IoT Device Identification Method by Using DNS Queries at Initial Phase

Haruki Ishimaru
Graduate School of Informatics
Osaka Metropolitan University
Osaka, Japan
sp25332y@st.omu.ac.jp

Manato Fujimoto
Graduate School of Informatics
Osaka Metropolitan University
Osaka, Japan
manato@omu.ac.jp

Shingo Ata
Graduate School of Informatics
Osaka Metropolitan University
Osaka, Japan
ata@omu.ac.jp

***Abstract*—In recent years, the Internet of Things (IoT), in which various devices are connected to networks, has become widespread and is bringing benefits to society. On the other hand, there is a risk of information leakage from IoT devices and unintended behavior of devices. In order to communicate securely, it is necessary to understand the devices, and device identification technology that identifies them based on their characteristics is required. Methods using traffic data from IoT devices have been proposed so far, but they do not communicate on a daily basis, which makes it time-consuming to acquire necessary data and causes the time required for identification to be unpredictable. This paper proposes a device identification method that does not require long-term data acquisition, focusing on DNS queries issued by IoT devices immediately after connecting to the network. When they connect to a network, their initial operation involves communication with a cloud server, and the target domains are characteristic of each device. We identify devices using traffic data within a short period after connection and measure the accuracy of this identification. As a result of the identification, we demonstrate that an identification accuracy of 99 % at the vendor level and 87 % at the device name level can be achieved using data within 60 seconds after connection.**

***Index Terms*—IoT, DNS, Machine Learning, Network.**

I. INTRODUCTION

In recent years, the Internet of Things (IoT), where various devices are connected to the internet, has been rapidly spreading. IoT devices range from general-purpose devices like smartphones and PCs to specific-purpose devices such as cameras and home appliances, and it is predicted that approximately 40.6 billion IoT devices will be connected to networks by 2034 [1]. They are useful in many fields, and in the transportation sector, inventory management systems using RF tags have been established [2].

While IoT offers significant benefits to society, there is a possibility of information leakage and unexpected device operations due to the vulnerabilities of IoT devices and the low security awareness of people [3]. For example, DDoS (Distributed Denial of Service) attacks by malware-infected IoT devices, and eavesdropping on communication data such as sniffing and snooping are conceivable [4]. To prevent these damages, it is required to accurately grasp their vulnerabilities and network connection status by administrators, and to appropriately isolate networks according to the type and purpose of the devices. However, manual registration of MAC addresses and device names, for example, poses challenges such as the occurrence of human errors and an increased burden on administrators. Therefore, in addition to detecting connected devices, IoT device identification that automatically identifies them based on their characteristics (type, purpose) is essential for security measures.

Previously, machine learning-based device identification methods using device traffic data have been proposed. However, these studies require long-term traffic data for learning, which not only increases the learning cost but also takes a long time for device identification as a whole. This paper proposes an IoT device identification method that does not require long-term traffic data. The goal is to reduce the time required for device identification and achieve it with stable accuracy.

II. RELATED WORK

A binary classification method for IoT/non-IoT devices based on domain analysis within DNS queries has been proposed [5]. Feature vectors were created using Word2Vec, and classification was performed using six machine learning models (Naive Bayes, Logistic Regression, k-means, SVM, Decision Tree, Random Forest). As a result, the accuracy when using Random Forest ranged from a highest of 99.1 % to a lowest of 83.0 %.

Regarding multi-class classification of IoT devices, a method has been proposed to identify devices by creating fingerprints for each device based on domain names in DNS queries and their frequencies, and calculating cosine similarity [6]. Each device's fingerprint included four elements: the number of time windows, the query probability for each domain, the IDF value for each domain, and a threshold. The results showed that when the threshold was set at 0.1 %, it was possible to accurately classify 50 out of 52 devices. Furthermore, a classification method has been proposed using a neural network trained with DNS queries as input [7]. Classification was performed by switching the power of the IoT device to make it perform its initial operation, and utilizing the DNS queries generated during that time. To treat DNS queries that differ only in their subdomains as the same, the

This work is partially supported by JSPS KAKENHI Grant Number 24K02934.

SLD (Second-Level Domain) was hashed and used as the input value for the learning model. As a result, the proposed method achieved 93 % accuracy at the vendor level and 82 % accuracy at the device level.

These studies require long-term packet observation, leading to high learning costs. This is because IoT devices do not communicate frequently, but rather communicate when specific operations (such as starting cleaning or switching power) are performed. To address this challenge, this paper focuses on the fact that IoT devices communicate with specific cloud servers during their initial network connection, and aims to identify them with low learning costs by using the DNS queries made at that time. Previous studies also used DNS queries immediately after connection [7]. However there are concerns about the cost of building a switching system and the impact on devices due to frequent switching, as the power of IoT devices is frequently turned on and off to acquire data. Furthermore, vectorization is performed on the SLD, which may lead to a loss of information for highly unique domains. This paper proposes a method that more accurately reflects domain uniqueness by using the entire domain within DNS queries and achieving network connection switching by performing packet filtering of IoT devices on the gateway router.

III. PROPOSED METHOD

A. Usefulness of DNS Queries

Devices connected to the network communicate by specifying the domain of the communication partner, and go through several DNS servers in the process. In the case of IoT devices, they often communicate with specific cloud servers, and unique domains can be confirmed within DNS queries depending on the device type and vendor. Furthermore, since communication behavior differs depending on the presence or absence of network connectivity, it is expected that IoT device identification can be achieved by observing DNS queries immediately after connection.

B. Configuration and Operation Sequence of IoT Devices

Most IoT devices are deployed on-site, and their vendors often provide smartphone applications to acquire device data and cloud services to store the collected data. Therefore, they are connected to cloud servers, and have a self-healing system against failures for sustainable operation. For example, if communication with the cloud server fails, after several retries, the device restarts and proceeds to the initial setup procedure. Conversely, by blocking such communication, it is also possible to intentionally transition the device to its initial state.

C. Communication Behavior of IoT Devices When Network is Disconnected

This paper focuses on the traffic generated when IoT devices are disconnected from the network and enter an initial state. Figure 1 shows the overall flow of communication patterns of IoT devices during network disconnection. When disconnected

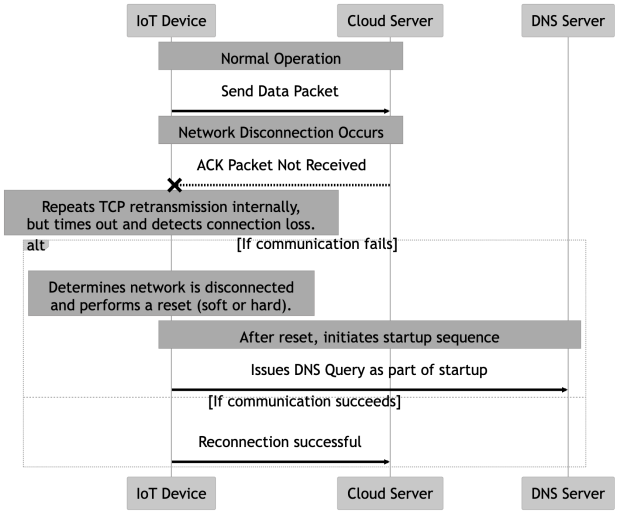

Fig. 1. Communication patterns of IoT devices during network disconnection

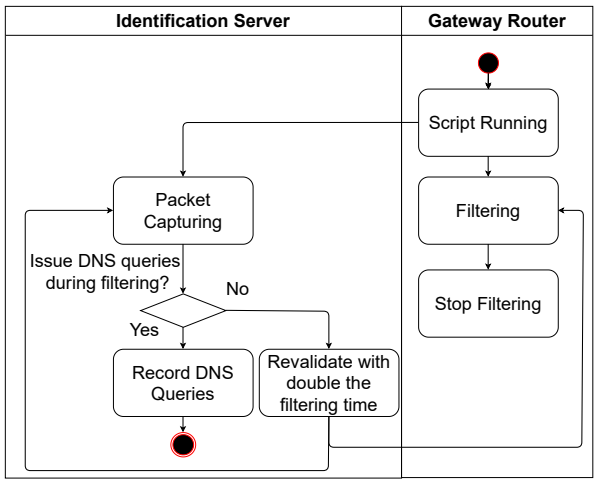

Fig. 2. Dynamic Filtering

from the network, the device can no longer receive ACK packets from the communication partner for the data packets it sent. It retransmits packets each time the TCP retransmission timer expires. If ACK packets are still not received, the TCP stack terminates the connection. At this point, it detects a communication failure from the network and performs a soft reset or a hard reset. Subsequently, as part of the initial operation to establish communication with the cloud server, it issues a DNS query to resolve the server domain name.

D. Acquiring DNS Queries through Packet Filtering

Considering the communication patterns described above, DNS queries are acquired by controlling the communication of IoT devices at the gateway router. However, the filtering time required for the initial state transition varies for each device, necessitating dynamic changes to the filtering time. This paper executes the flow shown in Figure 2 to acquire traffic data after connection. For IoT devices where DNS queries are confirmed during filtering, it is determined that

the device has been reset and has transitioned to its initial state. The DNS queries issued by the device after filtering is lifted (network reconnection) are then recorded. For IoT devices where no DNS queries are confirmed, it is considered that the filtering time is insufficient. Therefore, the operation is repeated by setting twice the current filtering time as the new filtering time. The initial value for the filtering time is set to 30 seconds.

E. Feature Vector and Dataset Creation

This section describes how to create feature vectors from domain names in DNS queries. Domain names consist of lowercase English letters (a-z), numbers (0-9), dots, and hyphens, totaling 38 distinct characters. For Bag-of-Words, we process each character individually, while for Word2Vec, we split the domain by dots. For example, the domain *www.company.com* is split as follows.

- Bag-of-Words : [”w”, ”w”, ”w”, ”.”, ”c”, ”o”, ”m”, ”p”, ”a”, ”n”, ”y”, ”.”, ”c”, ”o”, ”m”]
- Word2Vec : [”www”, ”company”, ”com”]

For Bag-of-Words, we prepare a 38-dimensional array (with all elements initially set to 0). When the K-th character among the 38 types (in the order of a-z, 0-9, ”.”, ”-”) appears in the domain name, we increment the K-th element of the array to create the feature vector. For Word2Vec, we use the Gensim library provided in Python, setting a window size of 3 to create 38-dimensional vectors. The internal algorithm used is CBOW. Furthermore, for data labeling, we perform two types of labels: device name and vendor name, and we will confirm the identification accuracy using three types of machine learning models: k-means, SVM, and Random Forest.

F. Conversion to a Single Vector

It is necessary to create a single feature vector from multiple DNS queries issued by a specific IoT device. In this paper, we consider the set of queried domains as a ”document” and the domains within it as ”words,” and calculate the TF-IDF value for each domain. By multiplying the feature vector created from the domains with the TF-IDF values, we create a single feature vector for each device.

IV. IMPLEMENTATION

A. Evaluation Environment

In this paper, we use the network constructed within the Sugimoto Campus of Osaka Metropolitan University as the evaluation environment. Figure 3 shows the network environment for acquiring traffic data from IoT devices. IoT devices installed in each room on the floor are connected to access points installed in the same room, and all access points are connected to one switch. An Identification Server for packet capture and a Gateway Router are placed upstream of the switch, and this router performs packet filtering for the IoT devices. The NAT function is disabled within the access points and the router, and each IoT device has a fixed IP address.

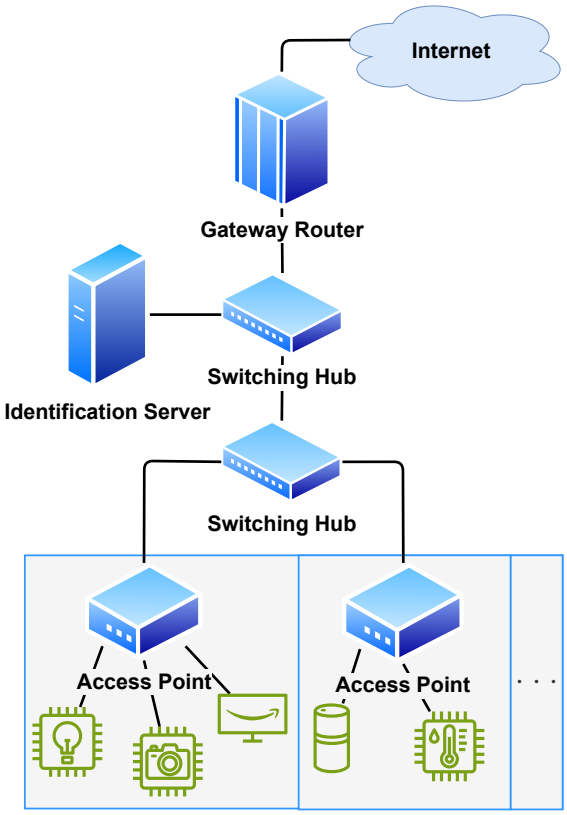

Fig. 3. Network environment for acquiring traffic data from IoT devices

TABLE I
LIST OF IOT DEVICES

Device Name	Vendor Name	Number of Devices
SmartHub	B	6
Plug1	B	3
Camera	D	1
Plug2	D	4
SmartTV	F	1
Cleaner2	G	2

B. Evaluation Metrics

As evaluation metrics for the learning model, we use accuracy, macro-averaged precision, macro-averaged recall, and macro-averaged F1-score. In the case of multi-class classification, it is necessary to average across classes, so we adopt the macro average.

C. Experimental Data

We collect traffic data of IoT devices using the network built within the Sugimoto Campus of Osaka Metropolitan University. In this experiment, we use *Scapy*, a Python library for packet capture on the Identification Server. The data acquisition period is 6 days, from June 6 (Fri) to June 11 (Wed). Of the data acquired, 80 % is used as training data to train the model, and the remaining 20 % is used as test data to measure the accuracy of device identification. Table I shows the IoT devices targeted for device identification.

TABLE II
LABELED BY VENDOR NAME

Vectorization	Model	Accuracy	Precision	Recall	F1-score
Bag-of-Words	k-means	0.9916	0.9952	0.9804	0.9876
	SVM	0.9895	0.9819	0.9919	0.9866
	RF	0.9895	0.9935	0.9930	0.9933
Word2Vec	k-means	0.8634	0.7245	0.6773	0.6909
	SVM	0.8739	0.8771	0.7018	0.7162
	RF	0.8761	0.7915	0.7211	0.7408

TABLE III
LABELED BY DEVICE NAME

Vectorization	Model	Accuracy	Precision	Recall	F1-score
Bag-of-Words	k-means	0.8634	0.9459	0.7949	0.7833
	SVM	0.8613	0.9417	0.7984	0.7821
	RF	0.8739	0.9526	0.8226	0.8211
Word2Vec	k-means	0.7269	0.5479	0.5636	0.5446
	SVM	0.7521	0.7413	0.5670	0.5485
	RF	0.7353	0.6011	0.5909	0.5861

V. RESULT

First, Section V-A describes the learning results of the proposed method. Next, Section V-B verifies the time required for stable device identification. Finally, Section V-C verifies device identification without network disconnection. In all experiments, 80 % of the packet data is randomly allocated for training data and 20 % for testing data. Learning is performed using two types of feature extraction algorithms (Bag-of-Words, Word2Vec) and three types of learning models (k-means, SVM, Random Forest). The learning models used are the machine learning library *Scikit Learn* provided in Python, and all parameters are set to their default values.

A. Identification of IoT Devices by Proposed Method

1) Identification Results: The results of training using data from 120 seconds after network connection are shown in Tables II and III. When labeling by vendor name, very high accuracy is obtained using Bag-of-Words. However, when using Word2Vec, all learning models result in an accuracy rate below 90 %.

Similarly, when labeling by device name, high accuracy is obtained using Bag-of-Words. However, the accuracy is slightly lower compared to when labeling by vendor name. On the other hand, when using Word2Vec, even with SVM learning which yields the best results, the accuracy rate remains slightly above 75 %, and the overall accuracy is low.

2) Discussion on Identification Results: Regarding the decrease in identification accuracy for each device name, the main cause is thought to be that IoT devices from the same vendor access common domains. The confusion matrices for the learning results using vendor names and device names for labeling, with Bag-of-Words and RandomForest, are shown in Figures 4 and 5, respectively. Figure 5 shows that the main misclassification in labeling by device name is the classification of data obtained from Plug1 as SmartHub. Upon examining the actual packet data, SmartHub and Plug1, which are from the same vendor, access common domains, and both issue queries only to those domains. In this case, we

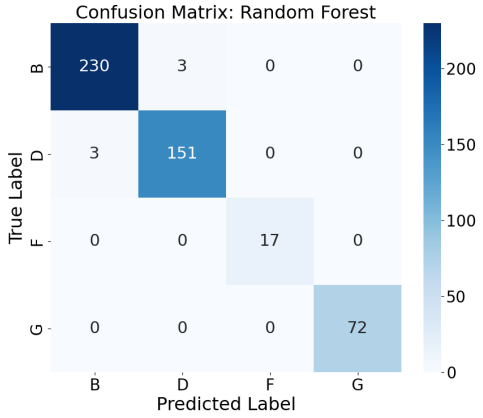

Fig. 4. Learning results of Random Forest model when labeled by vendor name (Bag-of-Words)

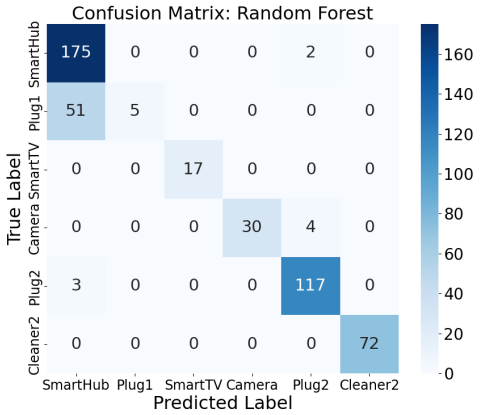

Fig. 5. Learning results of Random Forest model when labeled by device name (Bag-of-Words)

believe that misclassification occurred because the proposed method creates identical or very similar vectors between the two devices.

Furthermore, we will consider the point that the learning results when using Word2Vec are generally not favorable. Word2Vec creates feature vectors using a neural network. Since the weights in the intermediate layer can take negative values during the learning process, both positive and negative values appear in the elements of the generated feature vectors. In the proposed method, for a set of domains accessed by a specific device, after creating features from each domain, the TF-IDF for each domain is calculated and multiplied by the feature vector. In this case, the positive and negative elements of the feature vector may cancel each other out, and the original domain's characteristics may not be reflected. We believe that correct classification is more difficult with the final feature vector obtained from the set of domains.

B. Verification of Time Required for Stable Device Identification

In the previous section, device identification has been performed using packet data generated within 120 seconds after an IoT device reconnected to the network, achieving high accuracy for each vendor and device. This section verifies

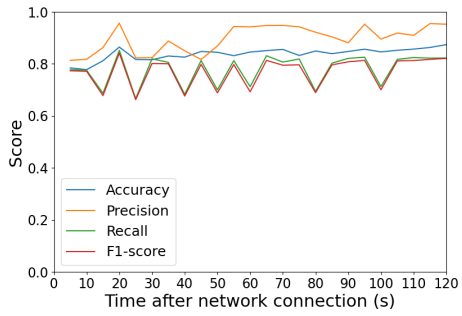

Fig. 6. Transition of learning results when varying the amount of data when using labels by device name and using Random Forest（Bag-of-Words）

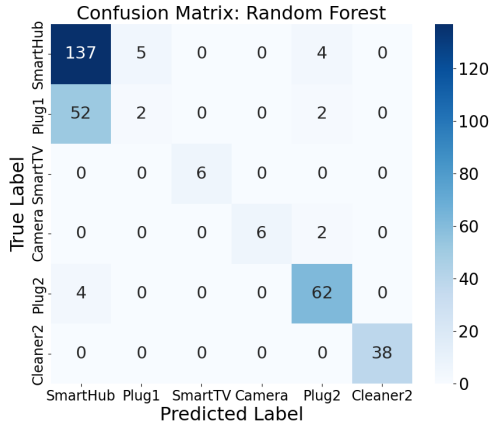

Fig. 7. Learning results by Random Forest when using packet data for 20 seconds from network connection（Bag-of-Words）

the time required for stable and accurate device identification. The verification is conducted using Random Forest, which has yielded the highest accuracy in labeling by device name.

1) Verification Procedure and Results: As a verification method, we increase the packet data used for training by 5 second intervals: 5 seconds, 10 seconds, and 15 seconds after network connection, and then confirm the classification accuracy for each case. The results of the training using Random Forest are shown in Figure 6. The Accuracy shows a stable trend from 30 seconds after network connection onwards, while the Precision shows a stable trend from 50 seconds onwards. Recall and F1-score show similar trends, with some numerical fluctuations for a period after network connection.

2) Discussion on Verification Results: Let's consider the point that a certain level of numerical values is obtained even within 5 seconds after network connection. Figure 7 shows the confusion matrix of the learning results by Random Forest when using packet data for the first 5 seconds after network connection. From Figure 7, it can be seen that a large number of queries occur even in a short time after connection. Therefore, the initial operation of IoT devices immediately after network connection is crucial for IoT device identification with low learning cost.

On the other hand, for SmartTV and Camera, the number of classified devices is small, and it cannot be said that device

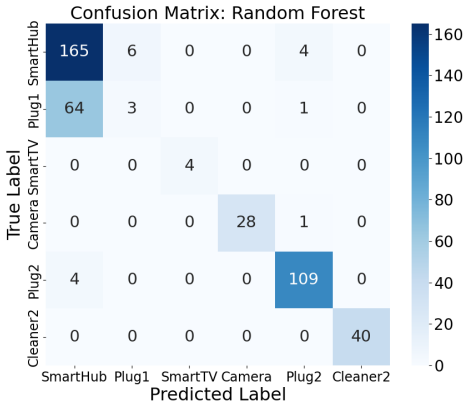

Fig. 8. Learning results by Random Forest when using packet data for 30 seconds from network connection（Bag-of-Words）

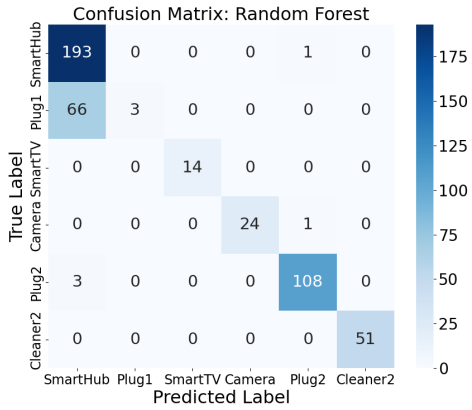

Fig. 9. Learning results by Random Forest when using packet data for 60 seconds from network connection（Bag-of-Words）

identification is sufficiently performed. Figures 8, 9 show the confusion matrices of the learning results by Random Forest when using data for 30 seconds and 60 seconds after connection, respectively. By using data for 30 seconds after connection, Camera achieves a classification count comparable to that in Figure 5, and by using data for 60 seconds after connection, SmartTV achieves a comparable classification count. Therefore, it can be said that most of the devices used in this experiment can be stably identified by utilizing traffic data for 60 seconds after network connection.

C. Device Identification Without Network Connection Switching

In previous experiments, devices have been identified using traffic data from their initial operation by disconnecting them from the network through filtering their communication within the Gateway Router. This section verifies device identification when using normal IoT device traffic data without performing such processing.

1) Verification Method: Traffic data was collected for six days from February 4 (Tue) to February 9 (Sat) from the network within the Sugimoto Campus of Osaka Metropolitan University. From the obtained traffic data, data for consecutive x seconds is extracted, and feature vectors are created using

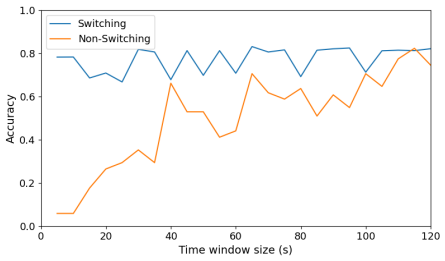

Fig. 10. Transition of learning results with and without network switching

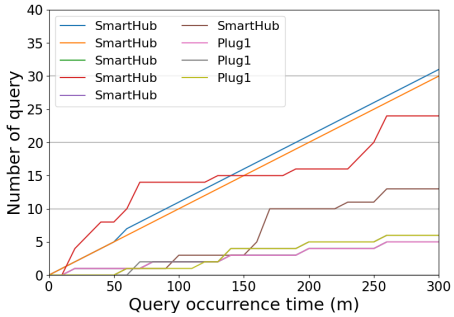

Fig. 11. Number of DNS queries and time of occurrence

Bag-of-Words and TF-IDF. The created dataset is given as input to three learning models (k-means, SVM, Random Forest) for learning. It is possible that the extracted dataset may not contain DNS queries from specific IoT devices, and not all devices may be subject to classification. If accuracy, which is the correct answer rate for the entire test data, is used as an evaluation metric for the learning results at this time, a high numerical value will be obtained as a result of the overall learning, even if there are IoT devices that are not classified. This is not appropriate from the perspective of IoT device identification.

Therefore, this time, the learning evaluation is performed by defining Accuracy as the average recall for each IoT device (0 if there is no data). This allows the presence of IoT devices that are not classified to be reflected in the results.

2) Verification Results: First, using the defined Accuracy as an evaluation metric, training for device identification is performed using traffic data from 60 seconds after network connection and Bag-of-Words. Next, training for device identification without switching network connections is performed, and these two results are shown in Figure 10. In the case where switching is not performed, significant variations in the results occur even when the window sizes do not differ much. On the other hand, when switching is performed, relatively stable accuracy is maintained.

3) Discussion on Verification Results: A contributing factor to the results is the frequency of DNS queries generated by IoT devices. For example, Figure 11 shows the number of DNS queries and their occurrence times for some IoT devices used in the experiment. It can be seen that considerable time is required to collect data necessary for device identification, as they do not issue DNS queries frequently.

On the other hand, since a certain amount of DNS queries are generated immediately after network connection, it can be said that utilizing traffic immediately after network connection is highly effective for rapid IoT device identification.

VI. CONCLUSION

We have proposed a method for identifying IoT devices using traffic data immediately after a device connects to the network. Immediately after connection, they perform initial operations that involve communication with specific servers. Consequently, they issue DNS queries to domains that are characteristic of each device. By extracting features from this traffic data and applying machine learning, we have demonstrated that device identification is possible with low training costs and high accuracy. As a result, we have achieved an accuracy exceeding 99 % at vendor level and over 87 % at device level. Furthermore, we have shown that stable device identification for all devices is possible within 60 seconds after an IoT device reconnects to the network.

We also have investigated device identification using DNS queries that IoT devices typically issue. The results have shown significant variations in accuracy compared to our proposed method, indicating that stable IoT device identification is difficult with data volumes of around 60 seconds. From these findings, it can be concluded that using DNS queries immediately after network connection is highly effective for rapid and low-cost IoT device identification.

Finally, we describe future challenges. The proposed method can identify devices with high accuracy, but it is difficult to accurately identify devices when different IoT devices access the same domain and issue queries only to that domain. Furthermore, the devices used in this paper are registered on a network built within university campus, and their types and numbers are limited. Therefore, it is necessary to conduct verification in a wider range of networks.

REFERENCES

[1] Statista Research Department, "Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2034," May 2025. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[2] Hitachi Solutions Create, Ltd., "What Are IoT Devices? Types, Fields of Use, Examples, etc." July 2022. [Online]. Available: https://www.hitachi-solutions-create.co.jp/column/iot/iot-device.html

[3] C. Hesselman, M. Kaeo, L. Chapin, K. Claffy, M. Seiden, D. McPherson, D. Piscitello, A. McConachie, T. April, J. Latour, and R. Rasmussen, "The DNS in IoT: Opportunities, Risks, and Challenges," *IEEE Internet Computing*, vol. 24, no. 4, pp. 23–32, July 2020.

[4] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions and Research Directions ," *Mobile Networks and Applications*, vol. 28, no. 1, pp. 296–312, February 2023.

[5] I. Ayoub, M. S. Lenders, B. Ampeau, S. Balakrichenan, K. Khawam, T. C. Schmidt, and M. Wählisch, "Understanding IoT Domain Names: Analysis and Classification Using Machine Learning," *arXiv preprint*, vol. abs/2404.15068, April 2024.

[6] R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis, "IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis," in *Proceedings of the 2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, September 2020, pp. 474–489.

[7] O. Thompson, A. M. Mandalari, and H. Haddadi, "Rapid IoT Device Identification at the Edge," in *Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning*, October 2021, pp. 22–28.