Context-Aware Behavioral Fingerprinting of IoT Devices via Network Traffic Analysis

Arjun Prasad*, Kevin Kanichery Biju*, Soumya Somani* and Barsha Mitra†
Department of CSIS, BITS Pilani, Hyderabad Campus, Hyderabad, India

Keywords: IoT Device Fingerprinting, Behavioral Fingerprint, Operating Context, Traffic Analysis, Packet Level Features.

Abstract: The large scale proliferation of IoT devices has necessitated the requirement of securing these devices from a massive spectrum of cyber security threats. IoT device fingerprinting is a defense strategy that can help to detect unauthorized device subversion and the consequent anomalous activities by identifying device behavior and characteristics. Device fingerprinting can be done by analyzing the network traffic features of the IoT devices present in a network, thereby creating a blueprint of normal device behavior and clearly distinguishing it from any kind of abnormal behavior. Since IoT devices operate under varying dynamic conditions, it is implicit that a single device exhibits different behavioral patterns under different contexts and operating modes. In this paper, we propose a context-aware behavioral fingerprinting of IoT devices that takes into account the circumstances or contexts under which the devices are operating. Each context results in a fingerprint and the complete behavioral fingerprint of an IoT device is the combination of all such fingerprints. We perform packet level feature engineering for finding the best possible set of features for performing device fingerprinting. Our fingerprinting strategy uses supervised learning for classifying the IoT devices. We have created an IoT test bed setup consisting of a gateway and several IoT devices. We have collected network traffic data of these IoT devices and have tested the efficacy of our proposed approach on these real data. Experimental results show that our fingerprinting technique is quite effective and is capable of identifying IoT devices with more than 94% accuracy.

1 INTRODUCTION

The Internet-of-Things (IoT) comprises of a collection of devices with network interfaces that can communicate with each other over a communication network, wired or wireless. This allows a variety of low cost, low processing power, energy efficient devices to be used to monitor various systems and environmental parameters, thereby improving the overall quality of human life. Today, IoT has found wide ranging applications in smart home devices, smart appliances, smart cities, smart transportation, connected cars etc. The recent diversification in the types and manufacturers of IoT devices has led to a large amount of data being exchanged over the network. With the personalization of these IoT devices, the network traffic, sometimes, carries private and sensitive user information. As a result, this wide array of devices become easy targets for intruders to gain access to user networks and perpetrate different types of attacks. Every year numerous attacks are detected against IoT devices. These attacks compromise user privacy by exposing sensitive information to malicious individuals and at times, are capable of disrupting social harmony.

Securing IoT devices using traditional security solutions like cryptographic techniques is impractical due to their limited capabilities in terms of memory capacity, processing power, battery etc. To compound these issues, users of IoT devices include people from varying demographics, not all of whom are aware of the risks associated with the IoT applications. Thus, IoT device security is often overlooked in the networks where they reside. In certain cases, the device firmware and software may not even be upgraded and the default access credentials are not changed by the users due to lack of awareness about the security concerns and their implications. Moreover, due to the increase in the number of IoT device manufacturers and

*Arjun Prasad, Kevin K. Biju and Soumya Somani have equal contribution
†Corresponding Author

Prasad, A., Biju, K., Somani, S. and Mitra, B.
Context-Aware Behavioral Fingerprinting of IoT Devices via Network Traffic Analysis.
DOI: 10.5220/0012056000003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 335-344
ISBN: 978-989-758-666-8; ISSN: 2184-7711

the huge device count involved, manufacturers themselves may at times overlook security concerns, even if unintentionally, leading to vulnerabilities in the devices that can be exploited.

To prevent subversion and exploitation of IoT devices, continuous monitoring of IoT devices and their network traffic using an automated device identification strategy is needed. Such a system creates a digital footprint of the IoT devices by determining their normal network behavior. Any deviation from this normal behavior is flagged as an anomaly. A system supporting IoT device fingerprinting can detect and identify any misbehaving or malfunctioning device and isolate it from the network, thereby limiting the potential of causing damage for any attack. Moreover, this requires minimal to no intervention from the user. The user may subsequently be alerted using a notification system to check the device and update or remove it from the network.

Identification of IoT devices can be realized using *Behavioral Device Fingerprinting*. This refers to creating unique device fingerprints for every device using its network traffic data, which can be either packets or flows. Once a set of fingerprints has been established, the network activity of the device can be monitored to ensure that it matches the device fingerprint. Any deviation from this behaviour can be classified as suspicious for that device. Such a scheme allows for the creation of a low energy, efficient and secret free security mechanism that can run even on resource-constrained IoT gateways and edge devices.

The creation of unique device fingerprints poses its own challenges due to the diversity of devices and protocols used in the IoT environment. The IoT devices also communicate with their manufacturers' servers using various encryption schemes and protocols. The IP address of a device may vary over time due to dynamic assignment and leasing using protocols such as DHCP. Additionally, an IP Address may easily be spoofed by a maliciously acting device. Thus, identities such as IP Address and packet data parsing schemes for creating fingerprints are not suitable in an IoT environment.

The device fingerprint needs to be created using the network data characteristics i.e., the characteristics and types of messages that the device exchanges over the network. The fingerprints must be automatically and remotely created by an entity that monitors the network traffic, such as a gateway or an edge device. These fingerprints must also be unique to a device and must not be masqueraded easily by an attacker. However, the fingerprints need to take into account that IoT devices operate under ever-changing dynamic conditions which in turn affect the device behaviors. Consequently, an IoT device functions differently under different operating conditions and therefore, exhibits different network traffic characteristics under these conditions. Thus, the fingerprinting technique should account for the difference in behavior for a particular device under different circumstances and should appropriately consider all the behavioral patterns. To address the aforementioned aspects related to security of IoT devices, we propose a *Context-Aware Behavioral Fingerprinting* technique for these devices.

The contributions of the paper are as follows:

- We propose a *Context-Aware IoT Device Fingerprinting* strategy that takes into consideration varied behavioral characteristics exhibited by the IoT devices under different functioning contexts. Each context generates a distinct fingerprint for a device. The overall behavioral fingerprint of the IoT device is the collection of all the context-specific fingerprints. Thus, our technique employs hierarchical approach for device fingerprint creation.
- Our proposed technique analyzes the network traffic characteristics of the IoT devices and uses several packet level features of the network traffic data. In this work, we use supervised learning for performing IoT device fingerprinting.
- The behavioral fingerprints are created considering a collection of packets rather than individual packets. For this, we employ a method called packet level aggregation.
- We have created an IoT test bed setup using a Raspberry Pi acting as an IoT gateway and 8 IoT devices. We have collected network traffic of each IoT device over a certain time window. These network data have been used for performing experiments to evaluate our proposed fingerprinting approach.
- Experimental results on the data collected from our test bed setup show that our proposed context-aware behavioral fingerprinting method is capable of effectively identifying different IoT devices and achieves an accuracy of more than 94%.

2 RELATED WORK

This section presents a survey of the existing literature on IoT device fingerprinting. In (Miettinen et al., 2017), the authors propose IoT Sentinel, a two step classification system to identify IoT devices. They also propose a Software Defined Networking (SDN) based policy enforcement strategy to

isolate compromised or malicious devices. The authors in (Sivanathan et al., 2017) propose a traffic activity based model for device fingerprinting. They use various network traffic attributes such as sleep time, active volume, average size of packet, peak rate, mean rate, active time, total number of servers contacted, total number of protocols used, DNS interval and NTP interval. Bezawada et al. (Bezawada et al., 2018) provide an approach that creates a behavioural profile for quantifying the various behaviours of devices belonging to a particular device type. They create initial device-specific behavioural profiles and continuously and automatically validate the device behaviours against the profiles. (Hamad et al., 2019) present a behavioral IoT device fingerprinting technique that extracts features from network flows. Sivanathan et al. (Sivanathan et al., 2019) propose a flow-based classification and device fingerprinting model. They created a test bed setup consisting of 28 different IoT devices along with several other non-IoT devices, all connected to a gateway. The work in (Thangavelu et al., 2019) presents DEFT, a supervised learning based technique for IoT device fingerprinting and identification of new devices.

Lorenz et al. (Lorenz et al., 2020) propose a hardware fingerprinting model based on PUF (Physically Unclonable Functions) based authentication. The authors in (Aftab et al., 2020) propose a clustering based tracking and categorization of online streaming applicable for embedded systems. Yadav et al. (Yadav et al., 2020), analyze the performance of several existing IoT fingerprinting methods to identify the most optimal factors and determine the machine learning models most suitable for fingerprinting. The authors in (Khandait et al., 2021) propose a flow-based classification technique called Deep Packet Inspection (DPI) that uses keywords related to names of devices, domain names, etc. Chakraborty et al. (Chakraborty et al., 2021) attempt to reduce the cost associated with the feature vector selected for device fingerprinting by presenting a cost-based feature selection strategy.

An IoT device identification model that uses 111 network packet based features to model device behaviour is presented in (Kostas et al., 2022). Thom et al. (Thom et al., 2022) propose a real-time IoT device identification strategy. Wan et al. in (Wan et al., 2022) present DevTag, a network packet based IoT device fingerprinting technique. DevTag incorporates both model-based and rule-based methods of fingerprinting. The authors in (Kuzniar et al., 2022) propose PoirIoT, a fast system capable of detecting and identifying IoT devices. IoTXray (Yan et al., 2022) is a device fingerprinting method that takes into account the reuse and re-branding of IoT devices. Aramini et al. (Aramini et al., 2022) propose a distributed on-device IoT fingerprinting technique. Kurmi and Matam in (Kurmi and Matam, 2022) focus on creating IoT device fingerprints from the network traffic traces. In (Abdallah et al., 2022), a method for identifying IoT and LoRa devices based on radio features is presented. Another radio frequency based device fingerprinting technique has been presented in (Ren et al., 2022) that makes use of semi-supervised deep learning models. Li et al. (Li and Cetin, 2021) propose a neural network based radio frequency oriented device fingerprinting method for IoT devices. In (Ma et al., 2022), Ma et al. have designed a spatial and temporal fingerprinting method that can handle large IoT networks. The authors in (He et al., 2022) put forth a network traffic analysis based fingerprinting for IoT platforms and have built IoTPF, a network traffic analysis tool. Other works related to IoT device fingerprinting include (Jiao et al., 2021) and (Wanode et al., 2022).

To the best of our knowledge, few works in the existing literature incorporate the notion of separate device fingerprints for different operating conditions and thus focus on context-specific behavioral fingerprinting of IoT devices. Hence, in this paper, we present a context-aware fingerprinting technique that considers the operating contexts of IoT devices and the resultant behaviors.

3 PROPOSED METHODOLOGY

In this section, we present our proposed behavioral fingerprinting method. We also discuss about the feature set extracted from the network traffic packets that are used for identifying the IoT devices.

3.1 Context-Aware Behavioral Fingerprinting

IoT networks are ever-changing and are quite dynamic in nature. Hence, the devices functioning in such networks operate under dynamic conditions. The behavior of an IoT device is determined by the condition as well as the context in which it functions. Every IoT device goes through different phases in its entire operating life cycle. Initially, when a device is introduced in the network for the very first time, it goes through a setup phase. After this, the device transitions to the functioning phase in which it performs its usual activities. This phase can also be considered as the transmission or sensing phase. In this phase, an IoT device can exhibit different behavior

depending on the operation it is performing and the type of data it is sensing. This in turn is guided by the type of device. Moreover, the nature of transmission can also vary across devices, like it can be broadcast, multicast, etc. Apart from these, a device can also be doing maintenance related activities in its operating life cycle. Thus, based on the context and the activities an IoT device carries out, it exhibits distinct behavioral characteristics. The incorporation of the context-specific behavioral characteristics in the device fingerprints helps to create unique fingerprints leading to more accurate device identification and distinction between normal and abnormal activities. For this reason, we present a context-aware behavioral fingerprinting technique for IoT devices that uses packet-level features. The details of the technique are presented below.

Let I be an IoT network that consists of m devices $D_1, D_2, \ldots, D_m$. Let D_i be one such device of the network ($1 \leq i \leq m$). Each IoT device performs specific network activities during a particular behavior phase and hence a distinct fingerprint is associated with each such phase. Suppose the number of possible behavior phases of device D_i is k. Let P denote the set of all such behavior phases of D_i. Thus, $P = \{P_1, P_2, \ldots, P_k\}$. Assume that device D_i performs a set of n network activities during a particular behavior phase P_j. It is to be noted here that we do not simply associate a single network activity with a particular behavior phase. Suppose N is the set of all such network activities of D_i for phase P_j. Thus, $N = \{ac_1, ac_2, \ldots, ac_n\}$ where each ac denotes a single network activity. Let FP_i^j be the fingerprint associated with behavior phase P_j (and hence the set N of network activities) of device D_i. We call such a fingerprint as *phase fingerprint*. If D_i goes through k phases, then there are k such phase fingerprints for D_i. Here, we assume that k is the maximum possible number of phases for any device in the IoT network I. Some of the devices may go through lesser number of phases. We assume that each phase fingerprint coupled with the corresponding phase constitute a *phase identifier*. We represent each phase identifier as a tuple $\langle FP_i^j, P_j \rangle$ for D_i. Each such phase identifier contributes to the behavioral fingerprint of D_i. Thus, the behavioral fingerprint BP_i of device D_i is the set of all the individual phase identifiers. Hence,

$$BP_i = \{\langle FP_i^1, P_1 \rangle, \langle FP_i^2, P_2 \rangle, \ldots, \langle FP_i^k, P_k \rangle\} \quad (1)$$

The behavioral fingerprint BP_i serves as the network-wide unique identifier for D_i. We denote the *behavioral identifier* of D_i as a tuple $\langle D_i, BP_i \rangle$. We define the behavioral profile of the entire IoT network I consisting of m devices as Network Profile and denote it as NPF. NPF is given as:

$$NPF = \{\langle D_1, BP_1 \rangle, \langle D_2, BP_2 \rangle, \ldots, \langle D_m, BP_m \rangle\} \quad (2)$$

The problem of IoT device fingerprinting can now be defined as: *Given a network activity ac and the corresponding fingerprint F_{ac} of a yet unidentified IoT device D_l, determine the behavioral identifier $\langle D_l, BP_l \rangle \in NPF$ such that F_{ac} corresponds to some phase identifier $\langle FP_l^j, P_j \rangle \in BP_l$ and consequently, $F_{ac} \in \langle D_l, BP_l \rangle$. If $F_{ac} \notin \langle D_l, BP_l \rangle \; \forall \, l, 1 \leq l \leq m$ (m = number of IoT devices), then classify F_{ac} to be the fingerprint of a new device D_{m+1}.*

This implies that we need to classify the given network traffic activity and the corresponding fingerprint into a particular device's behavior. It is to be noted here that if D_l is one of the devices present in I, then at the time of fingerprinting it is unknown that F_{ac} corresponds to the behavioral identifier of D_l. If F_{ac} does not correspond to the behavioral profile of any of the existing devices, then F_{ac} is identified as the fingerprint (possibly partial) of a new device. Subsequently, the behavioral identifier of the new device need to be created and D_l is added to the list of devices of I.

In this work, we have essentially designed a hierarchical fingerprinting approach where every operating phase generates a phase identifier and all such phase identifiers cumulatively constitute the behavioral identifier of a device. In this regard, our technique is different from the one presented in (Bezawada et al., 2018). The duration of operation of IoT devices is not fixed in a network. Hence, it may so happen that an IoT device remains operational for a short period of time and in that time window operates in a single behavior phase and thus exhibits a single phase-level fingerprint. In such a scenario, the phase identifier as well as the behavioral identifier of the device will consist of the single fingerprint. Irrespective of the number of individual fingerprints present in the behavioral fingerprint, a fingerprinting method should be capable of identifying the IoT devices.

3.2 Feature Selection

Feature selection is an important step in IoT device fingerprinting since the selected feature set affects the accuracy with which the devices are identified. In this work, we consider packet-level features for context-aware behavioral fingerprinting. Our objective is to select an optimal yet small sized feature set that is capable of accurately classifying IoT devices in a network.

We have selected features pertaining to packet header as well as packet payload. We have taken into account a total of 10 packet-level features, out

Table 1: Packet Header Features and their Values.

Feature	Possible Values
Data Link Layer Protocol	ARP
Network Layer Protocol	IP, ICMPv6
Port Identifiers	Source Port Type, Destination Port Type
Transport Layer Protocol	TCP, UDP
Application Layer Protocol	HTTP, DNS, MDNS, SSDP, BOOTPROTOCOL, DHCP

of which 5 are packet header features and the remaining are packet payload features. The packet header features selected for fingerprinting include, (i) Data Link Layer protocol, (ii) Network Layer protocol, (iii) Port Identifiers, (iv) Transport Layer protocol and (v) Application Layer protocol. Table 1 summarizes the possible values for the above mentioned features. These features can be considered as static features and have been considered in (Miettinen et al., 2017). The packet header features are binary in nature, implying that each one of them is either present (represented using a 1) or absent (represented using a 0).

We have also considered a set of 5 packet payload based features. These include (i) TCP Receive Window Size, (ii) TCP Payload Length, (iii) Packet Size, (iv) Packet Rawdata and (v) Payload Entropy. These features can be considered as dynamic features. Next, we discuss the intuition behind the inclusion of various payload based features in the feature vector used for device classification.

Payload Entropy provides information about the contents of a packet. Entropy is low for well-structured plaintext data and high for encoded (JSON, XML etc.) or encrypted (SSL/TLS) data. The packet payload is parsed as bytes, each byte taking a value between 0 and 255). Entropy E is then calculated as (Bezawada et al., 2018) - $E = -\sum_{i=0}^{255} pr_i * log(pr_i)$, where pr_i is the probability of the i^{th} byte value occurring in the payload and log denotes base-2 logarithm.

TCP Receive Window Size has been proposed as a device fingerprinting feature by the authors in (Sivanathan et al., 2019). It encapsulates the memory limits of the device. IoT devices are generally restricted in terms of memory and processing resources. Due to this, the authors use a smaller TCP Receive Window Size. Moreover, this feature has been shown to be highly variable in various device classes by Bezawada et al. (Bezawada et al., 2018) and hence contribute considerably towards the behavioral fingerprints of the devices. Packet Size and TCP Payload length represent the length of the messages sent by the IoT devices. This can also be highly variable depending on the processing and memory capabilities of the devices. This kind of variability is key to a good discrimination of devices (Bezawada et al., 2018).

4 TEST BED SETUP

In this section, we describe in detail the IoT test bed setup that we have created and the procedure using which we have collected network traffic traces of the IoT devices included in the setup.

4.1 Hardware Architecture

We designed an IoT network consisting of 8 IoT devices and a gateway. Each IoT device was created by connecting a sensor to a micro-controller board (NodeMCU) that reads the sensor data and broadcasts it over WiFi. We used the following 8 sensors to create our IoT devices - (i) Raindrop sensor, (ii) Vibration sensor, (iii) IR sensor, (iv) IR distance sensor, (v) Smoke sensor, (vi) Sound sensor, (vii) Photo sensor and (viii) Tilt sensor. Each IoT device was connected to a Raspberry Pi that acted as a gateway. The Raspberry Pi was operated using a set of peripherals that included a monitor, a mouse and a keyboard. Fig. 1 depicts the graphical representation of the overall setup architecture and the associated legends. Two USB hubs were also used for having the requisite number of USB ports for the connections.

We connected a sensor to a micro-controller board via the appropriate wiring. The board was powered via USB, and we powered the sensor from the board itself via jumper wires. The actual hardware components and the interconnections among them are shown in Fig. 2. In this figure, the 8 sensors constituting the 8 IoT devices, the Raspberry Pi along with the NodeMCUs and the USB hubs have been labelled.

4.2 Data Collection

After completing the hardware setup, we collected network traffic data for each of the IoT devices. For data collection, each NodeMCU (micro-controller board) was programmed using the Arduino IDE. The NodeMCU read the data from the sensor as programmed, and broadcasted it after connecting to the WiFi network. The code snippet running on the NodeMCU read the data from the sensor and then embedded it into an HTML webpage, along with the up time of the NodeMCU for debugging purposes. The HTML was then sent by the NodeMCU to the gateway device (Raspberry Pi). The webpage was refreshed automatically, getting new data every few seconds from the NodeMCU. This webpage was kept open in the browser of the Raspberry Pi.

We captured the packets transmitted by the IoT devices in a format that we can process and use for training machine learning models. We used the Pi for

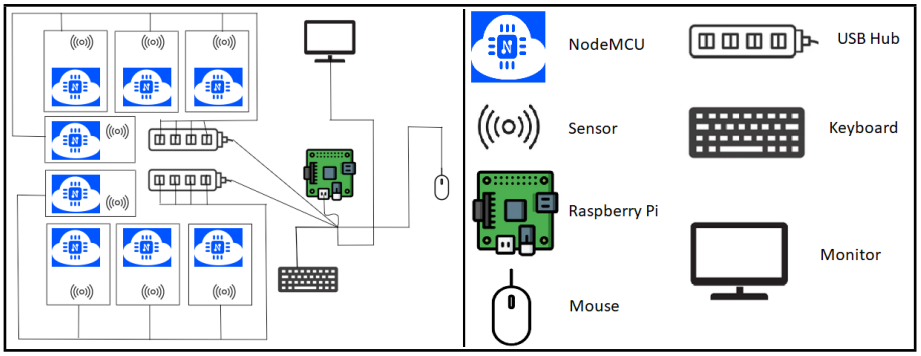

Figure 1: Graphical Representation of the IoT Test Bed Architecture.

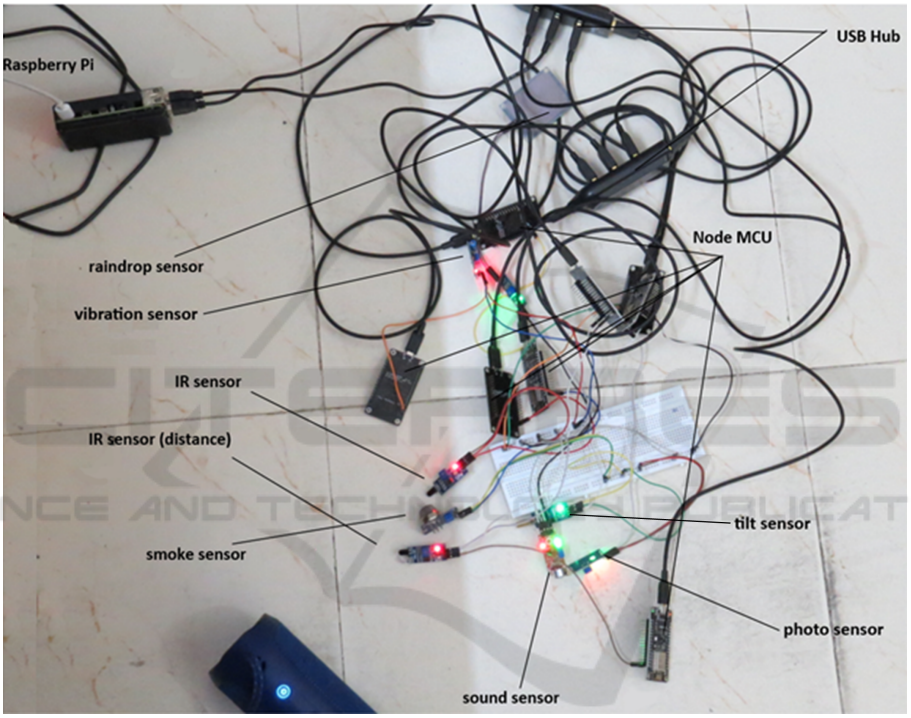

Figure 2: IoT Test Bed Setup consisting of 8 IoT devices and Raspberry Pi.

this, setting it up to capture packets from multiple devices using Wireshark. We kept the auto-refreshing website open and captured packets over a 6 hours time window for each device, and saved it as a .pcap file for further processing. We collected a total of 6,00,000 packets from the entire setup across all devices. However, some of these packets are the ones that were sent by the Raspberry Pi back to the devices and hence do not contribute to any device fingerprint. Table 2 lists the number of packets captured for each individual device that can be used for fingerprint generation. We name our devices as D1, D2 and so on. Henceforth, will refer to the devices using this naming scheme. We extract the features mentioned in Sub-section 3.2 from these data packets.

Table 2: Packet Count for Individual IoT Device.

Device Name	D1	D2	D3	D4
Packet Count	18,427	15,543	73,497	25,320
Device Name	D5	D6	D7	D8
Packet Count	72,751	19,064	11,273	23,574

5 PERFORMANCE EVALUATION

In this section, we discuss how we have prepared our captured dataset for running experiments and also present the results of our experimental evaluation. We conclude this section by analyzing the results and the overall performance of our proposed fingerprinting approach.

5.1 Dataset Preparation

The number of packets captured per device is not uniform for all the devices as can be observed from Table 2. Hence, we performed random sampling to ensure a balanced class distribution and prevent any sort of bias in model training which in turn can affect the performance of behavioral fingerprinting. We randomly sampled 10,000 packets for each IoT device and thus created the experimental data corpus consisting of a total of 80,000 packets for all the 8 devices.

Existing literature shows that creating fingerprints for IoT devices on a collection of data packets is more effective than a fingerprint created with a single packet. The cumulative pattern exhibited by a set of packets encapsulates a more distinguishing device behavior rather than a single packet. Some works highlighting this aspect include (Miettinen et al., 2017), (Bezawada et al., 2018), (Hamad et al., 2019), (Sivanathan et al., 2019). Hence, we have performed packet level aggregation by sorting the packets of each device as per their timestamps. It is to be noted here that packet aggregation was done after random sampling the data for each device. As a result of packet aggregation, we obtained an aggregated behavioral fingerprint for each device, i.e., an aggregation of say x packets contributing to a phase-level fingerprint of a device. We refer to x as *Packet Aggregation Count* (*PAC*). In fact, *PAC* seamlessly fits into our hierarchical fingerprinting strategy. One or more sets of aggregated packets create a phase identifier for a device. The value of *PAC* plays a crucial role in the overall fingerprinting accuracy.

5.2 Experimental Results

We have used supervised learning for performing context-aware behavioral fingerprinting. The dataset obtained after random sampling and packet aggregation was split into 80% and 20%, 80% of the data was used for model training and 20% was used for testing. Our fingerprinting approach used Decision Tree (Quinlan, 1987) and Random Forest (Breiman, 2001) for identifying the 8 IoT devices. We have used the features mentioned in Sub-section 3.2. We have employed one-hot encoding for creating the feature vector. As a result, the size of our final feature vector is 18. It can be noted here that apart from the packet header features listed in Table 1, other features were also captured in the traffic packets. However, these features were largely non-existent, as exhibited by the presence of 0s corresponding to those features in the feature vector. Hence, these features were dropped from the final feature vector. The implementation was carried out using the scikit-learn library and the experiments were executed on a laptop with Intel Core i7 processor, 16 GB RAM and running Windows 11 as the operating system.

We have experimented with different values of *PAC* (like 1, 5 and 10) and found that a *PAC* value of 5 gives the best results. Therefore, we present the results for this value of *PAC*. Each phase-level fingerprint of a device consists of several sets of 5 consecutive packets of the device. We had 2,000 fingerprints for each device and a total of 16,000 fingerprints across all the devices. Thus, the complete behavioral fingerprint of an IoT device consisted of all the 2000 fingerprints. It is to be noted here that, our devices were operational in two modes - sensing mode in which the device captured data specific to its type and passive mode in which the device was simply turned on but was not sensing anything. However, this is applicable for a subset of the devices like devices containing raindrop sensor, smoke sensor, tilt sensor, etc. Other devices like the ones containing sound sensor or photo sensor are capable of picking up environmental data continuously as long as they are powered on.

We evaluate the performance of our proposed fingerprinting technique in terms of the following metrics:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1-score = \frac{2*Precision*Recall}{Precision+Recall} \quad (6)$$

In Eqns. 3 - 6, TP, FP, TN and FN refer to True Positives, False Positives, True Negatives and False Negatives respectively. We have computed the metric values for each of the 8 device classes. The overall metric values for the proposed method have been calculated as macro averages of each of the corresponding class specific values. Table 3 summarizes our experimental results. The results show that Random Forest gives better performance for device fingerprinting than Decision Tree. For Random Forest, we are able

Table 3: Experimental Results for Behavioral Fingerprinting.

Metric	Classifier	
	Decision Tree	**Random Forest**
Accuracy	91.36%	94.66%
Precision	91.64%	94.68%
Recall	91.38%	94.65%
F1-Score	91.50%	94.66%

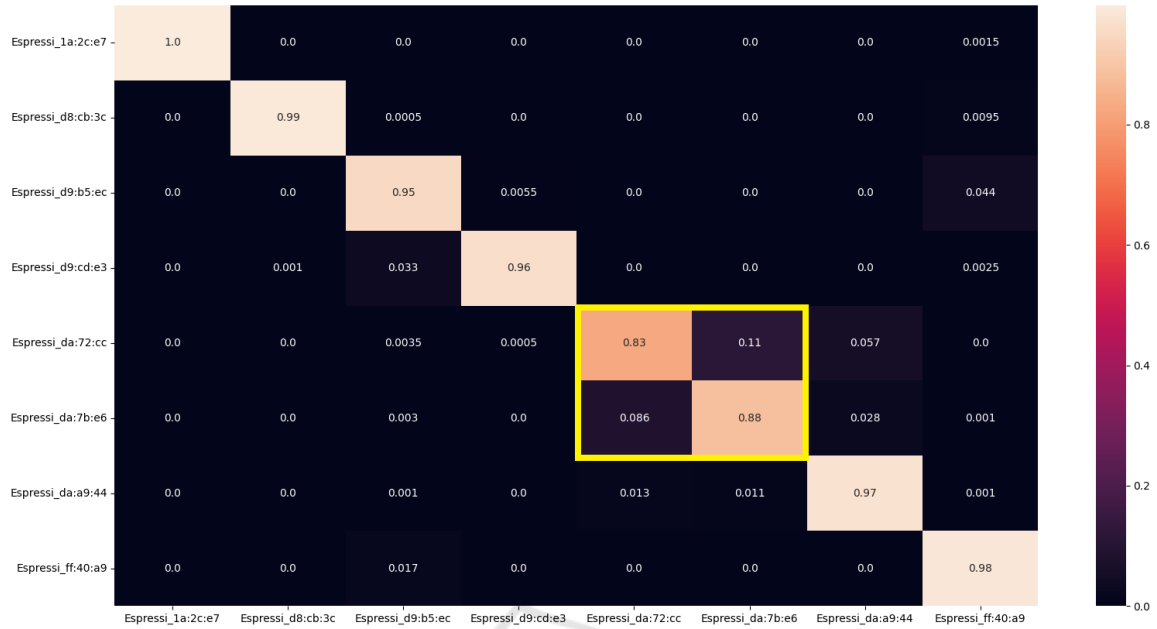

Figure 3: Confusion Matrix depicting Classification Issue for D2 and D6 for Random Forest.

ARP	IP	ICMP6	TCP	UDP	TCP_Window_Size	HTTP	DHCP	BOOTP	SSDP	DNS	MDNS	NTP	PortTyp_Src	PortTyp_Dst	Pck_Size	Pck_Rawdata	Payload_Len	Entropy	MAC_Src	Label	Device
0	1	0	1	0	1628	1	0	0	0	0	0	0	1	3	390	1	384	4.938631822	58:bf:25:da:7b:e6	Espressi_da:7b:e6	D6
0	1	0	1	0	1628	1	0	0	0	0	0	0	1	2	389	1	349	4.964494386	58:bf:25:da:72:cc	Espressi_da:72:cc	D2

Figure 4: Feature Vectors of 2 Typical Packets of D2 and D6.

to obtain more than 94% Accuracy, Precision, Recall and F1-score. In this regard, our proposed technique is capable of accurately identifying the IoT devices. Moreover, the packet-level features that we have selected is quite effective in terms of device identification. In the next sub-section, we provide an insight into the analysis of the experimental results.

5.3 Discussion

We analyze the results in order to determine the factors that affect the performance of our behavioral fingerprinting approach. We find that our technique is facing the issue of poor discrimination of 2 device classes, D2 and D6. This is also evident from the confusion matrix of Fig. 3 for Random Forest classifier. The labels along the horizontal and the vertical axes are the device labels, given as their MAC addresses. The yellow colored rectangle corresponds to the devices D2 and D6. For each of these 2 classes, the lowest device classification accuracies of 83% (for D2) and 88% (for D6) are observed. Moreover, 8.6% of the total number of packets of D2 are predicted as D6 and 11% of the total number of packets of D6 are predicted as D2. Thus, the classifier is not always able to distinguish correctly between the packets generated from these 2 devices. This, however, does not occur for the remaining 6 device classes.

In order to figure out the reason behind the poor classification performance for D2 and D6, we examine the packets from these devices. It is interesting to note that one device consists of a smoke sensor and another one consists of a sound sensor. The feature vectors of 2 typical packets of these two devices are shown in Fig. 4. In this figure, the feature vector in the first row is for D6 and the one in the second row is for D2. As is evident from the figure, the feature vectors for the 2 sample packets are very similar. The value of Entropy also varies by a small margin ($\approx$ 0.03) for these 2 devices. Fig. 5 shows a comparison of several feature vectors of D2 and D6. From the figure, it can be seen that a large number of feature vectors of these devices are similar implying that the feature vectors do not provide sufficient information to properly distinguish between D2 and D6. This leads to the relatively poor identification of these 2 classes. Moreover, their reporting interval is also similar. Due to these factors, the issue of device confusion occurs resulting in poor classification performance.

Our proposed context-aware behavioral IoT device fingerprinting strategy uses simple tree-based classifiers and is able to provide good performance

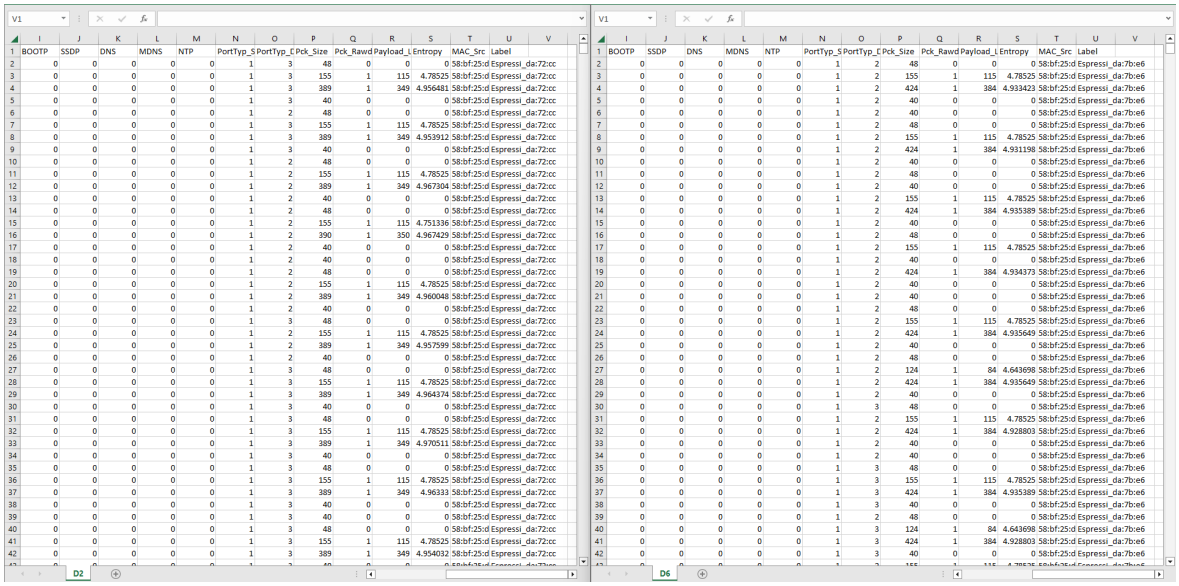

Figure 5: Feature Vectors of Multiple Packets from D2 and D6 (Left Half: D2, Right Half: D6).

for device identification. Hence, our approach is quite light-weight and can be run even on resource constrained gateways and edge devices. Moreover, we have used only 10 features for classifying the IoT devices. Even with one-hot encoding, the feature vector is of size 18 which can be easily handled by systems with low computational power. Our approach can also be extended to identify new IoT devices introduced in the network. Basically, any given fingerprint that cannot be mapped to an existing behavioral identifier can be considered as belonging to a new device. This will be useful to detect the presence of any malicious IoT device introduced in the network or any existing malfunctioning device. In either case, the packets generated from such devices will be classified as anomalous and malicious.

6 CONCLUSIONS

In this paper, we have proposed a machine learning enabled context-aware behavioral fingerprinting technique for classifying IoT devices. The proposed strategy takes into account the diverse phases encountered by an IoT device in its operating life cycle which result in the generation of distinct device fingerprint corresponding to each phase for a single device. The overall behavioral fingerprint is considered as the collection of all such phase-specific fingerprints. Our method analyzes the network traffic data from the IoT devices to extract the best possible feature set for device identification and uses simple tree-based classifiers for performing fingerprinting. We have also created an IoT test bed setup to collect network traffic from actual IoT devices. Experimental results on these real datasets show that we are able to achieve good device classification performance.

In future, we plan to scale up the test bed setup by adding more devices, maybe more complex ones and enhancing our existing data corpus. We plan on applying the proposed strategy for new as well as malicious device identification. In the present work, we have considered a packet feature based approach towards fingerprinting using a multi-class classification strategy. A future direction of work can be focused towards the use of one-vs-all classification strategy which can also include the distinction between IoT and non-IoT devices, identification of new IoT devices, exploring a packet flow-based approach and a combination of packet-based and flow-based strategies to solve the fingerprinting problem. We also intend to experiment with other types of features related to network traffic.

REFERENCES

Abdallah, A., de Abreu, M. F. B., Vieira, F. H. T., and Cardoso, K. V. (2022). Toward secured internet of things (IoT) networks: A new machine learning based technique for fingerprinting of radio devices. In *IEEE 19th Annual Consumer Communications & Networking Conf.*, pages 955–956.

Aftab, M., Chau, S. C., and Shenoy, P. (2020). Efficient online classification and tracking on resource-

constrained IoT devices. *ACM Trans. Internet Things*, 1(3):1–29.

Aramini, A., Arazzi, M., Facchinetti, T., Ngankem, L. S. Q. N., and Nocera, A. (2022). An enhanced behavioral fingerprinting approach for the internet of things. In *IEEE 18th International Conference on Factory Communication Systems*, pages 1–8.

Bezawada, B., Bachani, M., Peterson, J., Shirazi, H., Ray, I., and Ray, I. (2018). Behavioral fingerprinting of IoT devices. In *Workshop on Attacks and Solutions in Hardware Security*, page 41–50.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Chakraborty, B., Divakaran, D. M., Nevat, I., Peters, G. W., and Gurusamy, M. (2021). Cost-aware feature selection for IoT device classification. *IEEE Internet of Things Journal*, 8(14):11052–11064.

Hamad, S. A., Zhang, W. E., Sheng, Q. Z., and Nepal, S. (2019). IoT device identification via network-flow based fingerprinting and learning. In *18th IEEE Int. Conf. On Trust, Security And Privacy In Computing And Communications/13th IEEE Int. Conf. On Big Data Science And Engineering)*, pages 103–111.

He, X., Yang, Y., Zhou, W., Wang, W., Liu, P., and Zhang, Y. (2022). Fingerprinting mainstream IoT platforms using traffic analysis. *IEEE Internet of Things Journal*, 9(3):2083–2093.

Jiao, R., Liu, Z., Liu, L., Ge, C., and Hancke, G. (2021). Multi-level IoT device identification. In *IEEE 27th Int. Conf. on Parallel and Distributed Systems*, pages 538–547.

Khandait, P., Hubballi, N., and Mazumdar, B. (2021). IoTHunter: IoT network traffic classification using device specific keywords. *IET Networks Journal*, 10(2):59–75.

Kostas, K., Just, M., and Lones, M. A. (2022). IoTDevID: A behavior-based device identification method for the IoT. *IEEE Internet of Things Journal*, 9(23):23741–23749.

Kurmi, J. and Matam, R. (2022). Device identification in IoT networks using network trace fingerprinting. In *Int. Conf. on Smart Applications, Communications and Networking*, pages 1–6.

Kuzniar, C., Neves, M., Gurevich, V., and Haque, I. (2022). IoT device fingerprinting on commodity switches. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9.

Li, B. and Cetin, E. (2021). Design and evaluation of a graphical deep learning approach for rf fingerprinting. *IEEE Sensors Journal*, 21(17):19462–19468.

Lorenz, F., Thamsen, L., Wilke, A., Behnke, I., Waldmüller-Littke, J., Komarov, I., Kao, O., and Paeschke, M. (2020). Fingerprinting analog IoT sensors for secret-free authentication. In *29th Int. Conf. on Comp. Communications and Networks*, pages 1–6.

Ma, X., Qu, J., Li, J., Lui, J. C. S., Li, Z., Liu, W., and Guan, X. (2022). Inferring hidden IoT devices and user interactions via spatial-temporal traffic fingerprinting. *IEEE/ACM Trans. on Networking*, 30(1):394–408.

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A., and Tarkoma, S. (2017). IoT SENTINEL: Automated device-type identification for security enforcement in IoT. In *IEEE 37th Int. Conf. on Distributed Computing Systems*, pages 2177–2184.

Quinlan, J. (1987). Simplifying decision trees. *Int. Journal of Man-Machine Studies*, 27(3):221–234.

Ren, Z., Ren, P., and Zhang, T. (2022). Deep rf device fingerprinting by semi-supervised learning with meta pseudo time-frequency labels. In *IEEE Wireless Communications and Networking Conference*, pages 2369–2374.

Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., and Sivaraman, V. (2019). Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Trans. on Mobile Computing*, 18(8):1745–1759.

Sivanathan, A., Sherratt, D., Gharakheili, H. H., Radford, A., Wijenayake, C., Vishwanath, A., and Sivaraman, V. (2017). Characterizing and classifying IoT traffic in smart cities and campuses. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 559–564.

Thangavelu, V., Divakaran, D. M., Sairam, R., Bhunia, S. S., and Gurusamy, M. (2019). DEFT: A distributed IoT fingerprinting technique. *IEEE Internet of Things Journal*, 6(1):940–952.

Thom, J., Thom, N., Sengupta, S., and Hand, E. (2022). Smart recon: Network traffic fingerprinting for IoT device identification. In *IEEE 12th Annual Computing and Communication Workshop and Conf.*, pages 0072–0079.

Wan, S., Li, Q., Wang, H., Li, H., and Sun, L. (2022). Devtag: A benchmark for fingerprinting IoT devices. *IEEE Internet of Things Journal*, 10(7):6388–6399.

Wanode, S. S., Anand, M., and Mitra, B. (2022). Optimal feature set selection for IoT device fingerprinting on edge infrastructure using machine intelligence. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6.

Yadav, P., Feraudo, A., Arief, B., Shahandashti, S. F., and Vassilakis, V. G. (2020). Position paper: A systematic framework for categorising IoT device fingerprinting mechanisms. In *2nd Int. Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, pages 62–68.

Yan, Z., Li, Z., Li, H., Yang, S., Zhu, H., and Sun, L. (2022). Internet-scale fingerprinting the reusing and rebranding IoT devices in the cyberspace. *IEEE Trans. on Dependable and Secure Computing*, pages 1–18.