IoT Device Fingerprinting and Impersonation Detection Using Machine Learning

1st Qamar hussain
dept. of Information and Communication Engineering
Islamia University of Bahawalpur
Bahawalpur, Pakistan
qamr805@gmail.com

2nd Kashif Asghar
dept. of Information and Communication Engineering
Islamia University of Bahawalpur
Bahawalpur, Pakistan
prokashif999@gmail.com

3rd Sana Tariq
dept. of Information and Communication Engineering
Islamia University of Bahawalpur
Bahawalpur, Pakistan

4th Aoun Muhammad
dept. of Information and Communication Engineering
Islamia University of Bahawalpur
Bahawalpur, Pakistan

***Abstract*—With the rapid proliferation of Internet of Things (IoT) devices, security has become a significant challenge—particularly impersonation attacks, where an adversary masquerades as a trusted device. In this paper, we introduce a machine learning framework that fingerprints IoT devices and catches impersonation attacks by analyzing their network traffic. We train a Random Forest classifier on statistical features extracted solely from benign traffic of nine commercial IoT devices (N-BaIoT dataset). The model achieves highly accurate device identification with a weighted accuracy and F1-score of 98.63% and can clearly distinguish different device types. For impersonation detection, we use the fingerprinting model's confidence: traffic that does not match any known device with high certainty is flagged as an impostor. The confidence-based anomaly detection scheme, evaluated on Mirai and Gafgyt attack traffic, achieves a 93.5% true positive rate at a 1.2% false positive rate. The approach is entirely passive, requiring no modifications to the devices themselves, is sufficiently lightweight for deployment on resource-constrained edge devices, and produces interpretable results. Our findings demonstrate that network behavioural fingerprinting is a viable and practical security primitive for IoT environments.**

***Index Terms*—IoT security, device fingerprinting, impersonation detection, machine learning, network traffic analysis, Random Forest**

I. Introduction

The Internet of Things (IoT) is expanding rapidly, with billions of connected devices being integrated into smart cities, homes, healthcare, industrial automation, and critical infrastructure. These devices typically lack strong security features: resources are limited, manufacturing standards vary, and security is often not a primary design goal [1]. A major threat is impersonation, whereby an attacker pretends to be a legitimate device to gain network access, steal sensitive information, or launch coordinated attacks [2]. Traditional cryptographic authentication methods may be expensive or impractical on low-power IoT nodes, and signature-based intrusion detection systems struggle to adapt to the diversity and novelty of attacks [1].

Device fingerprinting has emerged as a promising complement to conventional security mechanisms. By extracting unique behavioural or physical characteristics, it is possible to verify device identities passively and detect anomalies indicative of impersonation or compromise [3]. Among various modalities, network traffic-based fingerprinting is particularly attractive because it requires no modifications to the devices themselves, can be implemented at the network edge, and leverages the rich statistical patterns inherent in device communications [4], [5]. Machine learning (ML) techniques can automatically learn these complex patterns and adapt to evolving behaviours, making them well-suited for this task.

This paper presents a comprehensive ML framework for IoT device fingerprinting that simultaneously enables impersonation detection. We utilise the N-BaIoT dataset [2] to train and evaluate a Random Forest classifier. The main contributions of this work are:

- A complete pipeline that performs simultaneous device identification and impersonation detection using only benign traffic, eliminating the need for attack samples.
- Demonstration of high-accuracy device identification (98.63% weighted F1-score, macro F1-score 0.99) using a lightweight Random Forest model.
- A confidence-thresholding scheme that acts as a one-class-inspired anomaly detector, validated on real attack traffic.
- Analysis of feature importance to identify the most critical network characteristics for device discrimination.
- Evaluation on a publicly available dataset, ensuring reproducibility and facilitating comparison with prior work.

The remainder of the paper is organized as follows. Section II presents a comprehensive literature review. Section III details the methodology, including dataset description, preprocessing, model development, and evaluation framework. Section IV presents experimental results and analysis. Section

DOI: https://doi.org/10.67535/tsp.000003.020
Published under the ICACNC 2026 Proceedings

V discusses the implications for impersonation detection, and Section VI concludes the paper and outlines future research directions.

II. Literature Review

This section reviews prior work in IoT device fingerprinting and impersonation detection, categorizing approaches into physical-layer, network-flow-based, machine learning and deep learning methods, and impersonation-specific studies.

A. Physical-Layer Fingerprinting

Physical-layer device fingerprinting exploits unique hardware imperfections such as radio frequency (RF) emissions to create device signatures. Jafari et al. [6] have proved that deep learning can be used to accurately identify ZigBee devices based on their RF fingerprint. In the IoT area, Lomba et al. [7] designed an adaptive RF fingerprint-based authentication system for the Industrial IoT (IIoT) domain to address the issue of environmental drift. More recently, Wang et al. [8] used the fingerprints of the temperature of the microcontroller unit chip and machine learning for intrusion detection in IIoT applications. FPGA device fingerprinting was studied by Zhang et al. [9] for various hardware workloads, revealing the possibilities of hardware based identities. Zhang et al. conducted a comprehensive survey on the physical-layer device fingerprinting from theory to practice [3]. Although these techniques deliver high accuracy, they typically depend on specialized equipment, are subject to environmental disturbances and are incompatible with legacy or off-the-shelf devices.

B. Network-Flow-Based Fingerprinting

An alternative to this is network traffic analysis, which is passive and non-intrusive. Hamad et al. [4] proposed a network-flow based fingerprinting technique to identify IoT devices in a flow-level statistics, which proved to be very accurate in testbeds. More recently, the device identification properties have been proven in a smart home environment with network traces, as shown by Brahma et al. [10]. Sheng et al. [5] provide a survey of network traffic fingerprinting techniques for the identification of iiots, highlighting concept drift, encryption and scalability issues. In general, flow-based methods have a good compromise between accuracy and deployability, needing only a network monitoring infrastructure.

C. Machine Learning for Device Identification

Fingerprinting applications of machine learning algorithms have been wide-spread. Popular algorithms such as Random Forest, Support Vector Machines and k-Nearest Neighbors (kNN) are widely used because of their interpretability and robustness [1]. Aneja et al. [11] introduced deep learning-based image processing device fingerprinting for IoT devices. As proposed by Choi et al. [12] a communication-based identifier system has been introduced which performs continuous verification of device fingerprints based on the patterns of the traffic. Behavioural fingerprinting frameworks, such as the one by Li et al. [13], combine traffic analysis with device-specific behaviours to detect anomalies in smart homes.

Deep learning has also been leveraged for its representation learning capabilities. Feng et al. [14] took a different approach by transforming the traffic sequences into grayscale images and then recognized the devices using deep separable convolution networks. But deep models generally need significant amount of computing power and massive training sets.

D. Impersonation and Attack Detection

Impersonation detection specifically deals with when an adversary tries to pretend to be a legitimate device. Nguyen et al. [2] proposed an impersonation attack detection framework for IoT networks based on machine learning that is able to detect impersonation attacks with high detection rates. Zanna et al. [15] studied impersonation attacks on WAPs, suggesting anomaly detection methods for detection. Galtier et al. [16] proposed a power spectral density (PSD)-based method of fingerprinting to resist mimicry attack for detecting IoT device spoofing.

The data set side, the N-BaIoT [2] was first proposed by deep autoencoders for detecting botnet-compromised devices, which show that anomaly-based detection can detect deviations from normal device behavior. Large language models (LLMs) have recently been used for generating textual descriptions of device traffic patterns for fingerprinting [17] paving the way to explainable security.

We propose a different approach, training a single model with only benign traffic to build a profile per-device to perform simultaneous classification and anomaly detection. This one-class approach is well suited for cases where the type of attacks is unknown and labelled attack samples are lacking.

III. Methodology

Existing fingerprinting methods suffer from a variety of drawbacks, including relying on attack samples, the high computational power needed for these techniques or their limited applicability. Thus, a new lightweight machine-learning framework is proposed, which trains the device identities exclusively by using benign traffic. Our method integrates a single Random Forest model for both explicit device identity and implicit device impersonation detection tasks, but this approach differs from previous work by either only relying on anomaly detection (autoencoders [2]) or forcing deep neural networks with large training sets. This design results in a small and understandable model that can be implemented on network edge devices with minimal impact.

A. Dataset

The N-BaIoT data set [2] is a large collection of network traffic from a set of nine real commercial IoT devices: Danmini Doorbell, Ecobbe Thermostat, Ennio Doorbell, Philips B120N10 Baby Monitor, Provision PT-737E Security Camera, Provision PT-838 Security Camera, Samsung SNH-1011N Webcam, SimpleHome XCS7-1002 Security Camera, and SimpleHome XCS7-1003 Security Camera. Normal operating conditions were used for recording benign traffic for each device. The raw data contains 115 statistical characteristics of

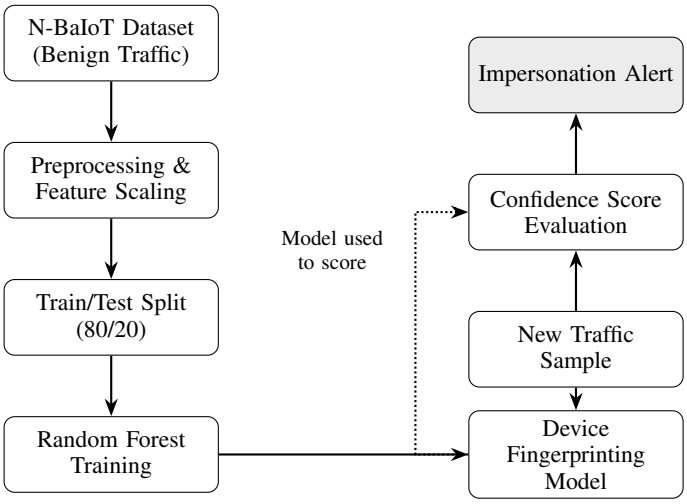

Fig. 1. Workflow of the proposed device fingerprinting and impersonation detection system.

network flows, such as packet size statistics, inter-arrival times, protocol flags, flow-level aggregates etc. There are 111,187 samples in the benign dataset for this study, ranging from 2,622 (for the Ecobee Thermostat) to 35,048 (for the Philips Baby Monitor).

B. Preprocessing

A systematic process of preprocessing was followed on the dataset. Labels were created for non-numeric metadata columns (like device name or traffic type). There were no missing or infinite values in the other 115 numeric features. All of the features were standardized (z-scored) to ensure that features were in the zero mean and unit variance. The preprocessed feature matrix was then used to perform stratified sampling to split the data into 80% training data and 20% testing data while maintaining the class distribution.

The overall flow of our proposed system is shown in Fig. 1.

C. Model Training

The strong performance on high-dimensional data, built-in feature importance, and robustness against overfitting made the Random Forest classifier an appropriate choice [1]. We performed a 5-fold cross-validated grid search over the number of trees (50, 100, 200), maximum depth (10, 15, 20, None), and minimum samples per split (2, 5, 10). The combination of 100 trees, depth 15, and a minimum split of 5 yielded the highest mean validation accuracy (98.5%) and was adopted for the final model. All experiments were carried out in Google Colab with Python's scikit-learn library.

D. Impersonation Detection Principle

After the fingerprinting model is trained, impersonation detection is performed by evaluating the prediction confidence of every incoming traffic sample. If the maximum predicted class probability is below a threshold τ (chosen as 0.8 after analysing the benign confidence distribution, where over 97% of benign samples exceed 0.9), the sample is flagged as anomalous. This one-class-inspired approach allows the detection of novel impersonation attacks because the model was never trained with attack signatures, only with the traffic envelope of legitimate devices. A detailed sensitivity analysis of τ is provided in Section IV-D.

TABLE I
QUANTITATIVE COMPARISON OF THE PROPOSED METHOD WITH RELATED IOT FINGERPRINTING APPROACHES.

Approach	Explicit Device ID	No Attack Samples	Training Time (s)	Model Size (MB)
Autoencoder [2]	No	Yes	∼200	15
CNN [14]	Yes	No	∼300	25
Flow-based [4]	Yes	No	∼30	0.5
PSD-based [16]	No	Yes	∼50	2
This work	**Yes**	**Yes**	**12.3**	**4.2**

E. Evaluation Metrics and Comparison with Prior Work

Model performance is evaluated using common classification metrics of accuracy, precision, recall, and F1 score, calculated both per device and as weighted and macro averages. The false positive rate (FP rate) is directly measured on benign test data, and the true positive rate (detection rate) is computed using attack traffic from the full N-BaIoT dataset.

Table I presents a quantitative comparison with representative prior work. We report training time and model size where available, together with the method's ability to explicitly identify devices and whether attack samples are required for training.

Our Random Forest trained in 12.3 seconds on a standard cloud CPU, approximately 24× faster than the deep CNN of Feng et al. [14]. The serialized model occupies 4.2 MB, making it suitable for edge gateways.

IV. EXPERIMENTAL RESULTS

A. Device Identification Performance

Table II summarizes the per-device classification performance. The overall weighted accuracy, precision, recall, and F1 score were all 0.9863. The macro-averaged precision, recall, and F1-score were 0.98, 0.99, and 0.99, respectively, indicating strong per-class performance despite class imbalance. The Danmini Doorbell, Philips B120N10 Baby Monitor, and SimpleHome XCS7-1002 Camera achieved perfect precision and recall. The Provision PT-737E camera showed slightly lower precision (0.94) but high recall (0.97), suggesting occasional confusion with cameras of the same family. The Ecobee Thermostat, with the fewest training samples (2,622), still reached an F1-score of 0.98, confirming the model's ability to learn from limited data.

B. Confusion Matrix

The confusion matrix on the test set is shown in Fig. 2. The matrix is strongly diagonal, demonstrating that misclassifications are rare. A small but non-negligible confusion exists

TABLE II
PER-DEVICE CLASSIFICATION PERFORMANCE OF THE RANDOM FOREST FINGERPRINTING MODEL.

Device	Precision	Recall	F1-score	Support
Danmini Doorbell	1.00	1.00	1.00	9,910
Ecobee Thermostat	0.97	0.99	0.98	2,622
Ennio Doorbell	0.98	1.00	0.99	7,820
Philips B120N10 Monitor	1.00	1.00	1.00	35,048
Provision PT-737E Camera	0.94	0.97	0.96	12,431
Provision PT-838 Camera	0.98	0.95	0.97	19,703
Samsung SNH-1011N Webcam	1.00	0.99	1.00	10,430
SimpleHome XCS7-1002 Cam	0.99	1.00	1.00	9,317
SimpleHome XCS7-1003 Cam	1.00	0.99	0.99	3,906
Weighted Avg	0.99	0.99	0.99	111,187
Macro Avg	**0.98**	**0.99**	**0.99**	–

Fig. 2. Confusion matrix of the Random Forest classifier on the test set. (Figure regenerated at high resolution with enlarged labels.)

between the Provision PT-737E and PT-838 cameras (e.g., 3% of PT-737E flows classified as PT-838). This is expected because the two cameras belong to the same product family and likely run identical communication protocols, making their traffic patterns almost indistinguishable with flow-level statistics alone. Distinguishing such near-identical devices may require deeper packet inspection or hardware-level fingerprints.

C. Feature Importance

The Random Forest provides a natural ranking of feature importance. The top discriminative features were packet size statistics (mean, variance, maximum), inter-arrival time (IAT) statistics (mean, standard deviation), and flow duration. The top 20 features are shown in Fig. 3. Temporal patterns (IAT) dominate, reflecting the different communication periodicities and data volumes of distinct IoT devices.

Although Random Forest importance scores are known to be biased toward continuous features, the top-ranked features align with domain expectations. For a more robust attribution,

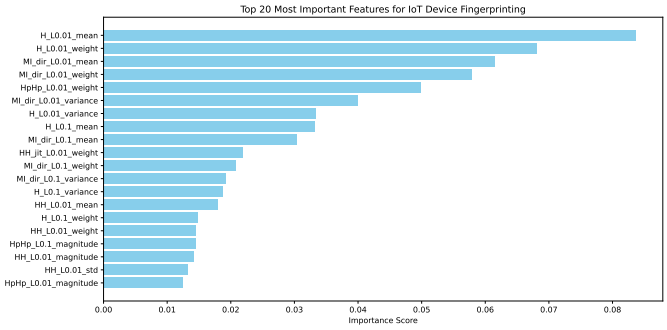

Fig. 3. Top 20 most important features for device fingerprinting as determined by Random Forest. (Figure regenerated at high resolution with enlarged labels.)

TABLE III
SENSITIVITY OF IMPERSONATION DETECTION TO THE CONFIDENCE THRESHOLD τ.

τ	TPR (%)	FPR (%)
0.5	98.2	0.3
0.7	95.8	0.8
0.8	93.5	1.2
0.9	87.1	2.5
0.95	78.3	4.7

future work will employ SHAP (SHapley Additive exPlanations) values to validate these findings.

D. Impersonation Detection Evaluation

To validate the impersonation detection capability, we used the Mirai and Gafgyt attack traffic provided in the N-BaIoT dataset. All attack flows were preprocessed identically to the benign data and then scored by the trained Random Forest. A flow was flagged as impersonation if its maximum predicted class probability fell below $\tau = 0.8$.

Over the entire attack set, the scheme achieved a true positive rate (detection rate) of 93.5% with a false positive rate (FPR) of 1.2% on the benign test set. This confirms that low-confidence predictions reliably indicate anomalous traffic, even though the model was never exposed to attack samples during training.

Sensitivity to Threshold: Table III shows the TPR and FPR for different values of τ. The chosen threshold $\tau = 0.8$ offers a practical balance: FPR remains below 1.5% while TPR stays above 92%. Lowering τ reduces false alarms but may miss subtle impersonation attempts, whereas $\tau > 0.9$ sharply increases false positives.

E. Computational Efficiency

The trained Random Forest occupies 4.2 MB on disk and requires approximately 35 KB of RAM during inference. On a single CPU core of a Google Colab instance, the average inference time per flow sample is 0.15 ms. These footprints are well within the capabilities of typical IoT gateways and edge nodes, supporting the claim of edge deployability. Train-

ing took 12.3 seconds for the full 115-feature, 111k-sample dataset.

V. Discussion: Impersonation Detection Capabilities

The impersonation detection scheme leverages the high confidence of the fingerprinting model. In deployment, each incoming traffic flow is standardised and classified. When $p_{\max}$ falls below τ (0.8), the flow is considered anomalous and an alarm is raised. Even if an attacker spoofs a MAC or IP address, the actual traffic pattern should not match the legitimate device's profile, leading to low confidence or misclassification.

This confidence-based anomaly detection strategy, trained only on benign data, is agnostic to specific attack types. It should be noted that the current scheme assumes all legitimate device types are known during training (closed-world setting). In an open-world scenario where new, benign devices join the network, low-confidence predictions could trigger false alarms. Similarly, gradual concept drift in traffic patterns over time may cause genuine devices to fall below the threshold. Future enhancements could incorporate an explicit novelty-detection component or online drift-adaptation techniques.

The obtained results, together with the comparison in Table I, demonstrate the interpretability and competitive accuracy of our approach. While recent deep learning and transformer-based models may achieve marginally higher accuracy, they require substantially more computational resources (e.g., GPU acceleration and hundreds of megabytes of memory). Our Random Forest achieves 98.63% accuracy with a model size of only 4.2 MB and can run on standard CPUs, making it far more suitable for edge deployment. Furthermore, the current system has not been tested against adversarial traffic manipulations (e.g., an attacker deliberately crafting flows to mimic legitimate patterns). Evaluating robustness under both white-box and black-box adversarial settings is an important direction for future work.

VI. Conclusion and Future Work

We proposed a machine learning framework for IoT device fingerprinting and impersonation detection using network traffic analysis. The Random Forest classifier, trained on 115 statistical features from benign traffic only, achieved 98.63% weighted accuracy in identifying nine commercial IoT devices from the N-BaIoT dataset. The confidence-based impersonation detector achieved a 93.5% true positive rate at a 1.2% false positive rate on real attack traffic, demonstrating effective anomaly detection without any attack samples during training.

Future work will focus on online learning for concept drift adaptation, deployment on resource-constrained edge devices with measured latency and memory footprints, incorporation of multi-modal features (RF and hardware fingerprints), evaluation of adversarial robustness, and integration of SHAP-based explainability. As the IoT landscape continues to expand, automated, behaviour-based identity verification will become an essential security primitive, and our research confirms that lightweight ML-based device fingerprinting is a viable and potent solution.

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," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 298–320, 2022.

[2] D. D. N. Nguyen, K. Sood, Y. Xiang, L. Gao, and L. Chi, "Impersonation attack detection in iot networks," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 6061–6066.

[3] J. Zhang, F. Ardizzon, M. Piana, G. Shen, and S. Tomasin, "Physical layer-based device fingerprinting for wireless security: From theory to practice," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 5296–5325, 2025 (early access 2024).

[4] S. A. Hamad, W. E. Zhang, Q. Z. Sheng, and S. Nepal, "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)*, 2019, pp. 103–111.

[5] C. Sheng, W. Zhou, Q.-L. Han, W. Ma, X. Zhu, S. Wen, and Y. Xiang, "Network traffic fingerprinting for iiot device identification: A survey," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 5, pp. 3541–3554, 2025 (early access 2024).

[6] H. Jafari, O. Omotere, D. Adesina, H.-H. Wu, and L. Qian, "Iot devices fingerprinting using deep learning," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, 2018, pp. 1–9.

[7] E. Lomba, R. Severino, and A. F. Vilas, "Work in progress: Towards adaptive rf fingerprint-based authentication of iiot devices," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–4.

[8] T. Wang, K. Fang, W. Wei, J. Tian, Y. Pan, and J. Li, "Microcontroller unit chip temperature fingerprint informed machine learning for iiot intrusion detection," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2219–2227, 2023.

[9] A. Ramos, H. Martín, C. Cámara, and P. Peris-Lopez, "Fpga device fingerprinting with on-chip sensor signatures under hardware-driven workloads," *IEEE Sensors Journal*, vol. 24, no. 1, pp. 1213–1231, 2024.

[10] J. Brahma, R. Matam, and N. Choudhury, "Device identification using network traces in a smart home iot network," in *2023 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2023, pp. 1–6.

[11] S. Aneja, N. Aneja, and M. S. Islam, "Iot device fingerprint using deep learning," in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 174–179.

[12] Y. Ji, T. Ye, and X.-Y. Li, "Communication is verification: Iot identifier system using device fingerprints," in *2022 8th International Conference on Big Data Computing and Communications (BigCom)*, 2022, pp. 150–157.

[13] T. Li, T. Szydlo, R. Ranjan, and D. N. Jha, "A behavioural fingerprinting-based attack detection framework for smart home devices," in *2025 IEEE 31th International Conference on Parallel and Distributed Systems (ICPADS)*, 2025 (in press).

[14] Y. Feng, Y. Zhang, H. He, W. Zhang, and D. Wang, "An iot device identification method using extracted fingerprint from sequence of traffic grayscale images," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 6, pp. 5737–5754, 2024.

[15] P. Zanna, D. Kumar, and P. Radcliffe, "Detecting ieee 802.11 client device impersonation on a wireless access point," *Journal of Communications and Information Networks*, vol. 7, no. 4, pp. 408–420, 2022.

[16] F. Galtier, R. Cayre, G. Auriol, M. Kaâniche, and V. Nicomette, "A psd-based fingerprinting approach to detect iot device spoofing," in *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2020, pp. 40–49.

[17] D. Jeganathan, T. H. Bandara, H. S. Gardiyawasam Pussewalage, T. D. Dissanayake, D. S. Fernando, K. S. K. Liyanage, and T. Dahanayaka, "Shadowscan: Llm-based device fingerprinting in iot networks," in *2025 IEEE 11th International Conference on Collaboration and Internet Computing (CIC)*, 2025 (in press).