2024 20th International Conference on Network and Service Management (CNSM)

IoT Device and State Identification based on Usage Patterns

Jeffrey A. Adjei
Faculty of Computer Science
Dalhousie University
Halifax, Canada
jeffrey.adjei@dal.ca

Nur Zincir Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Canada
zincir@cs.dal.ca

Biswajit Nandy
Solana Networks
Ottawa, Canada
bnandy@solananetworks.com

Nabil Seddigh
Solana Networks
Ottawa, Canada
nseddigh@solananetworks.com

***Abstract*—In this paper, we explore usage patterns for the identification of IoT devices and their corresponding states. Machine Learning (ML) methods are trained on IoT device traffic patterns to recognize the state that the device is in. Three device states are the focus of this study - Power-up, Idle and Active. Devices are visible and open to cyber attacks from the moment they are powered on. Previous studies have focused primarily on identifying IoT devices which are in the active state. This study advances the research domain by exploring all three states of an IoT device. Eight different ML algorithms are evaluated using three different feature sets extracted from device network traffic, using flow analysis tools - Tranalyzer2, NFStream and Zeek. They are rigorously assessed to accurately identify diverse IoT devices under normal operational conditions over the aforementioned three states. .**

***Index Terms*—IoT Identification, State Detection, AI/ML, Robustness, Generalization**

I. Introduction

By 2025, industry reports estimate that there will be a staggering 27 billion connected IoT devices worldwide [1]. In this new era of a hyper-connected world, the ability to identify and model the behaviours of IoT devices is an important step to guarantee smooth and secure network operations.

In this research, we continue from our previous work [2] by exploring the intricacies of IoT device identification across three states, namely Power-On, Idle and Active. Our goal is to provide valuable insights that can guide strategies for efficient network operations, robust security measures, and streamlined governance of the IoT ecosystem. To this end, we evaluated nine widely used IoT devices namely - Amcrest Camera, Smarter Coffeemaker, Ring Doorbell, Amazon Echodot, Google Nestcam, Google Nestmini, Kasa Powerstrip, Samsung 32 inch Smart Television (TV), and Amazon Smartplug, under three states. IoT network traffic data was generated and captured on our testbed where all communication occurred over IEEE 802.11 (WiFi) in 2.4GHz channels. Eight Machine Learning (ML) algorithms were studied and trained using statistical and metadata exported by the Tranalyzer2 [3], NFStream [4] and Zeek [5] flow analysis tools. IoT devices in the testbed were instrumented such that both manual and automated methods were utilized for data generation and collection.

The main objective of this research is to assess the robustness, generalization and efficacy of ML models utilized to identify IoT devices and their state based on their network traffic characteristics, i.e. without deep packet inspection (DPI). We believe that this will enable the models to generalize well from one network to another. The proposed approach can also operate successfully in environments where DPI faces challenges such as when IoT devices utilize encrypted communication. During the study, the default settings of both the ML models and flow analysis tools were utilized, allowing evaluation of performance in real-world settings without any manual tuning or adjustment. The following methodology was utilized to address the following five challenges:

- Creating a realistic testbed to emulate real-world usage of devices under their three states.
- Evaluating the best metadata and ML model combination for IoT device and state usage pattern identification.
- Analyzing the most important features contributing to the identification of IoT devices and states.
- Demonstrating the generalization and robustness of device and state identification by (i) evaluating them on unseen data from the same network, and (ii) on unseen data captured from different networks.
- Providing the datasets publicly available to the research community for the reproducibility and extensibility of the research efforts in this field [1]

To the best of authors' knowledge, this is one of the first times that the robustness and generalization of ML based models are explored for IoT device and state identification based on usage patterns without using deep packet inspection. The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 introduces the methodology. Section 4 details the evaluations and results. Finally, conclusions and the future work are discussed in Section 5.

II. Related Work

This section highlights existing solutions in the literature for device identification in IoT networks within the scope of

[1] The datasets can be access at: https://ieee-dataport.org/documents/nims-benign-dataset-2024-2

978-3-903176-66-9 ©2024 IFIP

this work. Meidan et al [8] applied ML models on network traffic data for accurate identification of IoT devices connected to a network. Using captured network traffic data from nine IoT devices, two PCs and two Smartphones, they trained and evaluated a two-stage classification system. Kolcun et al [9] compared the accuracy of previously proposed ML models for identifying IoT devices. They used packet traces collected over a period of six months from a large IoT test-bed. They used a combination of classifiers with 19 flow features. Mainuddin et al [10] developed a ML based IoT device identification scheme based on 70 features of TCP flows. Ren et al. [12] reported on a multidimensional analysis of information exposure from IoT devices between IoT labs in the US and UK. They performed both automated and manually controlled experiments, characterizing information exposure based on Internet traffic destinations, encryption of communication contents, inference of IoT-device interactions from such content, and identification of unexpected exposures of private or sensitive information. Loginov et al. [11] employed an ML based approach on four publicly available consumer IoT traffic traces to explore the nature and extent of potential data exposure as well. They proposed a feature set for use with the models and evaluated it against other feature sets. Zain et al. [7] utilized the NFStream flow analyzer to extract 85 features from PCAP files. They selected 20 features using the information gain method, and trained six classifiers on four publicly available datasets. Erefani et al. [14] proposed a framework to enrich IoT datasets in two directions: Vertical and Horizontal. The former direction merged different public IoT datasets while the latter proposed a new set of features to represent the behavior of IoT devices across different environments. Sivanathan et al. [15] developed a ML based classification approach and provided insights for operators to monitor and secure IoT assets in smart environments without the need for specialized tools or devices. In summary, existing literature has conducted research into methods to analyze network traffic in order to differentiate IoT traffic from non-IoT traffic and identify different IoT devices within the traffic. In contrast, in this paper, we extend our previous work [2] for detecting the state of an IoT device identified using machine learning based approach. Furthermore, we evaluate the robustness and generalization of the proposed identification model on new test datasets (not seen during training) from the same and different networks.

III. Methodology

This section introduces the proposed framework used in the study, the capture of IoT device and state traffic, extraction of metadata, utilization of data engineering techniques, and evaluation of ML models.

A. Proposed Framework

The Figure 1 illustrates an overview of the proposed framework. In the framework, the testbed we used for this study is built upon our previous work [2]. The nine IoT devices deployed in the testbed were connected via a Wi-Fi based IP network. They collectively constitute to five categories, and represent seven brands.

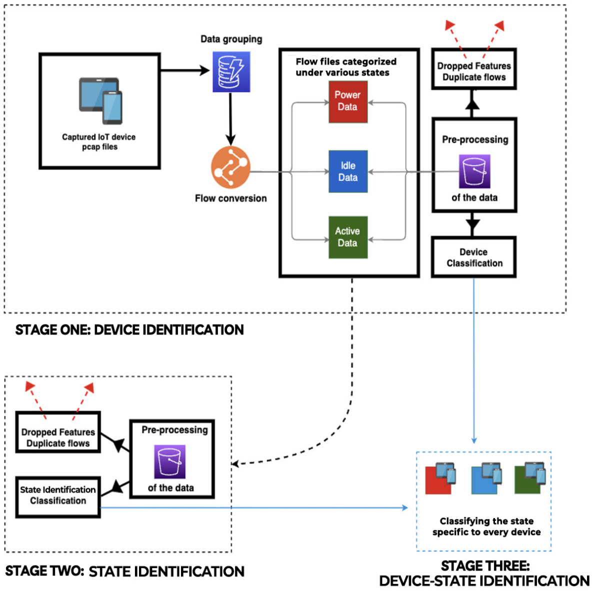
Data grouping
Captured IoT device pcap files
Flow conversion
Flow files categorized under various states
Power Data
Idle Data
Active Data
Dropped Features Duplicate flows
Pre-processing of the data
Device Classification
STAGE ONE: DEVICE IDENTIFICATION
Dropped Features Duplicate flows
Pre-processing of the data
State Identification Classification
STAGE TWO: STATE IDENTIFICATION
Classifying the state specific to every device
STAGE THREE: DEVICE-STATE IDENTIFICATION

Fig. 1: Proposed Framework

Significant effort was expended to generate traffic and capture it for the training and testing of ML models. The capture process utilized a Lenovo Legion desktop and a Lenovo ThinkPad Laptop, both running Ubuntu 20.04, as servers. Each system was equipped with two network interfaces, with one specifically designated for communicating with the IoT devices.

B. Capturing Device State

Traffic generation and capture was performed for all devices individually without any interference from other devices or networks. The captured traffic represented all three states of interest. The parameters used for defining the states are as follows:

- **Power-up:** Captured during the initial 50 seconds from the moment the device is powered on, allowing for the establishment of necessary connections. This process was repeated 10 times for each device.
- **Idle:** Captures span 50 minutes of idle time in a quiet environment. The captures were executed 10 times for each device. This resulted in a cumulative total of 500 minutes of idle data for each device.
- **Active:** Similar to the Idle state, this involved capturing 50 minutes of real human activities and communication with a device. These were repeated 10 times. Multiple continuous realistic activities were executed to generate traffic throughout the data capture. These activities include background noise, music, visible movement, intermittent commands for audio speakers such as setting alarms or asking for directions, bell notifications, communications with the devices as well as audio and visual streaming.

In short, traffic was captured for 50 seconds for Power-up and 50 minutes for Idle and Active. This capture was repeated 10 consecutive times for each device. The background traffic from non-IoT devices was also captured during this time. Default configurations were utilized for all the devices. Tables I, II, and III show the captured device statistics for all three states.

TABLE I: Power: Device Statistics - 10 iterations combined

Devices	**Packets**	**T2 Flows**	**NFStream**	**Zeek**
Amcrest	2059	657	355	345
Coffeemaker	518	82	31	31
Doorbell	3085	324	168	171
Echodot	10872	1283	667	675
Nestcam	55585	679	334	325
Nestmini	15787	995	536	526
Powerstrip	1321	238	110	115
Samsung TV	61573	1987	1028	1018
Smartplug	265	48	15	15
Total	151065	6293	3244	3221

TABLE II: Idle: Device Statistics - 10 iterations combined

Devices	**Packets**	**T2 Flows**	**NFStream**	**Zeek**
Amcrest	11981	2965	1487	1465
Coffeemaker	3018	78	39	28
Doorbell	1154	359	119	30
Echodot	97336	2677	1506	1463
Nestcam	41871	538	302	280
Nestmini	37505	2657	1468	4406
Powerstrip	1784	604	231	366
Samsung TV	429615	19973	16839	16753
Smartplug	4159	133	59	49
Total	628423	29984	22050	24840

TABLE III: Active: Device Statistics - 10 iterations combined

Devices	**Packets**	**T2 Flows**	**NFStream**	**Zeek**
Amcrest	3375368	3211	1694	1656
Coffeemaker	4680	78	39	29
Doorbell	3521834	914	446	389
Echodot	683569	4145	2222	2147
Nestcam	5175502	10018	5121	5098
Nestmini	615622	4796	2614	5457
Powerstrip	1723	587	229	367
Samsung TV	7757693	25987	20399	20506
Smartplug	4091	124	54	44
Total	21140082	49860	32818	35693

C. Extracting Metadata

In this work, metadata is extracted using three open-source flow analysis tools (NF-Stream, Tranalyzer2, and Zeek) were deployed with their default configuration. Tables IV and V provide a summary of the captured traffic with and without the background traffic.

Exporting Flows with NF-Stream: Following the results in [2], NFStream was installed using its default configuration, which resulted in the tool extracting 38 features for each 5-tuple flow. During data engineering, 18 potentially biased features were identified and excluded, resulting in 20 per-flow features utilized for model building, validation and testing. .

Exporting Flows with Tranalyzer2 (T2): T2 was installed using its default configuration of 10 plugins which resulted in the tool extracting 110 features for each 5-tuple flow. During data engineering, 36 potentially biased features were identified and excluded, resulting in 74 per-flow features utilized for model building, validation and testing.

Exporting Flows with Zeek: Zeek was deployed as the third system in the proposed framework. By default Zeek extracted 21 features for each 5-tuple flow. During data engineering, 8 potentially biased features were identified and excluded, resulting in 13 per-flow features utilized for model building, validation and testing.

TABLE IV: Packets and Flows count with background traffic

Devices	**Packets**	**T2**	**NFStream**	**Zeek**
Amcrest	4315562	47280	25114	25958
Coffeemaker	2968871	13134	7384	7591
Doorbell	2364955	44860	25129	27386
Echodot	1477270	10930	6367	6228
Nestcam	4198297	3907	2347	2829
Nestmini	1433127	12138	6982	9103
Powerstrip	910925	3990	2247	2294
Samsung TV	2249597	12960	8382	8255
Smartplug	2922809	23266	12945	13047

TABLE V: Packets and Flows count with no background traffic

Devices	**Packets**	**T2**	**NFStream**	**Zeek**
Amcrest	1348168	1734	938	905
Coffeemaker	1917	26	22	13
Doorbell	366841	154	124	116
Echodot	706099	2508	1310	1283
Nestcam	1868505	198	129	275
Nestmini	466827	1943	1208	2924
Powerstrip	363	34	22	19
Samsung TV	1559366	7002	5144	5055
Smartplug	2431	70	44	35

D. Data Engineering

There is a lack of clear standards, and guidelines for data preprocessing pipelines for IoT traffic in the literature [18]. This study seeks to contribute insights into useful elements to be included in such pipelines which are utilized for AI-based modeling endeavours. In this work, our goal is not Deep Packet Inspection, rather identifying usage patterns on the basis of 5-tuple flow metadata - primarily extracted in the form of statistics which represent flow traffic characteristics. We believe that this will not only enable our ML models to generalize well from one network to another, but also will potentially enable us to employ such an approach for the identification of IoT device and state even when the device utilizes encrypted traffic communication. As a result, MAC addresses, IP addresses, port numbers, and similar identifiers are removed from the metadata once the packets are exported to flows. We also dropped duplicate flows to eliminate redundancy.

IV. Experiments and Results

In this paper, evaluations are conducted using three flow analysis tools and eight ML models. The ML models employed are based on the following algorithms: Decision Tree (DT), K-Nearest Neighbor (KNN), Naive Bayes (NB), Support Vector Machine (SVM), Multi-layer Perceptron (MLP), Random Forest (RF), Logistic Regression (LR) and Gradient Boost (GB). All the tools and models are set up using default parameters. ML models are trained on a randomly selected subset of the data captured (70%) and tested on the remaining data which was not used during training (30%). Additional datasets are used for further testing purposes to evaluate the generalization and robustness of the best features and ML model combination. The weighted average of the following metrics are used to evaluate performance: Precision (1), Recall (2), and F1-Score (3).

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{3}$$

TP: Instances correctly classified as belonging to a specific class.
FN: Instances incorrectly classified as not belonging to a specific class.
FP: Instances incorrectly classified as belonging to a specific class.

A. Analyzing ML models on Flow Exported Metadata

Tables VI, VII, and VIII show the weighted average of the evaluations performed to identify IoT devices using eight ML models and the T2 traffic flow metadata as the feature set under Power-On, Idle and Active states respectively. Based on these results, the RF model achieves best performance for all three states across all devices. As an example, Figure 2 depicts the confusion matrix for the RF model with the T2 features for the IoT devices in active state on the test data. Additionally, evaluations were conducted using NFStream and Zeek traffic flow metadata, where RF again achieved the highest performance-results are presented in Table IX. Due to the maximum page constraints, we present only the best performing model results for NFStream and Zeek. Overall, these results demonstrate that it is possible to identify an IoT device and its state with a high Precision, Recall, and F1-score. Figure 3 presents a Venn diagram presenting the top contributing T2 features used by the RF model for identifying usage patterns of IoT devices and their corresponding states. The top features are selected based on Gini impurity. Table X presents the performance of the best metadata and ML model combination per IoT device and state. To this end, the RF model utilizing T2 based metadata achieves the highest performance for identifying the device

TABLE VI: Device Usage Pattern Identification: Under Power State using T2 Features

Classifier	Split	F1-Score	Precision	Recall	Accuracy
DT	Training	99.88%	99.88%	99.88%	99.88%
	Testing	96.64%	96.64%	96.65%	96.65%
KNN	Training	77.53%	78.37%	77.73%	77.73%
	Testing	68.63%	69.15%	69.15%	69.15%
NB	Training	6.57%	9.03%	9.68%	9.68%
	Testing	6.18%	8.33%	9.45%	9.45%
SVM	Training	21.25%	19.09%	33.46%	33.46%
	Testing	20.61%	18.22%	32.63%	32.63%
MLP	Training	34.86%	45.39%	36.47%	36.47%
	Testing	35.36%	44.97%	36.52%	36.52%
RF	Training	99.88%	99.88%	99.88%	99.88%
	Testing	98.16%	98.16%	98.16%	98.16%
LR	Training	19.20%	14.64%	27.90%	27.90%
	Testing	19.98%	15.23%	29.01%	29.01%
GB	Training	99.84%	99.84%	99.84%	99.84%
	Testing	97.36%	97.36%	97.35%	97.35%

TABLE VII: Device Usage Pattern Identification: Under Idle State using T2 Features

Classifier	Split	F1-Score	Precision	Recall	Accuracy
DT	Training	99.87%	99.88%	99.87%	99.87%
	Testing	98.53%	98.55%	98.52%	98.52%
KNN	Training	94.63%	94.68%	94.66%	94.66%
	Testing	92.62%	92.67%	92.68%	92.68%
NB	Training	5.91%	8.68%	12.84%	12.84%
	Testing	6.03%	8.59%	12.96%	12.96%
SVM	Training	59.64%	54.94%	70.23%	70.23%
	Testing	59.79%	54.92%	70.32%	70.32%
MLP	Training	67.82%	72.22%	73.87%	73.87%
	Testing	68.34%	74.75%	74.30%	74.30%
RF	Training	99.87%	99.87%	99.87%	99.87%
	Testing	98.98%	98.99%	98.98%	98.98%
LR	Training	53.79%	49.55%	59.87%	59.87%
	Testing	53.74%	49.60%	59.75%	59.75%
GB	Training	99.42%	99.43%	99.42%	99.42%
	Testing	98.84%	98.86%	98.84%	98.84%

TABLE VIII: Device Usage Pattern Identification: Under Active State using T2 Features

Classifier	Split	F1-Score	Precision	Recall	Accuracy
DT	Training	99.96%	99.97%	99.96%	99.96%
	Testing	98.86%	98.86%	98.86%	98.86%
KNN	Training	92.78%	93.23%	92.83%	92.83%
	Testing	91.26%	91.68%	91.32%	91.32%
NB	Training	7.76%	12.76%	11.26%	11.26%
	Testing	7.82%	13.13%	11.27%	11.27%
SVM	Training	44.07%	41.44%	56.07%	56.07%
	Testing	44.10%	41.76%	56.12%	56.12%
MLP	Training	66.40%	76.02%	65.36%	65.36%
	Testing	66.46%	76.03%	65.36%	65.36%
RF	Training	99.96%	99.96%	99.96%	99.96%
	Testing	99.43%	99.45%	99.43%	99.43%
LR	Training	41.49%	36.33%	48.57%	48.57%
	Testing	41.49%	36.28%	48.63%	48.63%
GB	Training	99.58%	99.59%	99.58%	99.58%
	Testing	99.10%	99.10%	99.10%	99.10%

and state usage patterns. These results show that RF with the 74 feature of T2 achieves an average F1-score of 93%.

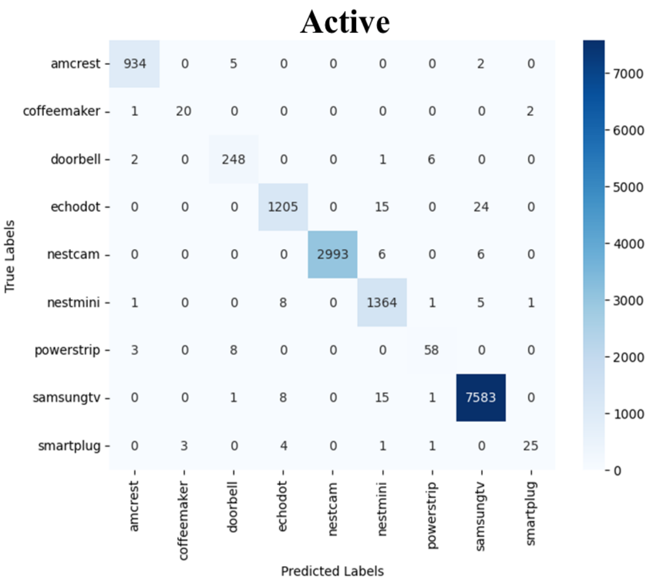

Fig. 2: Confusion Matrix of the Test Data for Device Usage Pattern Identification under the Active State with RF using T2 Metadata

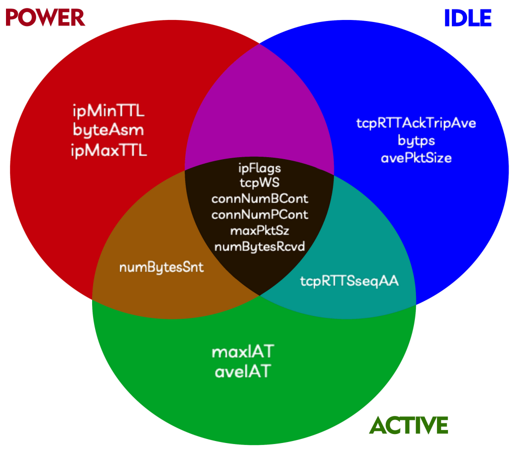

Fig. 3: Top T2 Features used for State Identification of Devices

B. Generalization and Robustness

While these results are very impressive, we sought to better understand how well the models will perform, in terms of generalization and robustness. Thus, we evaluated the best performing ML models under two additional new test conditions: (i) New Test Data from the same network, and (ii) New Test Data from a different Network [12]. In the following, we develop 3 different RF models, one each with T2 features, NFStream features and Zeek features.

TABLE IX: Device Identification: RF with NFStream and Zeek based Metadata Features under three States

RF-State	Split	F1-Score	Precision	Recall	Accuracy
NFStream					
RF-Power	Training	100.00%	100.00%	100.00%	100.00%
	Testing	100.00%	100.00%	100.00%	100.00%
RF-Idle	Training	99.66%	99.66%	99.66%	99.66%
	Testing	98.75%	98.76%	98.76%	98.76%
RF-Active	Training	100.00%	100.00%	100.00%	100.00%
	Testing	99.93%	99.94%	99.94%	99.94%
Zeek					
RF-Power	Training	97.76%	98.04%	97.91%	97.91%
	Testing	93.55%	93.90%	93.69%	93.69%
RF-Idle	Training	98.94%	98.98%	98.97%	98.97%
	Testing	97.88%	97.88%	97.91%	97.91%
RF-Active	Training	99.65	99.67%	99.65%	99.65%
	Testing	98.61%	98.66%	98.64%	98.64%

TABLE X: Device-State Results using T2 features

Label	F1-Score	Precision	Recall	Accuracy
amcrest_active	0.90	0.91	0.89	0.89
amcrest_idle	0.90	0.89	0.91	0.91
amcrest_power	0.99	0.99	0.99	0.99
coffeemaker_active	0.69	0.67	0.70	0.70
coffeemaker_idle	0.74	0.80	0.70	0.70
coffeemaker_power	0.85	0.94	0.77	0.77
doorbell_active	0.89	0.90	0.88	0.88
doorbell_idle	0.66	0.64	0.68	0.68
doorbell_power	0.81	0.77	0.86	0.86
echodot_active	0.90	0.89	0.90	0.90
echodot_idle	0.83	0.84	0.82	0.82
echodot_power	0.94	0.93	0.94	0.94
nestcam_active	1.00	1.00	1.00	1.00
nestcam_idle	0.88	0.88	0.89	0.89
nestcam_power	0.89	0.89	0.89	0.89
nestmini_active	0.90	0.90	0.91	0.91
nestmini_idle	0.81	0.80	0.82	0.82
nestmini_power	0.93	0.93	0.94	0.94
powerstrip_active	0.45	0.42	0.49	0.49
powerstrip_idle	0.41	0.43	0.39	0.39
powerstrip_power	0.97	0.96	0.97	0.97
samsungtv_active	0.95	0.95	0.95	0.95
samsungtv_idle	0.94	0.94	0.93	0.93
samsungtv_power	0.94	0.93	0.94	0.94
smartplug_active	0.34	0.42	0.29	0.29
smartplug_idle	0.27	0.30	0.24	0.24
smartplug_power	0.90	0.90	0.90	0.90
Macro avg	0.81	0.81	0.81	0.81
Weighted avg	0.93	0.93	0.93	0.93

1) New Test Data - Same Network: In this case, we evaluated the model on a new test traffic dataset captured in our testbed (same network). However, this new test data is captured at a different time period, collected under a different setup, and configuration of the testbed. This time around, a Raspberry Pi is added to the testbed in addition to the nine IoT devices used. Then, the devices are assigned static IP addresses, and traffic is generated for continuous five (5) hours. Statistics of the data can be found in Tables IV and V. The trained RF model used in the evaluations above is then tested on this new test data for identification of IoT devices and their states. Three RF models are developed - one each for T2, NF-Stream and Zeek features. Table XI shows the results on this new test data from the same network (testbed) which includes a different time period and under a different setup as discussed above. While performances of models using NFStream and Zeek based features decreases, the model with T2 features generalize well for the Idle and Active states. Figure 4 shows the confusion matrix of the model for the IoT devices in Active state. This demonstrates the high performance of the model on this new test data while also showing where the model could be improved further in terms of Echodot and Nestmini with Samsungtv. We do note, however, for the Power-on state, performance of all three RF models for different flow tool features all perform poorly.

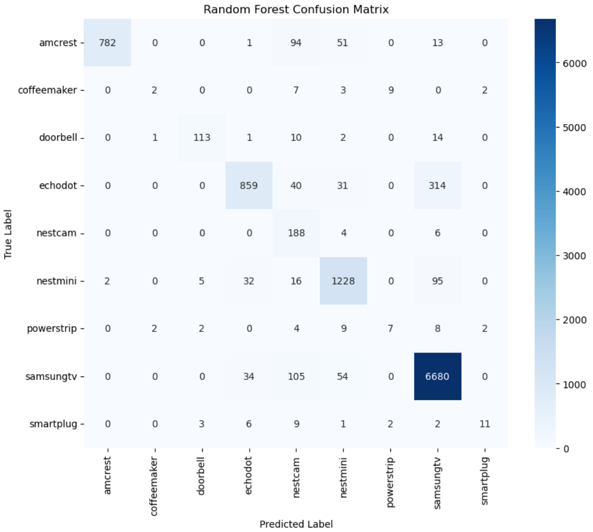

Fig. 4: Confusion Matrix of the New Test Data - Same Network: RF with T2 in Active State

TABLE XI: Performance of trained RF model on New Test Data - Same Network

RF-State	F1-Score	Precision	Recall	Accuracy
Tranalyzer				
RF-Power	36.44%	63.47%	35.27%	35.27%
RF-Idle	87.93%	88.97%	88.65%	88.65%
RF-Active	91.13%	92.20%	91.10%	91.10%
NFStream				
RF-Power	2.05%	1.13%	10.64%	10.64%
RF-Idle	0.67%	0.35%	5.94%	5.94%
RF-Active	0.06%	0.03%	1.70%	1.70%
Zeek				
RF-Power	30.76%	36.72%	35.30%	35.30%
RF-Idle	52.05%	58.91%	51.93%	51.93%
RF-Active	14.51%	19.84%	12.87%	12.87%

2) New Test Data - Different Network: To further test the robustness of our models, we conducted an additional evaluation using a public dataset [12]. In this case, the public dataset was collected on a different network with different configuration, at a different time. This new test data included six devices that were similar to the devices employed on our testbed network, namely Amcrest Camera, Echodot, Nestmini, Ring Doorbell, Samsung TV, and Coffeemaker. Despite their similar characteristics and functions, the devices exhibited notable differences in their emitted data. For instance, in the public dataset, the Samsung TV and Amcrest Camera utilized Ethernet connections, whereas on our testbed they used wireless connections. Additionally, the Nestmini and Coffeemaker were of different or previous generations compared to the ones used on our network. In the robustness evaluations, our goal is to explore whether these differences would affect the performance of the previously trained ML models with T2 features on this new test data from a different network, and to what extent. Table XII shows that among the ML models with T2 based metadata features trained previously on our testbed network data, RF with T2 is again the best performing model on this new test data from a different network and time period. However, the performance of RF with T2 model drops from 91% F1-Score test performance on 'new test data from the same network' to 65% F1-Score test performance on 'new test data from a different network' for identifying devices in active state. This seems to indicate that differences in network connections, and IoT device editions do affect the robustness and generalization of the RF model. We believe that these differences result in different IoT device usage patterns in the metadata yielding into almost a 25% drop in the performance of the trained model in identifying IoT devices and states on new test data from a different network. Further research will explore how to augment the training data to enhance the best identified models.

TABLE XII: Performances of trained ML Models with T2 on New Test Data - Different Network

Classifier	F1-Score	Precision	Recall	Accuracy
DT	47.74%	47.74%	40.36%	40.36%
KNN	19.08%	71.26%	12.73%	12.73%
NB	2.74%	1.55%	12.08%	12.08%
SVM	64.16%	58.14%	71.58%	71.58%
MLP	41.18%	58.26%	32.53%	32.53%
RF	64.65%	69.56%	61.16%	61.16%
LR	47.26%	52.29%	43.39%	43.39%
GB	44.10%	70.92%	33.50%	33.50%

V. Conclusion and Future Work

The Internet of Things (IoT) stands at the forefront of the hyper-connected world of networks, yet its growth outpaces the security and defence mechanisms, which precipitates the need for more comprehensive research across the field of IoT networks. With IoT devices expanding in quantity and type rapidly across homes, hospitals, offices, and various industries, understanding the cybersecurity implications is very important. Thus, the objective of this research was to evaluate the robustness, generalization and efficacy of ML models for identifying different IoT devices and their states based on their network traffic usage patterns. This was explored without employing deep packet inspection, enabling the approach to be utilized for the identification IoT devices and states even when they utilize encrypted communication. To achieve this, we designed, developed and evaluated: (i) A testbed network to emulate real-world usage of IoT devices under three states, namely Power-up, Idle and Active; (ii) Explored the best flow features and ML model combination for IoT device and state usage pattern identification; (iii) Analyzed the most important features contributing to the identification of IoT devices and their states; (iv) Analyzed the generalization and robustness of device and state identification by evaluating the best model on unseen data from the same network, and on unseen data from a different network; and (v) Made the datasets generated publicly available to the research community. To this end, a testbed network was deployed with nine IoT devices and traffic was captured during the three states recording over 1000 minutes per device. The evaluations included three flow analysis tools with eight ML models trained and used to identify the IoT devices and states. The results show that it is possible to identify consumer IoT devices and states using only traffic flow metadata without reliance on deep packet inspection, source/destination MAC addresses, IP addresses or Port numbers. The RF model with T2 based metadata features achieved the best performance in terms of F1-Score across all test datasets. In particular, this trained model achieved a 91% F1-Score for IoT device identification when in the Active state- in scenarios where the new test data from the same network as the one it was trained on. This shows that the model generalized well for the same network even though the new test data was captured at a different time period and network setup. However, the same model achieved an F1-Score of 65% when tested on new data from a totally different network (time period and set up). This seems to indicate that the model's robustness decreases as the network connections and IoT devices editions change. For future work, it is crucial to continue to test the strength and feasibility of the models on data captured or generated in different environments. To this end, we aim to explore how to augment the training data to enhance the generalization and robustness capabilities of the model. Furthermore, we aim to study the complexity and explanability of the ML models to improve the generalization ability. Another avenue that will be explored is the drifts and shifts in the usage patterns over time. We believe that these will equip the security experts with a comprehensive view of device and state identification under various configurations, environments and attack scenarios.

Acknowledgement

This research is supported by the Mitacs and Solana Networks funding programs. The research is conducted as part of the Dalhousie NIMS Lab: https://projects.cs.dal.ca/projectx/.

References

[1] Average Order Value, “Number Of Internet Of Things (IoT) Connected Devices Worldwide,” https://aovup.com/stats/iot/.

[2] J. Adjei, N. Z. Heywood, B. Nandy and N. Seddigh, ”Identifying IoT Devices: A Machine Learning Analysis Using Traffic Flow Metadata,” NOMS 2024-2024 IEEE Network Operations and Management Symposium, Seoul, Korea, Republic of, 2024, pp. 1-7, doi: 10.1109/NOMS59830.2024.10575442.

[3] “Tranalyzer,” 2022, https://tranalyzer.com/, Accessed: 2023-04-20.

[4] “NFStream: Flexible Network Data Analysis Framework,” 2022, https://www.nfstream.org/, Accessed: 2023-04-20.

[5] ”Zeek: An Open Source Network Security Monitoring Tool,” 2024,https://zeek.org//, Accessed: 2024-04-20.

[6] F. Haddadi and N. Zincir-Heywood, ”Benchmarking the Effect of Flow Exporters and Protocol Filters on Botnet Traffic Classification,” in IEEE Systems Journal, vol. 10, no. 4, pp. 1390-1401, Dec. 2016, doi: 10.1109/JSYST.2014.2364743.

[7] A. Zain, F. Hussain, S. Ghazanfar, M. Husnain, S. Zahid and G. A. Shah, ”A Generic Machine Learning Approach for IoT Device Identification,” 2021 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan, 2021, pp. 118-123, doi: 10.1109/ICCWS53234.2021.9702983.

[8] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. 2017. ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis. In Proceedings of the Symposium on Applied Computing (SAC ’17). Association for Computing Machinery, New York, NY, USA, 506–509. https://doi.org/10.1145/3019612.3019878

[9] Kolcun, R., Popescu, D.A., Safronov, V., Yadav, P., Mandalari, A.M., Mortier, R., Haddadi, H. (2021). Revisiting IoT Device Identification. ArXiv, abs/2107.07818.

[10] M. Mainuddin, Z. Duan, Y. Dong, S. Salman and T. Taami, ”IoT Device Identification Based on Network Traffic Characteristics,” GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 6067-6072, doi: 10.1109/GLOBECOM48099.2022.10001639.

[11] A. Loginov, J. Adjei, N. Zincir-Heywood, S. Sampalli, K. de Snayer and T. Dougall, ”Preliminary Results on Exploring Data Exhaust of Consumer Internet of Things Devices,” 2023 19th International Conference on Network and Service Management (CNSM), Niagara Falls, ON, Canada, 2023, pp. 1-5, doi: 10.23919/CNSM59352.2023.10327914.

[12] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi. 2019. ”Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach”, In the 2019 Internet Measurement Conference (IMC ’19). Association for Computing Machinery, New York, NY, USA, 267–279. https://doi.org/10.1145/3355369.3355577.

[13] D. K. Andrews, R. K. Agrawal, S. J. Matthews and A. S. Mentis, ”Comparing Machine Learning Techniques for Zeek Log Analysis,” 2019 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, USA, 2019, pp. 1-4, doi: 10.1109/URTC49097.2019.9660501.

[14] M. Erfani, F. Shoeleh, S. Dadkhah, B. Kaur, P. Xiong, S. Iqbal, S. Ray, and A. A. Ghorbani, ”A feature exploration approach for IoT attack type classification,” 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, AB, Canada, 2021, pp. 582-588, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00101.

[15] A. Sivanathan, H.H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, ”Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics,” in IEEE Transactions on Mobile Computing, vol. 18, no. 8, pp. 1745-1759, 1 Aug. 2019, doi: 10.1109/TMC.2018.2866249.

[16] A. S. Khatouni and N. Zincir-Heywood, ”Integrating Machine Learning with Off-the-Shelf Traffic Flow Features for HTTP/HTTPS Traffic Classification,” 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1-7, doi: 10.1109/ISCC47284.2019.8969578.

[17] A. S. Khatouni, N. Seddigh, B. Nandy, and N. Zincir-Heywood, ”Machine Learning Based Classification Accuracy of Encrypted Service Channels: Analysis of Various Factors,” Journal of Network and Systems Management, vol. 29, no. 8, pp. 1-7, 2021, doi: 10.1007/s10922-020-09566-5.

[18] A. Tawakuli and T. Engel, ”Towards Normalizing The Design Phase of Data Preprocessing Pipelines For IoT Data,” 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 4589-4594, doi: 10.1109/BigData55660.2022.10020312.

[19] M. Mainuddin, Z. Duan, Y. Dong, S. Salman and T. Taami, ”IoT Device Identification Based on Network Traffic Characteristics,” GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 6067-6072, doi: 10.1109/GLOBECOM48099.2022.10001639. keywords: Performance evaluation;Network servers;Telecommunication traffic;Smart homes;Machine learning;Feature extraction;Internet of Things;Internet of Things;IoT Devices;IoT Device Identification;IoT Security.