

Network Traffic Analysis based IoT Device Identification

Rajarshi Roy Chowdhury
Faculty of Integrated Technologies
19h0901@ubd.edu.bn

Sandhya Aneja
Faculty of Integrated Technologies
sandhya.aneja@ubd.edu.bn

Nagender Aneja
Institute of Applied Data Analytics
nagender.aneja@ubd.edu.bn

Emeroylariffion Abas
Faculty of Integrated Technologies
emeroylariffion.abas@ubd.edu.bn

ABSTRACT

Device identification is the process of identifying a device on Internet without using its assigned network or other credentials. The sharp rise of usage in Internet of Things (IoT) devices has imposed new challenges in device identification due to a wide variety of devices, protocols and control interfaces. In a network, conventional IoT devices identify each other by utilizing IP or MAC addresses, which are prone to spoofing. Moreover, IoT devices are low power devices with minimal embedded security solution. To mitigate the issue in IoT devices, fingerprint (DFP) for device identification can be used. DFP identifies a device by using implicit identifiers, such as network traffic (or packets), radio signal, which a device used for its communication over the network. These identifiers are closely related to the device hardware and software features. In this paper, we exploit TCP/IP packet header features to create a device fingerprint utilizing device originated network packets. We present a set of three metrics which separate some features from a packet which contribute actively for device identification. To evaluate our approach, we used publicly accessible two datasets. We observed the accuracy of device genre classification 99.37% and 83.35% of accuracy in the identification of an individual device from IoT Sentinel dataset. However, using UNSW dataset device type identification accuracy reached up to 97.78%.

CCS Concepts

• Networks → Cyber-physical networks;

Keywords

Internet of Things (IoT); Device Identification; Network Traffic Analysis; Feature Evaluation; Machine Learning.

1. INTRODUCTION

Device identification is the process of identifying a device on Internet without using its assigned network or other credentials

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

BDIOT 2020, August 22–24, 2020, Singapore, Singapore

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7550-4/20/08...\$15.00

<https://doi.org/10.1145/3421537.3421545>

such as IP address, Medium Access Control (MAC) address, Electronic Serial Number (ESN), International Mobile Station Equipment Identity (IMEI) number or Mobile Identification Number (MIN). These identifiers can be altered or manipulated by using expert knowledge of networking or software, which can create significant security threats where devices identification play key security roles. DFP exploits device-specific signature or packet information (feature set) which the device uses for communication over the network [3]. Effective device fingerprint must assure two attributes (i) the features are hard to forge, (ii) the DFP remains stable even when devices move from one network to another network [17]. DFP has emerged as a significant solution for device identification (or authentication) due to its resistance against vulnerabilities such as node forgery or masquerading in IoT network. Features used for DFP are classified based on their extraction methods. There are two techniques namely active and passive which have been observed to impact the identification of devices. In active fingerprinting, devices are probed with different types of packets and the subsequent received responses are analyzed to attain unique fingerprints. In contrast, in passive fingerprinting, a profiler observes ongoing communication of the target system without any obvious querying into the system, by analyzing and extracting information from transmitted packets. In this paper, we used passively observed datasets to evaluate the proposed method, our DFP approach is for passive fingerprinting.

A DFP approach proposed based on the analysis of (each packet) 23 features from 12 network packets by Miettinen et al. [10]. The set of features were represented as binary (0 or 1) and integer values, where binary value 1 define the protocol used during communication. Overall, the scheme achieved identification accuracy of 81.50% (global ratio) over the 27 devices using the IoT Sentinel dataset [10].

The single packet-based DFP method [1] uses genetic algorithm (GA) to determine the feature subset which contributes substantially, for generation of device fingerprint used for device identification. This approach has been shown to achieve approximate classification accuracy of over 95% for device genre and 82% for individual device type using 23 IoT devices from IoT Sentinel dataset [10].

We worked on a device fingerprinting approach using packets header information (distinct protocols header features) from the target devices to extract a unique set of features (actual feature values) to generate device fingerprint. Features are assessed using three metrics: the variability of a feature, the stability of a feature, and the suitability of a feature, to learn the most suitable feature

subset. Metrics evaluation results assign a score to each of the features to form a feature vector. The feature set is subsequently processed to filter feature subset according to the user-defined threshold value λ whereas a lambda value ranged between 0 and less than 1. Finally, we used machine learning (ML) classification algorithms to classify individual IoT device type or genre by using selected feature subset vectors of each device. While two algorithms, such as J48 and PART, outperformed compare to other classification algorithms. However, based on the user expected accuracy λ value can be changed empirically. To evaluate our proposed DFP method we used two publicly available datasets: IoT Sentinel dataset [10] and UNSW dataset [16]. While we achieved approximate classification accuracy to 99.37% for device genre and 83.35% for individual device type by using IoT Sentinel dataset [10] and 97.78% of precision achieved using 10678 instances from UNSW dataset [16] for identification of individual device type. The contributions of this paper are the following:

- We present a conceptual IoT network model incorporating device identification with a wide range of IoT devices (Section 2).
- We introduce metric-entropy based packet header features assessment methodology using three metrics for device fingerprinting and design of our proposed DFP approach (Section 5).
- We examine proposed DFP method using two datasets, which consists of various real-world IoT devices (Section 6).

The rest of this paper is organized as follows: Section 2 describes a conceptual IoT network model to depict the DFP incorporated in the IoT network. In Section 3, notations and device information are presented, and preliminary of the metrics evaluation process is discussed in Section 4. The proposed DFP approach is illustrated in Section 5, while experimental results with the different datasets are discussed in Section 6. Section 7 addresses related work and Section 8 presents the conclusion and the future direction of work.

2. IOT NETWORK MODEL INCORPORATING DEVICE IDENTIFICATION

We presume that our network model comprises of various types of IoT devices, which are used in smart homes, smart buildings or enterprises [14]. In a network, most of the devices, such as IP cameras (TP-LinkCam, EdimaxCam), smart bulbs, connect to the user's network (a single access point gateway) directly either through WiFi (wireless fidelity) or Ethernet connection to access the Internet services.

Most of the devices are identified by using explicit identifiers, such as IP address, MAC address and other network identities. Devices use these identifiers to communicate on the network. Unfortunately, all these identifiers have been exhibited to be easily mutable by using software [5]. Spoofing is one of the techniques which can forge a device explicit identifier to gain illegitimate access to restricted resources in the network. For example, an ioctl system call can be used to forge or modify MAC address of a device network interface card (NIC) [17] to access a network (e.g. a corporate office) where devices are recognized by using their MAC addresses. Similarly, IoT Node using spoofed IP address can launch crucial attacks while communicating with an access point (AP) to get required services. Initially, each device

on the IoT network is identified by using their unique IPv4 address, however, even IP address is spoofed than also it would be unique. Moreover the range of spoofed IP address is restricted by type of the network e.g. 256 in case of Type C network.

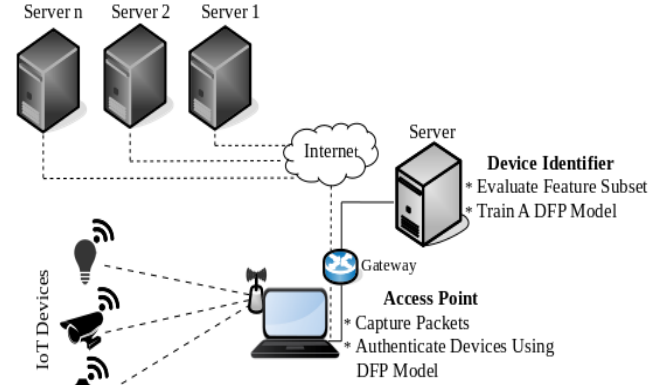


Figure 1. Conceptual IoT network model of the proposed method

IoT devices are classified as resource-constrain devices [6] in terms of processing power, communication capability, memory and energy, while complex cryptography algorithms can be used to secure the device identity. Therefore, while communicating the encrypted data, the processing power of devices must be preserved. In other words, a secure method is required to identify devices without using the traditional identifiers, such as IP address or MAC address. An intuition that network traffic provides device-specific signature since devices comprised different types of hardware-software and used distinct communication protocols in a network. DFP is a technique to identify IoT devices using network traffic information from the target devices. Conceptual network model is shown in Figure 1 wherein the proposed DFP method can be used to generate unique identifiers of IoT devices to overcome the spoofed network identities of the device.

We assume that our proposed DFP scheme is able to capture all types of network traffic (inbound and outbound traffic) passively on the access point (AP). The IoT Node communicate for various services e.g. DNS to translate domain name of Internet resources, SSL/TLS certificates to encrypt and decrypt data, NTP server to synchronize device time, and Manufacture server to upgrade firmware services. This communication of IoT Node is captured on the AP shown in Figure 1. In the proposed model packets payload information are not considered, which can be plain text, compressed, binary or encrypted data. We only focused on packet header features (or fields) [4], [10], [1]. An intuition that distinct protocols features from the packet header provide a significant fingerprint for device identification. This fingerprint is generated using header information of all the packets and regularly updated on the device identifier station. Device identifier evaluates the feature set from n number of packets from each device to determine the most suitable subset of features and then train a model for devices classification using an ML classification algorithm. Finally, the AP uses this classification model to authenticate devices in the IoT network.

3. NOTATIONS AND DATASETS

A list of notations used to describe our proposed DFP approach are listed in Table 1 along with a description of each notation.

Table 1. A list of notations

Notation	Description
pkt	: A TCP/IP packet
t	: Total number of packets or instances
r	: r^{th} number of packet
f	: A feature in a packet
m	: Total number of features in a packet
j	: j^{th} number of feature in a packet
d	: An IoT Device
n	: Total number of devices in a dataset
blk	: A block of packets
k	: Total number of packets in a block
b	: Total number of blocks
str	: A string (concatenated features values)
c	: Total number of distinct characters in a string
i	: i^{th} number of character in a string
$P(str_{f_{ji}})$: Discrete probability density function
$H(str_{f_{ji}})$: Metric-entropy value calculation function
L	: Total length of a string
sd	: Individual device stability vector
v	: Variability vector of the features
s	: Stability vector of the features
u	: Suitability vector of the features
l	: l^{th} number of device in a dataset
//	: Concatenation operation
λ	: A threshold value (Lambda)
*	: Multiplication operation

We utilized publicly available two datasets to evaluate our proposed DFP method. The datasets are IoT Sentinel [10] dataset and UNSW [16] dataset containing different types of IoT devices. We use a total of 27 IoT devices from IoT Sentinel dataset. There are a total 12 genres or groups of devices in IoT Sentinel [10]: Aria, D-Link (D-LinkCam, D-LinkDayCam, D-LinkDoorSensor, D-LinkHomeHub, D-LinkSensor, D-LinkSiren, D-LinkSwitch and D-LinkWaterSensor), Hue (HueSwitch and HueBridge), Smarter (iKettle2 and SmarterCoffee), Lightify, MAXGateway, TP-Link (TP-LinkPlugHS100 and TP-LinkPlugHS110), WeMo (WeMoInsightSwitch, WeMoLink and WeMoSwitch) and Withings. We use a total of 19 devices from UNSW dataset. All these devices information are listed in Table 2 and Table 3.

[Note: ★ indicates devices are from the same manufacturer as above]

4. PRELIMINARIES

Robyns et al. [13] defined three metrics: (a) the variability of a bit (b) the stability of a bit and (c) the suitability of a bit for bitwise entropy evaluation of a Probe-Request frame on a WiFi network.

The defined metric variability attains ideal value 1 when the feature (particular field of packet format) over the multiple transmission from the devices is highly variable over all the devices and therefore uniquely contribute to fingerprint of the device e.g IP address of a device is unique and therefore contribute highly to the feature set for fingerprinting the device.

Table 2. IoT Sentinel dataset- the list of devices [10]

Manufacturer	Device	Connectivity
Fitbit	Aria	WiFi
D-Link	D-LinkCam	WiFi
★	D-LinkDayCam	WiFi/Ethernet
★	D-LinkDoorSensor	Z-Wave
★	D-LinkSensor	WiFi
★	D-LinkWaterSensor	WiFi
★	D-LinkSiren	WiFi
★	D-LinkHomeHub	WiFi/Ethernet/Z-wave
★	D-LinkSwitch	WiFi
Edimax	EdimaxCam1	WiFi/Ethernet
★	EdimaxPlug1101W	WiFi
★	EdimaxPlug2101W	WiFi
Ednet	EdnetCam1	WiFi/Ethernet
Ednet.Living	EdnetGateway	WiFi/Other
HomeMatic	HomeMaticPlug	Other
Philips Hue	HueBridge	ZigBee/Ethernet
★	HueSwitch	ZigBee
Smarter	iKettle2	WiFi
★	SmarterCoffee	WiFi
Osram	Lightify	WiFi/ZigBee
eQ-3	MAXGateway	Ethernet/Other
TP-Link	TP-LinkPlugHS100	WiFi
★	TP-LinkPlugHS110	WiFi
Belkin (Wemo)	WeMoInsightSwitch	WiFi
★	WeMoLink	WiFi/ZigBee
★	WeMoSwitch	WiFi
Withings	Withings	WiFi

Similarly, the defined metric stability attains ideal value 1 when the feature among other features of the device uniquely identify the device and remains stable over the multiple transmissions from the device. The defined metric suitability uses variability, and stability by multiplying to calculate the feature suitability. Robyns et al. evaluated suitability of features e.g. SSID, vendor specific data, HT capabilities in a frame to identify the mobile devices. The authors included the MAC address while evaluating frame for the metrics.

We define the metrics at character-level for packet header analysis on 212 set of features and improve the IoT Sentinel dataset classification compared to [1] without IP address and the method outperformed with IP address. The character-level metric is more promising, as the bit-wise metric evaluate the probability density

of frame vector over $\{0, 1, \text{none}\}$ while character level metric evaluate the probability density value of feature vector over $\{0, 1, \dots, 9, \text{none}\}$. The bit wise evaluation [13] drops the accuracy gradually e.g, the reported accuracy of 10-100 devices including MAC Address is 60-87%.

Table 3. UNSW dataset- the list of devices [16]

Manufacturer	Device	Connectivity
Amazone	AmazoneEcho	WiFi
Belkin	BelkinMotion	WiFi
★	BelkinSwitch	WiFi
Blipcare	BlipcareBpMeter	WiFi
Google Nest	DropCam	WiFi
HP	HP-Printer	WiFi
iHome	iHomePlug	WiFi
Lifx	LiFX	WiFi
Nest	NestSmoke	WiFi
Netatmo	NetatmoCam	WiFi
★	NetatmoWeather	WiFi
Pixstar	PixstarPhotoFrame	WiFi
Samsung	SamsungCam	WiFi
SmartThings	SmartThings	Wired
TP-Link	TP-LinkCam	WiFi
★	TP-LinkPlug	WiFi
Triby	TribySpeaker	WiFi
Withings	WithingsScale	WiFi
★	WithingsSleep	WiFi

5. DEVICE IDENTIFICATION- DEVICE FINGERPRINTING APPROACH

In this section, we describe the major components of our device fingerprinting method to identify IoT device type or genre based on a single TCP/IP packet. This method appraises Network layer (IP and ICMP), Transport layer (TCP and UDP) and Application layer (DNS, HTTP, TLS/SSL and DHCP) protocols header fields (or features) to learn the most suitable subset of features among a large number of features, which can be used for device fingerprinting. We define three character-level metrics, variability of a feature, stability of a feature, and suitability of a feature to assess the feature set. This process identifies a suitable subset of features which are unique to each device, while a lambda value is utilized to filter selected feature subset according to the expected accuracy. Then we perform a machine learning classification algorithm to identify or classify individual IoT device type or genre by using a selected subset of features. The entire process of our proposed DFP scheme is illustrated in Figure 2.

5.1 Packet Header Features

In the proposed system, we used device originated network traffic or packets to extract a list of features from each packet header fields. A total $m = 218$ features extract from a single packet $pkt_r = (f_1, f_2, f_3, f_4, f_5, f_6, \dots, f_{m-1}, f_m)$ based on the contents of a packet header information, where $r \in (1, 2, 3, \dots, t-1, t)$. Most of the values of the features are of integer type except from some

features from the DHCP protocol, which include both integer and string values. For a range of string values, we convert to nominal values by using *StringToNominal* function in WEKA tool [7]. Additionally, we eschew IP address and IP checksum for the proposed DFP. The set of features used for our DFP scheme is partially listed in Table 4, where all features are based on distinct protocols header fields to ensure that the proposed method can be used for all types of network traffic like as plain-text, compressed data, binary or encrypted traffic.

Table 4. List of the protocols header fields (or features)

Layers	Protocols	Features
Link Layer	- - -	- - -
Network Layer	IP, ICMP (33 features)	ip.hdr_len, ip.dsfield, ip.dsfield.dsccp, ip.proto, ip.ttl, ip.opt.len, ip.opt.ra, ip.opt.type, icmp.code, icmp.checksum, icmp.type, icmp.seq, etc.
Transport Layer	TCP, UDP (64 features)	tcp.srcport, tcp.dstport, tcp.hdr_len, tcp.window_size, tcp.checksum, tcp.connection.syn, tcp.analysis.ack_rtt, udp.srcport, udp.checksum, udp.checksum.status, udp.stream, etc.
Application Layer	DNS, HTTP, SSL/TSL, DHCP (121 features)	dns.flags, dns.id, dns.qry.name.len, dns.count.queries, http.request, http.response, http.request_number, tls.handshake.version, tls.handshake, dhcp.id, dhcp.option.hostname, etc.

5.2 Feature Subset Selection

We noticed that from our literature studies [1], [4], [2], [12], [11] a small number of features is sufficient to identify most of the devices with high accuracy. Therefore, instead of manually or randomly selecting a suitable subset of features for DFP, it would be rational to identify the most suitable feature subset automatically from a large number of protocol header features. To accomplish this goal our proposed DFP methodology comprises of the character-level analysis on packet header features using the three metrics [13]: (a) the variability metric of a feature (b) the

stability metric of a feature and (c) the suitability metric of a feature.

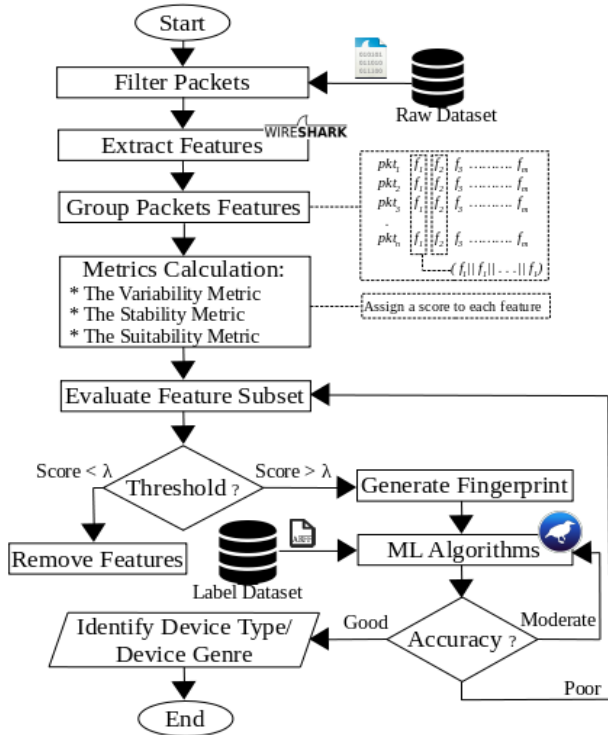


Figure 2. A workflow of the proposed DFP method

We measure on three character-level metrics of a feature obtained from b blocks of packets transmitted by the device. Hence, from every device b number of blocks values are concatenated individually based on each feature, such as $blk_1(f_j)||blk_2(f_j)|| \dots ||blk_{b-1}(f_j)||blk_b(f_j)$, $j \in (1, 2, 3, \dots, m-1, m)$, to determine a metric-entropy [15] value of that particular feature, wherein, a block comprises of k number of packets transmitted by an IoT device over the network during communication. A feature values from k number of packets are concatenated, like as $pkt_1(f_j)||pkt_2(f_j)|| \dots ||pkt_{k-1}(f_j)||pkt_k(f_j)$, to form a string (str_{f_j}) to compute metric-entropy (Eq. 1) value of that feature.

$$H(str_{f_j}) = \frac{-\sum_{i=1}^c P(str_{f_{ji}}) \times \log_2 P(str_{f_{ji}})}{L} \quad (1)$$

Where metric-entropy defines randomness of a feature string, $P(str_{f_{ji}})$ characterizes discrete probability density of individual character from a given string and a total number of unique character in a string is delimited by c . Each metric-entropy value ranges from 0 to less than 1, wherein 0 means no entropy value.

Calculation of The Variability Metric: A feature variability is assessed based on the metric-entropy value of that feature calculated from a different set of devices, where equal number blocks of packets were chosen from each device for assessment. A feature metric-entropy value calculates from n number of blocks, while blocks are concatenated, such as

$$d_1blk_1str_{f_1} || d_2blk_1str_{f_1} || d_3blk_1str_{f_1} || \dots || d_nblk_1str_{f_1},$$

$$d_1blk_1str_{f_2} || d_2blk_1str_{f_2} || d_3blk_1str_{f_2} || \dots || d_nblk_1str_{f_2}, \dots$$

$$d_1blk_1str_{f_m} || d_2blk_1str_{f_m} || d_3blk_1str_{f_m} || \dots || d_nblk_1str_{f_m},$$

individually to compute variability (Eq. 1) value of that feature. The variability vector v can be demonstrated as:

$$v = [H_{str_{f_1}}, H_{str_{f_2}}, H_{str_{f_3}}, H_{str_{f_4}}, H_{str_{f_5}}, \dots, H_{str_{f_m}}]$$

The objective of this variability metric is to identify the least number of features which provide a unique contribution for device fingerprinting.

Calculation of The Stability Metric: The stability vector s define how likely feature values remain stable over multiple transmission from the same device, which can be used for device fingerprinting. Consequently, first we measure individual device stability vector sd_i represented as:

$$sd_i = [1 - H_{str_{f_1}}, 1 - H_{str_{f_2}}, 1 - H_{str_{f_3}}, \dots, 1 - H_{str_{f_m}}]$$

Then, device-specific stability vector from all devices are averaged to assess the final stability vector s . The goal of the stability metric is to identify a set of features that are associated with devices and varies less frequently.

$$s(f_j) = \frac{\sum_{i=1}^n sd_i(f_j)}{n} \quad (2)$$

Calculation of The Suitability Metric: We determine the most suitable set of features by combining the variability and stability metrics. Hypothetically, the features utilized for our DFP approach should be highly variable and stable to achieve higher classification accuracy. For instance, a feature metric-entropy value with 0.5 variability and 0.99 stability provides approximately higher suitability, whereas a feature with zero variability and 0.99 stability is considered unsuitable to use for fingerprint. We incorporate these two metrics by using multiplication operation to attain the suitability vector u of the features (Eq. 3):

$$u = v * s \quad (3)$$

The suitability vector u can be represented as $u = [v_{f_1} * s_{f_1}, v_{f_2} * s_{f_2}, v_{f_3} * s_{f_3}, v_{f_4} * s_{f_4}, \dots, v_{f_{m-1}} * s_{f_{m-1}}, v_{f_m} * s_{f_m}]$. Henceforth, a threshold value λ is used to experimentally assess the feature subset according to the user's expected accuracy and to include those features which have suitability value greater than the threshold. A λ value influences the trade-off between a feature uniqueness and suitability to be used for device fingerprint. Table 5 presents assessment results of some features evaluated by using the three metrics, while features are randomly selected from the IoT Sentinel dataset and a λ value defined experimentally to exclude (x) or include (✓) features for device fingerprinting.

5.3 Device Classification

In the preceding subsection, we explained how to choose the most suitable subset of features from packet header fields that can be used for devices classification by using ML algorithms. We used WEKA (Waikato Environment for Knowledge Analysis) tool [7] to perform various ML classification algorithms, such as PART and J48 Decision Trees, to classify IoT devices based on distinct protocol header features from a single packet of a device. We observed that PART and J48 algorithms provide higher accuracy than other algorithms available in WEKA libraries. Our proposed method utilizes real values of the features to facilitate both algorithms to identify devices with high accuracy.

Table 5. Randomly selected few features suitability metric evaluation results

Features	u	λ
	$u > \lambda$	0.0
ip.hdr_len	0.0002	✓
ip.dsfield	0.0	x
ip.dsfield.dscp	0.0001	✓
ip.proto	0.0005	✓
ip.ttl	0.0004	✓
ip.opt.len	0.0	x
ip.opt.ra	0.0	x
ip.opt.type	0.005	✓
icmp.checksum	0.0427	✓
icmp.type	0.0086	✓
icmp.seq	0.2114	✓
icmp.code	0.0	x
tcp.srcport	0.0005	✓
tcp.dstport	0.0006	✓
tcp.hdr_len	0.0006	✓
tcp.window_size	0.0005	✓
tcp.checksum	0.0002	✓
tcp.connection.syn	0.0	x
tcp.analysis.ack_rtt	0.0	x
udp.srcport	0.0008	✓
udp.checksum	0.0003	✓
udp.stream	0.0022	✓
dns.flags	0.0003	✓
dns.id	0.0003	✓
dns.qry.name.len1	0.0066	✓
dns.count.queries	0.0052	✓
http.request	0.0	x
http.response_number	0.0162	✓
http.chunk_size1	0.1049	✓
tls.handshake.version	0.0045	✓
tls.handshake3	0.0	x
dhcp.id	0.0017	✓

5.3.1 Identification of device genre and individual device type

- **Identification of device genre:** We use a total of 27 IoT devices from IoT Sentinel dataset, those devices are categorized based on their manufacturer name or device name (if a single device from manufacture) and then apply proposed DFP method to classify individual device genre.

- **Identification of individual device type** (based on device name): In this scenario, we utilize both datasets to identify specific device type using our proposed DFP approach.

6. EXPERIMENTAL RESULTS

In this section, we describe our experimental process and results based on the two datasets. First, we extracted all the devices originated network packets or instances from the individual device (.pcap files) by using T-shark [8], with each device identified based on their MAC address or IP address. Then, all the device instances were filtered according to the selected protocols used for our DFP approach and were also labeled according to the individual device type and genre.

All the devices information (46 IoT devices) from both datasets are listed in Table 2 and Table 3, on which our proposed DFP approach was carried out. In our experiment to evaluate the features, we used approximately 100 packets from each device to compute the three metrics explained in Section 5.

However, from our observation, we noticed that in most of the cases, the total number of distinct values of a feature varies from 0 to 100, as shown in Figure 3 e.g. IP address has one distinct value for each device. Hence, we do not consider a feature with a large number of distinct values in our metrics computation separately. We assume that these types of features always provide a prominent result for DFP.

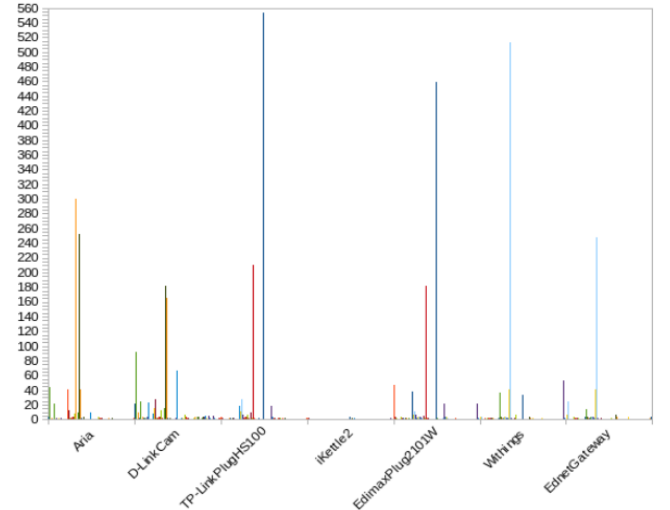


Figure 3. Individual device total number of distinct values of a feature

Table 6. Total number of instances (or packets) used for training and validation

Dataset	Devices	Training	Validation	Total
		70%	30%	100%
IoT Sentinel	27	69489	29782	99271
UNSW	19	7474	3204	10678

6.1 Classification Performance

Every dataset instances (or packets) were randomly divided into a training dataset (70% of the dataset) and a validation dataset (30% of the dataset) by using WEKA tool to perform device

classification or identification based on ML algorithms. We used unsupervised-resample function to split the datasets accordingly. Initially, we select the suitable feature subset from the attributes list in WEKA preprocess stage based on the metrics evaluation results and a threshold value calculated from the dataset to remove insignificant features. We performed metrics evaluation process two thousand times with different sets of packets and finally averaged the suitability metric values to assign a score for each feature. Table 6 lists the total number of instances used for our DFP approach according to different datasets and Table 7 reports the total number of features selected according to the threshold values in our experiment. From our observation, we notice that when the threshold value decreased devices identification accuracy and the number of features increased respect to the individual dataset.

Table 7. Total number of selected features based on λ value

Dataset	Total Number of Features	Threshold (λ)	Selected Features
IoT Sentinel	218	0.0	161
UNSW	218	0.0004	86

To assess the proposed DFP method classification performance, we used several ML classification algorithms to identify individual device genre and device type based on a single packet from each device. By using IoT Sentinel dataset [10] our method achieved mean classification accuracy of 99.37% for device genre with threshold value 0.0 experimentally, as shown in Figure 4. There were total 12 genres or groups of devices namely Aria, D-Link (D-LinkCam, D-LinkDayCam, D-LinkDoorSensor, D-LinkHomeHub, D-LinkSensor, D-LinkSiren, D-LinkSwitch and D-LinkWaterSensor), Hue (HueSwitch and HueBridge), Smarter (iKettle2 and SmarterCoffee), Lightify, MAXGateway, TP-Link (TP-LinkPlugHS100 and TP-LinkPlugHS110), WeMo (WeMoInsightSwitch, WeMoLink and WeMoSwitch) and Withings. We grouped the devices based on their manufacture name or device name if a single device from the manufacture in the dataset.

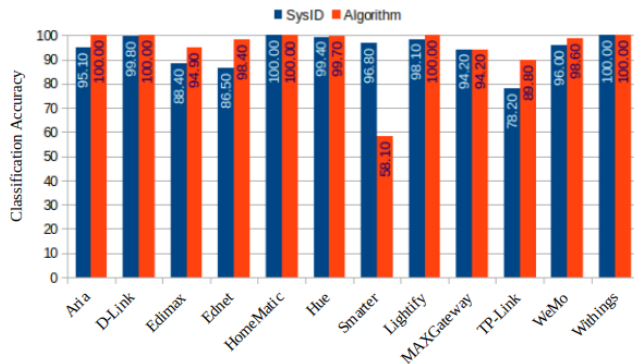


Figure 4. IoT Sentinel dataset: device genre classification performance

Figure 4 presents our proposed DFP scheme attained higher classification accuracy compare with SysID. However, the Smarter group of devices exhibited lower classification performance by using the proposed method. However, overall the

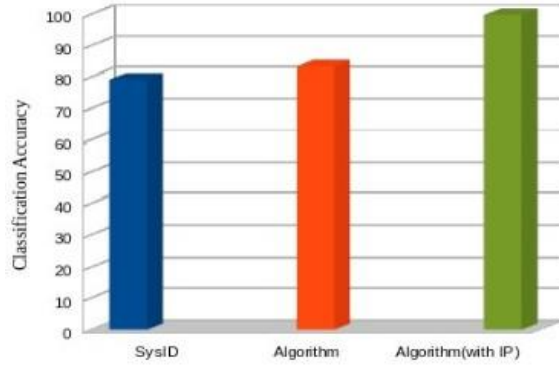
proposed method reached a higher classification rate to identify individual manufacturer with multiple devices.

In terms of individual device type classification performance, the proposed scheme achieved overall 83.35% of accuracy using IoT Sentinel dataset [10] as shown in Figure 5(a) Although the method was unable to provide higher rate of classification accuracy on some of the devices but the proposed scheme combined with IP address (as a feature) reached maximum accuracy up to 99.77%. On the other hand, SysID achieved approximately 79.20% of accuracy using the same dataset. Figure 7 exhibits the confusion matrix of the proposed algorithm (83.35% accuracy), while rows define individual device type (or real device) and columns show predicted classification. It would be the optimal case when all the instances from a dataset intersect at the point of both actual and predicted classes. For example, 179 instances from HomeMaticPlug were precisely classified as HomeMaticPlug. However, for the SmarterCoffee, 9 packets were classified correctly as SmarterCoffee and other 7 packets were identified incorrectly as Lightify, which indicated the wrong classification.

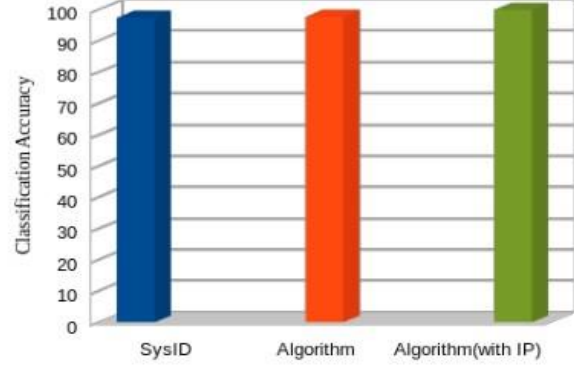
Furthermore, Figure 6(a) exhibits individual device classification accuracy based on the proposed method (with and without IP address) and SysID. Although the proposed scheme (without IP address) classified most of the devices correctly compare with SysID, in some cases like as D-LinkSwitch, TP-LinkPlugHS100, WeMoSwitch, the scheme was not able to identify accurately as SysID. In the experiment on UNSW dataset [16], we observed that our proposed method and SysID provided almost similar results, as shown in Figure 5(b) and Figure 6(b), whereas both methods achieved classification accuracy over 97%. From the both figures, it was noticeable that our proposed method able to achieve 99.93% of accuracy when we included IP address (such as *ip.src*) as a feature with the selected subset of features. However, Figure 6(b) presents both methods (SysID and the proposed scheme) performed higher classification precision except for some cases, like as BelkinSwitch and BlicareBpMeter. From the experimental results on UNSW dataset, we noticed that our proposed DFP method achieved approximately similar accuracy as SysID. Even though both algorithms used a single TCP/IP packet for device identification. SysID performed higher precision when device types varies, but when devices from the same manufacturer performance decline gradually. On the other hand, our proposed DFP scheme exhibits comparatively good result even when a manufacturer with multiple devices.

Experimental results shown in Figure 6 indicate that the accuracy of the proposed DFP method is improved by using IP address as one of the features to incorporate with the selected feature subset. In a network, every device must have a unique IP address to communicate over the network nevertheless, it is spoofed or altered. Our DFP scheme can utilize this IP address with other features to detect a device type, which helps to mitigate spoofing attack. Hence, our proposed method achieved 99.77% accuracy from 83.35% to classify individual device type using IP address as an additional feature on IoT Sentinel dataset. Furthermore tested on UNSW dataset including IP address the proposed DFP approach achieved approximate 99.93% of accuracy.

7. RELATED WORK



(a) IoT Sentinel dataset



(b) UNSW dataset

Figure 5. Overall classification performance: (a) IoT Sentinel dataset and (b) UNSW dataset

The process of gathering device-specific signature from the analysis of network traffic to form a device fingerprint and using them by various classification algorithms to identify individual device type or genre provide significant solutions to distinct research issues. In our study, we mainly focus on network traffic (or packets) analysis based DFP approaches and publicly available commonly used two datasets namely IoT Sentinel [10] dataset and UNSW [16] dataset, which consists of different types of IoT devices.

A DFP approach proposed based on the analysis of 12 consecutive network packets from a target device by Miettinen et al. [10]. The authors extracted 23 features from a packet header includes Link layer (ARP, LLC), Network layer (IP, ICMP, ICMPv6, EAPoL), Transport layer (TCP, UDP) and Application layer (HTTP, HTTPS, DHCP, BOOTP, SSDP, DNS, MDNS, NTP) protocols, IP options, Packet content, IP address and Port class to form $(12 * 23) = 276$ -dimensional feature vector for a device fingerprint. The set of features represented as binary (0 or 1) and integer values, whereas binary values 1 define some protocols are used during communication. However, from each device, a set of fingerprints are utilized to train a model using Random Forest classification algorithm to perform device identification. The scheme generates one classifier per device type. While a new device type fingerprints captured by the scheme, it is required to train a new classifier to avoid any modification of the existing classifiers. Overall, the scheme achieved identification accuracy of 81.50% (global ratio) over the 27 devices from IoT Sentinel dataset [10]. In comparison, our scheme uses less number of protocol header information from a single packet to construct a device fingerprint and we are able to classify devices (type or genre) by using a single classifier.

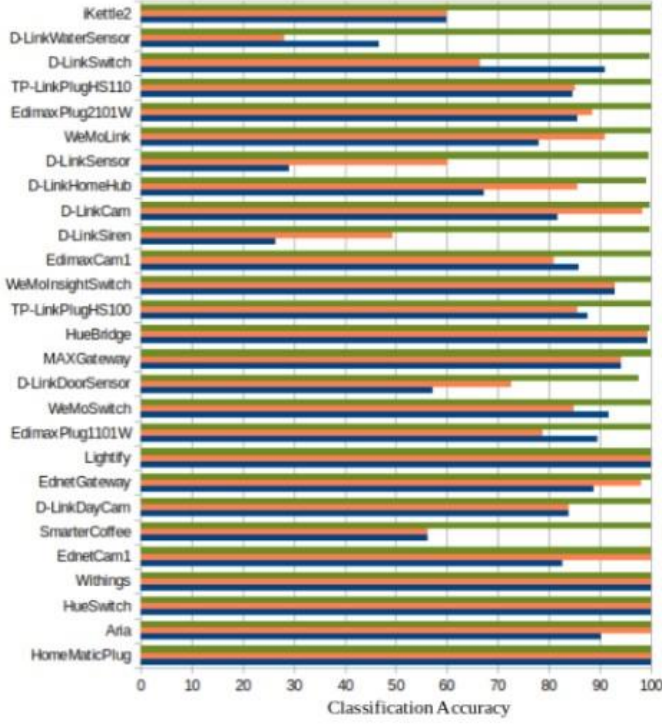
Bezawada et al. [4] presented a device identification mechanism based on the analysis of devices behavioral fingerprint. The researchers extracted 20 features from a packet, which includes distinct protocols from the Link layer, Network layer, Transport layer and Application layer, IP options, entropy of payload, TCP payload length and TCP windows size. While the 17 features are utilized as binary values of 0 or 1 to represent the absence or presence of a particular feature and the 3 other features use as numerical values. The authors grouped device originated 5 packets (as a session) to form a feature vector of $(5 * 20) = 100$ features for a device fingerprinting. Finally, these feature vectors are used to train a machine learning classification algorithm to predict the device type. From the test-bed of 10 unique devices,

the scheme achieved a true positive rate of 99.0% to 100.0% using all the selected features. Though, the identification accuracy decreased on average 2% without using payload entropy. In contrast, we use only the packet header information to preserve the privacy of users' data and generate a feature vector from a single TCP/IP packet.

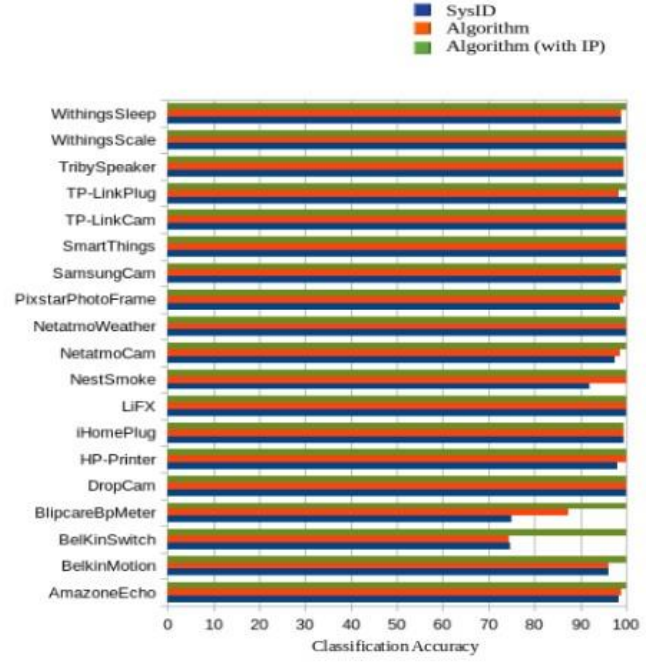
Ammar et al. [2] described a DFP approach based on the analysis of service discovery protocols (mDNS and SSDP), DHCP and HTTP protocols on IoT Sentinel [10] dataset and their own dataset. The authors extracted features as textual information from the first few sets of device originated packets. This textual information from each device modeled as a binary data by using the Bag of Words (BOW) technique for device identification. The scheme first extracts a list of unique words from all devices and then each device information turn into a feature vector, whereas a binary value set to 1 if the word present in the device information and set to 0 if not. Then all the devices information are used to form a matrix while each column defines a unique word and each row represents a device. For classification, devices are evaluated using a similarity verification process. The scheme identified 31 devices among 33 of the unique test devices. However, in some cases, accuracy can be decreased when IoT devices may not use any discovery protocol. Comparing to this scheme, our method uses packet header features of protocols in Table 4 to train a model using ML classification algorithm for device identification.

Guo et al. [9] discussed a device identification approach using network traffic analysis and knowledge of devices manufacturer server names. The scheme tracked a list of server names each device communicate regularly and then defined a threshold number of server names require for device identification. Server names are filtered to exclude third-party and human-facing servers

to improve accuracy. The authors examined the presence of server names in network traffic from a given IP address to identify what types of IoT devices present in the network. From a controlled experiment on 10 IoT devices and 15 non-IoT devices, the reported accuracy of the scheme was 96%. Our method has some significant differences. We use only packet header features to identify devices without tracking any server names along with IP address. Our scheme can be used to identify IoT devices in a network where manufacturer of the devices are different. When the manufacturer is same in our scheme e.g. 8 D-Link devices the individual device identification is not sufficiently competent 85.35% without IP address and 99.77% with IP address.



(a) IoT Sentinel dataset



(b) UNSW dataset

Figure 6: Individual device type classification performance: (a) IoT Sentinel dataset and (b) UNSW dataset

Aksoy et al. [1] proposed a DFP method to identify an IoT device (type or genre) using any single TCP/IP packet content from the device. The scheme analyzed the Network layer (IP, ICMP), Transport layer (TCP, UDP) and Application layer (DNS, HTTP and SSL) protocols header information to extract 212 features from each packet. Then applied a genetic algorithm (GA) to select feature subsets to create device fingerprint. These selected feature subsets used to train various ML classification algorithms to identify devices. The scheme utilizes two-level of classification to improve accuracy. Whereas first determine the manufacture of devices and then identify individual device type from the same manufacturer using different classifiers. The experiment on IoT Sentinel [10] dataset over the 23 IoT devices the average classification accuracy was 82% for the cumulative of all the devices and over 95% of precision for device genre classification. However, the scheme was not able to achieve a higher rate of accuracy since some devices had similar network behaviors from the same manufacturer. The main difference between our scheme and SysID is feature subset evaluation process. We use character-level analysis of each feature based on the three metrics, while

SysID uses a GA to select feature subsets to generate device fingerprinting.

Sivanathan et al. [16] described a framework for IoT devices classification approach by using statistical analysis of network traffic characteristics. The authors extracted network traffic passively to compute traffic characteristics. To characterize network traffic from a device, the authors converted all the raw pcap files into flows using the Joy tool on an hourly basis and then calculated statistical attributes of traffic activity patterns (flow volume, flow duration, average flow rate and device sleep time) and signaling patterns (server port numbers, DNS queries, NTP queries and Cipher suite). This framework consists of two stages, in Stage-0 each multi-valued attributes are fed into

corresponding Naive Bayes classifiers in the form of BoW to generate two outputs - a tentative class and a confidence level and outputs are provided into the next stage. In Stage-1, all the quantitative attributes from traffic flow and outputs of Stage-0 use to train a classifier based on the Random Forest algorithm to perform device identification. The accuracy of the experiment was above 99% for a set of 28 IoT devices. However, the scheme requires a large number of packets from a network flow for statistical analysis. In contrast, our scheme uses a single TCP/IP packet header information for classification of devices, while features are evaluated using the three metrics.

Robyns et al. [13] presented mobile devices fingerprinting scheme based on a single Probe-Request frame. The scheme used per bit entropy analysis of the frames from different devices using the three metrics evaluation process to construct device fingerprint (bit patterns), while frames captured by monitoring stations (MSs) without concern of device users. This non-cooperative MAC layer fingerprinting method uses to track mobile devices location. The accuracy of the scheme achieved between 80.0% to 67.6% for a small dataset of 50 to 100 devices, while for a large dataset of 1000 to 10000 devices the accuracy varied from 33.0% to 15.1%. The scheme provides a mechanism to defeat MAC address randomization problem conjunction with some temporal information. In comparison, our method uses in a home network to capture network packets on an AP and features are extracted from each packet header information to construct suitable feature subset for device fingerprinting. In our DFP scheme, features are evaluated on character-level using the three metrics to identify the most suitable feature subset.

8. CONCLUSION

In this paper, we presented a DFP approach using device originated network traffic (or packets) and machine learning

classification algorithms to identify individual device type or genre based on a single packet header features. We learned the most suitable feature subset automatically by utilizing three metrics assessment process of the features. This process assigned a score to every feature according to their significance for device identification, whereas a user-defined threshold value helped to remove noisy features and improve classification accuracy. Overall, the proposed DFP scheme achieved 99.37% of precision to identify each device genre and 83.35% of accuracy to classify individual device type from IoT Sentinel dataset. However, for some devices, the proposed method was not able to achieve a higher rate of accuracy since some devices have similar network behaviors from the same manufacturer. On the other hand, the scheme reached a mean accuracy of 97.78% using UNSW dataset, where most of the devices were from the different manufacturers.

In future, we will investigate other packet header features logged from the network protocols during communication to identify unique feature subset to improve accuracy of the same manufacturer e.g. D-Link 20 devices. Moreover, the problem will be more intense when finding feature for the same device from the same manufacturer e.g. 20 D-Link Cam.

9. REFERENCES

- [1] Ahmet Aksoy and Mehmet Hadi Gunes. 2019. Automated iot device identification using network traffic. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 1–7.
- [2] Nesrine Ammar, Ludovic Noirie, and Sébastien Tixeul. 2019. Network-Protocol-Based IoT Device Identification. In *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 204–209.
- [3] Sandhya Aneja, Nagender Aneja, and Md Shohidul Islam. 2018. IoT Device Fingerprint using Deep Learning. In *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*. IEEE, 174–179.
- [4] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. 2018. Behavioral fingerprinting of iot devices. In *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*. 41–50.
- [5] John R Douceur. 2002. The sybil attack. In *International workshop on peer-to-peer systems*. Springer, 251–260.
- [6] Garcia-Morchon et al. 2018. State-of-the-Art and Challenges for the Internet of Things Security draft-irtf-t2trg-iot-seccons-16 (Internet-Draft). URL: <https://tools.ietf.org/pdf/draft-irtf-t2trg-iot-seccons-16.pdf>.
- [7] Eibe Frank, Mark A. Hall, and Ian H. Witten. 2016. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", fourth edition ed. Morgan Kaufmann, 2016.
- [8] Combs et al. Gerald. 1998. D.2. tshark: Terminal-based Wireshark. URL: https://www.wireshark.org/docs/wsug_html_chunked/index.html. Accessed on: February 9, 2020.
- [9] Hang Guo and John Heidemann. 2018. IP-based IoT device detection. In *Proceedings of the 2018 Workshop on IoT Security and Privacy*. ACM, 36–42.
- [10] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. 2017. IoT Sentinel: Automated device-type identification for security enforcement in IoT. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2177–2184.
- [11] Antônio J Pinheiro, Jeandro de M Bezerra, Caio AP Burgardt, and Divanilson R Campelo. 2019. Identifying IoT devices and events based on packet length from encrypted traffic. *Computer Communications* 144 (2019), 8–17.
- [12] Sakthi Vignesh Radhakrishnan, A Selcuk Uluagac, and Raheem Beyah. 2014. GTID: A technique for physical deviceanddevice type fingerprinting. *IEEE Transactions on Dependable and Secure Computing* 12, 5 (2014), 519–532.
- [13] Pieter Robyns, Bram Bonné, Peter Quax, and Wim Lamotte. 2017. Noncooperative 802.11 mac layer fingerprinting and tracking of mobile devices. *Security and Communication Networks* 2017 (2017).
- [14] Noman Shahid and Sandhya Aneja. 2017. Internet of Things: Vision, application areas and research challenges. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 583–587.
- [15] Claude E Shannon. 1948. A mathematical theory of communication. *Bell system technical journal* 27, 3 (1948), 379–423.
- [16] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2018. Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing* 18, 8 (2018), 1745–1759.
- [17] Qiang Xu, Rong Zheng, Walid Saad, and Zhu Han. 2015. Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials* 18, 1 (2015), 94–104.

	Aria	D-LinkCam	D-LinkDayCam	D-LinkDoorSensor	D-LinkHomeHub	D-LinkSensor	D-LinkSiren	D-LinkSwitch	D-LinkWaterSensor	EdimaxCam1	EdimaxPlug1101W	EdimaxPlug2101W	EdnetCam1	EdnetGateway	HomeMaticPlug	HueBridge	HueSwitch	iKettle2	Lightify	MAXGateway	SmarterCoffee	TP-LinkPlugHS100	TP-LinkPlugHS110	WeMoInsightSwitch	WeMoLink	WeMoSwitch	Withings
Aria	163	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D-LinkCam	0	1822	0	0	19	0	0	9	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D-LinkDayCam	0	45	237	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D-LinkDoorSensor	0	21	2	484	6	1	12	107	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	
D-LinkHomeHub	0	53	0	54	2176	51	40	167	3	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	
D-LinkSensor	0	34	0	0	53	1150	302	331	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D-LinkSiren	0	66	0	0	45	417	909	378	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D-LinkSwitch	0	76	0	0	165	130	255	1270	8	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	
D-LinkWaterSensor	0	75	0	0	63	471	428	331	537	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
EdimaxCam1	0	20	0	0	0	0	0	0	0	114	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
EdimaxPlug1101W	0	0	0	0	0	0	0	0	0	0	237	61	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
EdimaxPlug2101W	1	2	0	0	0	0	0	0	0	0	27	232	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
EdnetCam1	0	0	0	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	0	0	0	0	0	0	0	
EdnetGateway	0	0	0	0	0	0	0	0	0	0	0	0	0	194	0	0	0	0	4	0	0	0	0	0	0	0	
HomeMaticPlug	0	0	0	0	0	0	0	0	0	0	0	0	0	0	179	0	0	0	0	0	0	0	0	0	0	0	
HueBridge	0	13	0	0	2	4	0	1	0	0	0	2	0	0	0	3982	4	0	0	0	0	0	0	0	1	3	
HueSwitch	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5519	0	0	0	0	0	0	0	0	0	
iKettle2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	6	0	0	0	0	0	0	0	
Lightify	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1232	0	0	0	0	0	0	0	
MAXGateway	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	161	0	0	0	0	0	0	
SmarterCoffee	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	9	0	0	0	0	0	
TP-LinkPlugHS100	0	0	0	0	4	1	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	144	12	0	0	0	
TP-LinkPlugHS110	0	0	0	0	7	2	0	11	0	0	0	0	0	0	0	2	0	0	0	0	0	7	166	0	0	0	
WeMoInsightSwitch	0	49	0	0	0	0	0	0	0	0	1	31	0	0	0	0	0	0	0	0	0	0	0	1099	3	0	
WeMoLink	0	86	0	0	8	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	49	1804	32	0	
WeMoSwitch	0	124	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	10	20	920	0	
Withings	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	170	

Figure 7. IoT Sentinel dataset: confusion matrix