IoTSpot: Identifying the IoT Devices Using their Anonymous Network Traffic Data

Liangdong Deng, Yuzhou Feng, Dong Chen and Naphtali Rishe
School of Computing and Information Sciences
Florida International University

***Abstract*—The Internet of Things (IoT) has been erupting the world widely over the decade. Smart home owners and smart building managers are increasingly deploying IoT devices to monitor and control their environments due to the rapid decline in the price of IoT devices. The network traffic data produced by these IoT devices are collected by Internet Service Providers (ISPs) and telecom providers, and often shared with third-parties to maintain and promote user services. Such network traffic data is considered "anonymous" if it is not associated with identifying device information, e.g., MAC address and DHCP negotiation. Extensive prior work has shown that IoT devices are vulnerable to multiple cyber attacks. However, people do not believe that these attacks can be launched successfully without the knowledge of what IoT devices are deployed in their houses. Our key insight is that the network traffic data is not anonymous: IoT devices have unique network traffic patterns, and they embedded detailed device information. To explore the severity and extent of this privacy threat, we design IoTSpot to identify the IoT devices using their "anonymous" network traffic data. We evaluate IoTSpot on publicly-available network traffic data from 3 homes. We find that IoTSpot is able to identify 19 IoT devices with F1 accuracy of 0.984. More importantly, our approach only requires very limited data for training, as few as 40 minutes. IoTSpot paves the way for operators of smart homes and smart buildings to monitor the functionality, security and privacy threat without requiring any additional devices.**

I. Introduction

The Internet of Things (IoT) has been erupting the world widely over the decade. The rise of the IoT holds great promise to transform people's lives by making society more efficient in many areas, including smart home and building, smart grid, transportation, healthcare, manufacturing, etc. In each area, new IoT-enabled devices are rapidly being developed, while, in parallel, existing devices are also being augmented with IoT functionality, i.e., Internet connectivity, remote programmability, and automation. The total installed base of the IoT connected devices is projected to amount to 75.44 billion worldwide by 2025, a fivefold increase in ten years [1].

The network traffic data produced by these IoT devices are collected by Internet Service Providers (ISPs) and telecom providers, and shared with third-parties with the goal of maintaining and promoting the user services. "Any service that provides Internet access (ISPs, WiFi hotspots, and telecom providers) can obviously see what resources users are accessing. And even with encryption, traffic patterns provide some information about activity." [2]. ISPs like AT&T, Comcast, Time Warner, Sprint and Verizon are selling personal network traffic data like web browsing history without prior user consent before the Federal Communications Commission (FCC)'s broadband privacy protections due to come into force later this year [3]. They build substantial datasets on the online activities of certain demographics and can tell companies who want to market to. "It is very similar to Facebook, except the ISPs have a better understanding of who's using what device in the home, " noted Matt Hogan, CEO at DataCoup [3]. Before sharing and selling the network traffic data, it is recommended that ISPs and third-parties should replace or remove the MAC address and DHCP negotiation information for the IoT devices from their network traffic data to avoid any potential security and privacy issues. Such network traffic data is generally considered "anonymous" since it is not associated with identifying device information. In parallel, extensive prior research work [4], [5], [6], [7], [8] has shown that IoT devices are vulnerable to multiple cyber attacks and have significant privacy concerns. For instance, recently, researchers have found that Belkin Wemo Zero-Day vulnerability [9] and Samsung SmartThings hub vulnerabilities [10] could leave the door open for the IoT-based attack. However, people do not believe that these device-specific attacks can be launched successfully since the identifying information for their IoT devices has been removed or slipped from the shared dataset.

Our key insight is that the anonymized network traffic data is not anonymous: most of the IoT devices have unique network traffic signatures, and they embedded detailed device information. And based on this key insight, we can identify what devices are deployed in a home using its "anonymized" network traffic data. To explore the severity and extent of this privacy threat, we design IoTSpot to identify the IoT devices using their "anonymous" network traffic data. In doing so, this paper makes the following contributions.

Identification Challenges. We highlight the challenges to identify IoT devices using their "anonymized" network traffic data. We discuss the fundamental physical relationships that govern the network traffic data over time for the widely deployed IoT devices in detail.

IoTSpot Design. We present the design of IoTSpot, which can identify IoT devices for a smart home using its "anonymous" network traffic data. We leverage the Principal Component Analysis (PCA) algorithm to select the major features that affect device detection and Random Forest classifier to learn the device-specific network traffic signature.

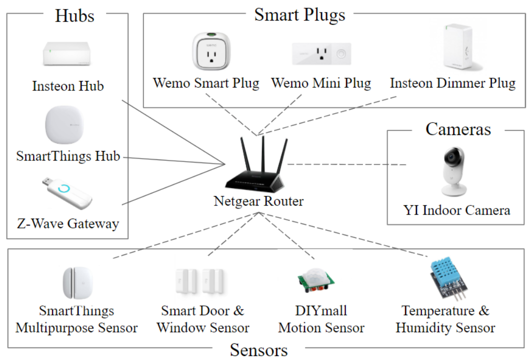


Fig. 1. The architecture of our data collection environment that is comprised of 11 different IoT devices.

Implementation and Evaluation. We experiment our IoTSpot in 3 real homes. The evaluation shows that our approach has the F1 score — a standard measure of a binary classifier's overall performance—of 0.984, which effectively identifies IoT devices. Interestingly, IoTSpot achieves similar accuracy without access to any network traffic data from testing homes as fully supervised approach with complete access to such training data.

II. Background

There is a significant amount of prior work on classifying general Internet traffic data. However, the research area that is aimed at characterizing IoT device network traffic is very new and still developing. We summarize them as follows,

Application based Classification. For example, recent work [11], [12] focus on network traffic pattern detection using application data, e.g., web browsing history, mails, etc. These work are not appliable to IoT device identification directly since they require specific application-driven data for training.

Fully Supervised Classification. These approaches typically require access to the network traffic data from testing sites to train a reasonable accurate machine learning models. [13], [14] use Bayesian analytics to classify Internet traffic and it requires manually labeled data to build a classification model. However, these labels are not always available, for example, new homes become online or new IoT devices are added.

Deep Learning based Classification. For instance, recent work [15], [16] employ Conventional Neural Networks (CNN) and Recurrent Neural Network to predict the unlabeled devices. The drawback of these approaches is that they require a significant amount of groundtruth traffic data for training.

Instead, we present a new Machine Learning (ML) – based unsupervised approach — IoTSpot that can accurately identify IoT devices using only very limited network traffic data, as few as $\sim$40 minutes traffic data. More importantly, our approach does not require access to the groundtruth device data from testing sites to train our ML models.

III. IoT Devices Network Traffic Analysis

In this section, we explain how we set up our "mock" smart homes to collect network traffic data and synthesize traffic from various IoT devices. We also identify the major features that can be potentially used to classify IoT devices.

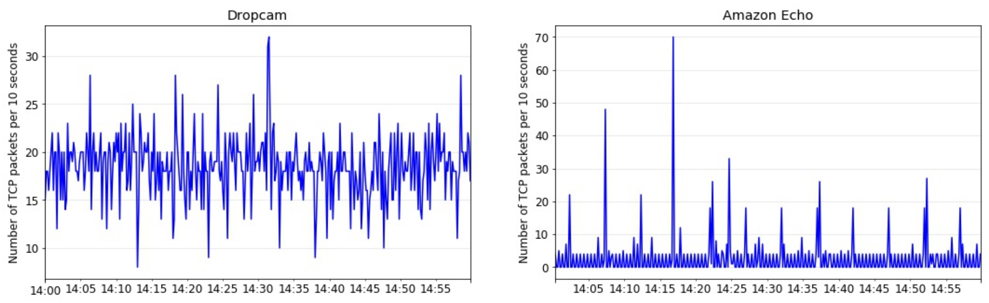
A. Smart Home Setup and Data Collection

We set up our IoT-based smart environment at two different locations. Home #1 is a private townhouse apartment that has 2 occupants and is located in Miami, Florida. Home #2 is our lab space that has 4 students staying in the room daily, which is on the FIU main campus. To collect network traffic data from all the IoT devices in our "mock" homes, we first deploy a NETGEAR AC1750 smart Wi-Fi router at each location and it serves as the internal switch and the gateway to the public Internet. We flash the router using DD-WRT — a Linux based alternative opensource firmware. We then install $tcpdump$ on this gateway to capture the network traffic data. We write $bash$ scripts to store the traffic data into $pcap$ files on an external 128GB USB hard drive that is attached to the gateway. We use $cronjob$ to automatically upload the locally stored data to our remote server in the backend. On the remote server, we parse the $pcap$ file to our system and group the packets basing on each device's IP address. We sort the traffic data using its timestamp for each device. We then remove the data of non-IoT devices and other irrelevant data from the dataset. Our smart home environment has a total of 19 IoT devices.

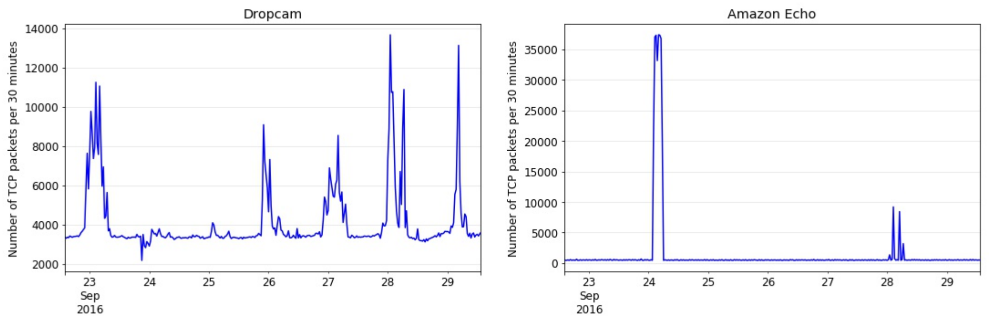
B. Traffic Data Analysis: Basics

Typically, the traffic data collected from a smart home in 24 hours can be as large as several gigabytes and contains millions of TCP/IP packets. To understand the "big" traffic data, we first visualize the amount of packets sent and received by each device to fingerprint its network traffic. Figure 2 shows the total number of TCP packets sent and received by IoT devices. Due to the space limitation, we only show 2 devices from different categories: Dropcam (camera) and Amazon Echo (hub). As we can see from Figure 2 (a) and (b), at fine granularity, such as ten seconds level, traffic data from an IoT device shows its strong pattern. Dropcam is streaming video to its cloud server, and the size of video streaming varies since the data compression ratio changes based on the content of the image captured by the camera. And Dropcam's IP packet counting follows certain distributions. While, Amazon Echo has a consistent data flow, but shows data burst from time to time. Amazon Echo is sending voice data to cloud servers for command parsing purposes, and the size of voice data varies depending on the voice and noise from the environment, which explains the data bursts and why they have different sizes. In contrast, at a coarse granularity, the traffic data may not reveal any observable pattern, as shown in Figure 2 (c) and (d). These graphs show the number of incoming and outgoing TCP packets in 30-minute intervals over 7 days. At this granularity, the pattern of traffic data is more related to human interactions. For example, someone probably connects to Dropcam several times to check the real-time video footage, which causes the peaks in Figure 2(c).

Observation: *To model an IoT device's traffic pattern, "raw "network traffic data is not dependable since at different granularities we may observe different patterns or not be able to learn their activity pattern.*



(a) Number of packets transmitted per 10 seconds



(b) Number of packets transmitted per 30 minutes

Fig. 2. Traffic Data Analysis: Basics

C. Traffic Data Analysis: Flows

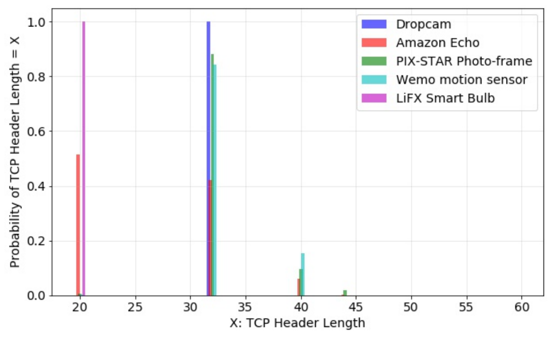
Since traffic packets are generated by IoT devices that are often interacted with humans in smart homes, the size of those traffic data packets typically varies for different devices, and thus it is challenging to apply data analytical approaches on these "raw" data directly. Besides, only very limited information can be learned directly from the "raw" network traffic data packets. The most useful information is the IP, TCP, and UDP related header information. We leverage the inner network IPs or the device IDs to isolate each device's traffic trace data. *Note that, unlike other existing approaches, our approach does not require the MAC address from each IoT device as inputs*.

We first group packets into TCP sessions (basing on the source and destination IP/port information for a packet), then break long TCP sessions into short packet streams (we define them as "flow"). We define the "flow" as a piece of TCP traffic data that is comprised of one or many data packets within a sliding window size ΔT, depending on how frequently a local device communicates with its remote server, for instance, cloud servers. To determine ΔT, we learn the relationship between our approach's accuracy and the size of ΔT using experiments that we will discuss in the evaluation section later. We find the optimal ΔT is 1 minute for regular IoT devices using empirical analytics. All the features are extracted at flow level. One flow is corresponding to one data record in our training dataset. For instance, assume we have the packets set $\{p_1$, p_2, p_3, ... $p_n\}$ in the raw traffic data, after transferring to the flow set $\{f_1$, f_2, f_3, ... $f_n\}$. Our approach guarantees that for any flow f_x, it contains the packet data: p_i, p_{i+1}, p_{i+2}, ... p_j, where 1 $\leq$ i $\leq$ j $\leq$ n and $(p_j - p_i) \cdot T_i \leq \Delta T$.

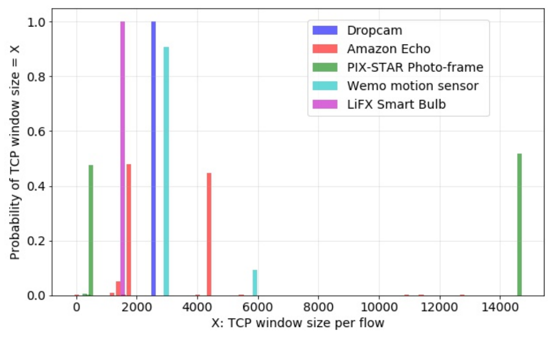
D. Traffic Data Analysis: features

After data preprocessing, now we have three layers of data: raw packets, TCP sessions, and flows. From each layer, we can extract multiple features for analysis. In Figure 3, we show four features—TCP header length, TCP window size per flow, outbound byte ratio, and outbound packet ratio. Dropcam and Smart Bulb have a fixed TCP header length of 32 and 20, respectively. The other three devices have several different lengths due to their different jobs running on these devices. Interestingly, as shown in Figure 3 (b), TCP initial window size is the first out-going packet's window size for a certain TCP session. The window size may be adjusted throughout the communication depending on multiple factors including network latency, data processing speed of the local and remote device, etc. And the initial window size of a TCP session is nearly a constant value for a certain device. This indicates that we may use this strong feature to identify some IoT devices.

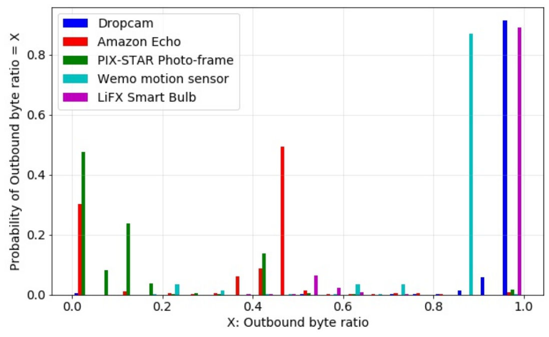
Figure 3 (c) shows the probability distribution of X/(X+Y), where X is the number of outbound packets in a flow, and Y is the number of inbound packets in the same flow. Figure 3 (d) shows similar results as in Figure 3 (c), the difference is that it shows the number of packets instead of the number of bytes. As we can see from these two figures, some devices show very strong patterns, while, some other devices show weak patterns. For example, Dropcam's outbound bytes ratio is close to 1.0 but the packet ratio is close to 0.5. This is because the camera traffic is more focusing on uploading than downloading, but it sends and receives the same amount of packets. We observe the same behaviors on LiFX Smart Bulb and Wemo Motion Sensor, while some devices have more complicated patterns that are not easy to explain or model.



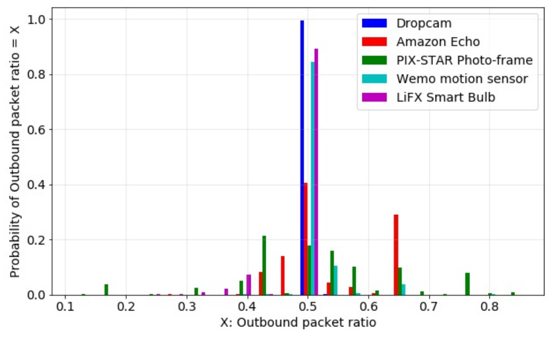
(a) TCP Header Length



(b) TCP window size per flow



(c) Outbound byte ratio



(d) Outbound packet ratio

Fig. 3. Traffic Data Analysis: features

Observation: *Different IoT devices may have different traffic patterns. Some devices have strong patterns, while, some do not. For those devices that have strong patterns, we may be able to model them physically, while others indicate a more comprehensive and sophisticated modeling such as machine learning models is necessary.*

IV. IoTSpot Design

Given only anonymous network traffic data, our system—IoTSpot can identify its source IoT device. IoTSpot's system design includes three key steps, which we detail below.

A. Selecting features

For each TCP flow, we extract all features that are available in TCP header per TCP flow, including—initOutH, initOutW, initOutD, initInH, outPackets, outEmptyPackets, outBytes, inPackets, outBytePerPacket, outEmptyPacketRatio, inBytePerPacket, duration, packetRatio, and byteRatio. To analyze the effects of these features, we leverage the output of their coefficient matrix to learn their relationships between each other. Table 1 shows correlation coefficients for each traffic metric using the Pearson product-moment correlation coefficient,

$$\rho_{x_1,x_2} = \frac{cov(x_1, x_2)}{\sigma_{x_1} \cdot \sigma_{x_2}} \tag{1}$$

. Here, $cov(x_1, x_2)$ is the covariance of the two metrics — x_1 and x_2, and σ_{x_1} and σ_{x_2} are the standard deviations. The higher the absolute value of the correlation coefficient, the stronger the correlation between the two traffic metrics—a positive correlation indicates an increasing linear relationship, while a negative correlation indicates a decreasing linear relationship. As shown in Table 1, we find that many features have strong correlations with other features. *The complex relationships between traffic metrics and device identification shown in this table motivate our study of automated prediction models using machine learning techniques in the next section.*

B. Identifying features

As discussed in the prior section, we can extract 19 traffic measurement metrics from TCP flows. However, the relationships among these metrics are complex, and many metrics show a high correlation with each other. Before building ML models to learn a device's traffic signature, we first use Principal Component Analysis (PCA) algorithm to identify the most important features. We leverage PCA to convert 19 dimensional data to low dimensional data by selecting the most important feathers that capture the maximum information in network traffic data. The advantages of this data processing are two-folds. First, we significantly reduce the training time of the ML algorithms using less number of features. Second, we mitigate the overfitting problem caused by the input of the duplicated or highly correlated features. We study PCA accuracy with different amount of features. Eventually, we find that 11 out of 19 features are important features.

C. Building a characteristic model

Our insight enables us to build a general network traffic signature model for an IoT device using it's network traffic data from one (or many) deployments where it is available, and use that model to accurately identify the same device at new locations. We train an ML classifier using Random

	initOutH	initOutW	initOutD	initInH	initInW	initInD	outPackets	outEmpty Packets	outBytes	inPackets	inEmpty Packets	inBytes	outByte PerPacket	outEmpty PacketRatio	inByte PerPacket	inEmpty PacketRatioA	duration	packetRatio	byteRatio
initOutH	1.00	0.21	-0.12	0.95	0.16	-0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.12	0.44	0.24	0.26	0.16	0.29	-0.36
initOutW	0.21	1.00	0.35	0.21	0.44	0.42	-0.01	0.00	0.00	-0.02	-0.01	-0.01	0.08	0.23	0.16	-0.12	-0.04	0.19	-0.09
initOutD	-0.12	0.35	1.00	-0.10	0.04	0.83	0.01	0.00	0.03	0.01	0.01	-0.01	0.29	-0.14	0.22	-0.10	-0.07	-0.05	0.19
initInH	0.95	0.21	-0.10	1.00	0.14	-0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.10	0.38	0.19	0.29	0.15	0.27	-0.31
initInW	0.16	0.44	0.04	0.14	1.00	0.13	-0.02	0.00	-0.01	-0.03	-0.02	-0.01	-0.03	0.25	-0.04	0.32	0.17	0.12	-0.23
initInD	-0.01	0.42	0.83	-0.01	0.13	1.00	0.00	0.01	0.00	0.01	-0.01	0.15	0.21	0.13	0.45	-0.20	-0.03	0.03	-0.08
outPackets	0.01	-0.01	0.01	0.01	-0.02	0.00	1.00	0.87	0.35	0.99	0.96	0.11	0.04	-0.03	0.00	0.02	0.02	-0.02	0.03
outEmptyPackets	0.01	0.00	0.00	0.01	0.00	0.01	0.87	1.00	0.01	0.90	0.90	0.10	0.00	0.01	0.02	0.00	0.02	0.00	0.00
outBytes	0.01	0.00	0.03	0.01	-0.01	0.00	0.35	0.01	1.00	0.24	0.24	0.01	0.11	-0.02	0.00	0.02	0.00	0.00	0.02
inPackets	0.01	-0.02	0.01	0.01	-0.03	0.01	0.99	0.90	0.24	1.00	0.98	0.18	0.03	-0.03	0.02	0.02	0.02	-0.03	0.03
inEmptyPackets	0.01	-0.01	0.01	0.01	-0.02	-0.01	0.96	0.90	0.24	0.98	1.00	0.02	0.03	-0.03	-0.01	0.03	0.01	-0.02	0.04
inBytes	0.01	-0.01	-0.01	0.01	-0.01	0.15	0.11	0.10	0.01	0.18	0.02	1.00	0.00	0.03	0.20	-0.03	0.00	-0.05	-0.04
outBytePerPacket	0.12	0.08	0.29	0.10	-0.03	0.21	0.04	0.00	0.11	0.03	0.03	0.00	1.00	-0.26	0.13	0.09	-0.05	-0.06	0.31
outEmptyPacketRatio	0.44	0.23	-0.14	0.38	0.25	0.13	-0.03	0.01	-0.02	-0.03	-0.03	0.03	-0.26	1.00	0.31	-0.31	0.18	0.48	-0.93
inBytePerPacket	0.24	0.16	0.22	0.19	-0.04	0.45	0.00	0.02	0.00	0.02	-0.01	0.20	0.13	0.31	1.00	-0.37	-0.05	0.05	-0.32
inEmptyPacketRatio	0.26	-0.12	-0.10	0.29	0.32	-0.20	0.02	0.00	0.02	0.02	0.03	-0.03	0.09	-0.31	-0.37	1.00	0.11	-0.45	0.25
duration	0.16	-0.04	-0.07	0.15	0.17	-0.03	0.02	0.02	0.00	0.02	0.01	0.00	-0.05	0.18	-0.05	0.11	1.00	0.15	-0.18
packetRatio	0.29	0.19	-0.05	0.27	0.12	0.03	-0.02	0.00	0.00	-0.03	-0.02	-0.05	-0.06	0.48	0.05	-0.45	0.15	1.00	-0.31
byteRatio	-0.36	-0.09	0.19	-0.31	-0.23	-0.08	0.03	0.00	0.02	0.03	0.04	-0.04	0.31	-0.93	-0.32	0.25	-0.18	-0.31	1.00

TABLE I

CORRELATION MATRIX SHOWING CORRELATION BETWEEN DIFFERENT IoT NETWORK TRAFFIC METRICS.

Forest to build a customized model for each IoT device using its network traffic features. Prior work has evaluated a wide range of ML models that can be potentially applied to identify IoT devices using their network traffic data, including Support Vector Machine (SVM), Random Forest, Least Squares Regression, and deep neural networks. *While we evaluate different modeling techniques in the evaluation section, our approach is orthogonal to the specific ML model.*

V. IMPLEMENTATION

We implement IoTSpot using *python*. We use the scikit-learn machine learning library in python to build our machine learning IoT device identification models. The library supports multiple techniques including Principal Component Analysis (PCA), Support Vector Machine (SVM), and Random Forest and multiple linear regression models. We also use NumPy and Pandas for the traffic data pre-processing.

VI. EXPERIMENTAL EVALUATION

We first evaluate IoTSpot's detecting accuracy across 3 smart homes using 1 month of second-level network traffic data. To quantify our IoTSpot's identification accuracy, we compute the F_1 score — a measure of a classifier's accuracy.

$$F_1 = 2TP/(2TP + FP + FN) \tag{2}$$

Here, TP is the true positive, FP is the false positive and FN is the false negative. The F_1 score is the harmonic average of the precision and recall where it reaches its best identification at 1 and worst at 0.

A. Dataset

We use network traffic data from 3 homes, including 2 homes that we instrument in Florida and 1 house in Australia where network traffic data is made publicly available online. We first anonymize the network traffic data by removing their device MAC addresses and DHCP negotiation data. We then use our proposed data processing approach in prior section to parse all the traffic data into small "flows".

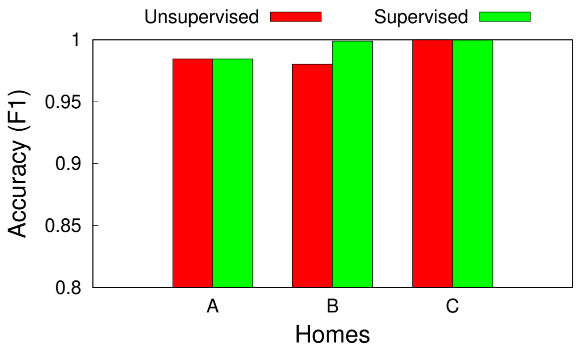


Fig. 4. Comparing with a Supervised Approach

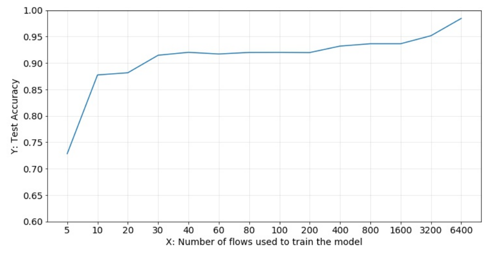


Fig. 5. IoTSpot achieves high accuracy only using 40 "flows".

B. Comparing with a Supervised Approach

We evaluate IoTSpot's accuracy to a fully supervised machine learning approach that has access to all the network traffic data from testing homes. Figure 4 compares IoTSpot's accuracy with that of the supervised approach for each of the 3 homes. For the supervised approach, we use the ratio 7:3 to split the training dataset. For the unsupervised approach, for a specific smart home, we train on the other 2 homes and test on this home. The graph shows that across all the homes, our system—IoTSpot's F1 accuracy does not change significantly. **Results**: *IoTSpot achieves similar accuracy without access to any traffic data from a testing home as a fully supervised approach with complete access to such training data.*

C. Quantifying IoTSpot's Accuracy

We next evaluate different conditions that affect IoTSpot's accuracy. First, we examine the effect of changing the size of the training dataset. Figure 5 shows the results. As expected,

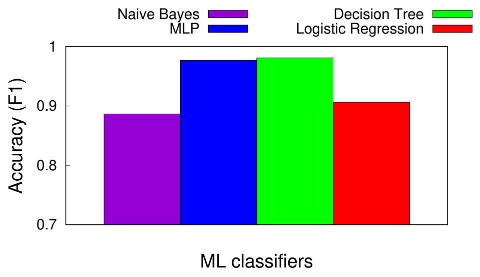


Fig. 6. IoTSpot is orthogonal to the specific ML model.

as the size of the training dataset increases, we see an increase in the F1 score of IoTSpot identification. Interestingly, after increasing to 40 flows, IoTSpot's accuracy becomes stable and stays around 0.92. As we discussed in prior sections, the upper bound for each flow is 1 minute, which says, on average, IoTSpot can achieve F1 accuracy as 0.92 using only 40 minutes of traffic data per device. This also indicates another insight—our approach can still identify IoT devices accurately when missing their traffic data and having at least 40 minutes of traffic data available.

Results: *IoTSpot achieves F1 accuracy as 0.92 when only using as few as 40 traffic flows. In addition, IoTSpot is potentially robust to missing traffic data.*

Eventually, we examine the accuracy effect of changing the ML classifiers. As shown in Figure 6, IoTSpot achieves similar accuracy (0.89∼0.99) when using a set of ML models, including Gaussian Naive Bayes, MLP (Neural Network), Decision Tree and Logistic Regression.

Results: *IoTSpot achieves similar F1 accuracy when using different ML models, Thus, IoTSpot is orthogonal to the specific ML model.*

Limitation: For IoT devices that have the same firmware, IoTSpot has F1 accuracy as ∼0.7. This is mainly because these devices have the same characteristics except their exterior shapes might look different. For instance, we have Belkin Wemo and Belkin Wemo Mini Smart Plug deployed in our "mock" smart homes. These two different IoT devices have the same hardware and firmware. Thus, in many situations, they are identified as the same device. While, of course, this results in the wrong reporting when computing F1 scores.

VII. Conclusion

This paper presents IoTSpot, which can identify IoT devices using their anonymized network traffic data. IoTSpot first extracts the major features using PCA, then leverages Random Forest modeling to build a customized network traffic model for each IoT device. We show that IoTSpot has an F1 score of 0.984, which can effectively identify 19 IoT devices. In addition, IoTSpot achieves similar accuracy without access to any network traffic data as fully supervised approach with complete access to such training data. IoTSpot enables numerous cyber attacks on smart homes that are not possible without knowing what devices are deployed in those homes. We plan to explore more IoT devices and also design a new technique to prevent this efficient device identification.

Acknowledgements. We thank School of Computing and Information Sciences, Florida International University for the travel award to present this work. This research is supported by NSF under Grant Nos. I/UCRC IP-1338922, III-Large IIS-1213026, MRI CNS-1429345, MRI CNS-1532061.

References

[1] Statista, "Internet of Things Connected Devices Installed base Worldwide from 2015 to 2025 (in billions)," https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, Mar 30th 2016.

[2] Mirimir, "Collection of User Data by ISPs and Telecom Providers, and Sharing with Third Parties," https://www.ivpn.net/blog/collection-of-user-data-by-isps-and-telecom-providers-and-sharing-with-third-parties, March 15th 2018.

[3] T. Brewster, "Now Those, Privacy Rules Are Gone, This Is How ISPs Will Actually Sell Your Personal Data," https://www.forbes.com/sites/thomasbrewster/2017/03/30/fcc-privacy-rules-how-isps-will-actually-sell-your-data/#2e30f75921d1, Mar 30th 2017.

[4] W. Ding and H. Hu, "On the safety of iot device physical interaction control," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 832–846.

[5] H. Yu, J. Lim, K. Kim, and S.-B. Lee, "Pinto: Enabling video privacy for commodity iot cameras," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 1089–1101.

[6] Q. Wang, W. U. Hassan, A. Bates, and C. Gunter, "Fear and logging in the internet of things," in *NDSS*, 2018.

[7] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhu, "Homonit: Monitoring smart home apps from encrypted traffic," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018.

[8] R. Schuster, V. Shmatikov, and E. Tromer, "Situational access control in the internet of things," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 1056–1073.

[9] S. Schick, "Belkin Wemo Zero-Day Vulnerability Could Leave the Door Open for Iot Attacks," https://securityintelligence.com/belkin-wemo-zero-day-vulnerability-could-leave-the-door-open-for-iot-attacks/, April 23rd 2019.

[10] R. Moore-Colyer, "Samsung SmartThings Hub Vulnerabilities Leave Smart Home Devices Open to Attack," https://www.theinquirer.net/inquirer/news/3036719/samsung-smartthings-hub-riddled-with-20-security-bugs, 2018.

[11] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: When randomness plays with you," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '07. New York, NY, USA: ACM, 2007, pp. 37–48.

[12] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '05. New York, NY, USA: ACM, 2005.

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

[14] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, 2018.

[15] J. Ortiz, C. Crawford, and F. Le, "Devicemien: Network device behavior modeling for identifying unknown iot devices," in *Proceedings of the International Conference on Internet of Things Design and Implementation*, ser. IoTDI '19. New York, NY, USA: ACM, 2019, pp. 106–117.

[16] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized iot devices using machine learning techniques," *arXiv preprint arXiv:1709.04647*, 2017.