WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IoT Device Recognition Using Device Fingerprinting

Dr. T.Sunil Kumar[1], G. Nishitha[2] , P. Vinuthna[3] , R. Sathvika Reddy[4] and T. Nikhitha[5]

[1]Professor, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, [2,3,4,5] Under Graduate Student, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad

A B S T R A C T

The Internet of Things (IoT) has multiplied in recent years, resulting in a profusion of linked devices in various areas, from home automation to industrial control systems. As a result, device recognition techniques must be accurate and effective. Device fingerprinting involves analyzing network behavior, communication patterns, and hardware features. The proposed method analyzes and categorizes these device fingerprints using machine learning algorithms. The model demonstrates exceptional accuracy in detecting a variety of devices, including sensors, actuators, and smart appliances, and determining whether the device is suspicious or legitimate based on the features mentioned above using two public datasets after rigorous testing and validation on several IoT device datasets. It also has a low computational overhead and can detect new and unknown devices, making it suitable for real-time deployment. The abstract concludes with a discussion of the advantages of employing device fingerprinting for IoT device identification, such as improved security, better network management, and greater visibility into IoT device behavior.

Keywords: Internet of Things (IoT), Device Recognition, Device Fingerprinting, Machine Learning, and IoT Security

1. Introduction

In the ever-evolving technology landscape, the IoT has emerged as an influential force connecting billions of smart devices to improve our lives and transform industries. As the IoT ecosystem grows, the demand for robust and efficient device recognition techniques becomes increasingly important. The essence of the IoT lies in the seamless exchange of data between interconnected devices. However, because of their huge number, variation, and lack of standardized identification protocols, recognizing and managing these various devices creates unique challenges.

Traditional approaches, such as IP-based tracking, MAC address filtering, and IMEI, are inadequate because of constrained resources including bandwidth, battery life, and computing power; they often fail to capture the complexities of the IoT ecosystem, limiting security, management, and optimization efforts.

This paper advances into the fascinating field of "IoT Device Recognition Using Device Fingerprinting", an innovative approach to identifying and classifying IoT devices in a network effectively through network packets. An effective DI system can identify devices in the network and enable the implementation of necessary security measures, such as upgrades, access restrictions, or isolating vulnerable devices. This model operates at the packet level using adaptable features, ensuring versatility and high detection accuracy.

This sophisticated technique helps for precise identification by using the unique traits and behavioral patterns of IoT devices, consequently improving network security, simplifying device administration, and facilitating personalized user experiences. This paper emphasizes the underlying principles and procedures of IoT device recognition utilizing device fingerprinting. We explore the key components of device fingerprinting, including data collection, feature extraction, and classification algorithms

2. Literature Survey

ML for the Detection and Identification of Internet of Things Devices [2022], This research analyzes machine learning methods for detecting and identifying Internet of Things (IoT) devices, including deep learning strategies like recurrent and convolutional neural networks, supervised, unsupervised, and semi supervised learning. It also explores challenges like limited computational resources, energy constraints, and data privacy concerns, evaluating their effectiveness, efficiency, and scalability and Zero-Bias Deep Learning for Accurate Identification of Internet-of-Things (IoT) Devices [2021]:

The study presents a novel approach to accurately identify IoT devices using zero-bias deep learning algorithms. This approach addresses the issue of unbalanced data distributions, resulting in biased predictions. It incorporates bias-correction layers and label-smoothing regularization, making it applicable across various IoT domains and sectors.

An Approach to IoT Device Identification Using Semi-Supervised Learning [2020] and Automated IoT device identification using network traffic

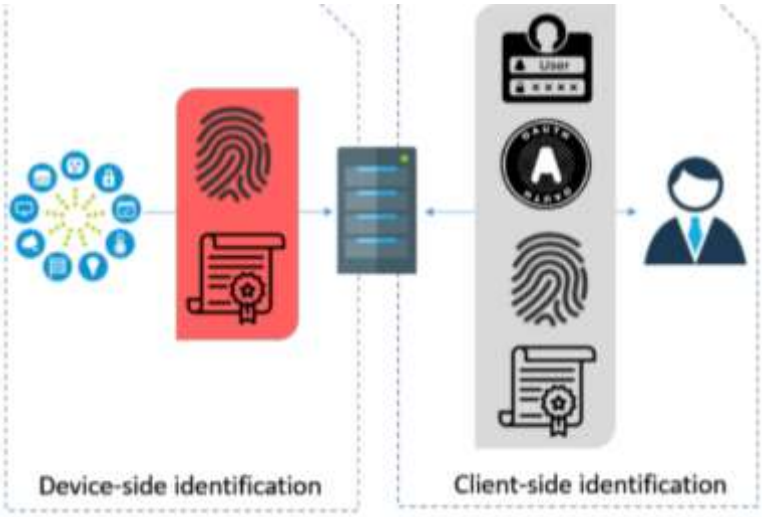
papers present two approaches to IoT device identification: one using semi-supervised learning, which improves accuracy and efficiency in identifying diverse devices within complex networks, and another using network traffic pattern analysis. Both methods demonstrate the importance of semi supervised learning in managing large-scale IoT deployments and addressing challenges caused by various device types and communication patterns. Both studies contribute to the advancement of IoT device identification and network management strategies.

Additionally, AuDI: Toward Autonomous IoT Device-Type Identification Using Periodic Communication The paper proposes AuDI, a machine learning approach that uses periodic communication patterns to identify IoT device types autonomously. This method, which uses unique temporal patterns in data transmissions, can be highly advantageous in situations where devices are not directly visible or interact with people, improving security protocols and network resource allocation.

3. Methodology

3.1 Device Fingerprinting for IoT Device Recognition:

The methodology employed for IoT device recognition using device fingerprinting centers on the concept of creating unique fingerprints for each IoT device based on their distinct characteristics and communication patterns. Without depending on more conventional identification techniques like IP or MAC addresses, this method allows precise and effective identification of IoT devices within the network. It uses feature extraction and selection to create unique fingerprints, machine learning algorithms like decision trees, SVM, or kNN, chosen based on the nature of the data and accuracy goals for classification, and thorough performance evaluation for validation. Following these steps enables the method to identify devices accurately and quickly, ensuring improved security and easy control within the IoT ecosystem Additionally, the comparison with previous works helps demonstrate this methodology's advancements and contributions to IoT device recognition.



3.2.1 Data Collection:

Then, the system asks the user for network traffic data generated by IoT devices. Here, we used two public datasets for data collection. Both contain real device data. First, in the Aalto dataset, there are 31 devices. Second, the UNSW dataset contains network traffic data that can be used to simulate IoT communication patterns. IP addresses are only widely used for network intrusion detection. We used the larger UNSW data set to measure the modeling approach's broader generality, while the smaller Aalto data set was used to formulate the modeling approach.

3.2.2 Individual, Aggregated, and Mixed Method Packets:

The system then processes the collected information and extracts relevant features. These extracted features form the basis for creating device fingerprints from network packets. The methods are individual, aggregated, and mixed approaches. Individual packets are based on unique identifiers like MAC or IP addresses to differentiate devices. However, this method can be limited in its ability to discriminate between devices and behaviors. To address this, an aggregated method where packets sharing the same identifier attribute and individual packet ML labels are grouped before merging This approach aims to enhance accuracy by disregarding mislabeled packets. The mixed method addresses issues with the aggregation process and involves using individual labels for exceptions. This method is proposed to deal with scenarios in which MAC addresses are affected by transfer issues, resulting in merging issues. Therefore, while these fingerprinting strategies offer a novel way to categorize network packets, careful consideration of their implications in different network contexts is necessary to ensure accurate device recognition.

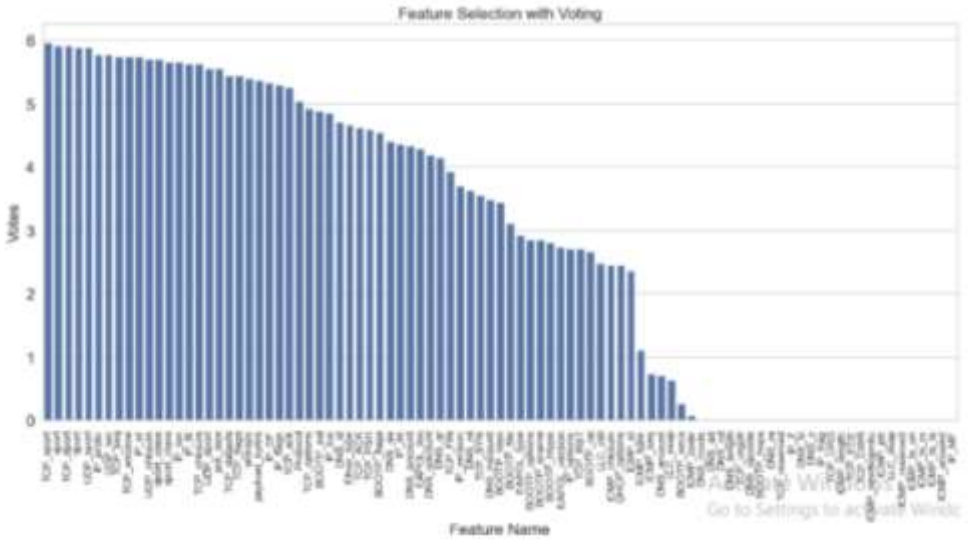
3.2.3 Feature Extraction:

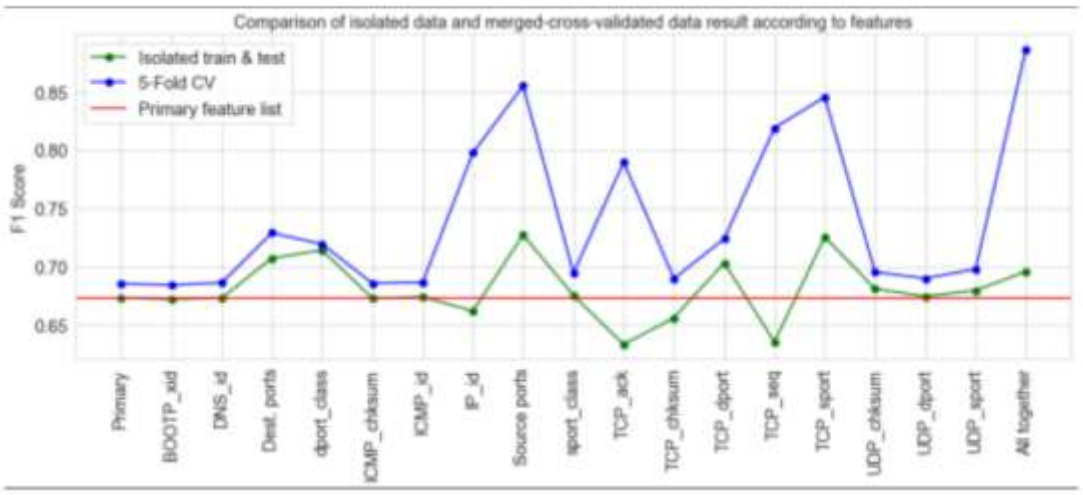
Features are extracted from packet headers in pcap (packet capture) files, focusing on network traffic analysis. Before this, the PCAP files were separated into distinct training and testing sets to ensure complete isolation. With 20 sessions per device, the Aalto dataset is divided into two subsets: 16 for training and 4 for testing, following an 80% training and 20% testing distribution. In a similar vein, training and testing data for the UNSW dataset are chosen from a number of days' worth of captures. 111 features that were taken from the packet headers and added to the payload entropy, source-destination port classes, and protocol information from the TCP-IP layers, all of which were shown to be useful in earlier studies. Details about payload characteristics are provided by payload entropy, while summaries of source and destination ports are provided by protocol and port class features. Since MAC and IP addresses uniquely identify devices but do not provide information about behavior, they are purposefully removed as features.

3.2.4 Feature Selection:

Once the features have been extracted, they need to be selected for the device fingerprint. The most relevant and distinctive features that significantly contribute to device identification are identified in this step. It also helps reduce computational overhead and enhance the efficiency of the next process.

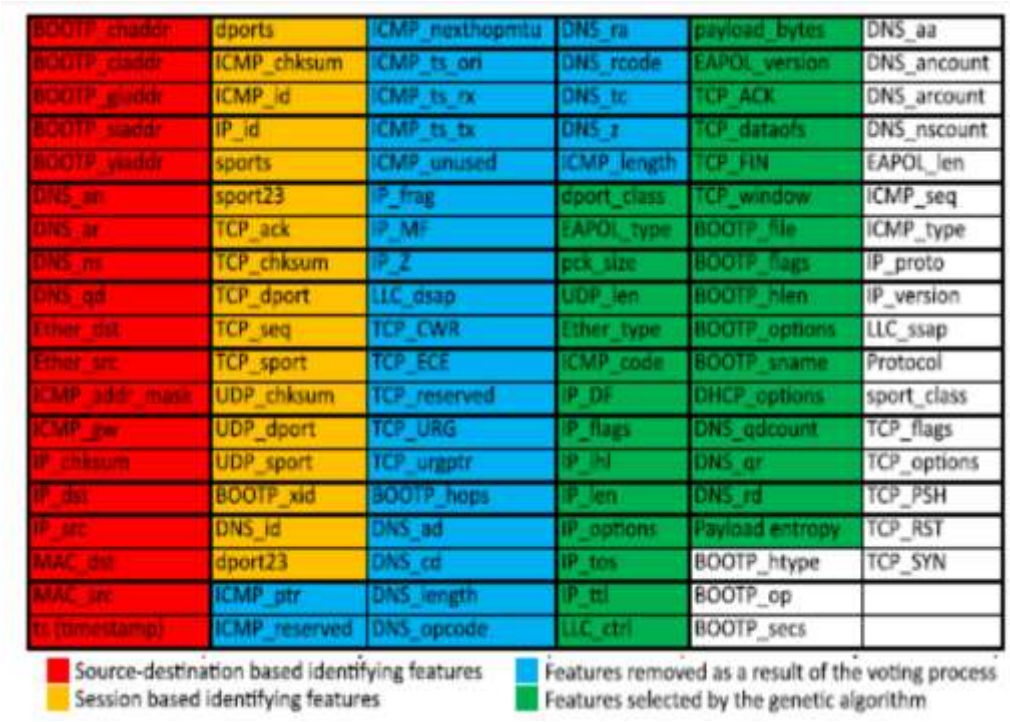
The initial feature pool obtained is refined by utilizing a voting method using the xverse package. This method uses six different scoring algorithms to evaluate the importance of features across devices and relies on user voting to decide which features should be included.





Additionally, we got rid of 26 features because none of the devices received any votes for them. Moreover, we discovered that characteristics based on ports showed potential for generalizing models. While a few port numbers were generalized among several device instances, most of them appeared to be session-specific. In order to resolve this, discrete values that represented port classes and protocols were mapped to raw port-based feature values. After removing redundant and identifying characteristics, a Genetic Algorithm (GA) was employed to choose the best feature subset from the remaining

pool. Acting as a wrapper method, the GA used a Decision Tree (DT) classifier to verify the utility of the feature set, which resulted in a refined feature subset that not only improved detection performance but also decreased model complexity.

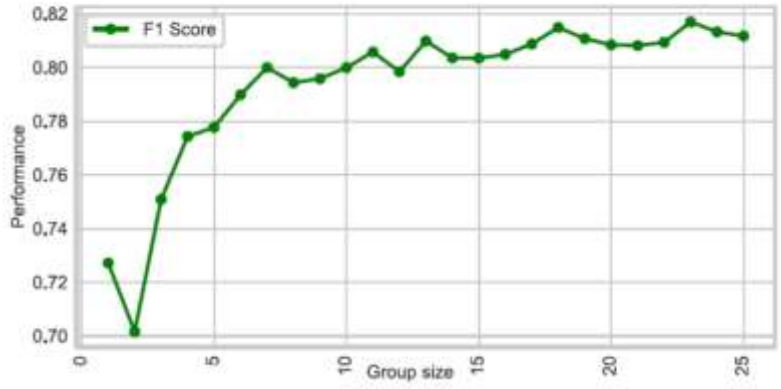


3.2.5 Algorithm selection:

This work explores machine learning (ML) algorithms for predicting device types from extracted features. Six algorithms, including RF, kNN, GB, DT, NB, and SVM, are evaluated using the Aalto data set. Employing random search and nested cross-validation, the study tunes hyper parameters for each algorithm. RF, DT, and GB emerge as the top performers in terms of device identification (DI). DT and NB are the fastest in inference time, but NB's accuracy is notably low. Despite kNN and GB's decent accuracy, their slow processing speeds render them impractical for real-time usage. Similarly, SVM's speed and accuracy aren't ideal. DT is chosen as the most suitable algorithm for further investigation. Its blend of speed and accuracy makes it well-suited for real time device detection systems operating in fast-paced network traffic scenarios.

4. Results and Analysis

The results are shown for three different versions of the method—individual, aggregated, and mixed—depending on the model that was chosen. Initially, we looked into how the context of the aggregation algorithm affected the group size vs performance relationship. There is a positive correlation, shown in Fig. 1, indicating that larger groups are more effective. Larger group sizes may not always be feasible, though many IoT devices communicate occasionally. In considering this, a group size of 13 was selected, which is roughly the point at which performance begins to level off.



The connection between performance and total group size, It seems that the plateau is at g = 13.

The overall outcomes of the two datasets are shown in the table.

Method	Dataset	Accuracy	F1 score	Test-t	Alg-t
Individual	Aalto	0.705±0.001	0.727±0.001	0.004	0.000
	UNSW	0.853±0.010	0.834±0.012	0.008	0.000
Aggregated	Aalto	0.745±0.011	0.809±0.005	0.007	0.164
	UNSW	0.943±0.012	0.937±0.017	0.017	0.425
Mixed	Aalto	0.833±0.002	0.861±0.004	0.008	0.216
	UNSW	0.941±0.012	0.935±0.017	0.022	0.479

Table Results:

Within the UNSW and Aalto datasets, an F1 score of 73% and 83%, respectively, was obtained using the individual packet technique. Aggregation, however, was found to significantly improve both dataset's capacity for accurately identifying devices. The overall F1 score increased to 81% on the Aalto dataset and nearly 94% on the UNSW dataset. Given that feature extraction on the UNSW dataset was limited to the Aalto dataset, it is extremely intriguing to witness such a high degree of discrimination.

This is a clear indication that the selected feature set is highly adaptable to various IoT environments. The table shows the average device-level discrimination performance over the DT models for the Aalto dataset. Table 1 illustrates how highly unbalanced the data is, so in order to provide a meaningful picture of the model's performance at the device level, we need to select a metric that is relatively insensitive to class size imbalance. This is why we use the F1 scores. This has previously been done with accuracy, a metric that isn't suitable for unbalanced datasets

	Packet statistics		Packet discrimination		
Device name	Packets	Percent	Individual	Aggregated	Mixed
Aria	441	0.420	0.932	1.000	1.000
D-LinkCam	6244	5.940	0.891	1.000	0.988
D-LinkDayCam	1063	1.010	0.864	1.000	1.000
D-LinkDoorSensor	1892	1.800	0.762	0.057	0.788
D-LinkHomeHub	8595	8.180	0.681	0.797	0.777
D-LinkSensor	6549	6.230	0.382	0.644	0.626
D-LinkSiren	6186	5.890	0.367	0.640	0.631
D-LinkSwitch	6519	6.200	0.665	0.969	0.964
D-LinkWaterSensor	6435	6.120	0.392	0.624	0.622
EdimaxCam	831	0.790	0.872	1.000	1.000
EdimaxPlug1101W	1160	1.100	0.601	0.827	0.824
EdimaxPlug2101W	1010	0.960	0.453	0.708	0.699
EdnetCam	408	0.390	0.833	1.000	1.000
EdnetGateway	683	0.650	0.908	1.000	1.000
HomeMaticPlug	611	0.580	1.000	1.000	1.000
HueBridge	13936	13.260	0.810	0.217	0.815
HueSwitch	18448	17.560	0.891	0.720	0.891
IKettle2	145	0.140	0.727	1.000	1.000
Lightify	4149	3.950	0.977	1.000	1.000
MAXGateway	567	0.540	0.964	1.000	1.000
SmarterCoffee	149	0.140	0.727	0.983	0.981
TP-LinkPlugHS100	667	0.630	0.693	0.748	0.746
TP-LinkPlugHS110	636	0.610	0.429	0.336	0.328
WeMoInsightSwitch	5962	5.670	0.667	0.874	0.866
WeMoLink	6625	6.300	0.638	0.929	0.924
WeMoSwitch	4477	4.260	0.511	0.769	0.768
Withings	688	0.650	1.000	1.000	1.000

Table Device proportions over complete dataset Examining the device-based results of the Aalto data set, it is found that almost all devices benefit from the aggregation algorithm, but four are negatively impacted. It is because of the transfer problem that affects the pairing that makes up these four devices (they share the same MAC address). To address this, a hybrid method of adding an exception to the aggregate mechanism was employed. It can be observed from Table IV that, when the hybrid strategy was used, the total F1 score of the dataset increased from 81% to 86% on the Aalto dataset. Since

the UNSW dataset does not have a transfer problem, this strategy has little effect on the outcome. (Table VI). It should be noted that the Aalto dataset outperforms the UNSW dataset significantly. It appears to be the result of specific device groupings.

A confusion matrix is shown in Fig. 1, which contains only low-performance devices. In this case, the devices in the subgroup have certain things in common: either they are different models of the same product (like the red and green groups in Fig. 5) or they are similar-purpose products manufactured by the same companies (like the yellow, blue, and orange groups in Fig. 5). It doesn't seem possible to completely separate these devices based on their behavior at the network level. However, they are likely to employ extremely similar hardware and software, exhibiting comparable behavior along with similarities in vulnerabilities and avoidance. Therefore, it would be possible to classify these devices together under a single label from a DI perspective. When the same is done, the accuracy of the Aalto dataset rises from 73% to 100%.

In the context of phishing detection, the streamlined architecture adopted here facilitates efficient training and skip connections play a key role in preserving spatial information. This approach, which achieves good results with fewer parameters, suggests a promising avenue for future research in developing resource-efficient models for phishing detection. The exploration of modules capturing spatial and semantic features, alongside the investigation of alternative network architectures beyond standard CNNs, is encouraged to enhance overall detection performance. Emphasizing resource-saving strategies will be crucial for scaling models and effectively countering evolving phishing threats.

5. Conclusion

Device fingerprinting is a powerful tool for identifying IoT devices. It can be used to track devices, restrict unauthorized access, and enhance network security. There are a variety of methods for generating device fingerprints, including statistical analysis, machine learning, and deep learning. Device finger printing's accuracy is dependent upon the method used and the amount of data available. The use of device fingerprinting for IoT security has gained popularity in recent years. This is due to the proliferation of IoT devices and the sophistication of cyberattacks. The use of device fingerprints ensures a robust and scalable recognition process, facilitating seamless integration and secure management of the interconnected IoT ecosystem. The system's real-time identification capabilities give customers quick access to device-specific data while strengthening security protocols and effectively allocating network resources. The methodology's adaptability and ability to handle class imbalances and evolving IoT environments highlight its potential for real-world implementation.

By providing an autonomous and reliable device recognition solution, IoT Device Recognition using Device Fingerprinting strengthens the foundation for building smarter, interconnected environments in diverse domains, including industrial automation, healthcare, smart cities, etc. To sum it up, IoT device recognition utilizing device fingerprinting has enormous potential to change the IoT landscape by building more interconnected, secure, and effective ecosystems.

6. References

[1] 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," in IEEE Internet of Things Journal, pp. 298–320, 2022.

[2] What is IoT? https://www.oracle.com/in/internet-ofthings/what-is-iot/

[3] What Is Device Fingerprinting, and How Exactly Does It Work? https://seon.io/resources/devicefingerprinting/

[4] Y. Liu et al., "Zero-Bias Deep Learning for Accurate Identification of Internet-of-Things (IoT) Devices," in IEEE Internet of Things Journal, pp. 2627–2634, 2021.

[5] L. Fan et al., "An IoT Device Identification Method Based on Semi-supervised Learning," 16th International Conference on Network and Service Management (CNSM), pp. 1–7, 2020.

[6] Aneja, S., Aneja, N., & Islam, M. S., "IoT Device Fingerprinting using Deep Learning," ArXiv. https://doi.org/10.1109/IOTAIS.2018.860082 4

[7] Aksoy, Ahmet, and Mehmet Hadi Gunes, "Automated IoT device identification using network traffic," in IEEE International Conference on Communications (ICC), 2019.

[8] S. Marchal, M. Miettinen, T. D. Nguyen, A.-R. Sadeghi, and N. Asokan, "AuDI: Toward Autonomous IoT Device-Type Identification Using Periodic Communication," in IEEE Journal on Selected Areas in Communications, pp. 1402-1412, 2019.

[9] C. Duan, H. Gao, G. Song, J., and Z. Wang, "ByteIoT: A Practical IoT Device Identification System Based on Packet Length Distribution," in IEEE Transactions on Network and Service Management, 2022.

[10] Vian Adnan Ferman and Mohammed Ali Tawfeeq, Early Generation and Detection of Efficient IoT Device Fingerprints Using Machine Learning," International Journal on Advanced Science, Engineering, and Information Technology, 2022.