Efficient IoT Device Identification via Network Behavior Analysis Based on Time Series Dictionary

Jianjin Zhao, *Student Member, IEEE*, Qi Li, Jintao Sun, Mianxiong Dong, *Senior Member, IEEE*, Kaoru Ota, *Member, IEEE*, Meng Shen *Member, IEEE*

***Abstract*—Due to hardware limitations, Internet of Things (IoT) devices without integrated security become easy targets for network attacks. IoT device identification is significant for network security management. Despite many efforts, previous studies either require excessive features raising concerns about efficiency and privacy, or underutilize the data resources to fulfill the potential of simple features. Moreover, the severe data imbalance problem is unaddressed. In this paper, we present IoTProfile, an efficient IoT device identification framework via time series dictionary. It only considers simple packet-level attributes and maps them into different time windows. On this basis, it further follows a shuffle&split organization scheme to structure the imbalanced data as multi-channel time series. By performing random convolutional kernel transformations in two ways and aggregations, IoTProfile captures discriminative patterns and forms the frequency count of recurring patterns to profile the network behaviors of IoT devices over a period of time. The experimental results show that IoTProfile is superior to the other state-of-the-art methods in terms of both identification effectiveness and time overhead, achieving 99.81% and 97.65% Macro-F1 scores on UNSW and UNB datasets in under four minutes.**

***Index Terms*—IoT device identification, machine learning, traffic analysis.**

I. Introduction

As a crucial paradigm for the future form of the Internet, IoT technology [1]–[3] extends network connectivity beyond standard devices to any range of things. The proliferation of IoT devices significantly facilitates people's lives. However, it also raises security issues [4]–[6]. Due to the limited computation and storage resources, IoT devices without integrated security become easy targets for cyber infiltration. This susceptible nature, coupled with the massive number of IoT devices, can lead to large-scale threat activities, posing severe security threats.

To ensure IoT security across data, devices and network, network administrators should be aware of connected IoT devices and monitor corresponding network behaviors. As a critical preliminary work for subsequent anomaly detection, discriminating heterogeneous IoT devices of different types and vendors within the network enables prompt detection and isolation of suspicious/compromised devices and effectively prevents threat activities from the outset. Therefore, it is imperative to realize efficient and accurate IoT device identification for network security management.

Given the significance of IoT device identification, many studies have been presented and achieved considerable progress [7]–[9]. Nevertheless, previous studies share common issues in the following three aspects. (1) *Excessive Feature Extraction.* Cumbersome feature extraction requires heavy computations, thus hardly meeting the efficiency demand in real-time. Additionally, an excessive feature set not only has poor generalization across datasets and environments but also poses significant privacy concerns. (2) *Underutilized Data Resource.* Insufficient feature analysis results in the loss of potentially valuable information. Taking one simple feature, packet lengths with directions, as an example, prior studies mostly conduct statistical or volume-based analysis, while few consider the time series patterns of packet length sequences. Many simple features have not yet fulfilled their full potential in previous studies. (3) *Severe Imbalanced Data.* The activity levels of heterogeneous IoT devices are significantly different. Even identical devices' activities vary in different periods. How to derive balanced and structured features from the grossly imbalanced dataset is still a problem to be solved.

To address these concerns, we propose an efficient IoT device identification system, namely IoTProfile, based on time series dictionary to profile the simplicity nature of IoT devices. Motivated by the observation that the functionalities and corresponding network behaviours of IoT devices are relative simplistic and singular due to hardware constraints, we aim to employ time series dictionary to count the frequencies of communication patterns to discriminate IoT devices. To the best of our knowledge, this work is the first attempt to take advantage of time series dictionary for IoT device identification, which derives simple yet discriminative features from packet-level attributes with low computational expense and achieves competitive identification performance across datasets with many recently proposed methods. We also design a shuffle&split organization scheme to evenly represent the network behaviors of IoT devices over time and effectively solve the data imbalance problem.

(Corresponding authors: Qi Li and Mianxiong Dong.)

Jianjin Zhao is with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China. He was with the Department of Sciences and Informatics, Muroran Institute of Technology, Muroran 050-0085, Japan. (e-mail: jianjinzhao@bupt.edu.cn).

Jianjin Zhao, Qi Li and Jintao Sun are with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876 China. (e-mails: {liqi2001, jintaosun}@bupt.edu.cn).

Mianxiong Dong and Kaoru Ota are with the Department of Sciences and Informatics, Muroran Institute of Technology, Muroran 050-0085, Japan. (e-mails: {mx.dong, ota}@csse.muroran-it.ac.jp).

Meng Shen is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China. (e-mail: shenmeng@bit.edu.cn).

Specifically, IoTProfile first extracts packet-level attributes (i.e., packet lengths with directions) of IoT devices as time series, and maps the time series into different time windows. To obtain structured features and profile IoT devices evenly, it shuffles the time series within each time window and splits into multi-channel time series with a threshold. Then it performs random convolutional kernel transformations at flow level to capture discriminative patterns and count repeated patterns. On the basis, IoTProfile further aggregates flow-level pattern statistics within each time window respectively to comprehensively profile the network behaviors of IoT devices over a period of time. With a well-trained logistic regression classifier, the final identification results are obtained. The contributions of this paper are summarized as follows:

- We propose to model simple packet-level features (i.e., packet lengths with directions) as time series instead of relying on complex features related to various protocol fields and turn IoT device identification problem into a Time Series Classification (TSC) problem.
- We design a shuffle&split organization scheme to balance the network flows in different time windows, which not only enriches the network behavior representations of IoT devices, but also effectively addresses the data imbalance problem.
- We perform random convolutional kernel transformations and time-window based aggregations on well-organized packet length sequences to capture discriminative patterns and count recurring patterns to profile the network behaviors of IoT devices over a period of time.
- We develop an efficient IoT device identification system, namely IoTProfile, and comprehensively compare it with several recently proposed methods. Extensive experimental results demonstrate the superiority of IoTProfile both in accuracy and efficiency across two datasets.

The remainder of this paper is organized as follows. In Section II, the related work is reviewed. In Section III, our proposed IoTProfile is presented. The performance evaluation of our work against several recently proposed methods are provided in Section IV. The limitations and potential future directions are discussed in Section V. Finally, we conclude in Section VI.

II. Related Work

In this section, we review the relevant studies in the following two aspects: IoT device identification and Rocket variants, and Hydra.

A. IoT Device Identification

In light of the rapidly expanding landscape of IoT devices, accurate and efficient device identification is of great significance to enable effective network management and facilitate the application of privacy and security schemes. In 2018, Sivanathan *et al.* [10] first present a systematic IoT device classification framework within smart environments. Through the collection of six-month network traces of IoT devices, extensive features are extracted to depict the network behaviors, including activity patterns (e.g., volume and time distributions) and signaling patterns (e.g., DNS and NTP query and cipher suites in TLS communications), laying the groundwork for subsequent studies.

Building upon this foundation, Fan *et al.* [11] propose a novel IoT device identification model, namely AutoIoT, which incorporates four categories of features related to time interval, traffic volume, protocol, and TLS encryption. By utilizing Compact Clustering via Label Propagation (CCLP) [12], AutoIoT achieves satisfactory performance with less labeled data. To handle open environments, Fan *et al.* develop an IoT/non-IoT device identification model, namely EvoIoT, with great scalability and sustainability. EvoIoT [13] designs representative device selection and model update schemes to address the concept drift caused by constantly emerging IoT devices. Though many studies have achieved satisfactory identification performance in ideal experimental environments, it is relatively difficult to realize accurate identification in real-world complex environments. In this context, Fan *et al.* [14] further propose GraphIoT to model IoT device identification as a heterogeneous graph representation learning problem, achieving promising performance in real-world complex environments. Parallelly, Kostas *et al.* [15] present a machine learning-based IoT device identification method, namely IoTDevID. They rigorously utilize an ensemble of feature selection techniques and a genetic algorithm to eliminate redundant features and select the most compelling feature set. From a more general and realistic aspect, IoTDevID realizes high identification accuracy of IoT devices at packet level. Despite these advancements, one common issue that still persists is the computational burden resulting from excessive feature extraction, which leads to tedious feature selection and complex feature engineering.

In the broader scope of encrypted traffic classification [16]–[18], many studies have shifted their focus towards leveraging simple features that are not tied to specific protocols. For instance, Shen *et al.* [18] demonstrate that packet length information can be informative and distinctive in website fingerprinting task. Similarly, an increasing number of studies also consider realizing accurate IoT device identification with a simple feature set to satisfy real-time requirements. Charyyev *et al.* [19] develop a simple and efficient fuzzy matching method employing Locality-Sensitive Hash (LSH) function to generate network flow signatures for IoT devices, which requires no complex feature extraction, parameter tuning and model retraining. Marchal *et al.* [20] present AuDI for autonomous IoT device identification. They perform discrete Fourier transform and signal auto-correlation on packet length sequences to model IoT devices' periodic communications and utilize the *k*-NN algorithm to realize autonomous device identification without human intervention. Trimanada *et al.* [21] design a universal IoT device profiling tool to extract packet-level signatures, namely PingPong. The key insight of PingPong is that simple packet length sequences are unique and can be exploited as signatures to profile IoT devices. Wang *et al.* [22] design an efficient IoT device classification model with CNN backbone, namely Ulfar. By recovering the format-related bytes from multi-scale n-gram features, Ulfar realize accurate network traffic classification of IoT devices.

Duan *et al.* [23] build a practical IoT device identification system, namely BYTEIOT. They count the frequency distribution of unique packet lengths as features and utilize k-NN algorithm as the classifier. Experimental results show that these simple yet well-organized features can effectively discriminate IoT devices.

In 2022, Kumar *et al.* [24] conduct a thorough experimental analysis of IoT network traffic classification using machine learning algorithms. They categorize the commonly used features into three levels: packet, flow, and behavior, upon which they perform rigorous evaluations considering both accuracy and efficiency. Safi *et al.* [25] comprehensively review previous studies on IoT device profiling. They perform more fine-grained categorizations of the features used in IoT profiling-related problems, which are classified into high, medium, and low levels based on the computational expenses during extraction.

In this work, we aim to realize efficient IoT device identification by determining simple yet effective features. Our key insight is that due to the hardware limitation of IoT devices, their functionalities and corresponding communication patterns are relatively simple and single. To this end, we model IoT devices' network behaviors with packet length sequences and introduce time series dictionary to count the frequencies of communication patterns.

B. Rocket Variants and Hydra

Time series, defined as a set of quantitative values in chronological order, is a ubiquitous form of data. Time series classification problems have been extensively studied and successfully deployed in virtually every scientific field. Among all these studies, the recently proposed algorithm, ROCKET, should be highlighted for its exceptionally fast and accurate performance in time series classification.

In 2020, Dempster *et al.* [26] first propose ROCKET to perform random convolutional kernel transformations on time series. Unlike previous studies focusing on shape, frequency, or variance, ROCKET utilizes convolutional kernels to capture basic patterns in time series. Moreover, the kernel weights are randomly determined rather than optimized by training. They demonstrate that various random convolutional kernels with different lengths, weights, dilations, and biases can capture complex patterns at different scales. By computing two aggregate features from the convolution outputs in combination as features, even simple linear classifiers can effectively discriminate time series. ROCKET is competitive with the most advanced algorithm and is faster than previous studies by several orders of magnitude.

Dempster *et al.* [27] reformulate ROCKET from two aspects: speeding up the transformations and removing practically all randomness and propose MINIROCKET in 2021. Specifically, MINIROCKET generates a set of almost deterministic kernels for time series transformations and only reserves one aggregate feature (i.e., the proportion of positive values). Benefiting from the deterministic kernels, they present four optimizations to speed up the transformations significantly. MINIROCKET achieves the same accuracy as ROCKET while with much less computational time.

Based on MINIROCKET, Tan *et al.* [28] propose MULTIROCKET in 2022, which improves on MINIROCKET by performing transformations on both original time series and corresponding first-order differences and adding multiple pooling operators. With sacrificing little computational expense, MULTIROCKET achieves significant improvement in accuracy and is competitive with the most advanced TSC algorithm HIVE-COTE 2.0 [29].

In 2022, Dempster *et al.* [30] present HYDRA for a hybrid dictionary and ROCKET architecture, which establishes a connection between random convolution kernel transformations and conventional time series dictionary-based methods. Unlike ROCKET variants, HYDRA arranges kernels in groups and assumes groups and kernels as dictionaries and patterns. In this way, HYDRA counts the best matching kernels in each group at each time point of the input time series to count repeated patterns of time series. Empirical evidence demonstrates that HYDRA is fast and accurate in TSC and can be further optimized in combination with ROCKET variants.

III. METHODOLOGY

A. Framework Overview

In this work, we propose a time series dictionary-based approach to comprehensively profile IoT devices' network behaviors and develop an efficient system for device identification, namely IOTPROFILE, which involves extracting simple packet-level features as time series, incurring minimal computational overhead while capturing distinctive and recurring patterns to characterize the network behavior of IoT devices over a period of time. By profiling the simplicity nature of IoT devices, IOTPROFILE achieves accurate and efficient identification to meet practical requirements. As illustrated in Fig. 1, the workflow of IOTPROFILE can be mainly divided into the following steps.

Multi-Channel Time Series Organization. IOTPROFILE intercepts traffic from IoT networks, aggregates packets into 2-tuple (i.e., source and destination IP addresses) network flows, and extracts packet length sequences with directions as time series. Following the shuffle&split organization scheme, IOTPROFILE maps these univariate time series into different time windows, subsequently shuffling and splitting them based on a given threshold to represent the network behaviors of IoT devices within each time window via multiple multi-channel time series. The details of multi-channel time series organization are presented in Section III-B.

Random Convolutional Kernel Transformation. With these multi-channel time series, we perform random convolutional kernel transformations on each channel in two ways to respectively profile IoT devices' network behaviors from certain perspectives. Specifically, discrete kernels capture discriminative patterns at different scales, while kernels organized in groups count repeated patterns based on time series dictionary. The details of random convolution kernel transformation are described in Section III-C.

Cross-Flow Feature Aggregation. We aggregates flow-level features to comprehensively profile the communication patterns of IoT devices over a period of time. For both

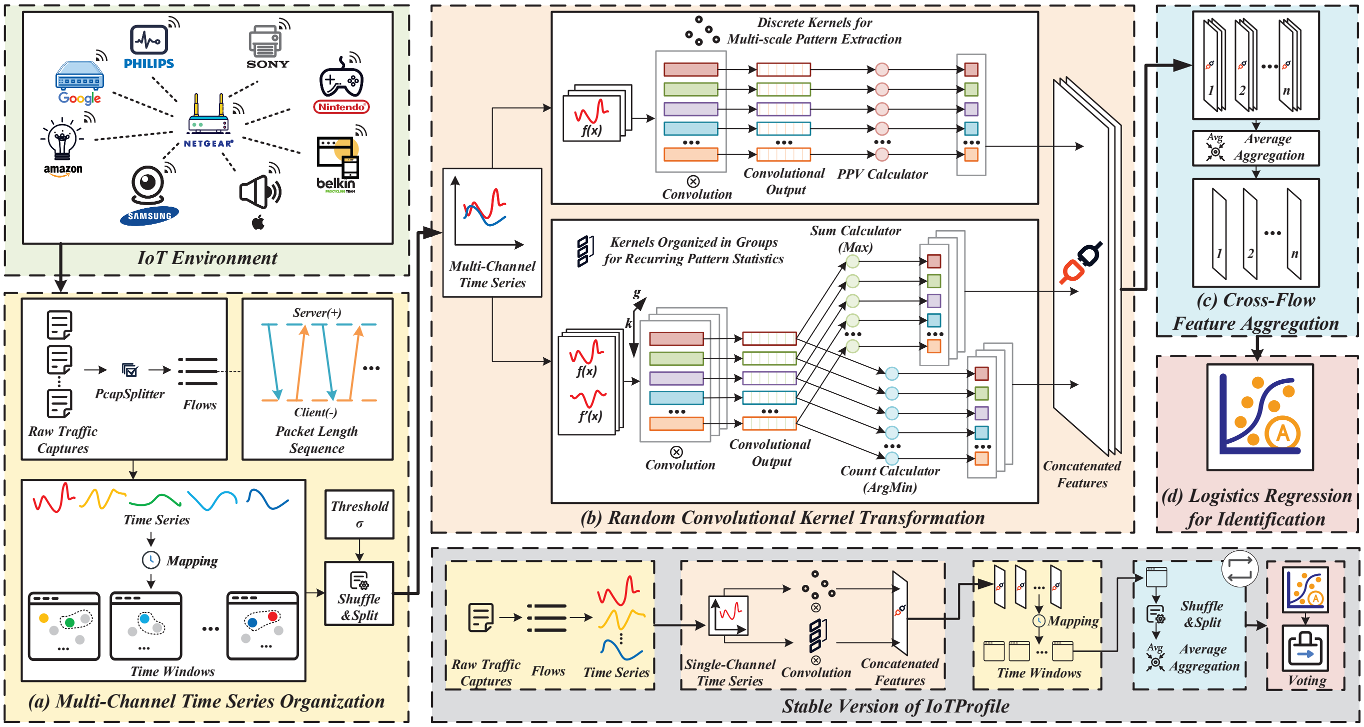

PHILIPS
SONY
Google
Nintendo
NETGEAR
amazon
SAMSUNG
belkin
IoT Environment
Raw Traffic Captures
PcapSplitter
Flows
Server(+)
Client(-)
Packet Length Sequence
Time Series
Mapping
Time Windows
Threshold σ
Shuffle &Split
(a) Multi-Channel Time Series Organization
Multi-Channel Time Series
Discrete Kernels for Multi-scale Pattern Extraction
f(x)
Convolution
Convolutional Output
PPV Calculator
Kernels Organized in Groups for Recurring Pattern Statistics
Sum Calculator (Max)
g
k
f(x)
f'(x)
Count Calculator (ArgMin)
Concatenated Features
(b) Random Convolutional Kernel Transformation
1
2
n
Avg
Average Aggregation
(c) Cross-Flow Feature Aggregation
(d) Logistics Regression for Identification
Single-Channel Time Series
Voting
Stable Version of IoTProfile

Fig. 1: The overall framework of IoTProfile contains four components. (a) Multi-Channel Time Series Organization; (b) Random Convolutional Kernel Transformation; (c) Cross-Flow Feature Aggregation; (d) Logistic Regression for Identification. In addition, we also provide a stable version of IoTProfile at the bottom right corner.

discriminative patterns and recurring pattern statistics, we calculate the average values of multi-channels per dimension as the final representations for identification. The details of cross-flow feature aggregation are described in Section III-D.

Logistics Regression for Identification. With a well-trained logistic regression classifier, IoTProfile achieves satisfactory performance in device identification. Furthermore, We provide a stable version of IoTProfile, denoted as IoTProfile$_{\text{stable}}$. As depicted in the bottom right corner of Fig. 1, IoTProfile$_{\text{stable}}$ repeats the shuffle&split procedure to enrich the representation of IoT devices' network behaviours within each time window and obtain the final IoT device identification results by a majority vote. In this context, the workflow is adjusted to improve efficiency. We will describe the details of logistic regression for identification and its stable version in Section III-E.

B. Mutli-Channel Time Series Organization

IoTProfile first captures traffic from IoT networks, where packets from different flows generated by various devices are mixed. We reorganize discrete packets into network flows and extracts the packet-level feature, i.e., packet length with direction, to profile fine-grained network behaviors of IoT devices at flow level. In each flow, the lengths of uplink and downlink packets are set as positive and negative to indicate the directions. Notably, we define each network flow as a series of packets with the same 2-tuple in this paper, which is different from previous studies [31]–[33]. Both TCP and UDP flows are utilized for subsequent IoT device identification. We take Amazon Echo as an example and randomly sample a specific flow to illustrate the interactions in Fig. 2.

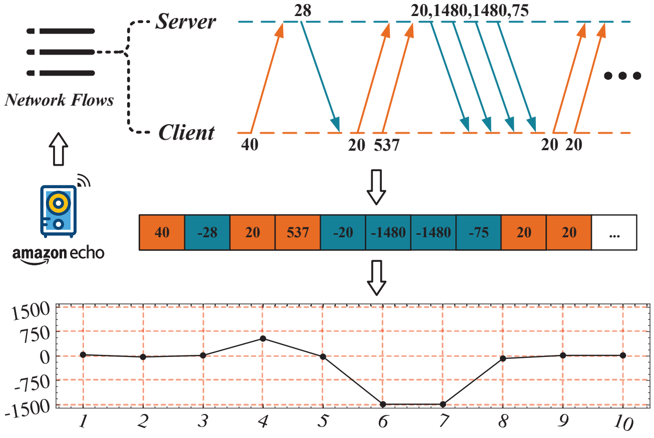


Fig. 2: Packet-level interactions (i.e., packet lengths with directions) for network flows as time series.

Due to the voluminous nature of IoT devices, even heterogeneous IoT devices of different types and vendors may generate similar network traffic. Identifying IoT devices at flow level could lead to significant misclassification and unsatisfactory performance. Similar to previous studies, we comprehensively consider the network flows over a period of time to profile the network behaviors of IoT devices. However, the activity levels of heterogeneous IoT devices are significantly different.

Even identical devices' activities vary in different periods. To balance the network flows of different IoT devices within each time window, instead of random sampling a fixed-size set of network flows, we design a shuffle&split organization scheme. The scheme first models per-packet feature sequences as time series and maps them into different time windows. Then it shuffles the time series within each time window and splits them into multiple instances[1] with a threshold ϵ. For those instances with less than ϵ series, we perform sampling with replacement to complement them. The motivations are two-fold. First, the time windows with different numbers of network flows are split and structured into multiple instances with a fixed threshold ϵ, which effectively solves the data imbalance problem. Second, the random combination of the time series within each time window enriches IoT devices' network behavior representations. For the time series within each time window, we organize them as multi-channel representations other than concatenate them to avoid introducing artificial patterns.

Specifically, for each 2-tuple network flow, we initially extract the packet length sequence and then divide it into univariate time series of a fixed length L with zero padding. Subsequently, we shuffle these univariate time series within each time window and split them with a threshold ϵ into multi-channel time series serving as multiple instances to represent the network behaviors of IoT devices. Given time window t and IoT device D, such an instance can be denoted as

$$F_D^t = [f_D^{t,1}; \dots; f_D^{t,\epsilon}] \quad = \begin{bmatrix} p_D^{t,1,1} & \dots & p_D^{t,1,L} \\ \vdots & \ddots & \vdots \\ p_D^{t,\epsilon,1} & \dots & p_D^{t,\epsilon,L} \end{bmatrix}, \qquad (1)$$

where $p_D^{t,i,j}$ is the j^{th} packet length of i^{th} flow generated by device D in time window t, $p_D^{t,i,j} > 0$ indicates uplink, otherwise downlink. With the well-organized multi-channel time series, further transformations and analysis can be performed to realize accurate and efficient IoT device identification.

C. Random Convolutional Kernel Transformation

To comprehensively profile the network behaviors of IoT devices over a period of time, we perform random convolutional kernel transformations on the multi-channel time series to simultaneously capture discriminative patterns and count repeated patterns. Notably, all transformations are performed at flow level in parallel and independently, enabling IoTProfile to be highly efficient and applicable in practical IoT network environments.

1) Multi-Scale Pattern Extraction: To capture the discriminative communication patterns of IoT devices at different scales, we generate a set of random convolutional kernels with different dilations and biases almost determinately, as recommended in [27].

The attributes of these discrete kernels are as follows:

- *Kernel Length:* Kernel length is fixed to 9 for all kernels.
- *Kernel Weights:* Kernel weights are restricted to two values α and β with fixed length 9. There are in total $2^9 = 512$ possible two-valued kernels for selection, which can be further categorized by the number of weights with the value of α in each kernel. Specifically, we set $\alpha = -1$, $\beta = 2$, and select the subset of kernels with six values of α and three values of β. $C_9^6 = 84$ kernels are finally determined for subsequent transformations.
- *Dilation Rate:* To expand the reception field size of convolutional kernels, dilations are utilized to incorporate global context without introducing extra parameters or computation costs. Each type of kernel has the same set of dilation rates in the range $\{\lfloor 2^0 \rfloor, \dots, \lfloor 2^{\max} \rfloor\}$ with the exponent d drawn from a uniform distribution $\mathcal{U}(0, \max = \log_2 \frac{|\mathrm{x}|-1}{|\mathrm{w}|-1})$ where $|\mathrm{x}|$ is the input time series length, and $|\mathrm{w}|$ is the kernel length (i.e., 9).
- *Bias:* Bias values are drawn from the convolution output of a randomly selected training example X. Given a kernel with weight W and dilation d, the output of sample X is denoted as $W_d * X$. We take the quantiles of $W_d * X$ as bias values.
- *Padding:* Padding operation is alternated with the combination of kernels and dilations. Overall, half kernels enable zero padding, and half are disabled.
- *Kernel Number:* Kernel number is set to the multiple of 84 nearest to 10,000 (i.e., 9,996[2]).

With the generated convolutional kernels, we perform convolutional kernel transformations on the input time series to extract multi-scale patterns. Given time series X and kernel W with dilation d and bias b, the convolution output Z at position i is denoted as:

$$Z_i = W_d * X_i = \left(\sum_{j=0}^{|\mathrm{w}|-1} X_{(i+j\times d)} \times W_j \right) + b. \qquad (2)$$

For each convolution output Z, we compute the Proportion of Positive Value (PPV) as the feature:

$$\mathbf{PPV}(Z) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(Z_i > 0), \qquad (3)$$

where n in the length of convolution output Z, and $\mathbb{I}(\cdot)$ is an indicator function such that $\mathbb{I} = 1$ if the condition is true, otherwise $\mathbb{I} = 0$.

With 10K discrete convolutional kernels with various weights and dilations to transform input time series, we compute corresponding PPVs of 10K convolution outputs to represent the multi-scale patterns of input time series.

2) Recurring Pattern Statistics: The multi-scale patterns are discriminative in discriminating IoT devices of different types or different vendors, however, which may fail in fine-grained device instance identification. The reason behind this is that different device instances of identical type or identical vendor may share similar or the same communication patterns.

[1] During the training phase, the split instances are all separately utilized for training. During the evaluation phase, a simple majority voting is utilized to predict the class of split instances within the same time window.

[2] For simplicity, we refer the number of kernels, 9,996, to the nearest 10,000 or 10K throughout the paper.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3305585

TABLE I: The details of MINIROCKET and HYDRA

	MINIROCKET	HYDRA ($k \times g = 512$)
length	9	$\sim$
weight	$\{-1, 2\}$	$\mathcal{N}(0, 1)$
number	10K	$k \times g \times \lvert d\rvert$
dilation rate	$max = log_2 \frac{\lvert \mathrm{x}\rvert - 1}{\lvert \mathrm{w}\rvert - 1}$	$\sim$
bias	from convolution output	not used
padding	alternatively enabled	always enabled
operator	PPV	$\max + \arg\min$
dimension	10K	$2 \times k \times g \times \lvert d\rvert$

Thus merely extracting representative communication patterns is insufficient to identify specific IoT device instances. IoT devices' network behaviors can be divided into spontaneous and manually-triggered communications. The former like Network Time Protocol (NTP), used to synchronize time with remote servers, is of little significance in instance identification. The key idea to realize accurate IoT device instance identification is to perform statistical analysis of manual-triggered communications by counting discriminative patterns in time series.

Specifically, we also perform random convolutional kernel transformations to count recurring patterns. The difference is that the kernels in recurring pattern statistics are organized in groups. Each group acts as a dictionary, and each kernel in the group acts as a pattern in the dictionary.

The attributes of these kernels organized in groups are as follows:

- *Kernel Length:* Kernel length remains the same as III-C1.
- *Kernel Weights:* Kernel weights are drawn from a Gaussian distribution $\mathcal{N}(0, 1)$.
- *Dilation Rate:* Dilation rate remains the same as III-C1.
- *Padding:* Padding operation is enabled for all kernels.
- *Kernel Number Per Group and Group Number:* The product of kernel number per group k and group number g is fixed to 512 (i.e., $k \times g = 512$).

It is notable that there are k kernels per group, g groups per dilation rate, $|d|$ dilation rates, and $k \times g \times |d|$ kernels in total. With the well-organized kernels for transformations, both the kernels with the maximum and minimum convolutional output values per group and per dilation are counted in the following two different forms:

- *Soft Counting (max):* Soft counting sums the output of the kernel with the maximum convolutional output at each time point.
- *Hard Counting (argmin):* Hard counting increments the count of the kernel with the minimum convolutional output at each time point.

The *max/min* competition is restricted to the kernels per group per dilation. With $k \times g \times |d|$ kernels organized into $|d|$ dilation rates, g groups per dilation rate, and k kernels per group, we perform soft counting and hard counting on the maximum and minimum convolution output, respectively, and finally, output $(2 \times k \times g \times |d|)$-dimension vectors to form the frequency counts of repeated patterns (i.e., most similar kernels) of input time series. We generate these kernels as recommended in MINIROCKET [27] and HYDRA [30]. The details are presented in Table I.

D. Cross-Flow Feature Aggregation

Through two types of random convolutional transformations, we have obtained the flow-level features in each channel, which consists of two parts: 1) discriminative communication patterns represented by PPVs in each channel; and 2) recurring communication pattern statistics represented by soft/hard counting in each channel. Since we aim to realize efficient IoT device identification via network behavior analysis at the granularity of specific time intervals, we further perform cross-flow feature aggregation on the multi-channel flow-level features.

For both discriminative patterns denoted by PPV values and recurring pattern statistics denoted by soft/hard countings, we can simply calculate the average values of multi-channels per dimension since we have fixed the channels (i.e., flows per time window) and dimensions (packets per flow) of the inputs. In this way, we aggregate multi-channel features within each time window to comprehensively profile IoT devices' network behaviors over a period of time.

E. Logistic Regression for Identification

As recommended by ROCKETs [26]–[28] and Hydra [30], we use a simple linear classifier, i.e., logistic regression with stochastic gradient descent, to identify IoT devices for our large datasets, which enables IOTPROFILE to efficiently mine valuable information from high-dimension features generated by large numbers of random convolutional kernels. Due to the memory limit, we integrate the convolutional transformations into mini-batch training of logistic regression for fast convergence.

To remove randomness and ensure reliable performance, we provide a stable version of IOTPROFILE with additional computational expense. Specifically, we repeat the shuffle&split operations within each time window. Therefore the final IoT device identification results are by a majority vote of the classifications of all instances. Notably, the stable version's workflow is different from the original variant. For the stable version of IOTPROFILE, the transformations are first performed on all time series to extract flow-level features for subsequent shuffling, splitting and aggregation to avoid repetitive transformations. As illustrated at the bottom right corner of Fig. 1, convolutional transformations, cross-flow feature aggregations, and model updates are no longer integrated into each mini-batch stochastic gradient descent.

IV. EVALUATION

A. Dataset and Setups

1) Dataset: To comprehensively validate the accuracy and efficiency of IOTPROFILE, we select two datasets: UNSW dataset and UNB dataset, both of which are publicly available and extensively applied for the evaluations of IoT device identification.

UNSW dataset [10] is released in 2018 by researchers from University of New South Wales (UNSW). They build a smart environment with 28 unique IoT devices of various types (e.g., cameras, sensors, healthcare equipment) and collect the network traces of the IoT devices across 26 weeks. Currently, 20-day network trace data are publicly available, consisting of 20 PCAP files with a total size of 11.70 GB.

UNB dataset [34] is collected by Canadian Institute for Cybersecurity at University of New Brunswick (UNB), which is used to analzyze security properties of IoT devices in smart home environments over a period of 30 days during idle stage. The IoT devices in UNB dataset is mainly divided into four categories (i.e., audio, camera, home automation and others). The dataset consists of 30 PCAP files with a total size of 6.06 GB.

Our proposed IOTPROFILE identifies IoT devices at the granularity of a specific time window. Therefore, with different time windows, we have different numbers of 2-tuple flows and instances. The detailed description of UNSW and UNB datasets are presented in Table II. Since many previous studies set the time window as 30 minutes, we follow this setting in IOTPROFILE. Notaly, all IoT devices with less than 10 instances are filtered in both datasets. The details of the specific IoT devices and corresponding instance numbers of UNSW and UNB datasets are presented in Table III and Table IV.

2) Setups: IOTPROFILE is basically implemented in C++ and Python. During multi-channel time series organization, we exploit PcapPlusPlus (version 22.05), a multi-platform C++ library with easy-to-use APIs for traffic analysis, to split raw traffic captures into 2-tuple flows and parse per-packet features. Then we leverage Python (version 3.9.13), and Pytorch [35] (version 1.13.0) to perform convolutional transformation, cross-channel aggregation, and logistic regression for IoT device identification.

In IOTPROFILE, we identify IoT devices by profiling the network behaviors over a period of time. Therefore one instance is defined as the traffic generated by a device during a time window. We set the time window as 30 minutes. The total number of instances is 9,856 in UNSW dataset, and 183,065 in UNB dataset. We split all two datasets into training sets, validation sets, and testing sets with a ratio of 5:2:3. It is noteworthy that the activities of different IoT devices vary, causing severe data imbalance issues. We utilize scikit-learn [36] (version 1.1.2) to conduct Synthetic Minority Over-sampling Technique (SMOTE) on the training dataset to balance the class distribution.

TABLE II: Dataset Descriptions

Datasets	Devices	Flows	Instances	Time Window
UNSW	19	200,279	22,356	10 Minutes
		129,213	13,351	20 Minutes
		101,631	9,856	30 Minutes
UNB	28	352,458	44,153	10 Minutes
		228,219	24,726	20 Minutes
		183,065	17,369	30 Minutes

TABLE III: IoT Device Descriptions of UNSW Dataset

Device, Instances
Smart Things 0.78K, Amazon Echo 0.89K, Netatmo Welcome 0.73K,
TP-Link Camera 0.60K, Samsung SmartCam 0.86K, Dropcam 20,
Insteon Camera 0.64K, Anonymous 0.39K, Withings Monitor 0.74K
Belkin Wemo Switch 0.40K, TP-LINK Plug 68, iHome 0.86K,
Belkin Wemo Motion Sensor 22, NEST Protect Smoke Alarm 0.72K,
Blipcare Blood Pressure Meter 0.48K, Withings Sleep Sensor 0.61K,
LiFX Bulbs 0.50K, Triby Speaker 0.43K, PIX-STAR Photo-frame 0.14K

TABLE IV: IoT Device Descriptions of UNB Dataset

Device, Instances
Amazon Dot 1.28K, Amazon Spot 0.66K, Amazon Studio 0.63K,
Google Nest 0.64K, Sonos One Speaker 0.66K, AMCREST Camera 0.61K,
Arlo Base Station 0.59K, Arlo Q Camera 0.19K, Borun/Sichuan Camera 0.66K,
D-Link Mini Camera 0.10K, HeimVision Smart WiFi Camera 0.37K,
Home Eye Camera 0.54K, Luohe Cam Dog 0.52K,
Netatmo Camera 0.55K, SIMCAM 1S 0.61K Amazon Plug 0.33K,
Atomi Coffee Maker 0.61K, Eufy HomeBase 2 0.56K,
Gosund ESP Socket/Plug 2.92K, HeimVision SmartLife Radio/Lamp 0.61K,
Philips Hue 0.61K, Ring Base Station AC 0.22K, iRobot Roomba 0.39K,
Smart Board 0.09K, Teckin Plug 1.09K, Yutron Plug 1.09K
D-Link Water Sensor 14, Netatmo Weather Station 0.25K

In this work, we assess the model performance utilizing four typical metrics: accuracy (AC), precision (PR), recall (RC) and F1 score (F1). The four metrics are defined upon four fundamental concepts, including True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN), where TP denotes the number of correctly classified positive samples, TN denotes the number of correctly classified negative samples, FP denotes the number of incorrectly classified negative samples, FN denotes the number of incorrectly classified positive samples. In the context of binary classification, AC, PR, RC and F1 are defined as follows:

$$\mathrm{AC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{4}$$

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{5}$$

$$\mathrm{RC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{6}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{PR} \times \mathrm{RC}}{\mathrm{PR} + \mathrm{RC}}. \tag{7}$$

To extend these evaluation metrics to multi-class classification scenario, we assume a one-vs-all classification formulation to calculate $\mathrm{AC_i}$, $\mathrm{PR_i}$, $\mathrm{RC_i}$ and $\mathrm{F1_i}$ for each class i (k classes in total). To circumvent potential bias due to the imbalance in data distribution across different classes, we employ Macro Average to calculate the mean value of the aforementioned evaluation metrics as follows:

$$\mathrm{PR} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{PR}_i, \tag{8}$$

$$\mathrm{RC} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{RC}_i, \tag{9}$$

$$\mathrm{F1} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{F1}_i. \tag{10}$$

In this manner, we ensure a comprehensive and unbiased evaluation in multi-class classification scenario. All experiments are simulated on a DELL server with 24 cores Intel Xeon 6240R CPU @2.40GHz, Ubuntu 20.04, 64 GB memory, and 2×NVIDIA Quadro RTX 5000 GPU with 10-fold cross-validation to compute the average and standard variance of metrics to ensure generality.

TABLE V: Effectiveness Evaluation on both Datasets.

Dataset	Metric	IoTDevID	ByteIoT	IoTProfile	IoTProfile$_{\text{stable}}$
UNSW	AC	0.9714±0.0031	**0.9991±0.0008**	0.9990±0.0008	0.9989±0.0004
	PR	0.9629±0.0027	**0.9999±0.0001**	0.9992±0.0006	0.9992±0.0006
	RC	0.8692±0.0048	0.9697±0.0262	**0.9976±0.0023**	0.9974±0.0015
	F1	0.9121±0.0023	0.9696±0.0263	0.9984±0.0012	**0.9985±0.0006**
UNB	AC	0.9080±0.0059	0.9814±0.0016	0.9839±0.0047	**0.9867±0.0045**
	PR	0.8795±0.0042	0.9803±0.0020	0.9817±0.0061	**0.9837±0.0060**
	RC	0.9186±0.0084	0.9726±0.0015	0.9733±0.0053	**0.9745±0.0029**
	F1	0.8787±0.0036	0.9744±0.0017	0.9765±0.0019	**0.9780±0.0013**

B. Comparison Baselines

We select two recently proposed machine learning-based methods as baselines to compare with IoTProfile from two perspectives (i.e., identification effectiveness and time overhead).

IoTDevID [15] is a packet-level IoT device identification method that systematically studies packet-level features with xverse package to eliminate redundant features and a genetic algorithm to select the most discriminative feature set. With a well-designed aggregation algorithm considering both MAC addresses and model predictions, IoTDevID shows robust performances that can generalize across datasets. Since IoTDevID identifies IoT devices at the granularity of packet level, for fair comparisons, we map packets into different time windows and aggregate the packets' predictions within each time window by voting to obtain the labels of time window instances.

ByteIoT [23] is a practical IoT device identification system based on the frequency distribution of bidirectional packet lengths, which utilizes the k-nearest neighbors algorithm to classify IoT devices by calculating the Hellinger distances between packet distribution features of instances. Like IoTProfile, ByteIoT also identifies IoT devices at the granularity of time window level. It counts packet lengths of each device per time window in feature extraction, similar to ByteIoT in both concerned features and frequency counting operation, except that ByteIoT counts unique packet lengths. In contrast, Our proposed IoTProfile counts discriminative patterns (i.e., packet length sequences).

C. Performance Evaluation

In this subsection, we comprehensively evaluate the performance of our proposed IoTProfile in terms of both identification effectiveness and time overhead by comparing with two state-of-the-art methods on UNSW and UNB datasets.

1) Identification Effectiveness: As shown in Table III-IV, the instance amounts of different IoT devices vary significantly in both UNSW and UNB datasets. Merely using accuracy metric is inappropriate on such highly imbalanced datasets. To this end, we select four metrics (i.e., accuracy, macro-average precision, macro recall, and macro-average f1 score) to exhibit the identification performances intuitively. Table V shows the results of three methods on both datasets in cross-validations.

On both UNSW and UNB datasets, IoTProfile significantly outperforms IoTDevID and slightly outperforms ByteIoT in almost all metrics. In addition, IoTProfile$_{\text{stable}}$, the stable version of IoTProfile achieves the best performance since it repeats shuffle&split operation and further enriches the network behavior representations.

TABLE VI: Time Overhead Evaluation on Full UNSW Dataset

Phase	IoTDevID	ByteIoT	IoTProfile
Feature Extraction	> 6 h	686.06 s	117.22 s
Model Training	31.47 s	$\sim$	78.82 s
Model Execution	10.28 s	731.78 s	19.57 s
Total Duration	> 6 h	1417.84 s	215.61 s

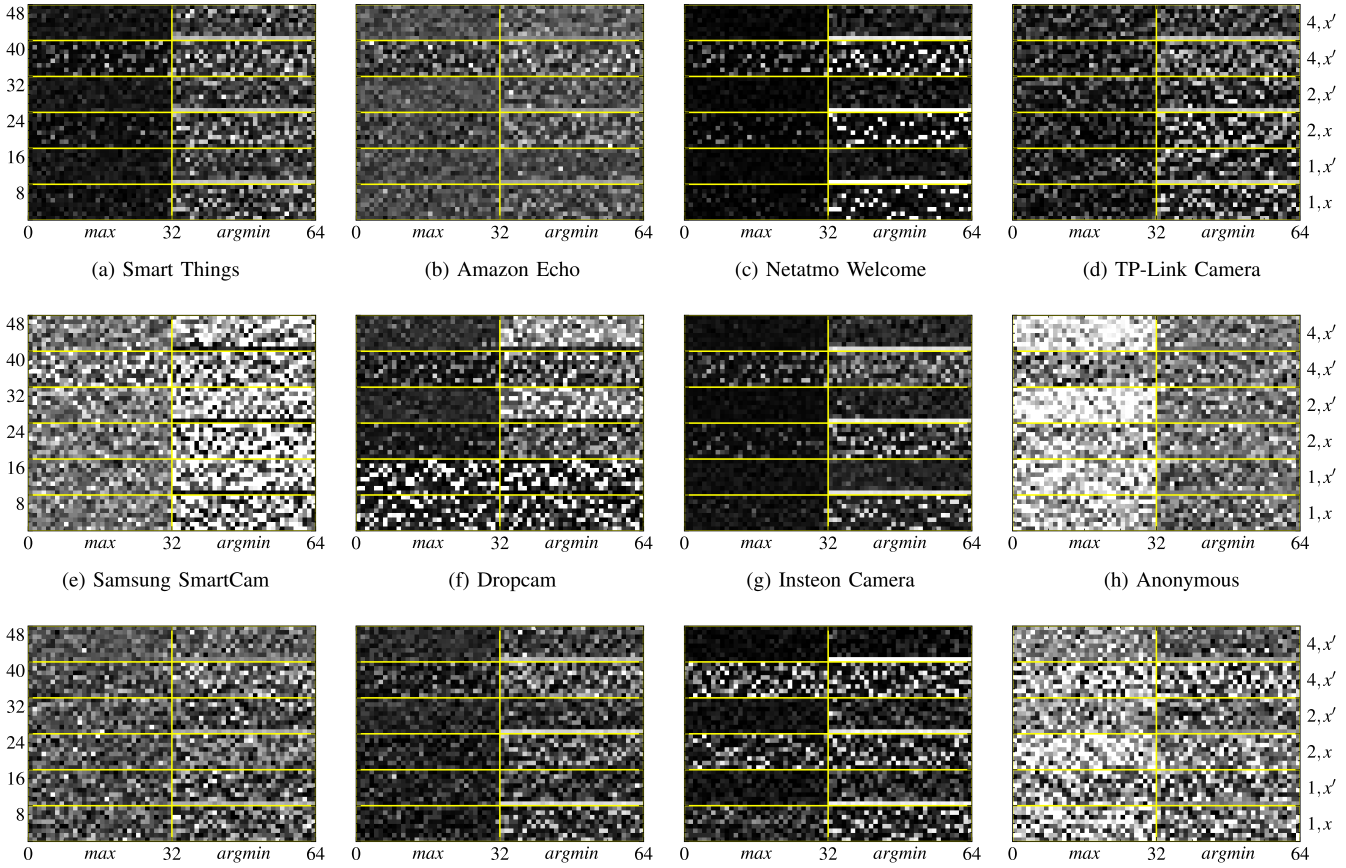
We observe that packet-level identification is error-prone since heterogeneous IoT devices tend to generate partially similar traffic. IoTDevID performs packet-level identification by designing an aggregation to take both the device MAC addresses and model's predictions into consideration, which effectively improves the identification performance compared with individual packet approach. However, posing label leakage concerns. Moreover, ByteIoT can also achieve comparable performance compared with IoTProfile on some metrics. The insight of ByteIoT is similar to IoTProfile. ByteIoT considers the unique packet length as patterns and IoTProfile considers the discriminative packet length subsequence to form the frequency counts respectively. The experimental results demonstrate that dictionary-based methods profile the simplicity of IoT devices, and the frequency counts can effectively discriminate IoT devices.

2) Time Overhead: To meet the practical requirements, we evaluate the time overhead of three methods in different phases: feature extraction, model training, and model execution. Specifically, this evaluation on time overhead is performed on 20-day traffic captures of UNSW dataset, where 50% for model training, 20% for model validation, and 30% for model execution. The time overheads of the three methods are presented in Table VI.

In sum, IoTProfile is significantly faster than both IoTDevID and ByteIoT in all phases. Though IoTProfile performs plenty of convolutional transformations on the packet length sequences, all the transformations are performed in parallel and independently. Moreover, IoTProfile only utilizes a simple linear model (i.e., logistic regression) to identify IoT devices, which does not require much computation overhead. The two reasons make IoTProfile highly efficient to be applicable to real-world environment. ByteIoT also utilizes simple features as we do. It utilizes *k*-NN algorithm requiring no training process, and the time overhead in the model training phase is zero. We evaluate the time overhead on the full UNSW dataset. We notice that when the scale of the dataset increases, the time overhead of *k*-NN algorithm in model execution phases significantly increases. As for

TABLE VII: Ablation Study of IoTProfile on UNSW and UNB Datasets.

Dataset	Metric	IoTProfile	w/o Agg	w/o S&S	w/o Rocket	w/o Hydra
UNSW	AC	0.9990±0.0008	0.6978±0.2037	0.9422±0.0804	**0.9992±0.0004**	0.9819±0.012
	PR	0.9992±0.0006	0.9175±0.0558	0.9496±0.0754	**0.9994±0.0003**	0.9851±0.0081
	RC	**0.9976±0.0023**	0.7353±0.1788	0.9341±0.0901	0.9969±0.0026	0.9790±0.0131
	F1	**0.9984±0.0012**	0.7195±0.1959	0.9145±0.1287	0.9981±0.0015	0.9809±0.0112
UNB	AC	0.9839±0.0047	0.7429±0.0873	0.9859±0.0034	**0.9874±0.0009**	0.9726±0.0051
	PR	0.9817±0.0061	0.8715±0.0239	0.9853±0.0052	**0.9868±0.0004**	0.9675±0.0054
	RC	**0.9733±0.0053**	0.7361±0.0342	0.9647±0.0073	0.9722±0.0078	0.9512±0.0077
	F1	0.9765±0.0019	0.7464±0.0428	0.9734±0.0061	**0.9784±0.0047**	0.9579±0.0066



(a) Smart Things (b) Amazon Echo (c) Netatmo Welcome (d) TP-Link Camera

(e) Samsung SmartCam (f) Dropcam (g) Insteon Camera (h) Anonymous

(i) Withings Monitor (j) TP-link Smart Plug (k) iHome (l) Belkin Motion

Fig. 3: The feature maps extracted by time series dictionary of 12 IoT devices from UNSW dataset

IoTDevID, it extracts many features related to specific protocols (e.g., BOOTP, DNS, ICMP) for identification, which is time-consuming compared to IoTProfile and ByteIoT. Both IoTProfile and ByteIoT prove that packet lengths effectively indicate IoT devices' functionality.

It is worth noting that both ByteIoT and IoTDevID are implemented based on their open-source code. The efficiencies of the components (e.g., feature extractor, *k*-NN algorithm) in these two methods can be further optimized. However, even taking these factors into account, IoTProfile still shows superior efficiency in comparison with these baselines.

D. Ablation Analysis

To further verify the effectiveness of our innovations, we conduct a careful ablation analysis on the key components of IoTProfile. Specifically, we compare the full IoTProfile with its four variants:

- w/o S&S: w/o S&S removes shuffle&split organization scheme. It solves the data imbalance problem by randomly sampling a fixed-size set of flows within each time window as instances for subsequent classification.
- w/o Agg: w/o Agg directly classifies IoT devices at the granularity of network flow, then obtains the final IoT device identification results by a majority vote of the classification results within each time window.
- w/o Rocket: w/o Rocket removes the discrete kernels to capture the discriminative patterns at different scales. It only performs transformations with kernels organized in groups to form the frequency count of recurring communication patterns of IoT devices.
- w/o Hydra: w/o Hydra removes the kernels organized

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3305585

in groups to form the frequency count of recurring patterns. It only performs transformations with discrete kernels to capture discriminative communication patterns of IoT devices at different scales.

We present the experimental results of IoTProfile and its variants on both datasets in Table VII. As can be seen, IoTProfile achieves the best identification performance on both datasets with excellent stability. The results of IoTProfile and w/o S&S validate the effectiveness of our proposed shuffle&split organization scheme. Instead of utilizing the full data in IoTProfile, w/o S&S merely samples partial data to solve the data imbalance problem. However, the randomness makes it unstable and inapplicable to real-world environments. Additionally, comparing IoTProfile with w/o Agg, we observe that heterogeneous IoT devices may inevitably generate a few similar network flows. Therefore identifying IoT devices based solely on one single network flow is hard. With cross-flow feature aggregation, we aggregate the pattern features of multi-channel time series to profile the network behaviors over a period of time, which has a high tolerance to those abnormal or similar flows. Finally, we evaluate the multi-scale patterns extracted by the discrete kernels and recurring patterns statistics extracted by the kernels organized in groups. We can see that w/o Rocket performs better than w/o Hydra since forming the frequency count of repeated patterns by time series dictionary can effectively profile the simplicity of IoT devices. Overall, IoTProfile outperforms all variants by integrating shuffle&split organization scheme, cross-flow feature aggregation, and different random convolutional kernel transformations to capture discriminative patterns and count repeated patterns simultaneously.

In this paper, we first introduce time series dictionary for IoT device identification. The results in Table VII have demonstrated the improvements when adding Hydra. To intuitively present the frequency differences of communication pattern of different IoT devices, we here select the first 12 IoT devices in UNSW dataset as examples to visualize the recurring pattern statistics similar to malware visualization [37].

Specifically, we reserve the first 60 packets of network flows. Through transformations and aggregations, we obtain 3072-dimension feature vectors for the frequency count of repeated communication patterns of IoT devices. We reshape and map the feature vectors to grayscale images in Fig. 3, where X-axis is defined as groups, and Y-axis is defined as kernels. We have 64 groups and 8 kernels per group, where half of the 64 groups are for the original time series, and the rest are for the first-order difference. For each group's convolution outputs, we perform hard and soft counting. Therefore, pixels ranging from 0 to 32 in X-axis represent soft counting for the maximums, and those ranging from 32 to 64 in X-axis represent hard counting for the minimums. On this basis, the statistics of different kernels on the original or the first-order difference are stacked along Y-axis. For example, pixels ranging from 0 to 8 in Y-axis represent the kernel groups with dilation rate $d = 1$ performed on the original time series. The dilation rate and time series type are marked on the right side of Fig. 3. The light pixels represent the kernels with a better match with the input communication patterns, and the dark pixels represent the kernels with a poorer match. Through the visualization, we can see that the recurring pattern statistics of different IoT devices differ markedly.

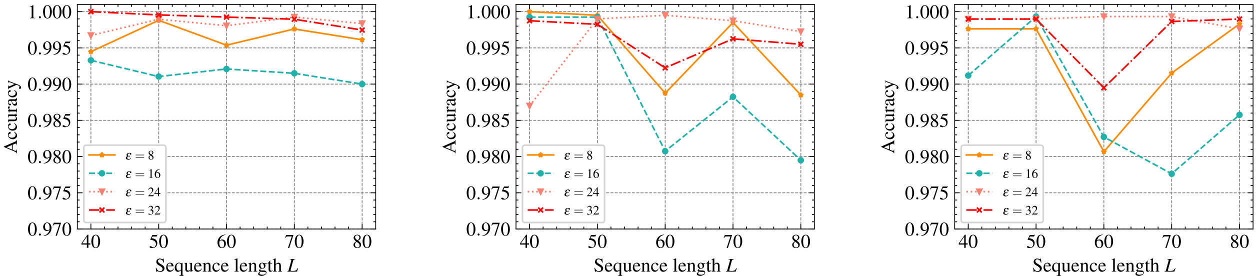
E. Hyper-parameter Tunning

To explore the effect of three critical hyper-parameters in IoTProfile, we perform hyper-parameter sensitivity analysis on time window T, sequence length L, and threshold ϵ. Specifically, we alter the values of T, L, and ϵ to see how they affect the accuracy of IoTProfile. We vary T from 10 to 30 (minutes) with interval 10, L from 40 to 80 with interval 10, and ϵ from 8 to 32 with interval 8. The IoT device identification results on UNSW and UNB datasets are illustrated in Fig. 4 and Fig. 5.

Time window T. We identify IoT devices by profiling their network behaviors over a period of time. The communication patterns of IoT devices within a short time window may sharply fluctuate, while a long time window will certainly introduce latency for identification. We observe that IoTProfile performs better with a short time window on UNSW dataset. While on UNB dataset performs better with a longer time window. Specifically, on UNSW dataset, IoTProfile performs more stably when the time window T is set as 10 minutes, which is valuable since a shorter time window means more instances for classification. However, it achieves the best identification accuracy with $T = 30$ minutes without consideration of other hyper-parameters. The reason may be that IoTProfile utilizes time series dictionary to count recurring communications patterns. A longer time window is more sufficient to incorporate the repetitive network behaviors of IoT devices. On UNB datasets, IoTProfile performs more stably when the time window T is set as 20 or 30 minutes. The variance is primarily attributable to the duration difference of network flows. Following previous studies, we set the time window T as 30 minutes on both datasets for fair comparisons.

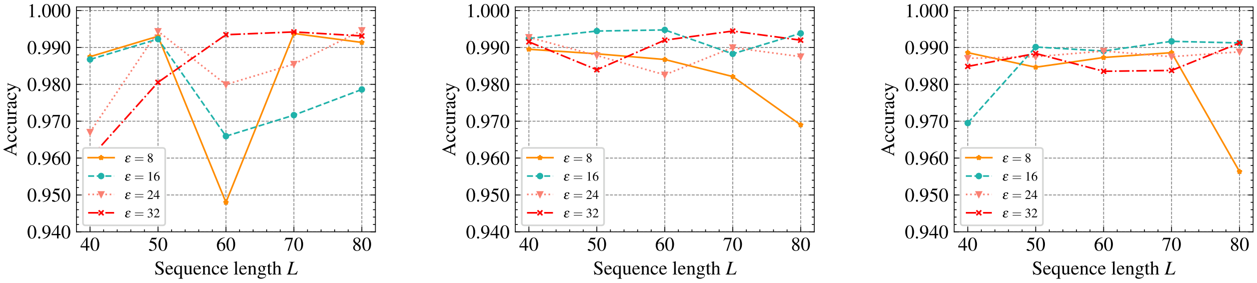
Sequence length L. We extract packet length sequences as time series to profile the network behaviors of IoT devices for identification. With the increase of sequence length L, the general accuracy trend is to increase first and then decrease in all settings on both datasets, which is easily explained. If we retrieve longer packet sequences as representations, more complex communication patterns at different scales can be extracted for more accurate identification. However, network traffic is irregular, and there are large amounts of short network flows. To reserve more packets, we have to filter those short network flows or normalize them by padding zeros, which either cause information loss or introduce noise in profiling. Therefore, we must set a moderate sequence length to incorporate more information and avoid introducing noise. However, in some cases, the performance first decreases and then increases with the increase of sequence length, which seems counterintuitive. We find the reason by examining the average packet number of the network flows. IoTProfile utilizes 2-tuple network flows, aggregating different 5-tuple flows. An inappropriate sequence length setting may cause performance degradation due to the introduction of meaningless communications (e.g., three-Way handshake And four-Way wavehand

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3305585



(a) $T = 10$ (b) $T = 20$ (c) $T = 30$

Fig. 4: The accuracy of IoTProfile with different hyper-parameters on UNSW dataset.



(a) $T = 10$ (b) $T = 20$ (c) $T = 30$

Fig. 5: The accuracy of IoTProfile with different hyper-parameters on UNB dataset.

processes). In those cases, if the sequence length continues to increase, the effects of those meaningless communications are mitigated, and the performance of IoTProfile becomes better. In this work, we reserve the first 60 packets of each network flow with zero padding on both datasets.

Threshold ϵ. The numbers of flows in different time windows vary significantly. To solve the data imbalance problem, we propose a shuffle&split organization scheme. Specifically, we shuffle the network flows of each IoT device within each time window and split them into different instances with a preset threshold ϵ. Benefiting from our proposed shuffle&split organization scheme for the data imbalance problem, IoTProfile generalizes well with different time window settings, which is practical to meet real-world demands. From Fig. 4 and Fig. 5, we can observe that in almost all settings, the accuracy also shows a trend that first increases and then decreases, similar to the cases in time window tuning. It is probably because that IoTProfile forms the frequency count of repeated communication patterns of each flow and aggregates the results as instances to profile the network behaviors over a period of time. The instances with more flows are more comprehensive for IoT device identification. However, for those time windows with very few flows, the shuffle&split operation has less practical effect. Though we perform sampling with replacement, a too large threshold ϵ is inappropriate and may affect the performance. To this end, we set threshold ϵ as 24 to form instances for evaluations in two datasets.

V. Discussion

IoTProfile achieves superior accuracy and efficiency in IoT device identification, yet it still exposes some flaws. In this section, we discuss its several limitations and present potential solutions.

A. Decision Efficiency Optimization

We have validated the high efficiency of IoTProfile in both feature extraction and model execution compared with two previous studies in Section IV-C2, however, which can be further optimized. Recently, the emergence of programmable switches provides essential hardware support and makes it possible to deploy AI models directly on the network data plane, which can effectively improve efficiency and meet the performance requirements of the high-speed big data environment. It is worth considering integrating IoTProfile into an intelligent network data plane to further improve the effectiveness of our work.

B. Data Organization Refinement

IoTProfile splits packets into 2-tuple TCP and UDP flows as time series to model the network behaviors of IoT devices. In fact, the data organization of IoTProfile can be further refined from the following two aspects. (1) TCP and UDP flows are not separated in our work, where UDP flows also consist of DNS and ICMP queries. We can conduct a more fine-grained analysis to separately process these network flows with different functionality in the future. (2) IoTProfile only utilizes packet length sequences as time series. Though these time series within one time window are organized as multi-channel time series, the transformations are performed on uni-variate time series. More per-packet attributes (e.g., time intervals) without extra computational expense can be added to enrich the data as multi-variate time series.

C. System Scalability Enhancement

In IoTProfile, we adopt a supervised learning algorithm, logistic regression, to classify IoT devices, which performs well on the experimental datasets. However, it may be impractical when it comes to real-world network environments. On the one hand, supervised learning algorithms require large amounts of labeled data, and the time-consuming collection process is unpractical. On the other hand, IoTProfile cannot recognize unseen IoT devices, which is inapplicable in the continuously changing network environment. To enhance the system scalability, we can reserve the feature extraction modules and substitute the logistic regression model of IoTProfile by a semi-supervised algorithm, which is less dependent on the training data and can effectively handle unseen IoT devices.

D. Advanced Analysis Methods Introduction

IoTProfile is the first work that identifies IoT devices via network behavior analysis based on time series dictionary. Due to the restricted computational abilities and hardware limitations, IoT devices' network functionalities are relatively simple compared with non-IoT devices. Considering this nature, we introduce Rocket and Hydra to perform random convolutional kernel transformations to capture discriminative patterns and form the frequency counts of repeated patterns over a period of time, which has achieved satisfactory performance in IoT device identification. In future work, we can further categorize fine-grained network behaviors of IoT devices and introduce more advanced analysis methods in TSC to improve identification accuracy.

VI. Conclusion

In this paper, we propose an efficient IoT device identification system via network behavior analysis based on time series dictionary, namely IoTProfile. Precisely, we first extract per-packet attributes, i.e., packet length sequence, and design a shuffle&split organization scheme to solve the data imbalance problem and organize packet length sequences as multi-channel time series. Then we perform random convolutional kernel transformations at flow level in parallel to capture the discriminative patterns and count repeated patterns. We further aggregate flow-level features to profile the network behaviors over a period of time. With a well-trained logistic regression classifier, IoTProfile achieves superior performance in accuracy and efficiency.

A key contribution of our work is that we consider the simplicity nature of IoT devices and introduce time series dictionary to form the frequency counts of repeated communication patterns, which has been proved to be effective in profiling the network behaviors of IoT devices. Additionally, we design a shuffle&split organization scheme to solve the data imbalance problem, which is frequently overlooked in previous studies. Through extensive experiments, we demonstrate that IoTProfile is an accurate IoT device identification system with high efficiency and can be further explored for more fine-grained IoT network traffic analysis.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62172055, 61972039, U20B2045 and U1836103; in part by the National Key Research and Development Program under Grant 2019QY1404; in part by the China National Funds for Excellent Young Scientists under Grant 62222201; in part by the Beijing Nova Program under Grants Z201100006820006 and 20220484174; in part by the Beijing Natural Science Foundation under Grants M23020, L222098 and 7232041; in part by the Ningxia Natural Science Foundation under Grant 2022AAC03620; in part by the BUPT Scientific and Technological Innovation Capacity Improvement Project under Grant 2021XD-A09; in part by the JSPS KAKENHI under Grants JP19K20250, JP20H04174 and JP22K11989; in part by the Leading Initiative for Excellent Young Researchers (LEADER), MEXT, and JST, PRESTO under Grant JPMJPR21P3; and in part by the China Scholarship Council (CSC) under Grant 202106470009.

References

[1] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE network*, vol. 32, no. 1, pp. 96–101, 2018.

[2] J. Wu, M. Dong, K. Ota, J. Li, and W. Yang, "Application-aware consensus management for software-defined intelligent blockchain in iot," *IEEE Network*, vol. 34, no. 1, pp. 69–75, 2020.

[3] J. Zhou, Y. Wang, K. Ota, and M. Dong, "Aaiot: Accelerating artificial intelligence in iot systems," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 825–828, 2019.

[4] K. Yang, Y. Zhang, X. Lin, Z. Li, and L. Sun, "Characterizing heterogeneous internet of things devices at internet scale using semantic extraction," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5434–5446, 2021.

[5] X. Chen, W. Feng, Y. Luo, M. Shen, N. Ge, and X. Wang, "Defending against link flooding attacks in internet of things: A bayesian game approach," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 117–128, 2021.

[6] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "Corrauc: a malicious bot-iot traffic detection method in iot network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2020.

[7] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "Sok: Security evaluation of home-based iot deployments," in *2019 IEEE symposium on security and privacy (sp)*, pp. 1362–1380, IEEE, 2019.

[8] C. Zuo, H. Wen, Z. Lin, and Y. Zhang, "Automatic Fingerprinting of Vulnerable BLE IoT Devices with Static UUIDs from Mobile Apps," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, (New York, NY, USA), pp. 1469–1483, Association for Computing Machinery, 2019.

[9] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Machine learning for the detection and identification of internet of things devices: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 298–320, 2021.

[10] 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.

[11] L. Fan, L. He, Y. Wu, S. Zhang, Z. Wang, J. Li, J. Yang, C. Xiang, and X. Ma, "Autoiot: Automatically updated iot device identification with semi-supervised learning," *IEEE Transactions on Mobile Computing*, 2022.

[12] K. Kamnitsas, D. Castro, L. Le Folgoc, I. Walker, R. Tanno, D. Rueckert, B. Glocker, A. Criminisi, and A. Nori, "Semi-supervised learning via compact latent space clustering," in *International conference on machine learning*, pp. 2459–2468, PMLR, 2018.

[13] L. Fan, L. He, E. Dong, J. Yang, C. Li, J. Lin, and Z. Wang, "Evoiot: An evolutionary iot and non-iot classification model in open environments," *Computer Networks*, vol. 219, p. 109450, 2022.

[14] L. Fan, L. He, X. Sun, E. Dong, J. Yang, Z. Wang, J. Lin, and G. Song, "Graphiot: Accurate iot identification based on heterogeneous graph," in *Proceedings of the 31st IEEE/ACM International Symposium on Quality of Service (IWQoS)*, 2023.

[15] K. Kostas, M. Just, and M. A. Lones, "Iotdevid: A behavior-based device identification method for the iot," *IEEE Internet of Things Journal*, 2022.

[16] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle, "Website fingerprinting at internet scale.," in *NDSS*, 2016.

[17] M. Shen, M. Wei, L. Zhu, and M. Wang, "Classification of encrypted traffic with second-order markov chains and application attribute bi-grams," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1830–1843, 2017.

[18] M. Shen, Y. Liu, L. Zhu, X. Du, and J. Hu, "Fine-grained webpage fingerprinting using only packet length information of encrypted traffic," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2046–2059, 2020.

[19] B. Charyyev and M. H. Gunes, "Locality-sensitive iot network traffic fingerprinting for device identification," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1272–1281, 2020.

[20] S. Marchal, M. Miettinen, T. D. Nguyen, A.-R. Sadeghi, and N. Asokan, "Audi: Toward autonomous iot device-type identification using periodic communication," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1402–1412, 2019.

[21] R. Trimananda, J. Varmarken, A. Markopoulou, and B. Demsky, "Packet-level signatures for smart home devices," in *Network and Distributed Systems Security (NDSS) Symposium*, vol. 2020, 2020.

[22] Y. Wang, X. Yun, Y. Zhang, C. Zhao, and X. Liu, "A multi-scale feature attention approach to network traffic classification and its model explanation," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 875–889, 2022.

[23] C. Duan, H. Gao, G. Song, J. Yang, and Z. Wang, "Byteiot: a practical iot device identification system based on packet length distribution," *IEEE Transactions on Network and Service Management*, 2021.

[24] R. Kumar, M. Swarnkar, G. Singal, and N. Kumar, "Iot network traffic classification using machine learning algorithms: An experimental analysis," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 989–1008, 2021.

[25] M. Safi, S. Dadkhah, F. Shoeleh, H. Mahdikhani, H. Molyneaux, and A. A. Ghorbani, "A survey on iot profiling, fingerprinting, and identification," *ACM Transactions on Internet of Things*, 2022.

[26] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.

[27] A. Dempster, D. F. Schmidt, and G. I. Webb, "Minirocket: A very fast (almost) deterministic transform for time series classification," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 248–257, 2021.

[28] C. W. Tan, A. Dempster, C. Bergmeir, and G. I. Webb, "Multirocket: multiple pooling operators and transformations for fast and effective time series classification," *Data Mining and Knowledge Discovery*, vol. 36, no. 5, pp. 1623–1646, 2022.

[29] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "Hive-cote 2.0: a new meta ensemble for time series classification," *Machine Learning*, vol. 110, no. 11, pp. 3211–3243, 2021.

[30] A. Dempster, D. F. Schmidt, and G. I. Webb, "Hydra: Competing convolutional kernels for fast and accurate time series classification," *arXiv preprint arXiv:2203.13652*, 2022.

[31] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Du, "Accurate decentralized application identification via encrypted traffic analysis using graph neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2367–2380, 2021.

[32] J. Zhao, Q. Li, S. Liu, Y. Yang, and Y. Hong, "Towards traffic supervision in 6g: a graph neural network-based encrypted malicious traffic detection method," *SCIENTIA SINICA Informationis*, vol. 52, no. 2, pp. 270–286, 2022.

[33] C. Fu, Q. Li, M. Shen, and K. Xu, "Frequency domain feature based robust malicious traffic detection," *IEEE/ACM Transactions on Networking*, 2022.

[34] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, and A. A. Ghorbani, "Towards the development of a realistic multi-dimensional iot profiling dataset," in *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*, pp. 1–11, IEEE, 2022.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

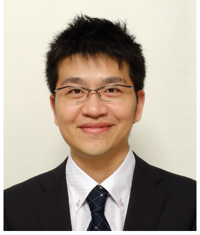
[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[37] Q. Li, J. Mi, W. Li, J. Wang, and M. Cheng, "Cnn-based malware variants detection method for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16946–16962, 2021.

Jianjin Zhao received the B.Eng. degree from Beijing University of Posts and Telecommunications, China, in 2019. He is currently pursuing the Ph.D. degree in Cyberspace Security at Beijing University of Posts and Telecommunications, China. He was a visiting student at Muroran Institute of Technology, Japan. His current research interests include encrypted traffic analysis and audit log analysis.

Qi Li received the Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2010. She is currently a Professor with the Information Security Center, State Key Laboratory of Networking and Switching Technology, School of Computer Science, Beijing University of Posts and Telecommunications. Her current research focuses on information systems and software.

Jintao Sun received the B.Eng. degree in computer science and technology from Yanshan University, China, in 2021. He is currently working towards the master's degree with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, China. His current research interests include IoT security and software vulnerability analysis.



Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is the Vice President and Professor of Muroran Institute of Technology, Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at the University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. He is the recipient of The 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017, Funai Research Award 2018, NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology, The Young Scientists' Award from MEXT in 2021, SUEMATSU-Yasuharu Award from IEICE in 2021, IEEE TCSC Middle Career Award in 2021. He is Clarivate Analytics 2019, 2021 Highly Cited Researcher (Web of Science) and Foreign Fellow of EAJ.

Kaoru Ota was born in Aizu-Wakamatsu, Japan. She received M.S. degree in Computer Science from Oklahoma State University, the USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively. Kaoru is a Professor and Ministry of Education, Culture, Sports, Science and Technology (MEXT) Excellent Young Researcher with the Department of Sciences and Informatics, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar at the University of Waterloo, Canada. Also, she was a Japan Society of the Promotion of Science (JSPS) research fellow at Tohoku University, Japan from April 2012 to April 2013. Kaoru is the recipient of IEEE TCSC Early Career Award 2017, The 13th IEEE ComSoc Asia-Pacific Young Researcher Award 2018, 2020 N2Women: Rising Stars in Computer Networking and Communications, 2020 KDDI Foundation Encouragement Award, and 2021 IEEE Sapporo Young Professionals Best Researcher Award. She is Clarivate Analytics 2019, 2021 Highly Cited Researcher (Web of Science) and is selected as JST-PRESTO researcher in 2021, Fellow of EAJ in 2022.

Meng Shen received the B.Eng. degree in computer science from Shandong University, Jinan, China, in 2009, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2014. He is a Professor with the Beijing Institute of Technology, Beijing. He has authored over 50 papers in top-level journals and conferences, such as ACM SIGCOMM, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. He has guest edited special issues on emerging technologies for data security and privacy technologies in IEEE NETWORK and IEEE INTERNET OF THINGS JOURNAL. His research interests include data privacy and security, blockchain applications, and encrypted traffic classification. He received the Best Paper RunnerUp Award at IEEE IPCCC 2014 and IEEE/ACM IWQoS 2020. He was selected by the Beijing Nova Program 2020 and was the winner of the ACM SIGCOMM China Rising Star Award 2019.