International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Aug-2022)

https://dx.doi.org/10.12785/ijcds/120144

Internet of Things Device Classification using Transport and Network Layers Communication Traffic Traces

Rajarshi Roy Chowdhury[1], Azam Che Idris[1] and Pg Emeroylariffion Abas[1]

[1] *Faculty of Integrated Technologies, Universiti Brunei Darussalam, Jalan Tungku Link, Gadong BE1410, Brunei Darussalam*

Received 16 Sep.2021, Revised 22 Jul. 2022, Accepted 31 Jul. 2022, Published 6 Aug. 2022

Abstract: In recent years, resource-constrained Internet of Things (IoT) devices have been incorporated in many domains. However, malicious users and attackers in the cyberspace have been taking advantage of these technological advancement, to gain unauthorized access to these devices. It is essential to identify all connected devices uniquely, to improve network security as well as preserve user's privacy and safety. In this paper, a device fingerprinting scheme have been proposed by utilizing device-originated communication traffic attributes from a single transmission control protocol (TCP)/internet protocol (IP) packet information. Nine features have been extracted for the proposed scheme. This approach has been evaluated using five machine learning algorithms: J48, Random Forest, Random Tree, Bagging, and Stacking, on three IoT datasets: the IoT Sentinel, UNSW, and D-Link IoT, to study the trade-off between classification performance and processing time. Experimental results have shown that the Bagging classifier achieves 96.6% precision, and 96.4% recall and f-measure using the D-Link IoT dataset, respectively, however, requiring a significant amount of time. On the other hand, the J48 classifier achieves comparable performance whilst requiring only a minimum time. The result is significant as the proposed device fingerprinting scheme can be used to increase security of an IoT network.

Keywords: Network Traffic Traces, Machine Learning Algorithm, Device Fingerprinting, Internet of Things, IoT Devices

1. Introduction

The Auto-ID center first instigated the term Internet of Things (IoT) in 1999, with a vision of identifying individual physical world object using globally unique identifier by means of radio frequency identification (RFID) [1], [2] tag, and allowing the objects to interact with one another over the Internet [3], [4]. Since then, IoT network has expanded rapidly and has been incorporated in many heterogeneous objects, technologies, and applications. Numerous communication protocols have also been developed to connect the physical world objects to the digital world. IHS Markit predicts that the total number of connected IoT devices worldwide will surge to approximately 125 billion by 2030 [5]. A wide range of IoT devices has made possible the development of different smart applications in our everyday life. These applications may be grouped into different domains, including smart city, smart home, smart agriculture, smart transportation, smart health, and fitness [3], [5], [6].

People are adapting to these technological advancements personally and socially, for industry and business purposes, due to their low cost, simplicity, and ease of use. For instance, smartphones and web applications now allow its user to control devices remotely over the network, and internet protocol (IP) camera can be used for the remote monitoring of home or office. However, these heterogeneous IoT devices [3], [7], which commonly are relatively resource-constrained in terms of memory, processing power and energy, are sometimes connected to the Internet with naive security configuration [7], [8]. This imposes new security and privacy challenges in the cyberspace, including device management, anomaly detection, and authentication. To mitigate these issues, device identification plays a vital role in an IoT network. In a network, communicating devices can be identified based on either explicit user-defined identifiers, such as internet protocol (IP) and media access control (MAC) addresses, or network traffic analysis, such as through analysis of packet, frame and radio signal attributes. Unfortunately, explicit identifiers have been shown to be easily mutable by knowledge of networking and even, with the use of some freely available software [9]. Some researchers [10] have successfully utilized radio signal as identifiers. In reference [10] the authors have utilized received signal strength indicator (RSSI) to identify position of a WiFi-enabled system in indoor un-ideal environment, with the classifier achieving respectable accuracy. However, these radio signal-based approaches commonly require an expensive hardware tool [11] for implementation. Consequently, many researchers have proposed distinct device fingerprinting (DFP) methods based on network traffic analysis (packet or frame), by utilizing machine learning (ML) or deep learning (DL) algorithms for device

E-mail address: rajarshiry@gmail.com or 19h0901@ubd.edu.bn, azam.idris@ubd.edu.bn, emeroylariffion.abas@ubd.edu

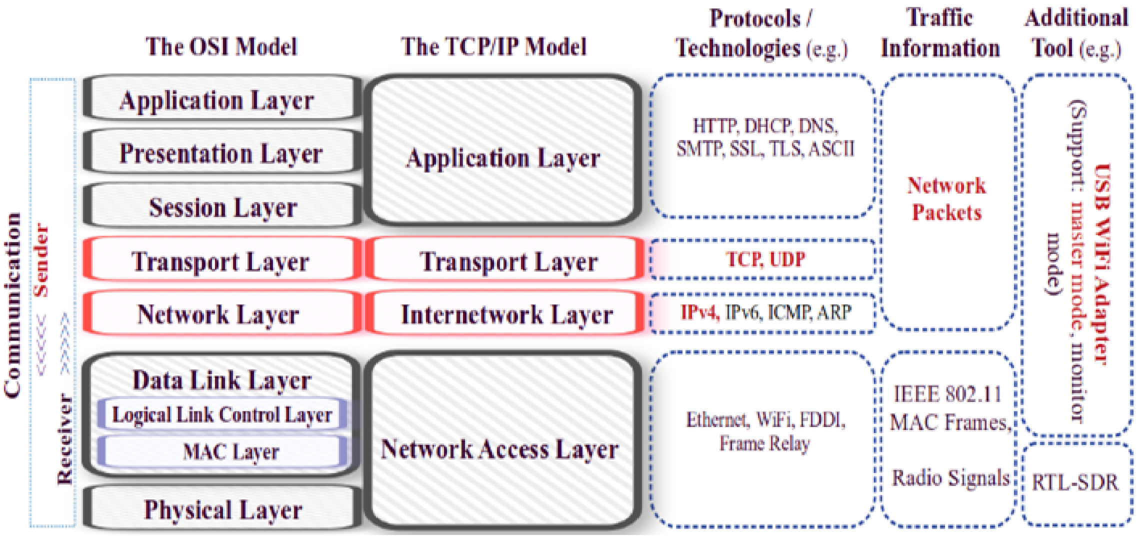


Figure 1. Basic structure of the standard communication models

identification [12], [13], [14].

Device fingerprinting [9], [15], [16] is one of the techniques that may be used to identify both connected IoT (specific-purpose devices, such as door sensor, smart switch/plug, smart bulb) and non-IoT (general computing devices, such as smart phone, tablet, and computer) devices in a network or the Internet without utilizing explicit identifiers. It has emerged as a significant solution in improving network security, due to its resistance against vulnerabilities, such as node forgery and masquerading attack. DFP can be categorized into either active or passive fingerprinting approaches [9]. In an active fingerprinting approach, a profiler is required to send a request to target systems, in order to capture corresponding responses for DFP analysis, whereas in a passive fingerprinting approach, a system captures target network communication traffic traces (both inbound and outbound network traffic traces), without actively making any request to any particular system [9], [15].

In this paper, a DFP approach to classify IoT devices based on the analysis of passively observed network packet traces, which devices use to communicate over the network, has been proposed. DFP may be extracted from different layers of the communication models, including the open systems interconnection (OSI) model and the transmission control protocol/internet protocol (TCP/IP) model [17]. The network and transport layers protocols are used to generate DFP for classification purpose in the proposed DFP model. Subsequently, these selected fingerprints have been used to train various ML models, including J48, Random Forest (RF), Random Tree (RT), Bagging (BG), and Stacking (ST), and it has been shown that the proposed DFP model achieves better performance by utilizing the J48 algorithm, in terms of device identification and processing time. 91.4% precision, and 91.1% recall and f-measure have been achieved by the J48 algorithm using IoT Sentinel dataset within 19 seconds, whilst utilizing the UNSW dataset precision and f-measure reach up to 96.7% and 94.6% in 331 seconds, respectively. On the D-Link IoT dataset, the scheme attains over 95.7% precision along with 95.2% recall and 95.1% f-measure within 1,079 seconds. The main contributions of this work are:

- Identifying a set of key attributes (TCP, UDP, IP protocols features) from network traffic traces, which can be used to characterize a cohort of IoT devices uniquely.
- Evaluate the proposed DFP model performance based on the selected nine features, to classify IoT devices using different supervised machine learning algorithms.
- Performance metrics (precision, recall, f-measure and RMSE) have been used, to find a trade-off between processing time (including training and testing times) and accuracy.

The remainder of this paper is organized as follows. Section 2 describes related works to the topic. IoT device data and collection process, network traffic analysis, the proposed machine learning based IoT device identification model, and performance measures are given in Section 3, followed by an exploration of the model performance using different datasets and ML algorithms in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

Device fingerprint or signature for IoT devices can be generated from different layers of the communication models [17], [18], as shown in Figure 1, based on the analysis of distinct feature vectors, such as the network packets [12], [15], [19] in the network layer, MAC frame [14], [20] in the data link layer, and radio signal [21], [22] in the physical layer. Many researchers have proposed different DFP models utilizing only network traffic traces, due to the availability of network traffic in a local area network (LAN) or wide area network (WAN), as well as the low-cost hardware required to capture communication traffic traces for analysis.

Miettinen et al. [12] have proposed an automated IoT device identification framework based on the analysis of passively observed network traffic traces, in order to enforce security and privacy in IoT networks. To generate unique device fingerprinting, 276-dimensional feature vectors (12 x 23 features) are extracted from 12 consecutive network packets, including from the link, network, transport, and application layers protocols information. These feature vectors are then utilized to train a machine learning model per device type, in order to uniquely classify IoT devices. The scheme achieves 81.5% accuracy (global ratio) over 27 IoT devices. The relatively low model performance is due to the presence of multiple devices from the same manufacturer in their dataset.

In reference [23], the authors have used five packets (forming a session) to construct DFP (5 x 20 features), from packet header protocols and payload attributes. The features are then used to train ML algorithms to give 99% accuracy over 10 devices from different manufacturers. Statistical set of features of attributes from individual device communication flows (n number of packets), have been extracted in references [24], [25], [26], to characterize the devices. These selected features are then utilized to train various ML models for classification.

Aksoy and Gunes [27] have proposed a DFP model referred to as SysID, based on the analysis of 212 features from a single TCP/IP packet information, with features extracted from different layers including from the network, transport, and application layers distinct protocols features. Accuracy of 82% has been reported using the IoT Sentinel dataset (23 devices), with key features selected using a genetic algorithm to improve the classification accuracy. Following this scheme, reference [15] has utilized metric entropy calculation to identify the suitable subset of features for DFP to reduce complexity.

3. Methodology

A. Datasets

The proposed method has been evaluated using two publicly available online datasets [12], [24], and an experimental testbed dataset of D-Link IoT devices [28], as listed in Table I. The IoT Sentinel dataset [12] comprises of 31 smart home IoT devices from different manufacturers, including from D-Link, Edimax, Fitbit, Ednet, Belkin, Withings, TP-Link, HomeMatic, PhilipsHue, and Smater, with the dataset consisting of individual device setup phase communication traffic only. In this dataset, network traffic traces have been captured from cameras, smart-bults, smart-plugs, smart-switches, and smart electronic gadgets. On the other hand, the UNSW dataset [24] incorporates both IoT (22 devices) and non-IoT (7 devices) devices traffic traces that have been captured for a long period of time from a smart laboratory setup environment. Lastly, the D-Link IoT dataset [28] is comprised of 14 IoT devices from the same manufacturer but of different categories, including cameras, door-sensors, hub, smart-plugs, as well as Z-Wave-enabled IoT devices (two D-Link door sensors). All the communication traffic traces have been captured in a controlled laboratory environment. The data collection setup for the D-Link IoT dataset is described in the next section.

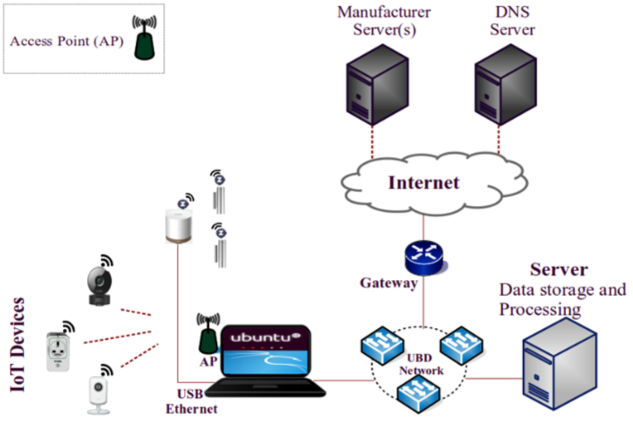


Figure 2. An experimental testbed of an IoT network

B. Data Collection Methodology

Network traffic traces are digital footprints of connected devices in a network. Basically, IoT devices leave traffic traces during communication either with other network-connected devices, or with servers in two cases [29]: (i) autonomous traffic, including domain name translation, clock synchronization, and device-to-device interaction, and (ii) traffic generated by human/object interactions, including remotely on-off operation of a smart device, pre-defined activities triggered by motion sensor, and video transfer over the network from IP camera.

In this paper, traffic traces from both scenarios have been collected passively, with experimental testbed shown in Figure 2, which forms the D-Link IoT dataset. All inbound and outbound traffic traces from the connected IoT devices have been captured on the access point (AP) using tcpdump (4.9.2.-4) utility [12], [28], [29] (an open-source command-line packet sniffer or analyzer). The utility allows the reading of packet contents via a network interface (WiFi or Ethernet interfaces) [30]. For instance, the following command is used to capture and store network packets from a network interface (*-i* flag): *tcpdump -i interface_name -w file_name.pcap*. A laptop, running Ubuntu (18.04) as host operating system (OS) and Kali Linux (2020.3) as guest OS over VMware Workstation Player (15.5.2), has been configured as an AP using software packages – hostapd

TABLE I. Details of the datasets

Dataset	Devices	Instances	TE	IO	Traffic	PCM	AT	AP	Source
IoT Sentinel	31	102,226	Real	All	Setup	tcpdump	–	Laptop	[12]
UNSW	22	6,844,821	Real	All	All	tcpdump	Cron	Gateway	[24]
D-Link IoT	12	15,308,711	Real	All	All	tcpdump	Cron	Laptop	[28]

Note: Testbed Environment - TEP, Inbound & Outbound - IO, Packet Capture Module - PCM, Automate Tasks - AT, Access Point - AP

[31] and dnsmasq [32], with a WiFi adapter attached to the laptop functioning as a WiFi interface (interface mode - master) for the Kali Linux.

Additionally, an external Ethernet adapter is connected to the same system, which acts as the Ethernet interface. The laptop gets its internet services through the university network over its built-in Ethernet port. A local server is connected to the same network for data storage and processing. This experimental testbed has been setup in the network systems and signal processing (NSSP) laboratory, Universiti Brunei Darussalam (UBD), with 14 D-Link IoT devices, as listed in Table II.

TABLE II. List of D-Link IoT devices

Category	Model	Devices	Connectivity
Camera	DCS-936L	2	WiFi
	DCS-930L	6	WiFi/Ethernet
Smart Plug	DSP-W215	3	WiFi
Home Hub	DCH-G022	1	WiFi/Ethernet /Z-Wave
Door Sensor	DCH-Z112	2	Z-Wave

C. Network Traffic Features Analysis

Feature vectors for DFP can be attained from different dimension of network traffic traces, including single packet information [15], [27], sequence of packets [12], [33], statistical features of packets [19], [33], [34] and combination of statistical and measurement values [34]. Traffic traces originating from devices are filtered by utilizing individual IoT device MAC addresses. These traces may carry unique characteristics of a device communication pattern, which may be used for DFP. From the collected traffic traces, 82 features are extracted from the network (IP) and transport (transmission control protocol (TCP) and user datagram protocol (UDP)) layers of the communication models, according to individual packet information. TShark utility [35] has been used for extraction.

Subsequently, these features are evaluated, by utilizing an attribute evaluator (GainRatioAttributeEval) and a search algorithm (Ranker) from Weka tool [36], to identify a significant subset of features that can be used for device fingerprinting. Subsequently, features uncorrelated to devices with gain ratio threshold value <= 0, have been removed. Time dependent attributes (such as *tcp.options.timestamp.tsval, tcp.options.timestamp.tse-cr*), attributes with very limited values (such as *tcp.option_kind*, and *ip.dsfield.dscp*), attributes with negative/hexadecimal/binary values (such as *tcp.window_size_scalefactor, udp.checksum,* and *tcp.flags.ack*), have been removed from the feature list. Finally, the feature list has been narrowed down to only nine features: *ip.len, ip.ttl, ip.proto, tcp.srcport, tcp.stream, tcp.ack, tcp.window_size, udp.srcport, udp.stream*, to be used as the proposed DFP feature set for the IoT devices. These features are deemed to carry significant information on the individual device characteristics. For instance, *tcp.window_size* [23] values depend on the individual device internal memory capacity and processing speed, whilst *ip.len* specifies each packet length in bytes, including total bytes in an IP header and data without considering actual information of a packet.

D. Proposed DFP Model

The proposed DFP scheme architecture, to classify IoT devices using network traffic traces is shown in Figure 3, with network traffic traces originating from any of the three datasets: IoT Sentinel, UNSW, and D-Link IoT datasets. This flowchart depicts a complete process; from data collection to device identification/ classification, including the proposed DFP model. These devices originated traffic traces are filtered, to extract a set of the selected nine features, which have been deemed to be significant for DFP of the IoT devices. Each of the IoT datasets is then randomly split into two subsets: 80% for training, and the remaining 20% for testing. The labelled training subsets are used to train the different ML classification algorithms, and subsequently, the trained ML classification algorithms are tested, by classifying the testing subsets for the classification task. Different performance measures are calculated from the outcome of the classification tasks.

These training and testing datasets are then used to train and test different ML classification algorithms using a shell script (a command line interpreter is designed for running a computer program by using the Unix shell [37]), to measure the performance of the proposed DFP model.

E. Classification Models

Five ML classifiers: J48 (C4.5), Random Forest (RF), Random Tree (RT), Bagging (BG), and Stacking (ST), have been used to evaluate the proposed DFP model. In this study, only trees-based different ML algorithms are considered for classifying devices. From the existing works

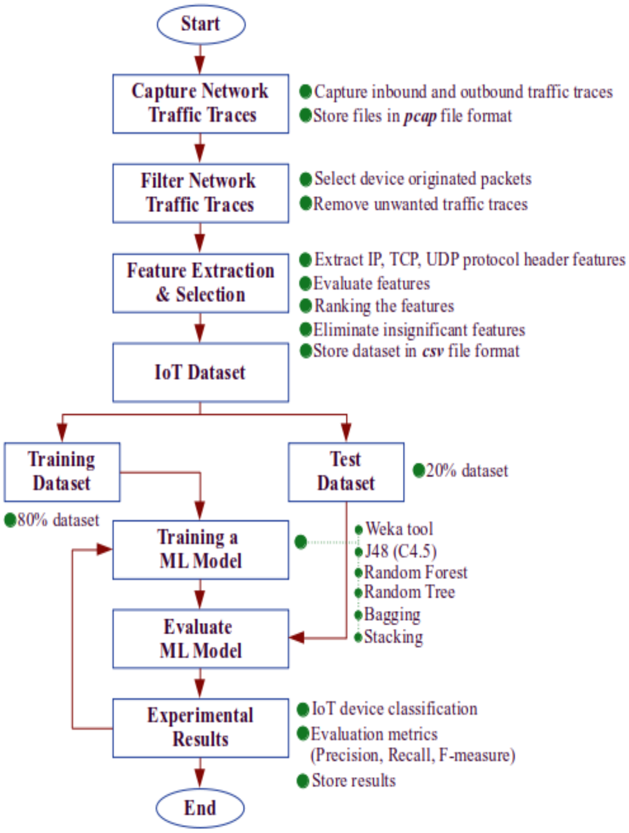

Start
Capture Network Traffic Traces
● Capture inbound and outbound traffic traces
● Store files in *pcap* file format
Filter Network Traffic Traces
● Select device originated packets
● Remove unwanted traffic traces
Feature Extraction & Selection
● Extract IP, TCP, UDP protocol header features
● Evaluate features
● Ranking the features
● Eliminate insignificant features
● Store dataset in *csv* file format
IoT Dataset
Training Dataset
● 80% dataset
Test Dataset
● 20% dataset
Training a ML Model
● Weka tool
● J48 (C4.5)
● Random Forest
● Random Tree
● Bagging
● Stacking
Evaluate ML Model
Experimental Results
● IoT device classification
● Evaluation metrics (Precision, Recall, F-measure)
● Store results
End

Figure 3. A flowchart of the proposed IoT device classification model architecture

in references [10], [12], [15], [24], [27], [38], [39], it has been observed that these trees-based ML models give better classification performances compared with other types of ML algorithms, such as rules-based, Bayes theorem-based.

- *J48 (C4.5)*: J48 is an extended version of the Iterative Dichotomiser 3 (ID3) classifier. In Weka, the C4.5 classifier is known as J48, which produces a decision tree based on information theory for the classification approach [36]. It allows investigating a significant subset of features, to utilize for classification [15].

- *Random Forest (RF)*: RF classifier is a supervised ML algorithm, which can be used for both classification and regression problems. It produces decision based on a set of decision trees estimations, either using majority voting or average results of n number of decision tree [12], [24], [36].

- *Random Tree (RT)*: RT classifier consists of a combination of single model trees and RF algorithms, which is used for classification and regression problems [36]. A model accuracy improves by using a set of uniformly distributed random trees. For a classification problem, this classifier measures a final decision based on the majority voting of ensemble trees [40].

- *Bagging (GB)*: Bootstrap aggregating (or Bagging) is an ensemble meta-algorithm utilize for both classification [41] and regression. It assists in reducing variance and over-fitting problem of a model. For classification, this classifier predicts actual value based on average probability estimations [40].

- *Stacking (ST)*: This classifier can be used for both classification and regression based on stacking [42] homogeneous and different classifiers [36]. Instead of averaging or voting classifiers results to estimate a final result, the Stacking classifier trains a meta-learner based on all the classifiers results as input to a meta-learner to produce an ensemble result.

F. *Performance Measures*

The performance of the DFP model with the different classification algorithms, have been measured in terms of device classification performance as well as processing time. Three evaluation metrics have been used: precision (Equation 1), recall (Equation 2), f-measure (Equation 3), and RMSE (Equation 4), to measure classification performance. These metrics quantify the effectiveness of the DFP model using the different ML classification algorithms, for device classification.

$$Precision = \frac{TP}{(TP + FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

$$F - measure = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{3}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x}_i)^2}{N}} \tag{4}$$

where true-positive (TP) is the total number of positive samples correctly classified. False-positive (FP) and false-negative (FN) represent the total number of positive and negative instances classified incorrectly, respectively. In Equation 4, x_i and $\bar{x}_i$ represent actual and predicted values, respectively, of the available traffic instances in a dataset and N defines the total number of instances available in this dataset.

4. EVALUATIONS AND RESULTS

The proposed IoT device identification model has been evaluated on a Dual core Intel i5-5200U CPU at 2.20GHz, Dual channel DDR3L-1600MHz 8GB RAM x 2, and an addlink SATA SSD 512GB hard-drive, running an Ubuntu (18.04) operating system (OS) with waikato environment for knowledge analysis (Weka) tool (3.8.5) [36].

TABLE III. LIST OF HYPER-PARAMETERS TUNED FOR THE CLASSIFIERS

Alg.	B. Size	D	BP	C	MC	Num. F	Num. I	CF
J48	100	1	–	–	–	3	–	0.25
RF	100	1	100	–	–	–	100	–
RT	100	1	–	–	–	0	–	–
BG	100	1	100	J48	–	–	10	–
ST	100	1	–	J48,BG	J48	10	–	–

Note: Num. – Number, CF – Confidence Factor, B. Size – Batch Size, D – Seed, C – Classifier, MC - Meta Classifier, F- Folds, I – Iterations, Alg. – Algorithm, BP – Bag Size Percent

TABLE IV. THE PROPOSED DFP MODEL CLASSIFICATION PERFORMANCES USING DIFFERENT DATASETS AND ALGORITHMS

Alg.		Precision			Recall			F-Measure	
J48	.914	.967	.957	.911	.938	.952	.911	.946	.951
RF	.904	.972	.954	.903	.937	.953	.903	.947	.953
RT	.879	.968	.946	.878	.933	.945	.878	.942	.945
BG	.924	.967	.966	.921	.943	.964	.921	.949	.964
ST	.924	.968	.967	.922	.938	.963	.921	.947	.963

Note: Alg. – Algorithm, Bagging – BG, Stacking – ST, Datasets (color) – IoT Sentinel, UNSW, D-Link IoT

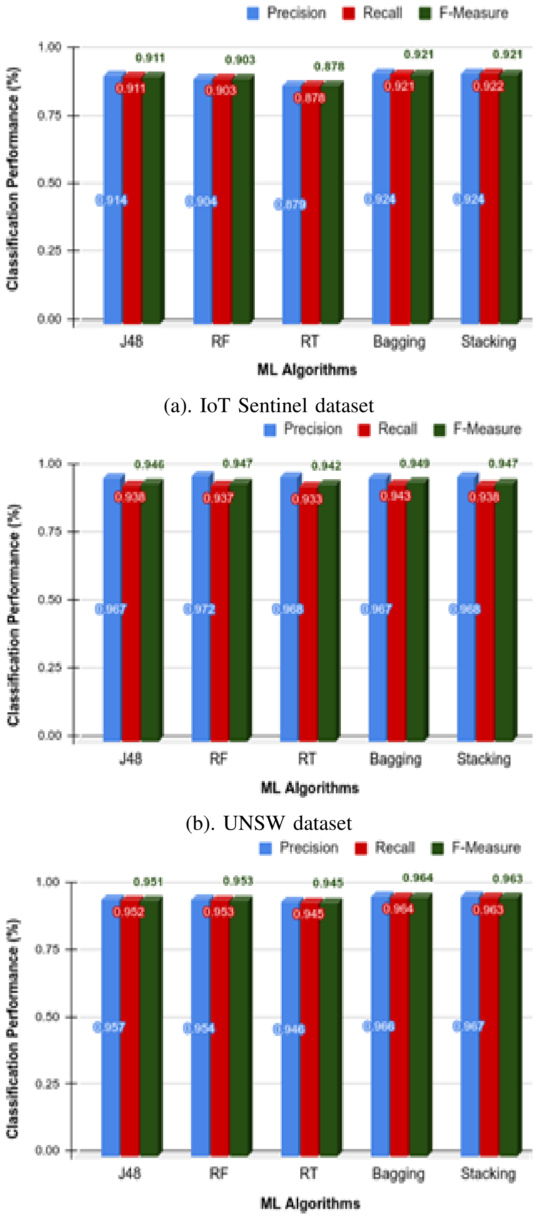
Two online datasets: the IoT Sentinel and UNSW datasets, and one experimental dataset: the D-Link IoT dataset, have been used to measure the performance of the proposed DFP model. These datasets have been divided into 80:20 training:testing, using the unsupervised instance resample function (weka.filters.unsupervised.instance.Resample [43], which subsamples dataset instances randomly. The proposed DFP model then extracted 9 carefully selected features from the network and transport layers. These nine features are deemed to carry significant information which can be used to uniquely identify the IoT devices. The features were then used for training the ML classifiers, which have been obtained from the workbench ML open-source (Weka) tool [36].

Performance of the DFP model have been measured in terms of precision, recall, f-measure, and RMSE. Additionally individual classifier processing time (combination of training time and testing time) has also been considered in the context of a specific system, to investigate the trade-off between classification results and computation time. Table III shows the list of hyper-parameters, which have been tuned to control the learning process of the classification algorithms.

Figure 4 and Table IV show classification performances of the proposed DFP model on the IoT Sentinel, UNSW, and D-Link IoT datasets, respectively, using different ML classification models. As can be seen, all the ML algorithms are able to classify individual IoT devices effectively based on the selected subset of features by using the proposed DFP model. Random Tree (RT) classifier gives the lowest performance in recall and f-measure as compared to all the other classifiers, while the Bagging algorithm classifies IoT device with the highest accuracies on all three experimental datasets. Bagging and Stacking machine learning classifiers attain more than 92% precision, recall, and f-measure, while J48 classifier attain over 91%.

In reference [12], the proposed automated IoT device identification method achieves overall accuracy of 81.5% using 27 devices out of the available 31 devices in the IoT Sentinel dataset. Using similar dataset, average classification accuracy of the SysID has been shown to be 82% by considering 23 devices [27]. As can be seen from Figure 4a, the proposed DFP model gives precision of up to 92%, particularly using the Bagging and Stacking algorithms, despite the presence of devices from the same manufacturer in the dataset. However, it should be noted that for performance evaluation, two IoT devices: iKettle2 and SmarterCoffee devices, have been excluded due to the minimal number of traffic traces available in the dataset as compared to other devices.

On the UNSW dataset, as shown in Figure 4b, the J48 and RT classifiers give low performance in precision as compared to the Random Forest (RF) algorithm, although the difference between the algorithms are less than 1%. It can also be seen from Figure 4b that the Bagging algorithm effectively identifies all the IoT devices, to give over 94% for all performance measures using the selected set of features based on a single TCP/IP packet information. In reference [24], the researchers achieve over 99% accuracy using a set of n number of packets information (network flow) from different layers of the communication models on the UNSW dataset. However, network features need to be computed on an hourly basis to generate device fingerprints,



(a). IoT Sentinel dataset

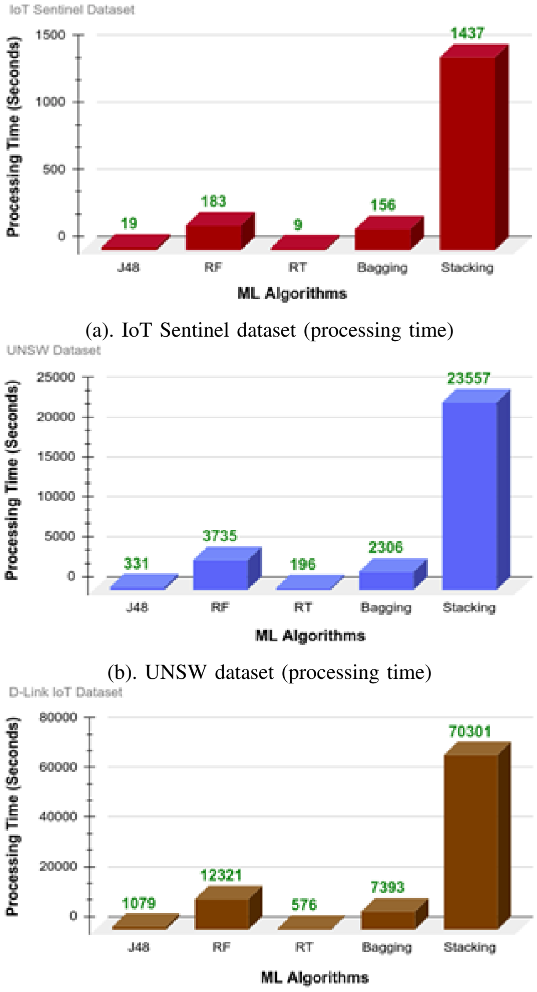
(b). UNSW dataset

(c). D-Link IoT dataset

Figure 4. Classification performance results using different datasets: (a) IoT Sentinel (b) UNSW, and (c) D-Link IoT datasets

which somehow limit the effectiveness of the method.

Figure 4c depicts individual IoT device classification performance metrics using the D-Link IoT dataset which has been collected in the laboratory. All the selected ML algorithms show good performance results in identifying the IoT devices, despite the devices coming from the same manufacturer. Using the Bagging and Stacking classifiers, the proposed DFP model attain over 96% classification results based on the selected subset of features, while the J48 and RF classifiers attain over 95%. These results have demonstrated that the proposed model is very effective in identifying IoT devices, irrespective of the devices coming from the same manufacturer or of similar categories of devices.



(a). IoT Sentinel dataset (processing time)

(b). UNSW dataset (processing time)

(c). D-Link IoT dataset (processing time)

Figure 5. Processing times (including training and testing times) of ML classification algorithms: (a) IoT Sentinel (b) UNSW, and (c) D-Link IoT datasets

In Figure 5 and Table V, processing times of the selected ML classification algorithms on the different datasets are presented. Time scales for the different datasets vary between 0 – 1,800 seconds for the IoT Sentinel dataset, 0 – 25,000 seconds for the UNSW dataset, and 0 – 80,000 seconds for the D-Link IoT dataset, based on the recorded processing times. Figures 5a, 5b, and 5c follow similar patterns for the different algorithms, irrespective of the datasets, with the Stacking and Random Tree classifiers representing the slowest and the fastest classifiers, respectively. Although the Random Tree classifier performs the fastest, the algorithm gives lower performances as compared to the other algorithms. On the IoT Sentinel dataset, the RT

TABLE V. THE PROPOSED DFP MODEL CLASSIFICATION PERFORMANCES BASED ON PROCESSING TIME

		IoT Datasets	
Algorithm	IoT Sentinel	UNSW	D-Link IoT
J48	19 sec	331 sec	1,079 sec
RF	183 sec	3,735 sec	12,321 sec
RT	9 sec	196 sec	576 sec
BG	156 sec	2,306 sec	7,393 sec
ST	1,437 sec	23,557 sec	70,301 sec

Note: Bagging – BG, Stacking – ST, Second – Sec

classifier requires only 9 seconds for data processing but achieves only over 87% identification accuracies. On the other hand, the Bagging classifier achieves 92% accuracies but requires 156 seconds for processing. Overall, the Stacking classifier requires much longer processing time but has not provided significant performance improvements over the other classifiers. On the D-Link IoT dataset, the stacking classifier requires 70,301 seconds of processing time to give 96% performance. On the other hand, the Bagging classier achieves similar result whilst requiring only 7,393 seconds of processing time, as shown in Figure 5c. Additionally, it has also been observed that the J48 classifier exhibits good performance results in identifying IoT devices within a short time scale.

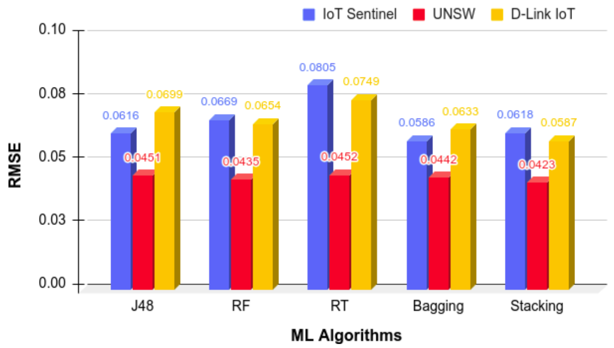


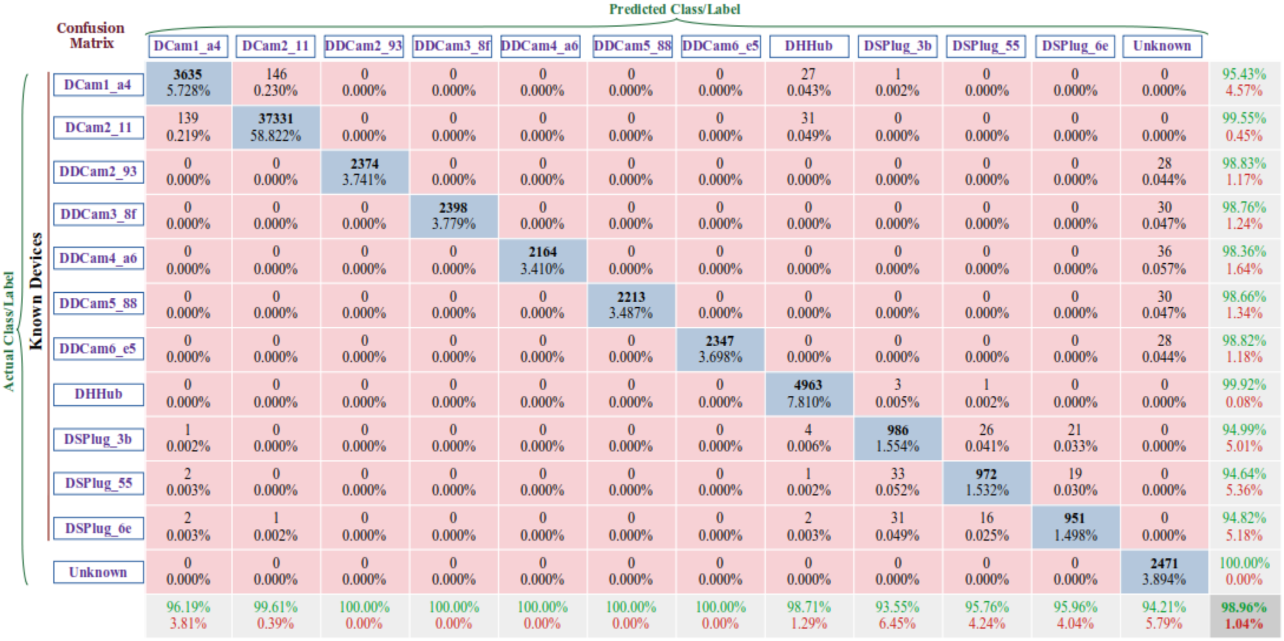
Figure 6. Evaluation of the proposed DFP model using RMSE

In Figure 6, classification performances of the selected five ML algorithms, in terms of their RMSE values, are presented on the three different IoT datasets. The proposed DFP model gains better performances in identifying individual IoT devices on the IoT Sentinel dataset by utilizing the Bagging classifier, with the classifier giving the lowest RMSE value of 0.0586 as compared to the other classifiers. However, the Stacking classifier gives RMSE values of 0.0423 and 0.0587 on the UNSW and D-Link IoT datasets, respectively, which are the minimum error rates as compared to the other four ML classifiers. On average the Stacking classifier gives the lowest RMSE value of 0.0543, whilst the RT classifier gives the highest RMSE value of 0.0669, in classifying IoT devices on all the experimental datasets.

Figure 7 shows the confusion matrix of the proposed DFP model classification performances in identifying known and unknown (intruder) IoT devices on the D-Link IoT datasets. 11 known and 1 unknown IoT devices, with 305,051 and 12,270 instances, respectively, have been considered. The supervised ML J48 classifier has been selected due to its minimal requirement on processing time during training and testing, as compared to the other 4 ML classifiers. Overall, 98.96% accuracy and 0.0328 RMSE value have been obtained using the J48 classifier. It can be observed that the proposed DFP model is able to identify unknown D-Link IoT instances with 100% accuracy, despite misclassifying some of the known IoT device instances as an unknown device. For instance, 28 instances from the DDCam2_93 device have been incorrectly classified as unknown instances. On the UNSW dataset (21 devices) with the addition of 1 unknown device, accuracy and RMSE value 97.38% and 0.0383, respectively, have been obtained. The proposed DFP model reported 99.8% accuracy in identifying unknown instances on the UNSW dataset. It is noted that in the context of identifying unknown instances, the IoT Sentinel dataset is not suitable to use for demonstration, as this dataset only consists of setup traffic traces. For identification of unknown instances with high accuracy, it is required that the ML model learn known IoT devices behaviours appropriately along with setup traffic instances.

5. Conclusion

Identification of heterogeneous IoT devices, which may come from different manufacturers, connected in a network is essential to network security. Despite being a necessity for communication over a network, explicit device identifiers such as IP/MAC addresses, are not suitable to uniquely identify all these devices due to MAC address randomization and IP/MAC addresses spoofing attacks. In this work, a DFP approach has been proposed to uniquely identify devices in a network, based on the analysis of device originated communication traffic from two layers. A subset of 9 features has been extracted from different protocols headers information without considering deep packet inspection, and hence, the method preserves users' data privacy and safety. An IoT network with fourteen devices has been setup in a laboratory to collect network traffic traces for analysis. Different ML classification algorithms have been used to evaluate the proposed model performance on three IoT datasets. Results have illustrated that the proposed DFP model achieves over 96.6% precision, and 96.4% recall and f-measure, respectively, on the D-Link IoT dataset using the Bagging classifier, albeit requiring a significant amount of time. On the other hand, the J48 classifier obtains almost similar result (less than 1%) as compared to Bagging algorithm, but within a shorter time scale. The proposed DFP method reports overall accuracy of 98.96% in identifying known and unknown IoT devices on the D-Link IoT dataset. These illustrate that the proposed DFP model is relevant and useful for network administrators or operators in improving network security in the context of identification of unknown traffic traces, and significantly


Predicted Class/Label
Actual Class/Label
Known Devices

Confusion Matrix	DCam1_a4	DCam2_11	DDCam2_93	DDCam3_8f	DDCam4_a6	DDCam5_88	DDCam6_e5	DHHub	DSPlug_3b	DSPlug_55	DSPlug_6e	Unknown	
DCam1_a4	3635 5.728%	146 0.230%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	27 0.043%	1 0.002%	0 0.000%	0 0.000%	0 0.000%	95.43% 4.57%
DCam2_11	139 0.219%	37331 58.822%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	31 0.049%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	99.55% 0.45%
DDCam2_93	0 0.000%	0 0.000%	2374 3.741%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	28 0.044%	98.83% 1.17%
DDCam3_8f	0 0.000%	0 0.000%	0 0.000%	2398 3.779%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	30 0.047%	98.76% 1.24%
DDCam4_a6	0 0.000%	0 0.000%	0 0.000%	0 0.000%	2164 3.410%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	36 0.057%	98.36% 1.64%
DDCam5_88	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	2213 3.487%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	30 0.047%	98.66% 1.34%
DDCam6_e5	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	2347 3.698%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	28 0.044%	98.82% 1.18%
DHHub	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	4963 7.810%	3 0.005%	1 0.002%	0 0.000%	0 0.000%	99.92% 0.08%
DSPlug_3b	1 0.002%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	4 0.006%	986 1.554%	26 0.041%	21 0.033%	0 0.000%	94.99% 5.01%
DSPlug_55	2 0.003%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	1 0.002%	33 0.052%	972 1.532%	19 0.030%	0 0.000%	94.64% 5.36%
DSPlug_6e	2 0.003%	1 0.002%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	2 0.003%	31 0.049%	16 0.025%	951 1.498%	0 0.000%	94.82% 5.18%
Unknown	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	0 0.000%	2471 3.894%	100.00% 0.00%
	96.19% 3.81%	99.61% 0.39%	100.00% 0.00%	100.00% 0.00%	100.00% 0.00%	100.00% 0.00%	100.00% 0.00%	98.71% 1.29%	93.55% 6.45%	95.76% 4.24%	95.96% 4.04%	94.21% 5.79%	**98.96%** **1.04%**

Figure 7. Confusion matrix of the proposed DFP model for classifying known and unknown IoT devices from the D-Link IoT dataset

point to the applicability of the proposed DFP method for device identification in an IoT network.

As a future direction, more device-specific features may be analyzed to increase accuracy of the device identification task even further. Furthermore, different non-IoT devices datasets as well as IoT devices datasets with multiple intruder IoT devices (or malicious devices) may be considered, to further evaluate the performance of the method.

Acknowledgment

The authors are profoundly grateful to the Faculty of Integrated Technologies (FIT), Universiti Brunei Darussalam (UBD), for supporting this research work, as well as to UBD for awarding the UBD Graduate Scholarship (UGS) to the first author.

References

[1] S. Madakam, V. Lake, V. Lake, V. Lake *et al.*, "Internet of things (iot): A literature review," *Journal of Computer and Communications*, vol. 3, no. 05, p. 164, 2015.

[2] R. R. Chowdhury and M. Ansary, "A secured mutual authentication protocol for rfid system," *international journal of scientific & technology research*, vol. 3, no. 5, pp. 52–56, 2014.

[3] O. Garcia-Morchon, S. Kumar, and M. Sethi, "State of the art and challenges for the internet of things security, draft irtf t2trg iot seccons 15," 2018.

[4] R. R. Chowdhury, "Security in cloud computing," *International Journal of Computer Applications*, vol. 96, no. 15, 2014.

[5] I. Markit, "The internet of things: a movement, not a market," *Critical IoT Insights*, pp. 1–9, 2017.

[6] V. Sarasvathi, S. Smrithi *et al.*, "Air quality monitoring and predicting system for sustainable health management using multi-linear regression in iot." *International Journal of Computing and Digital Systems*, vol. 9, no. 3, pp. 419–432, 2020.

[7] B. J. Mohd, T. Hayajneh, and A. V. Vasilakos, "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues," *Journal of Network and Computer Applications*, vol. 58, pp. 73–93, 2015.

[8] J. Hou, L. Qu, and W. Shi, "A survey on internet of things security from data perspectives," *Computer Networks*, vol. 148, pp. 295–306, 2019.

[9] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 94–104, 2015.

[10] D. J. Suroso, A. S. Rudianto, M. Arifin, and S. Hawibowo, "Random forest and interpolation techniques for fingerprint-based indoor positioning system in un-ideal environment," *International Journal of Computing and Digital Systems*, 2021.

[11] A. S. Uluagac, S. V. Radhakrishnan, C. Corbett, A. Baca, and R. Beyah, "A passive technique for fingerprinting wireless devices with wired-side observations," in *2013 IEEE conference on communications and network security (CNS)*. IEEE, 2013, pp. 305–313.

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

[13] L. Babun, H. Aksu, L. Ryan, K. Akkaya, E. S. Bentley, and A. S. Uluagac, "Z-iot: Passive device-class fingerprinting of zigbee and z-wave iot devices," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.

[14] P. Robyns, B. Bonné, P. Quax, and W. Lamotte, "Noncooperative 802.11 mac layer fingerprinting and tracking of mobile devices," *Security and Communication Networks*, vol. 2017, 2017.

[15] R. R. Chowdhury, S. Aneja, N. Aneja, and E. Abas, "Network traffic analysis based iot device identification," in *Proceedings of the 2020 the 4th International Conference on Big Data and Internet of Things*, 2020, pp. 79–89.

[16] S. Aneja, N. Aneja, B. Bhargava, and R. R. Chowdhury, "Device fingerprinting using deep convolutional neural networks," *International Journal of Communication Networks and Distributed Systems*, vol. 28, no. 2, pp. 171–198, 2022.

[17] M. M. Alani, "Guide to osi and tcp/ip models," 2014.

[18] P. Ravali, "A comparative evaluation of osi and tcp/ip models," *International Journal of Science and Research*, vol. 4, no. 7, pp. 514–521, 2013.

[19] B. Charyyev and M. H. Gunes, "Iot event classification based on network traffic," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 854–859.

[20] X. Gu, W. Wu, X. Gu, Z. Ling, M. Yang, and A. Song, "Probe request based device identification attack and defense," *Sensors*, vol. 20, no. 16, p. 4620, 2020.

[21] G. Qing, H. Wang, and T. Zhang, "Radio frequency fingerprinting identification for zigbee via lightweight cnn," *Physical Communication*, vol. 44, p. 101250, 2021.

[22] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "Deep learning for rf fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

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

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

[25] T. Gu and P. Mohapatra, "Bf-iot: Securing the iot networks via fingerprinting-based device authentication," in *2018 IEEE 15Th international conference on mobile ad hoc and sensor systems (MASS)*. IEEE, 2018, pp. 254–262.

[26] I. Cvitić, D. Peraković, M. Periša, and B. Gupta, "Ensemble machine learning approach for classification of iot devices in smart home," *International Journal of Machine Learning and Cybernetics*, pp. 1–24, 2021.

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

[28] R. R. Chowdhury, S. Aneja, N. Aneja, and P. E. Abas, "Packet-level and ieee 802.11 mac frame-level network traffic traces data of the d-link iot devices," *Data in Brief*, p. 107208, 2021.

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

[30] T. Group *et al.*, "Tcpdump/libpcap public repository," 2020.

[31] I. Gentoo Foundation, "Hostapd - gentoo wiki," 2020. [Online]. Available: https://wiki.gentoo.org/wiki/Hostapd

[32] ArchWik, "dnsmasq," 2020. [Online]. Available: https://wiki.archlinux.org/index.php/dnsmasq

[33] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, "Gtid: A technique for physical device and device type fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 519–532, 2014.

[34] N. Yousefnezhad, M. Madhikermi, and K. Främling, "Medi: Measurement-based device identification framework for internet of things," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. IEEE, 2018, pp. 95–100.

[35] R. Sharpe, "tshark - the wireshark network analyzer 3.4.5," 2021. [Online]. Available: https://www.wireshark.org/docs/man-pages/tshark.html

[36] I. H. Witten, E. Frank, M. Hall, and C. Pal, "The weka workbench. online appendix for "data mining: practical machine learning tools and techniques"," in *Morgan Kaufmann*, 2016.

[37] S. Baird, *Sams teach yourself extreme programming in 24 hours*. Sams Publishing, 2002.

[38] O. Salman, I. H. Elhajj, A. Chehab, and A. Kayssi, "A machine learning based framework for iot device identification and abnormal traffic detection," *Transactions on Emerging Telecommunications Technologies*, p. e3743, 2019.

[39] E. Ganesan, I. Hwang, A. T. Liem, M. S. Ab-Rahman *et al.*, "Sdn-enabled fiwi-iot smart environment network traffic classification using supervised ml models," in *Photonics*, vol. 8, no. 6. Multidisciplinary Digital Publishing Institute, 2021, p. 201.

[40] S. Gayathri, A. K. Krishna, V. P. Gopi, and P. Palanisamy, "Automated binary and multiclass classification of diabetic retinopathy using haralick and multiresolution features," *IEEE Access*, vol. 8, pp. 57 497–57 504, 2020.

[41] B. A. Desai, D. M. Divakaran, I. Nevat, G. W. Peter, and M. Gurusamy, "A feature-ranking framework for iot device classification," in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, 2019, pp. 64–71.

[42] "Stacking (weka-dev 3.9.5 api)." [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/classifiers/meta/Stacking.html

[43] "Resample (weka-dev 3.9.5 api)." [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/filters/unsupervised/instance/Resample.html

Rajarshi Roy Chowdhury Rajarshi Roy Chowdhury is currently pursuing his PhD in Systems Engineering under the Faculty of Integrated Technologies, Universiti Brunei Darussalam. He obtained his Master's degree in Computer Science from Universiti Sains Malaysia, Malaysia in 2012. Later, he joined Sylhet International University, Bangladesh, as a lecturer in 2012. His research interest is Internet of Things (IoT), wireless sensor networks, networking, data analysis, and machine learning.

Dr Azam Che Idris Dr Azam Che Idris is a chartered engineer with a wide interest in technology. Originally trained in high-speed aerodynamics, he gained major exposure to IR4.0 technology during his tenure in a defence consultancy group. His current interest is utilizing machine learning to understand hypersonic flow physics and to control air-breathing engine in Mach 5. He holds a doctorate in Aerospace Engineering from University of Manchester, UK.

Pg Dr Emeroylariffion Abas Pg Dr Emeroylariffion Abas received his B.Eng. Information Systems Engineering from Imperial College, London in 2001, before obtaining his PhD Communication Systems in 2005 from the same institution. He is now working as an Assistant Professor in System Engineering, Faculty of Integrated Technologies, Universiti Brunei Darussalam. His present research interest are data analysis, security of info-communication systems and design of photonic crystal fiber in fiber optics communication.