9.28	9.27	9.25	18.79	10.82	29.05	0
Train	20211125	IDLE	1.49	10.98	9.25	28.65	9.25	9.25	9.29	9.29	9.29	9.27	9.25	21.05	10.82	28.61	0
Train	20211126	IDLE	1.50	11.00	9.25	28.43	9.25	9.26	9.28	9.29	9.29	9.27	9.25	18.91	10.82	29.32	0
Train	20211129	IDLE	1.50	11.01	9.32	19.83	9.34	9.34	9.38	9.34	9.34	9.34	9.33	20.23	10.88	23.21	0
Train	20211130	IDLE	1.49	11.16	9.25	19.26	0.00	9.25	9.25	9.26	9.25	9.25	9.25	18.75	10.84	23.20	0
Train	20211201	IDLE	1.49	11.03	9.24	21.10	9.26	9.25	9.25	9.26	9.25	9.25	9.25	18.75	10.84	23.24	0
Train	20211202	IDLE	2.53	11.16	9.25	20.06	9.28	9.25	9.25	9.27	9.24	9.25	9.25	18.84	10.85	23.69	0
Train	20211203	IDLE	1.48	11.34	9.25	28.62	9.27	9.25	9.25	9.27	9.25	9.25	9.33	18.80	10.85	23.51	0
Train	20211207	IDLE	1.48	11.28	9.24	20.42	9.26	9.25	9.25	9.26	9.25	9.25	9.25	20.20	10.84	23.64	0
Train	20211208	IDLE	1.48	11.02	9.25	19.89	1.38	1.32	1.33	1.38	1.33	1.33	1.33	20.07	10.85	23.84	0
Train	20211209	IDLE	1.49	11.30	9.25	20.10	9.26	9.24	9.25	9.25	9.25	9.25	9.25	19.11	10.85	23.40	0
Train	20211210	IDLE	1.48	10.94	9.24	21.11	9.27	9.24	9.25	9.26	9.25	9.25	9.25	18.72	10.84	23.33	0
Test	20211213	IDLE	1.48	11.03	9.25	20.22	9.28	9.24	9.25	9.26	9.25	9.25	9.25	18.82	10.84	23.72	0
Test	20211215	IDLE	1.47	11.43	9.24	27.99	9.26	9.24	9.26	9.24	9.25	9.25	9.25	18.67	10.84	23.40	0
Test	20211216	IDLE	1.46	11.01	9.25	21.12	9.26	9.24	9.25	9.25	9.25	9.25	9.25	18.68	10.85	23.40	0
Test	20211220	IDLE	1.47	11.01	9.24	27.55	9.26	9.24	9.24	9.25	9.25	9.25	9.24	18.66	10.84	23.18	0
Test	20211222	IDLE	1.46	11.02	9.24	27.83	9.26	9.24	9.24	9.25	9.25	9.25	9.24	18.77	10.84	23.18	0
Test	20211223	IDLE	1.46	11.24	9.24	27.58	9.25	9.24	9.24	9.25	9.25	9.25	9.24	18.83	10.85	23.19	0

ML	Date	Type	Luohe Cam Dog	Nest Indoor Camera	Netatmo Camera	Netatmo Weather Station	Philips Hue Bridge	Ring Base Station AC:1236	SIMCAM 1S (AMPAKTec)	Smart Board	Sonos One Speaker	Teckin Plug 1	Teckin Plug 2	Yutron Plug 1	Yutron Plug 2	iRobot Roomba
Train	20211102	ACTIVE	3.03	33.40	9.59	1.81	12.50	3.37	21.45	0.40	5.18	6.68	6.34	6.35	6.68	1.12
Train	20211103	ACTIVE	2.92	24.23	12.21	1.87	12.67	3.26	21.08	0.55	8.71	6.62	6.29	6.53	6.65	1.21
Train	20211105	ACTIVE	2.78	23.78	26.15	1.70	14.03	3.21	20.91	0.38	4.97	6.61	6.46	6.63	6.06	1.05
Train	20211108	ACTIVE	2.98	84.38	10.19	1.82	13.81	3.47	21.96	0.43	4.97	6.59	6.07	6.61	6.12	1.06
Train	20211109	ACTIVE	3.03	43.68	9.17	1.77	12.47	3.34	21.65	0.44	4.99	6.62	6.08	6.64	6.07	1.05
Train	20211110	ACTIVE	3.01	26.28	9.00	1.82	13.79	3.27	21.15	6.15	6.87	6.09	6.11	6.46	6.49	1.16
Train	20211112	ACTIVE	3.02	29.18	10.53	1.91	15.19	3.28	21.48	8.36	6.98	6.25	6.26	6.56	6.39	1.19
Train	20211115	ACTIVE	3.01	67.01	9.43	1.83	17.50	3.85	21.47	3.04	4.25	6.36	6.28	6.58	6.59	1.21
Train	20211116	ACTIVE	3.02	84.55	14.13	1.90	15.72	3.32	22.03	8.06	3.84	6.11	6.01	6.26	6.25	1.17
Train	20211117	ACTIVE	3.02	29.93	9.54	1.91	13.70	3.25	21.23	3.99	4.84	6.15	6.01	6.14	6.35	1.10
Test	20211118	ACTIVE	3.02	29.55	4.31	1.88	12.86	3.20	21.01	3.63	3.89	6.26	6.21	6.28	6.38	1.34
Test	20211119	ACTIVE	3.02	29.57	4.10	1.85	12.70	3.21	20.92	1.37	3.82	6.14	6.00	6.13	6.33	1.12
Test	20211122	ACTIVE	3.02	78.49	9.49	1.82	13.54	3.32	21.26	3.82	3.87	6.13	6.05	6.39	6.38	1.12
Test	20211123	ACTIVE	3.09	74.48	11.61	1.85	12.97	3.33	28.15	5.24	3.93	6.25	6.18	6.57	6.59	1.06
Test	20211124	ACTIVE	3.01	23.02	6.47	1.86	12.57	3.29	27.77	4.74	4.11	6.10	6.23	6.46	6.34	1.03
Test	20211125	ACTIVE	3.03	22.86	5.75	1.90	14.36	3.30	21.16	6.04	5.75	6.10	6.18	6.30	6.31	1.18
Test	20211126	ACTIVE	4.05	39.26	19.26	2.47	17.65	5.69	29.41	212	6.21	8.18	8.72	8.65	8.37	1.56
Test	20211206	ACTIVE	2.97	69.60	11.83	1.77	12.83	3.41	21.78	0.77	4.66	6.08	6.32	6.55	6.55	1.20
Test	20211207	ACTIVE	3.02	27.10	9.89	1.82	14.18	3.81	21.62	0.43	11.62	6.73	6.25	6.68	6.78	1.05
Test	20211208	ACTIVE	3.02	89.40	9.71	1.79	12.85	3.46	21.97	0.51	3.89	6.64	6.07	6.59	6.54	1.06
Train	20211223	ACTIVE	3.02	23.24	3.78	1.77	12.81	3.17	20.89	0.74	4.98	6.81	6.67	6.74	6.46	1.20
Train	20211225	ACTIVE	3.02	16.77	7.10	1.88	12.95	3.20	21.15	2.71	11.82	6.80	6.68	6.75	6.35	1.04
Train	20211228	ACTIVE	3.02	24.72	5.67	1.81	7.08	3.23	21.19	0.61	44.11	6.79	6.68	6.64	6.17	1.10
Train	20220103	ACTIVE	0.00	25.41	3.74	1.68	7.07	3.16	20.91	0.48	4.92	6.68	6.37	6.61	6.60	1.39
Test	20211102	IDLE	4.15	6.63	7.35	2.47	17.05	4.79	28.62	0.35	6.89	9.08	8.70	8.82	9.08	1.43
Test	20211103	IDLE	3.66	6.48	7.27	2.46	16.93	4.70	28.64	0.20	6.82	9.06	8.71	8.95	9.07	1.42
Test	20211104	IDLE	3.80	7.69	7.69	2.53	16.88	4.73	28.60	0.24	6.90	8.91	8.84	8.88	8.56	1.51
Test	20211108	IDLE	4.15	20.06	5.12	2.51	16.89	4.69	28.64	0.05	6.82	9.11	8.36	9.11	8.38	1.52
Test	20211110	IDLE	4.15	8.06	5.17	2.51	17.11	4.68	28.63	3.87	6.86	8.39	8.30	9.12	8.93	1.43
Test	20211112	IDLE	4.15	11.71	5.12	2.55	24.88	4.72	28.64	3.30	6.93	8.41	8.40	9.03	8.70	1.43
Test	20211115	IDLE	4.12	41.22	5.11	2.58	24.87	4.65	28.63	1.41	5.31	8.48	8.38	8.72	8.83	1.43
Test	20211116	IDLE	4.15	40.95	5.20	2.47	17.21	4.69	28.64	0.00	5.33	8.43	8.29	8.56	8.71	1.50
Test	20211117	IDLE	4.16	40.84	5.26	2.59	17.20	4.75	28.65	0.00	21.86	8.41	8.24	8.68	8.69	1.51
Train	20211118	IDLE	4.15	40.66	5.16	2.63	17.17	4.72	28.64	3.38	5.36	8.42	8.27	8.64	8.67	1.48
Train	20211119	IDLE	4.15	40.78	5.21	2.50	17.28	4.71	28.64	1.29	5.42	8.42	8.26	8.56	8.67	1.42
Train	20211122	IDLE	4.16	40.82	5.05	2.56	17.25	4.70	28.63	1.34	5.32	8.35	8.39	8.84	8.87	1.46
Train	20211124	IDLE	4.15	7.00	5.14	2.48	17.18	4.65	37.53	4.55	5.38	8.44	8.38	8.72	8.63	1.43
Train	20211125	IDLE	4.16	6.82	5.07	2.55	17.18	4.68	28.64	0.48	5.37	8.40	8.32	8.44	8.55	1.43
Train	20211126	IDLE	4.18	21.80	5.00	2.50	17.21	4.69	28.62	4.77	5.38	8.37	8.60	8.63	8.62	1.42
Train	20211129	IDLE	4.17	6.52	5.02	2.59	17.31	0.00	28.73	0.39	5.40	8.51	8.62	9.00	9.04	1.48
Train	20211130	IDLE	4.15	6.65	5.08	2.46	17.27	4.70	28.62	0.38	5.34	8.30	8.86	8.25	8.55	1.42
Train	20211201	IDLE	4.14	7.39	5.19	2.56	17.26	4.71	28.63	0.44	5.37	8.28	8.87	8.27	8.71	1.43
Train	20211202	IDLE	4.13	7.83	5.15	2.52	17.25	4.68	28.60	0.50	5.34	8.26	8.80	9.07	8.50	1.43
Train	20211203	IDLE	4.16	8.64	5.14	2.47	17.70	4.67	28.63	14.91	5.20	8.44	8.60	9.10	8.93	1.00
Train	20211207	IDLE	4.15	7.06	5.24	2.61	17.33	6.41	28.63	0.58	5.38	8.45	8.51	8.68	9.01	1.50
Train	20211208	IDLE	4.16	12.79	5.05	2.40	17.26	4.70	28.63	0.41	5.29	9.12	8.34	9.10	9.00	1.44
Train	20211209	IDLE	4.16	14.62	5.21	2.54	17.30	4.70	28.63	0.65	23.08	8.29	8.58	8.39	8.70	1.42
Train	20211210	IDLE	4.15	23.02	5.20	2.37	17.25	4.69	28.59	0.47	6.86	8.31	8.55	8.54	8.79	1.43
Test	20211213	IDLE	4.16	20.58	5.42	2.42	17.26	4.66	28.54	0.42	6.90	8.20	8.62	8.40	8.88	1.42
Test	20211215	IDLE	4.15	21.63	5.16	2.63	17.27	4.66	28.60	0.50	5.38	8.29	8.95	8.27	9.07	1.44
Test	20211216	IDLE	4.15	25.21	5.15	2.46	17.20	4.66	28.61	0.37	5.33	8.28	9.13	8.32	9.11	1.43
Test	20211220	IDLE	4.14	21.32	5.18	2.28	17.32	4.75	28.62	0.52	5.32	9.36	9.15	9.24	8.67	1.58
Test	20211222	IDLE	4.16	22.92	5.20	2.53	17.28	4.69	28.64	2.39	6.84	9.37	9.12	9.28	8.70	1.44
Test	20211223	IDLE	4.16	20.25	5.19	2.54	17.28	4.70	28.62	3.19	6.87	9.36	9.19	9.25	8.68	1.56