An Adaptive Ensembled Neural Network-Based Approach to IoT Device Identification

Jingrun Ma[1,2], Yafei Sang[1,2(✉)], Yongzheng Zhang[3], Xiaolin Xu[4], Beibei Feng[1,2], and Yuwei Zeng[5]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{majingrun,sangyafei}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] China Assets Cybersecurity Technology Co., Ltd., Beijing, China
[4] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
[5] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China

Abstract. The Internet of Things (IoT) has developed rapidly in recent years and has been widely used in our daily life. An online report claimed that the connected IoT devices will reach the scale of 14.4 billion globally at the end of 2022. With the rapid and large-scale deployment of such devices, however, some severe security problems and challenges arised as well, especially in the field of IoT device management. Device identification is a prerequisite procedure to mitigate the above issues. Therefore, accurately identifying the deployed IoT devices plays a vital role in network management and cyber security. In this work, we come up with a spatio-temporal-based method that characterizes IoT device behaviors by leveraging the packet sequence features of IoT traffic, which is able to automatically extract the high-level features from raw IoT traffic. The further evaluation indicates that our method is capable of identifying diverse IoT devices with satisfactory accuracy.

Keywords: IoT Identification · Traffic classification

1 Introduction

IoT refers to the tens of billions of low-cost devices that communicate with each other and remote servers on the Internet autonomously, which contains a great diversity of types, and it comprises devices, such as IP cameras, motion sensors, and Internet of Vehicles devices. According to an authoritative report, the number of connected IoT devices will reach 14.4 billion by the end of 2022 [1]. The number of IoT devices connected to the Internet has exploded, making the home smarter by providing convenient services. However, with the popularization of IoT devices, security risks caused by the potential vulnerable IoT devices arise as well.

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 461, pp. 214–230, 2022.
https://doi.org/10.1007/978-3-031-24386-8_12

It is known that IoT devices are easy to be penetrated and exploited due to the architecture design [12,17]. There are lots of works referred that adversary tends to exploit the vulnerable IoT devices to conduct network attacks [3,11,19,20]. It is important for the network manager to accurately identify the network-connected IoT devices, especially the vulnerable ones, while it is not a trivial thing. An automatic identification method is able to facilitate the management of connected IoT devices, which would minimise the threat of cyber-attack on enterprise network to some extents.

There is a variety of works on IoT device identification based on network traffic analysis using machine learning techniques. They establish machine learning frameworks by leveraging various network traffic characteristics to identify IoT devices on a network. Sivanathan et al. [21] inspect the payload of packets to extract domain names, port numbers and other numerical features from network flow in a time window, and then builds a two-state classifier to identify the IoT devices. Thangavelu et al. [22] propose a distributed IoT device fingerprinting technique that can identify the presence of common devices, and find new devices, by clustering device fingerprints. Hamad et al. [9] analyze packet sequence from high-level network traffic, and creates IoT device fingerprint by leveraging flow-based features. Pinheiro et al. [18] propose a solution that uses packet length statistics including mean, standard deviation, and the number of bytes to identify IoT devices and events.

These machine learning-based methods conduct the analysis relying on the statistical characteristics of network traces, which need expert knowledge to carry out feature engineering. Moreover, some methods need to inspect the payload of packets to extract features, which may bring privacy risks and will be useless when encrypted traffic introduced.

In this paper, we address the above problem by developing an adaptive ensembled neural network-based method that exploits spatio-temporal features to classify IoT devices. Our method is able to identify IoT devices efficiently as it relys only on the length and TCP window size of a few packets without any statistical computation and inspecting payloads. A traffic flow refers to a packet sequence that consists of multiple packets. We can extract multiple characteristics from each packet. Therefore, we represent a flow as a collection of characteristics sequence, which can characterize each IoT device from spatial dimension and temporal dimension, respectively. In the temporal domain, we use bi-LSTM to learn the timing relationship and extract temporal features between packets for each characteristics sequence. In the spatial domain, we use an adaptive ensemble of multiple bi-LSTM to learn the correlation between different characteristics sequences. In this paper, we use packet length sequence and TCP window size sequence for accurate IoT device identification. In general, other characteristics sequences can be used in the same way which makes the method scalable.

In this paper, we propose a robust approach to identify IoT devices. Our contributions are as follows:

– We analyze packet characteristic sequences such as packet length sequence and TCP window size sequence, and use them to characterize IoT devices.

- We propose an spatio-temporal-based approach that exploits the packet sequence features from spatial dimension and temporal dimension, respectively. It can characterize the behavior of IoT devices without inspecting packet payloads, which can avoid privacy leakage risks.
- We build an adaptive ensemble of neural networks to identify IoT devices, which is scalable.
- Our method reach the accuracy of 99% on multiple datasets.

The rest of this paper is organized as follows: Sect. 2 describes relevant prior work. In Sect. 3 we present the preprocess of IoT traffic data and analysis of packet sequence features. We introduce the model architecture detailedly in Sect. 4. Section 5 presents the experiments and results. Finally, Sect. 6 concludes this paper.

2 Related Work

With the widespread application of IoT devices, there is a lot of work in IoT device identification. Previous methods in this field mainly fall into two categories: fingerprint-based methods and machine learning-based methods.

Fingerprint-Based Methods. The fingerprint-based methods profile IoT devices by exploiting plaintext information in the IoT traffic or features from network protocols. They generate fingerprints for each type of IoT device and identify them in the network by leveraging these fingerprints. Feng et al. [7] proposed a method to identify IoT device types by leveraging plaintext in banner grabbing. They scaned ports in the network and analyzed the response information that includes device vendor name and device product model. However, they assumed that the IoT device information are included in the response packet. Their method would be less effective when the device does not respond or device information are not included in the response packet. Yang et al. [24] gave an observation that different IoT device manufacturers have a diversity of implementation for the same network service. In addition, they extracted features from three network layers and generate fingerprints by using a neural network. Trimananda et al. [23] proposed a solution that can automatically extract packet-level signatures to identify IoT devices. These signatures which can be generated without prior knowledge consisted of sequences of packet lengths and directions.

Machine Learning-Based Methods. There are a lot of methods relying on building machine learning models using statistics of IoT traffic. These methods usually leverage supervised learning to characterize traffic patterns for each IoT device. Sivanathan et al. [21] collected a dataset in a testbed environment, and analyzed the features which can profile the IoT device. They build a two-stage classifier by leveraging both plaintext features and statistical features. Nguyen et al. [16] proposed a unsupervised learning-based method to identify compromised IoT devices by utilizing a self-learning federated learning approach, which can profile IoT devices without labeled data. Ma et al. [13] proposed a solution that leverage CNN to learn spatio-temporal traffic fingerprints to identify the IoT

devices, which also can identify the devices that hidden behind NAT. Duan et al. [6] only used the frequency distribution of bidirectional packet length to characterize IoT devices, and classified IoT devices with the k-NN algorithm, which can work extensively and adaptively. Marchal et al. [14] proposed a method that leverages periodic communication traffic to characterize IoT devices. This method can identify previously unseen IoT devices without expert knowledge or labeled data by leveraging an unsupervised learning method. Yin et al. [25] proposed a deep learning-based method to identify IoT devices. They build an automatic end-to-end framework based on CNN and bi-LSTM model in the face of identifying IoT devices.

3 Data Preprocess and Feature Analysis

In this paper, a flow is considered as a detection sample. A flow is a collection of multiple associated packets [5] between two computer addresses using a particular protocol on a particular pair of ports. Packets with the same tuple of information (source IP, destination IP, source port, destination port, protocol) belong to the same flow. The flow is bidirectional which includes packets that client to server and server to client.

3.1 Data Preprocess

We extract packet features from pcap files of IoT traffic and reconstruct flows by leveraging 5-tuple information. Each flow consists of multiple packet feature sequences, which our sequence models will learn representative features from.

Moreover, the number of packets in traffic flows are not always the same, which may conflict with the requirement of our classification model which needs a uniform size of input data. Hence, the flows that we reconstruct before need to be processed into same format. The following unified preprocessing measures including padding and segmentation are applied to make input flows have a uniform size:

- If the number of packets in a flow exceeds a certain threshold that we set, we select the first M packets in a flow to represent it as a whole.
- If the number of packets in a flow is less than the threshold, we pad the characteristics sequence with zeros.

A traffic flow is preprocessed according to the above rules, which allows for reducing the amount of data and unifying the data size.

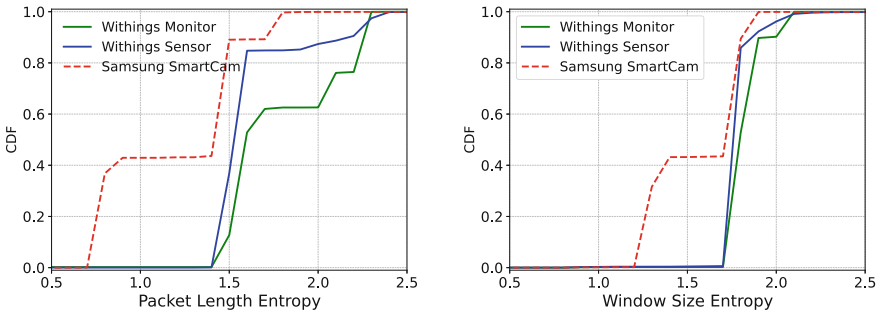
3.2 Packet Feature Analysis

In this paper, we choose two packet characteristics sequences to identify the IoT device correctly, which is scalable and other feature sequences can be added without major changes. The method exploits packet length sequence and TCP window size sequence derived from traffic flow, which is unnecessary to compute

Table 1. The breakdown of TMC-18

ID	IoT devices	Flow(#)	Connection type
1	Withings Monitor	5,591	Wired
2	Withings Sensor	3,584	Wireless
3	Samsung SmartCam	15,906	Wireless
4	TP-Link Cloud Camera	1,109	Wireless
5	HP Printer	151	Wireless
6	Amazon Echo	20,903	Wireless
7	Triby Speaker	149	Wireless
8	iHome	177	Wireless
9	Insteon Camera	4,073	Wired
10	Belkin Wemo Switch	7,642	Wireless
11	TP-Link Smart Plug	232	Wireless
12	Belkin Wemo Motion Sensor	78,761	Wireless
13	Netatmo Weather Station	2,347	Wireless
14	Netatmo Welcome	2,682	Wireless
15	PIX-STAR	1,139	Wireless

statistical features of flows. The premise is that packet length patterns of different IoT devices are distinct, which can be used to characterize the device [15]. In addition, TCP window size is also a distinction for different IoT traffic. TCP can govern the amount of data sent between client and server by using window size. The window size indicates an ability to receive data that can be buffered during a connection [2]. Different IoT devices have different capabilities of processing data and buffering data, and their corresponding servers are also diverse. We analyze these two packet characteristics in terms of size and sequence change on part data from [21] which is also used in the evaluation. The list of IoT devices are shown in Table 1.



(a) Packet length entropy distribution. (b) Window size entropy distribution.

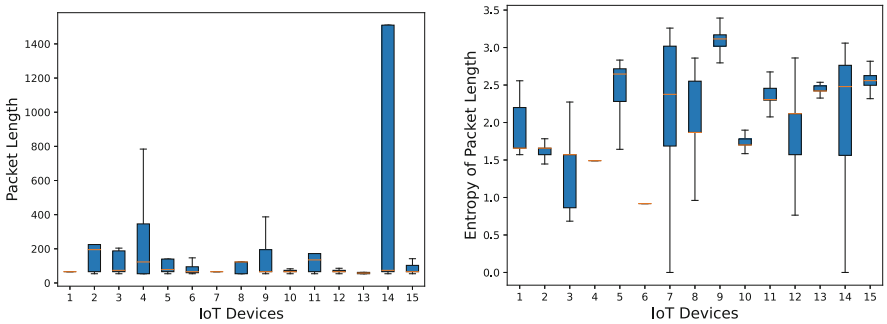
Fig. 1. Entropy distribution of three selected devices.

We select three devices including Withings Monitor, Withings Sensor and Samsung SmartCam from Table 1, and analyze their packet characteristics distribution. Among these three devices, Withings Monitor and Withings Sensor are two types of product with different functions from the same vendor. Withings Monitor and Samsung SmartCam are the same types of product which have similar functions from different vendors.

Here, we use entropy to measure the distribution of packet characteristics in a flow. Figure 1 shows the Cumulative Distribution Function (CDF) of packet length entropy and window size entropy. We can see that Withings Monitor and Withings Sensor have more similar packet characteristics distribution compared with Samsung SmartCam. Even though Withings Monitor and Samsung SmartCam are the same types of product, they have obvious differences in packet length distribution and window size distribution. For the same vendor's devices, there is an obvious distinction between Withings Monitor and Withings Sensor in the packet length distribution, but the distribution of the window size is not so obvious between these two devices. Next, we analyze these two packet characteristics on all the IoT devices in Table 1.

Packet Length. Different IoT devices carry a diversity of services, so the packet length may express different patterns when they communicate with the server. For example, smart camera devices usually generate flows with lots of packets and large packet lengths to transfer video streams. On the contrary, smart plug devices generate flows that consist of a few amount of packets and small packet lengths. We calculate the mean packet length of each flow for all IoT devices. Figure 2(a) illustrates the boxplot of mean packet length of each IoT device. We can see that the packet length of some devices is not obviously different, such as Withings Monitor, Triby Speaker, Belkin Wemo Switch, Belkin Wemo Motion Sensor, and Netatmo Weather Station. On the contrary, devices like TP-Link Cloud Camera and Netatmo Welcome have a conspicuous difference compared with other devices. It is confusing for those devices with similar mean packet length distribution. So we don't only use the packet length but introduce the packet length sequence which can reflect the change of the packet length over time. Figure 2(b) illustrates the boxplot of entropy of packet length for each IoT device. As shown in Fig. 2(b), entropy of packet length sequence is distinct between those confusing devices.

Window Size. The data processing ability of different IoT devices is different, so they have different patterns in window size. Because of low computing resources, IoT devices usually provide a relatively single service which leads to that they communicate with only a few servers. The window size characterizes the communication patterns between IoT devices and their servers. We also calculate the mean window size of each flow for all IoT devices. Figure 3(a) illustrates the boxplot of mean window size of each IoT device and Fig. 3(b) shows the boxplot of entropy of window size sequence. Most IoT devices have different patterns in window size and entropy of window size. We can see that devices

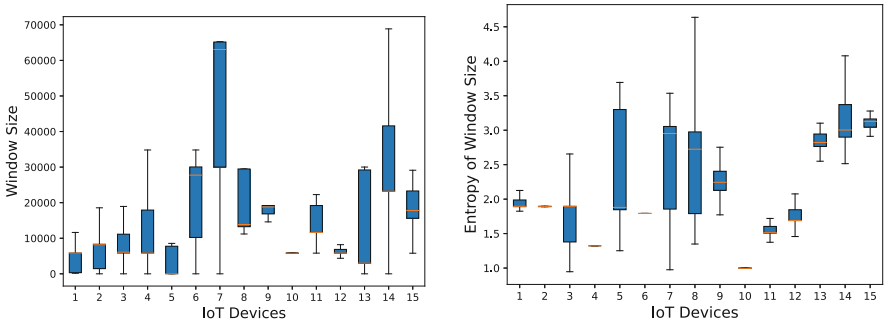


(a) Packet length of each IoT devices. (b) Entropy of packet length sequence.

Fig. 2. Various packet length patterns of different IoT devices.

with similar mean window size have different entropy of window size such as Belkin Wemo Switch and Belkin Wemo Motion Sensor. And devices with similar entropy of window size have different mean window size such as Withings Sensor and Amazon Echo.

These two characteristic sequences can work together for accurate IoT device identification.



(a) Window size of each IoT devices. (b) Entropy of window size sequence.

Fig. 3. Various window size patterns of different IoT devices.

4 Model Methodology

In this section, we build a adaptive ensembled neural network model based on deep learning. We introduce the overall structure of the model and the main network layers.

4.1 Overview

Our adaptive ensembled neural network is a hierarchical model as shown in Fig. 4. The packet length sequence and TCP window size sequence are extracted from flows during data preprocessing. Subsequently, the packet characteristics sequence is transformed into embedding vectors. Then the adaptive ensemble of sequence networks is designed to process the embedding feature to extract spatio-temporal features at the flow level. Finally, discrimination results are output through the fully connected network. Experiments show that this combination of multiple packet characteristics sequence can effectively classify IoT devices.

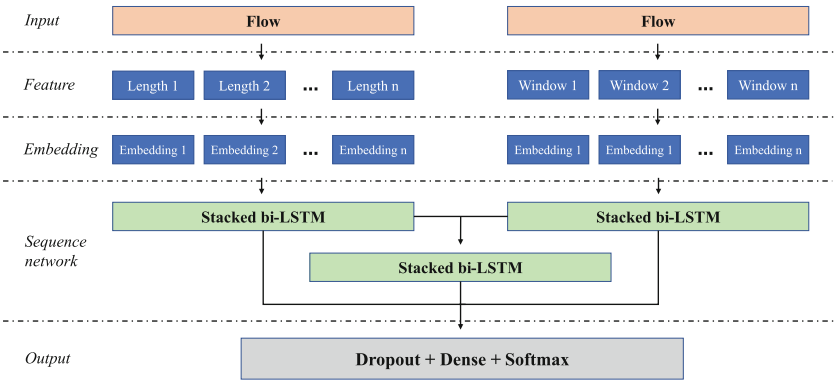


Fig. 4. Architecture of the neural network.

4.2 Embedding Layer

The input of embedding layer consists of a packet characteristic sequence. Inspired by embedding in natural language processing, we consider the sequence as a sentence, and convert the value of the characteristic sequence into a vector as the output of this layer.

Given a packet characteristic sequence with n elements $x = [F_1, F_2, ..., F_n]$ and dimension d of element embedding vectors, each element F_i, $i \in [1, n]$ need to be converted into a d-dimension vector. Finally, we can obtain the embedding matrix $E \in \mathbb{R}^{n \times d}$ for each packet characteristic sequence.

Converting the packet characteristic sequences into embedding vectors can integrate a large amount of valuable information, which can improve the ability of learning representative features and boost the performance of identifying IoT devices.

4.3 Adaptive Ensemble of Sequence Network

Network traffic is the conversation between the client and the server, which means that the packet sequence is naturally temporal. LSTM is a special kind

of Recurrent Neural Network (RNN) which can process sequence of inputs. Traditional RNNs have the disadvantages of long-term dependency problem which will result in gradient disappearance or explosion. LSTM uses gate structure to remember or forget information to avoid the long-term dependency problem. An LSTM has three gates to control the cell state including forget gate, input gate and output gate.

Before adding new information, The forget gate decides what information to be discarded from previous sequence. f_t outputs a value between 0 and 1 according to the hidden information h_{t-1} and the input x_t. The current cell keeps more information if f_t is close to 1 otherwise less.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

After discarding the information, the input gate decides what new information to be stored in current cell. A tanh layer creates new information according to the hidden information h_{t-1} and the input x_t.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

Based on f_t, i_t and $\tilde{C}_t$, cell updates C_{t-1} into the new state C_t.

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{4}$$

After the cell is updated, we need to determine the output h_t of the cell, which is controlled by the output gate o_t.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \times tanh(C_t) \tag{6}$$

To learn sequential features from both forward and backward directions, we use bidirectional LSTM (bi-LSTM) to incorporate the contextual features of the packet characteristics sequence. To improve the representation of the model, we adopt multi-layer bi-LSTM to learn the low-level and high-level features at the same time.

Many sequential features can be extracted from the traffic flow and we use multiple stacked bi-LSTM to learn different sequential features respectively. We use two stacked bi-LSTM to learn temporal features from packet length sequence and window size sequence. Moreover, another stacked bi-LSTM is used to learn spatial features. It takes the outputs of other stacked bi-LSTM as input, as shown in Fig. 4. This can learn the correlation between different packet characteristics sequence in the spatial dimension.

These stacked bi-LSTM form an adaptive ensemble of sequential neural networks. It learns spatio-temporal features from multiple packet characteristic sequences, which can characterize traffic patterns of different IoT devices. Moreover, it is scalable and other characteristic sequence can be added in easily.

4.4 Dense Layer

The last part of the model includes a dropout layer and a fully-connected layer followed by a softmax function to obtain a prediction vector $\hat{y_{ij}}$. We then use the cross-entropy loss function for calculating the loss L of the result, which can be expressed as:

$$L = -\sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij} \log \hat{y_{ij}} \tag{7}$$

where y_{ij} is the true label vector, $\hat{y_{ij}}$ is the prediction vector.

5 Evaluation

In prior sections, we analyze the traffic of IoT devices and preliminarily introduce the architecture of the model. This section sequentially introduces the experiment-used datasets, the evaluation metrics, and the performance comparison between the prior work and ours.

5.1 Datasets

We conduct experiments on four public datasets to verify the effectiveness of the method. Next, we give a brief description of these datasets. Table 2 shows the breakdown of these datasets.

Sivanathan et al. [21] publish a 20-day IoT traffic dataset, TMC-18, that collected in an experimental environment. To fit our experiment, we discard both the non-IoT device flows and the devices that own only a few flows here. Finally, 15 IoT devices, as shown in Table 1, are left to use in this experiment.

Hamza et al. [10] publish a 44-day IoT traffic dataset, SOSR-19, which includes attack traffic and benign traffic. Its collection environment is similar to TMC-18, and we select the benign traffic to evaluate our method.

Garcia et al. [8] create a labeled dataset with malicious and benign IoT network traffic. We also use benign IoT traffic to complete the evaluation.

Dadkhah et al. [4] generate a dataset for profiling, behavioural analysis of different IoT devices. It also has malicious and benign IoT traffic, and we evaluate on the benign IoT traffic.

Table 2. The breakdown of the datasets

Dataset	Devices (#)	Flow (#)	Public year
TMC-18 [21]	15	144,446	2018
SOSR-19 [10]	15	578,533	2019
IoT-23 [8]	3	559	2020
CIC-IoT [4]	37	199,033	2022

5.2 Metrics

Here, we use precision, recall, F1-score, and accuracy to evaluate the effectiveness of our approach. In our setting, for each IoT device, a correctly identified flow is treated as a true positive (TP); a flow identified as belonging to this device but actually not is treated as a false positive (FP); a flow of this device identified to other devices is treated as a false negative (FN). Based on these three definitions, three metrics are defined for each device as follows.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

We use macro averages of these three metrics and accuracy in evaluation. Accuracy is defined as follows.

$$Accuracy = \frac{\#\ of\ correct\ identification}{\#\ of\ total\ identification} \tag{11}$$

5.3 Experimental Setting

We take the packet feature sequences as the input of the model. The packet number is set to 32. The dimension of the packet feature embedding vector is set to 256. Besides, we set the dimension of hidden states of each bi-LSTM to 256 and take the 2 layer bi-LSTM in each sequence. Moreover, we take dropout with a 0.3 ratio to avoid over-fitting, and use 0.0001 as the learning rate of Adam optimizer. We implement our approach with pytorch and deploy it on a server with 32 CPU cores and 128 GB memory. The server uses a NVIDIA 1080Ti for accelerating computing.

5.4 Experiments and Results

Here, we use four well-known IoT traffic datasets to evaluate the performance of our method in terms of precision, recall, F1, and accuracy. After extracting features from IoT devices traffic to generate training and testing samples, we have randomly split these instances into three groups, one containing 80% of the instances for training, one containing 10% of the instances for validating, and the other containing 10% of the instances for testing. The adaptive ensemble of the sequence network is trained, and then evaluate to distinguish IoT devices. The evaluation includes fine-grained classification, and coarse-grained classification. The former distinguishes each single IoT device, and the latter distinguishes IoT device type. IoT devices with the same vendor and type are regarded as one class in the IoT device type identification. Moreover, we investigate the feature distribution on the IoT devices with the same vendor and type.

Table 3. Experiments on multiple datasets

Method	TMC-18				SOSR-19				IoT-23				CIC-IoT			
	A (%)	P (%)	R (%)	F1 (%)	A (%)	P (%)	R (%)	F1 (%)	A (%)	P (%)	R (%)	F1 (%)	A (%)	P (%)	R (%)	F1 (%)
KNN	74.59	77.03	55.87	57.68	91.25	85.23	68.75	73.13	99.86	99.32	98.43	98.86	40.30	53.46	38.60	36.90
DT	79.97	78.41	62.70	65.40	91.25	86.08	70.38	75.22	99.85	99.18	98.51	98.84	43.23	55.60	41.21	40.70
RF	79.32	79.30	62.61	65.37	91.25	86.48	70.13	75.34	**99.89**	**99.39**	**98.99**	**99.14**	43.56	55.38	41.70	40.55
SVM	50.24	30.35	18.70	16.53	66.38	26.74	14.38	13.65	90.89	90.65	46.69	53.63	13.80	10.54	6.27	4.32
MV	79.30	79.43	62.38	65.40	91.23	86.02	68.96	74.85	99.86	99.14	98.70	98.92	43.05	56.39	41.30	40.70
IoT ETEI	99.54	95.46	92.89	93.39	99.88	88.91	81.85	82.02	98.24	98.99	98.61	98.78	85.79	60.48	54.83	53.13
Our method	**99.97**	**96.73**	**93.47**	**94.61**	**99.99**	**98.12**	**98.68**	**98.18**	98.21	99.05	98.41	98.70	**86.06**	**65.73**	**65.81**	**64.59**

Fine-Grained Identification. Table 3 presents the experiment results on four datasets. Our method reaches the accuracy of 99.7%, 99.99%, and 98.21% on TMC-18, SOSR-19, and IoT-23 datasets, respectively. Precision, recall, and F1-score all exceed 93.47% on TMC-18, SOSR-19 and IoT-23. The evaluation on CIC-IoT dataset reaches an accuracy of 86.06%.

IoT devices in TMC-18 and SOSR-19 datasets are the same devices. We aggregate these two datasets and test our method on the mixed dataset. Figure 5 shows the confusion matrix on mixed dataset, in which rows represent actual labels and columns represent predictions. The diagonal values show the correct classifications. Figure 5 illustrates that our method performs well on each device and only a few samples are not correctly identified. Our method has reached an identifying accuracy of over 99.4% on most IoT devices. The experimental results clearly show that our approach has the ability to conduct accurate IoT device identification.

From Table 3 we can see that accuracy, precision, recall, and F1-score for fine-grained identification on CIC-IoT dataset is 86.06%, 65.73%, 65.81%, and 64.59%, respectively, which is obviously lower than on other datasets.

Coarse-Grained Identification. Our method performs a bit poor on the CIC-IoT dataset. Figure 6 illustrates the confusion matrix on CIC-IoT dataset. IoT Devices with low identification rates such as Gosund ESP series devices, are the same type of product that share the same vendor. And Amazon Echo Dot, and Yutron Plug series devices, are the same. Figure 7 shows the packet length entropy distribution of these three type devices. The entropy distribution is similar in each group. As for Amazon Echo Dot, entropy distribution is almost identical. Thus, IoT devices that are the same type of product from same vendor, use similar communication patterns making it difficult to identify. These products are not upgraded during the iterative process considering the communication protocol or mechanism, which causes the traffic generated during network communication is almost identical.

We regard devices with the same vendor and type as one class of device, to carry out coarse-grained identification. For example, we regard Gosund ESP_1 Socket, Gosund ESP_2 Plug, Gosund ESP_3 Socket, Gosund ESP_4 Plug, Gosund ESP_5 Socket, Gosund ESP_6 Plug, and Gosund ESP_7 Plug as one class, Gosund ESP Plug Series, and we carry out same operations on other IoT

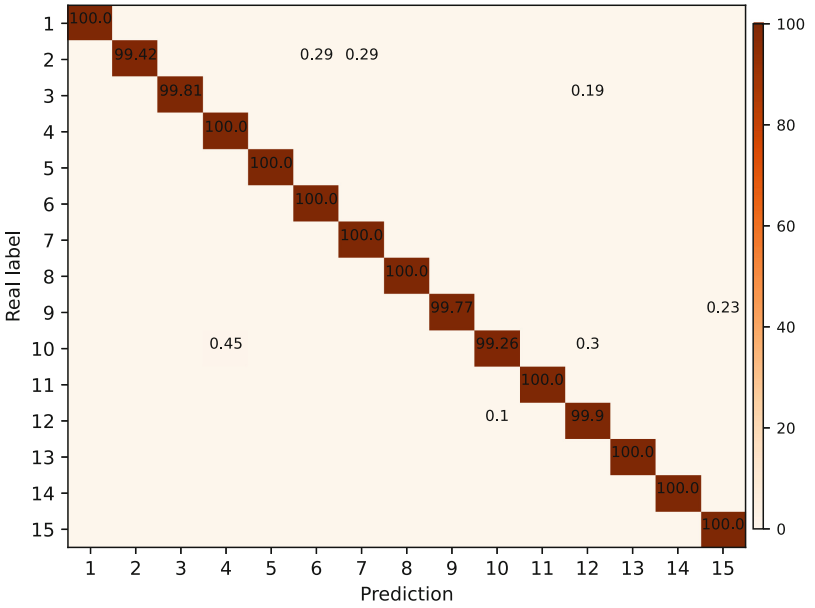


Fig. 5. Confusion matrix on TMC-18 and SOSR-19 dataset.

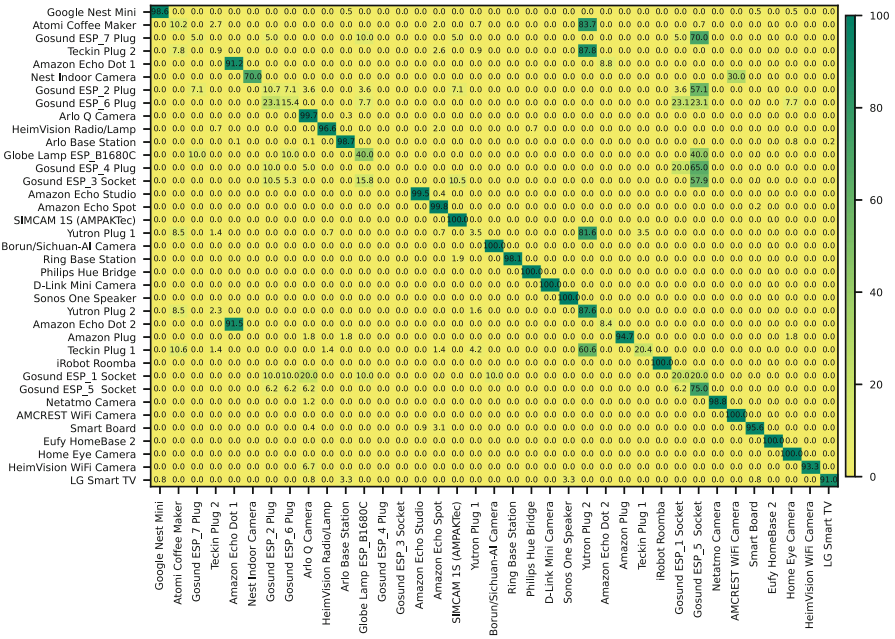
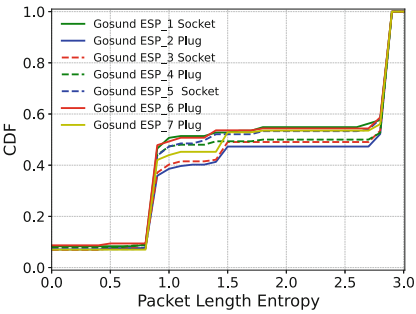
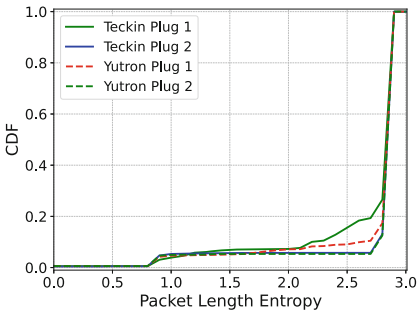


Fig. 6. Confusion matrix on CIC-IoT dataset.

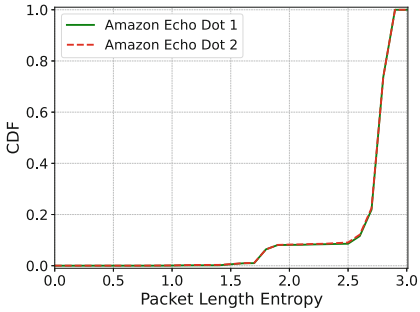
devices which are the same type of product of same vendor. After merging the labels, there are 23 IoT devices left. The accuracy, precision, recall, and F1-score reaches 99.05%, 96.11%, 94.37%, and 94.19%, respectively in coarse-grained identification, which is much higher than before.



(a) Packet length distribution of Gosund devices.



(b) Packet length distribution of Plug devices.



(c) Packet length distribution of Echo Dot devices.

Fig. 7. Packet length entropy distribution of confusing devices in CIC-IoT dataset.

Comparison with Prior Work. Pinheiro et al. [18] propose a solution that also uses packet length statistics to identify IoT devices. The solution uses only the statistical mean, the standard deviation, and the number of bytes transmitted over a one-second window. The solution carries out the identification by using five classification algorithm, including k-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM) and Majority Voting (MV). Table 3 shows the corresponding comparison results. Our method outperforms the prior work on TMC-18, SOSR-19, and CIC-IoT, while slightly lags behind the RF method on IoT-23. The problem lies in the fact that the scale of the IoT-23 dataset is a bit small, which contains only 3 devices and 559 traffic flows. However, with the increasing of dataset scale, the performance

of the RF method has declined gradually. Moreover, SVM reached the worst results in all scenarios.

Yin et al. [25] propose IoT ETEI, a deep learning-based method which leveraging CNN and bi-LSTM model to extract spatio-temporal features. It takes raw bytes of traffic flows as input, and removes irrelevant fields, such as MAC address and IP address. The method uses CNN to learn spatial features from raw data, and uses bi-LSTM to learn temporal features from packet sequence. Table 3 shows that our method outperforms theirs on SOSR-19 and CIC-IoT. We reach 10% more than IoT ETEI on precision, recall and F1-score except for precision on CIC-IoT. The results on TMC-18 and IoT-23 are similar to each other. Our method is more robust than IoT ETEI when the dataset scale is large and unbalanced.

Discussion. Our model can learn the spatio-temporal features of traffic data well, and is scalable. Experiments on multiple dataset show its effectiveness, accuracy, and robustness. However, the model still has some shortcomings. It performs a bit poor in fine-grained identification of IoT devices on the CIC-IoT dataset. IoT devices with same type and vendor follow similar traffic patterns when communicate with their servers, which are difficult to be identified. We need to improve this problem in the future.

6 Conclusion

In this paper, we propose an adaptive ensemble of sequence networks based on spatio-temporal features for IoT device identification, which takes network packet traces as input and automatically infers the IoT device type of the network traces. Our approach is purely based on network packet traces, and it jointly learns the representative features of typical IoT devices from the raw flow sequences. It takes stacked bi-LSTM to learn the representation of the flow sequence. Moreover, it can be easily extended with multi-attribute sequences as the input. It works well with both encrypted and plaintext flows. Our further evaluation demonstrates that our method is able to effectively identify the type of each IoT device from the mixed network traffic with high accuracy.

Acknowledgements. We thank the anonymous reviewers for their insightful comments. This work is supported by The National Key Research and Development Program of China under Grant No. 2019YFB1005201, No. 2019YFB1005203 and No. 2019YFB1005205.

References

1. Number of connected IoT devices growing 18% to 14.4 billion globally. https://iot-analytics.com/number-connected-iot-devices/
2. RFC 793 - transmission control protocol. https://datatracker.ietf.org/doc/html/rfc793

3. Antonakakis, M., et al.: Understanding the mirai botnet. In: 26th USENIX Security Symposium (USENIX Security 2017), pp. 1093–1110 (2017)
4. Dadkhah, S., Mahdikhani, H., Danso, P.K., Zohourian, A., Truong, K.A., Ghorbani, A.A.: Towards the development of a realistic multidimensional IoT profiling dataset. In: 2022 19th Annual International Conference on Privacy, Security & Trust (PST), pp. 1–11. IEEE (2022)
5. Dainotti, A., Pescape, A., Claffy, K.C.: Issues and future directions in traffic classification. IEEE Network **26**(1), 35–40 (2012)
6. Duan, C., Gao, H., Song, G., Yang, J., Wang, Z.: ByteIoT: a practical IoT device identification system based on packet length distribution. IEEE Trans. Netw. Service Manag. (2021)
7. Feng, X., Li, Q., Wang, H., Sun, L.: Acquisitional rule-based engine for discovering internet-of-thing devices. In: Proceedings of the 27th USENIX Conference on Security Symposium, pp. 327–341 (2018)
8. Garcia, S., Parmisano, A., Erquiaga, M.J.: IoT-23: a labeled dataset with malicious and benign IoT network traffic. Stratosphere Lab., Praha, Czech Republic, Technical report (2020)
9. Hamad, S.A., Zhang, W.E., Sheng, Q.Z., Nepal, S.: IoT device identification via network-flow based fingerprinting and learning. In: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), pp. 103–111. IEEE (2019)
10. Hamza, A., Gharakheili, H.H., Benson, T.A., Sivaraman, V.: Detecting volumetric attacks on lot devices via SDN-based monitoring of mud activity. In: Proceedings of the 2019 ACM Symposium on SDN Research, pp. 36–48 (2019)
11. Li, J., Li, Z., Tyson, G., Xie, G.: Your privilege gives your privacy away: an analysis of a home security camera service. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, pp. 387–396. IEEE (2020)
12. Loi, F., Sivanathan, A., Gharakheili, H.H., Radford, A., Sivaraman, V.: Systematically evaluating security and privacy for consumer IoT devices. In: Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, pp. 1–6 (2017)
13. Ma, X., Qu, J., Li, J., Lui, J.C., Li, Z., Guan, X.: Pinpointing hidden IoT devices via spatial-temporal traffic fingerprinting. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, pp. 894–903. IEEE (2020)
14. Marchal, S., Miettinen, M., Nguyen, T.D., Sadeghi, A.R., Asokan, N.: AuDI: toward autonomous IoT device-type identification using periodic communication. IEEE J. Sel. Areas Commun. **37**(6), 1402–1412 (2019)
15. Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.R., Tarkoma, S.: IoT sentinel: automated device-type identification for security enforcement in IoT. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 2177–2184. IEEE (2017)
16. Nguyen, T.D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A.R.: Dïot: a federated self-learning anomaly detection system for IoT. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 756–767. IEEE (2019)
17. Notra, S., Siddiqi, M., Gharakheili, H.H., Sivaraman, V., Boreli, R.: An experimental study of security and privacy risks with emerging household appliances. In: 2014 IEEE Conference on Communications and Network Security, pp. 79–84. IEEE (2014)

18. Pinheiro, A.J., Bezerra, J.D.M., Burgardt, C.A., Campelo, D.R.: Identifying IoT devices and events based on packet length from encrypted traffic. Comput. Commun. **144**, 8–17 (2019)
19. Ren, J., Dubois, D.J., Choffnes, D., Mandalari, A.M., Kolcun, R., Haddadi, H.: Information exposure from consumer IoT devices: a multidimensional, network-informed measurement approach. In: Proceedings of the Internet Measurement Conference, pp. 267–279 (2019)
20. Sadeghi, A.R., Wachsmann, C., Waidner, M.: Security and privacy challenges in industrial internet of things. In: 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6. IEEE (2015)
21. Sivanathan, A., et al.: Classifying IoT devices in smart environments using network traffic characteristics. IEEE Trans. Mob. Comput. **18**(8), 1745–1759 (2018)
22. Thangavelu, V., Divakaran, D.M., Sairam, R., Bhunia, S.S., Gurusamy, M.: Deft: a distributed IoT fingerprinting technique. IEEE Internet Things J. **6**(1), 940–952 (2018)
23. Trimananda, R., Varmarken, J., Markopoulou, A., Demsky, B.: Packet-level signatures for smart home devices. In: Network and Distributed Systems Security (NDSS) Symposium, vol. 2020 (2020)
24. Yang, K., Li, Q., Sun, L.: Towards automatic fingerprinting of IoT devices in the cyberspace. Comput. Netw. **148**, 318–327 (2019)
25. Yin, F., Yang, L., Wang, Y., Dai, J.: IoT ETEI: end-to-end IoT device identification method. In: 2021 IEEE Conference on Dependable and Secure Computing (DSC), pp. 1–8. IEEE (2021)