Temporal Features for IoT Devices: Out-of-Distribution Detection without Upper-Layer Dependencies

Jian Liu*, Huan Yan*, Jinyang Huang†, Xiang Zhang‡
*Guizhou Normal University
†Hefei University of Technology
‡University of Science and Technology of China
Email:liujian@gznu.edu.cn; yh1995.cs@gmail.com; hjy@hfut.edu.cn; zhangxiang@ieee.org

***Abstract*—The large-scale deployment of IoT devices accelerates intelligent applications but also brings significant security risks. Device detection helps mitigate these risks by identifying unauthorized or rogue devices and improving visibility into network activity. However, existing device detection methods based on network and transport layer protocols face two key challenges: encrypted traffic conceals protocol information, and most approaches fail to detect out-of-distribution (OOD) devices, limiting their effectiveness in real-world scenarios. To address these issues, this paper proposes an OOD detection method based on the 802.11 protocol. Specifically, we first extract intrinsic packet attributes from the 802.11 protocol headers, including transmission timing patterns and packet structure characteristics, without relying on any network or transport layer information. Then, these features are input into a bidirectional long short-term memory (LSTM) model to learn sequential dependencies, and the extracted feature embeddings are evaluated through k-nearest neighbor (KNN) distance calculation to detect both in-distribution (ID) and OOD samples. Experiments conducted on 12 commercial IoT devices spanning 8 categories demonstrate that the proposed method achieves effective device identification and OOD detection performance.**

***Index Terms*—802.11, out-of-distribution, k-nearest neighbor**

I. Introduction

With the rapid advancement of IoT technology, an ever-growing number of devices have entered our daily lives. Among various wireless communication protocols, Wi-Fi has emerged as the most dominant, accounting for approximately one-third of global IoT connections [1]. Although this widespread connectivity brings considerable convenience, the sharp increase in both the diversity and volume of IoT devices has significantly complicated network infrastructures. As a result, IoT ecosystems are increasingly exposed to security challenges of unprecedented scale [2].

As IoT applications continue to evolve, devices are no longer confined to traditional sensing roles. They are now capable of advanced perceptual tasks such as gesture recognition via wireless signals [3], [4]. While these advancements highlight the enhanced sensing adaptability of IoT systems, they also heighten security risks, as the increasing complexity of perceptual interactions and large-scale deployments introduces new attack surfaces and vulnerabilities. In such dynamic environments, unauthorized or rogue devices may easily blend in with legitimate traffic, posing significant threats to data integrity and network stability. Therefore, accurately identifying and monitoring the types of devices connected to the network has become a fundamental requirement for ensuring system security. To address these challenges, various device identification methods have been proposed. Some approaches leverage physical-layer features such as Wi-Fi channel state information (CSI) [5], [6], while others depend on upper-layer protocols information (e.g. ICMP, DNS, IP) [7], [8]. However, as IoT devices increasingly adopt encryption technologies such as WPA/WPA2 [9], upper-layer protocols often become inaccessible. Consequently, these limitations severely restrict the practicality of existing methods in real-world deployments. To overcome this, recent papers such as Lumos [10] and IoTBeholder [11] have proposed leveraging unencrypted features in 802.11 protocol headers. While these methods bypass encryption barriers and enable classification of known device types, they still fall short in addressing a more critical and practical need: the detection of out-of-distribution (OOD) devices. In dynamic IoT environments, previously unseen devices often join the network. Misclassifying these unknown devices as legitimate ones can introduce severe security risks. In real-world IoT environments, new and unknown devices frequently appear, posing potential security risks if they are misclassified as legitimate known devices. Therefore, effective OOD detection is not merely an extension of device identification but a fundamental capability necessary for building trustworthy IoT systems.

To address the aforementioned challenges, this paper conducts OOD detection based solely on features extracted from the 802.11 protocol. The approach leverages unencrypted 802.11 data frame headers to support detection in encrypted network environments. Specifically, the process consists of the following steps. First, 802.11 packets are passively captured, and retransmitted frames are filtered out to retain valid data frames. Second, the processed packet sequences are fed into a bidirectional long short-term memory (LSTM) [12] model to extract temporal feature patterns that characterize device behaviors. Finally, the extracted embeddings are compared against a reference feature set using a k-nearest neighbor (KNN) algorithm, enabling simultaneous classification of ID devices and detection of OOD samples based on distance thresholds. In summary, the main contributions of this paper are as follows:

- We break away from dependence on upper-layer protocols, accomplishing IoT device detection solely through the 802.11 protocol.
- We integrate OOD detection into traditional detection methods, enabling effective identification of unknown IoT devices in networks.
- In complex real-world environments, when half of the test devices are OOD, the proposed method achieves an area under the receiver operating characteristic curve (AUROC) of 0.9031, with a false positive rate (FPR) of only 0.094 at 80% true positive rate (TPR).

II. Related Work

In the field of IoT device detection, a range of methods have been proposed, focusing on identifying devices based on network traffic characteristics. Many papers have relied on network-layer and transport-layer protocols to extract temporal and spatial features for classification. For example, Ma et al. [7] leveraged protocol information from the network and transport layers to obtain spatiotemporal traffic patterns, enabling the identification of IoT devices and the estimation of devices hidden behind network address translators (NATs). Bai et al. [8] utilized DNS and ICMP protocols, employing deep learning models for automatic IoT device classification. Meidan et al. [13] applied a random forest algorithm to features extracted from continuous TCP sessions to detect unauthorized IoT devices. Guo et al. [14] proposed a method based on analyzing communication patterns at the IP layer to identify device types. Similarly, IoTAthena [15] analyzed raw IP-layer traffic with timestamps to reveal device activities. Although these methods have demonstrated strong device identification capabilities, they rely heavily on upper-layer protocol data. In encrypted network environments, their effectiveness is significantly limited. Moreover, many machine learning-based approaches inherently assume that all input samples belong to known classes, making them prone to misclassifying OOD devices as known types, which can introduce critical security risks.

To overcome the limitations of upper-layer dependency, several papers have explored using the 802.11 protocol, whose frame headers remain visible under encryption. Lumos [10] manually selected partial 802.11 header features and trained machine learning models for device identification and classification. IoTBeholder [11] further aggregated 802.11 packets into traffic bursts and used machine learning models to identify device types. Alyami et al. [16] leveraged only 802.11 frame header information, employing multiple machine learning techniques to fingerprint and infer the states of 12 popular IoT devices. These papers confirm that IoT device identification is feasible without relying on upper-layer protocols. However, similar to earlier approaches, they primarily focus on classifying known devices and overlook the challenge of OOD detection. In real-world deployments, where previously unseen devices frequently emerge, the absence of OOD detection mechanisms can result in unknown devices being mistakenly classified as legitimate, thus posing serious security threats.

III. 802.11 Protocol

The 802.11 protocol defines multiple frame types, including control frames, management frames, and data frames. Since data frames serve as the primary frame type for IoT devices to transmit data and carry core information about device traffic characteristics, we primarily capture data frames in the 802.11 protocol [11]. Among these, a special type called the null function data frame (NFDF), also referred to as a null packet, which is mainly used to transmit control information such as device power-saving mode switching and triggering of special network activities [17]. Although its structure is similar to that of a standard data frame, it does not contain a payload and is composed only of a frame header and a frame trailer. Given that these frames do not transmit actual data and the number of such frames sent may vary due to functional differences among IoT devices, they are retained but not recorded as valid data packets (valid data packets refer to data frames containing payloads). Additionally, when the sender does not receive an acknowledgment frame within a specific time window after transmitting a data packet, it triggers retransmission. Retransmitted packets can significantly interfere with the learning of a device's baseline behavior. For example, the actual arrival time of packets is affected by the number of retransmissions, introducing uncertainty into packet arrival timestamps that may disrupt the learning of temporal patterns in device behavior. Frequent retransmissions also increase captured traffic fluctuations, making it difficult to identify normal traffic patterns and impacting the accuracy of device behavior modeling. Therefore, we chose to filter out retransmitted packets to mitigate these issues.

IV. Out-of-Distribution Detection

The process is shown in Fig. 1. The Packet Flow Acquisition is primarily responsible for the real-time capture of 802.11 protocol packets. The Data Preprocessing performs batch padding on data across different batches. The Sequential Traffic Analyzer is responsible for extracting temporal features from the processed data streams, which is implemented using a bidirectional LSTM. The OOD Decision determines whether a sample belongs to the OOD category by calculating the Euclidean distance between the sample and the reference dataset.

The operation of the entire IoT device OOD detection method is divided into two phases: training and detection. During the training phase, data extracted from raw packet streams (categorized by MAC addresses) undergoes padding processing. This is because the number of packets per MAC address is variable during capture, resulting in variable-length data streams. The processed data streams are then input

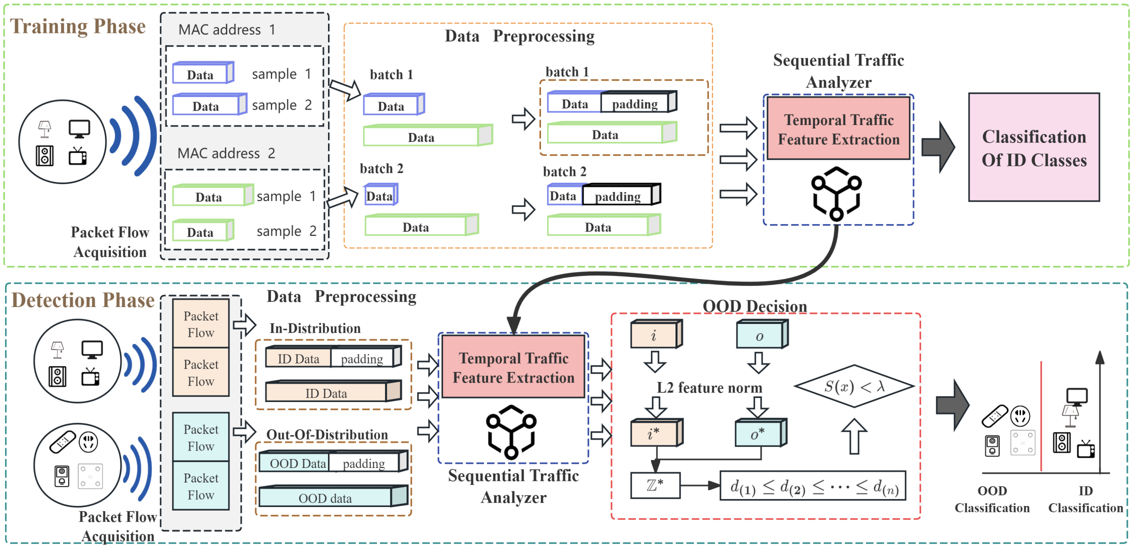


Fig. 1. Overview of detection. It includes a training phase and detection phase.

into the sequential traffic analyzer. Through this process, the sequential traffic analyzer learns the temporal features patterns of different IoT devices.

In the detection phase, when the input packet stream contains OOD samples, the sequential traffic analyzer extracts temporal features from each sample. These features are normalized, and the euclidean distance between each feature and the reference dataset is calculated. Based on this distance, the system determines whether a sample belongs to the OOD category. If it is identified as OOD, the sample is labeled as an unknown device. Otherwise, it is classified into a specific known device category.

A. Packet Flow Acquisition

In this section, we capture all 802.11 packet streams transmitted by IoT devices and categorize them by each device's MAC address for storage. To avoid redundant data, we focus on retaining core traffic characteristics by filtering out retransmitted packets. For each sample, we collect 100 valid data packets, defined as the sequence $y = \langle x_1, x_2, \cdots, x_{100} \rangle$, where y represents the sample and x_i denotes individual valid packets within this sequence. Additionally, we retain null packets, which indicate that the IoT device is engaged in special network activities (e.g., power-saving mode) but carry no effective payload information. Although these packets contain no useful payload, they reflect specific behavioral patterns of certain IoT devices and are therefore included in the dataset while excluded from valid data packet counts. Consequently, each sample consists of 100 valid data packets interspersed with j null packets, forming the sequence $y_i = \langle x_1, n_1, n_2, x_2, \cdots, n_j, x_{100} \rangle$, where n_j represents null packets. Fig. 2 illustrates a representative input sample composed of 100 valid data packets interspersed with j null packets. The sample may belong to either an ID or OOD category.

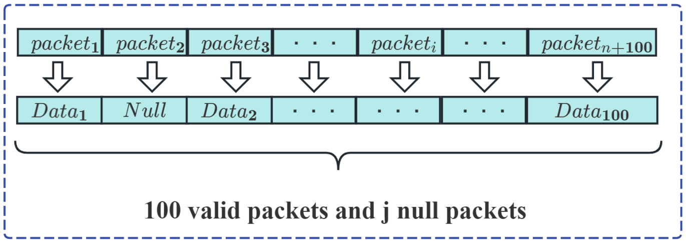


Fig. 2. An input sample with 100 valid packets and j null packets, labeled as either ID or OOD.

B. Data Preprocessing

This section focuses on extracting coarse-grained features from raw packet sequences and applying appropriate padding to meet the input requirements for model training. The extracted features include inter-packet time intervals, packet lengths, frame subtypes, transmission directions (incoming/outgoing), and quality of service (QoS) control field. These features effectively capture the behavioral patterns of IoT devices within a specific time window. Taking typical devices as examples, the packet flow from a camera generally exhibits short time intervals, large packet sizes, fixed subtypes, predominantly outgoing direction and a QoS control field labeled as a video stream. In contrast, the packet flow from a table lamp tends to show longer time intervals, moderate packet sizes and a high frequency of null packets. Inspired by a previous paper on traffic burst patterns [11], our scheme sets the burst interval threshold to 1 second, with each burst cycle containing 100 effective packets. This configuration allows us to capture fine-grained temporal features within burst traffic, such as short-interval high-frequency transmission patterns, while also accounting for macro temporal differences between burst cycles, including periods of silence or low-frequency interaction characteristics.

C. Sequential Traffic Analyzer

This section presents the temporal feature extraction method for characterizing IoT device behaviors from 802.11 protocol traffic. The primary objective is to learn discriminative sequential patterns from packet streams that may contain both valid data packets and null packets. To achieve this, we adopt a deep sequential model based on bidirectional LSTM model. The model processes each input sequence containing 100 valid data packets, which may be interspersed with null packets representing power-saving states. The bidirectional LSTM, which consists of two layers with 64 hidden units each, learns both forward and backward dependencies in the packet sequences. This design enables the model to maintain contextual awareness across null packets and capture comprehensive temporal relationships, including periodic transmissions, burst behaviors, and characteristic silent periods. The gating mechanism of LSTM cells proves particularly effective for handling these irregular sequences, as it selectively retains relevant state information while processing valid packets and skipping null packets.

D. OOD Decision

This section describes the OOD detection strategy employed in our method. The goal is to detect whether a test sample originates from an OOD device. To achieve this, we adopt a non-parametric detection approach inspired by the deep nearest neighbor detection method proposed in [18].

The underlying idea is to estimate the density of the learned feature space without assuming a global parametric distribution. Specifically, we construct a reference dataset $Z = \langle \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n \rangle$, where $\mathbf{z}_i$ denotes the feature embedding of a training sample, and Z corresponds to the embeddings of known IoT device categories. Each $\mathbf{z}_i$ is obtained from a sample sequence y_i in the training set. During inference, each test sample y_o is processed by the trained model to obtain its feature embedding $\mathbf{o}$, which is then compared against the reference set using KNN.

To reduce the impact of feature norm variations and ensure comparability in the distance space, both the reference embeddings and the test embedding are normalized via L2 normalization:

$$\mathbf{z}_i^* = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2}, \quad \mathbf{o}^* = \frac{\mathbf{o}}{\|\mathbf{o}\|_2} \tag{1}$$

The Euclidean distance between the normalized test sample and each reference embedding is then computed as follows:

$$d(\mathbf{o}^*, \mathbf{z}_i^*) = \|\mathbf{o}^* - \mathbf{z}_i^*\|_2, \quad i = 1, 2, \ldots, n \tag{2}$$

Let $d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(n)}$ be the sorted distances between $\mathbf{o}^*$ and all reference points. We define the k-th nearest neighbor distance as:

$$d_k(\mathbf{o}^*) = d_{(k)} \tag{3}$$

An OOD detection score is then defined as the negative distance to the k-th nearest neighbor:

$$S(\mathbf{o}) = -d_k(\mathbf{o}^*) \tag{4}$$

A higher score indicates a closer distance to the ID reference distribution. Given a threshold τ, the decision rule is defined as follows: if $S(\mathbf{o}) > \tau$, the sample is classified as an ID sample and is passed to the classifier for category prediction. Otherwise, it is classified as an OOD sample and rejected as unknown.

V. EXPERIMENTS AND RESULTS

In this section, we introduce the performance of our method in detecting OOD devices from real-world captured IoT device traffic.

A. Experimental Setup

We collected traffic generated by 12 IoT devices across 8 categories: smart socket, smart power strip, camera, doorbell, smart speaker, table lamp, smart screen, and body fat scale. Given the variability in traffic volume and timing across different devices, we applied differentiated sampling over a one-week period. For high-rate devices, packet flows were collected intermittently, while for low-rate devices, continuous collection was used. This process resulted in a total of 27,766 samples, with more than 1,100 samples per category. To evaluate the model's ability to detect unknown devices, we first conducted single-category OOD experiments in which each IoT device category was treated as out-of-distribution in turn. We then assessed performance in more challenging scenarios by designating half of the categories as OOD, simulating high-diversity conditions where behavioral overlap among device types may occur. We defined two experimental setups:

- Case 1: Randomly select half of the categories as the OOD samples and the remaining half as the ID samples, modeling complex networks with a high proportion of unknown devices.
- Case 2: Swap the OOD and ID samples from Case 1 to validate the feasibility of the method in scenarios where the distribution of known/unknown categories is reversed.

These experimental setups include single-category OOD detection and balanced-category partitioning through Case 1 and Case 2. This combination helps verify the model's effectiveness in both fine-grained unknown detection and broader category distribution scenarios. The corresponding device categorizations are shown in Table I.

The performance of our method is evaluated using the following metrics: (1) the false positive rate (FPR95) of OOD samples at an ID sample TPR of 95%; (2) the false positive rate (FPR80) of OOD samples when ID sample TPR is 80%; (3) AUROC is used to evaluate the overall ability of the model to distinguish between ID and OOD samples.

TABLE I
OOD AND ID SAMPLES SETUP UNDER COMPLEX ENVIRONMENT

Case	ID Category	OOD Category
Case 1	Camera Doorbell Table Lamp Body Fat Scale	Smart Socket Smart Power Strip Smart Speaker Smart Screen
Case 2	Smart Socket Smart Power Strip Smart Speaker Smart Screen	Camera Doorbell Table Lamp Body Fat Scale

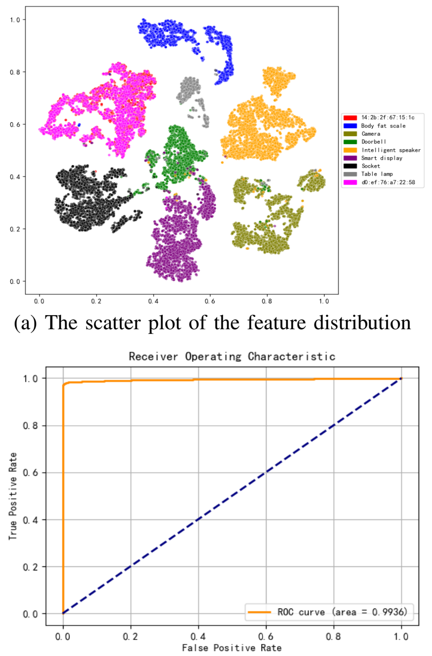
B. Experimental Results

When using single-category IoT device samples as OOD samples, the results are shown in Table II (through cross-validation, we selected $K = 5$), which presents the detection results for scenarios where each OOD sample consists of a single device category. According to the metrics, the smart power strip demonstrates the best performance. Its FPR95 is only 0.0067, FPR80 is 0, and AUROC reaches 0.9936, leading across all three indicators. The table lamp, body fat scale, and smart screen also achieve an FPR80 of 0, with AUROC values all exceeding 0.96, reflecting excellent detection effectiveness. The doorbell and smart socket exhibit moderate detection performance, while the camera and smart speaker show relatively higher FPR95 and FPR80 and lower AUROC. Nevertheless, their AUROC values all exceed 0.91. It can be seen that when the OOD samples belong to a single category, the detection performance is generally excellent.

TABLE II
COMPARISON OF EXPERIMENTAL RESULTS OF SINGLE-CATEGORY OOD DETECTION. ↑ INDICATES LARGER VALUES ARE BETTER AND VICE VERSA.

OOD Category	FPR95↓	FPR80↓	AUROC↑
Camera	0.3380	0.1546	0.9186
Doorbell	0.1463	0.0643	0.9564
Table Lamp	0.0193	0.0	0.9896
Body Fat Scale	0.0378	0.0	0.9619
Smart Socket	0.2397	0.0038	0.9676
Smart Power Strip	0.0067	0.0	0.9936
Smart Speaker	0.2879	0.1229	0.9143
Smart Screen	0.0169	0.0	0.9853

Overall, detection performance varies when different IoT device categories serve as OOD samples. Some devices are easily detectable due to distinct feature differences from ID data, while others present greater challenges due to feature similarities. As shown in Fig. 3(a) presents a scatter plot of features extracted from ID and OOD data after t-SNE dimensionality reduction. The smart power strip category forms a separate cluster with distinct boundaries, clearly distinguishable from other ID categories. This demonstrates the effectiveness of our method in detecting the smart power strip as an OOD sample by leveraging the temporal feature differences between OOD and ID categories. In Fig. 3(b), the AUROC curve further confirms this, with a score of 0.9936, indicating the method's excellent ability to distinguish between OOD (smart power strip) and ID samples.



(a) The scatter plot of the feature distribution

(b) ROC curve with its corresponding AUC value

Fig. 3. Distribution plot and plot displaying the ROC curve with its corresponding AUC value for the smart power strip as the OOD sample

In the Case 1 setup, a scatter plot was obtained after t-SNE dimensionality reduction, as shown in Fig. 4. The plot includes four ID IoT device categories labeled by product names and four OOD categories identified by MAC addresses, with the OOD categories consisting of two smart speakers, two smart power strips, two smart sockets, and one smart display. Our method successfully classifies IoT devices, with each device type forming distinct clusters in the feature space. However, due to similarities in network activity between some devices and unknown IoT devices, such as the video doorbell, which exhibits similar activity patterns to devices like cameras and smart displays due to its ability to transmit video, feature space overlaps occur. Additionally, like the smart socket, the video doorbell sends null packets to indicate power-saving mode. These temporal feature similarities lead to overlaps in the feature space, contributing to a decline in the performance of OOD detection for Case 1.

TABLE III
COMPARISON OF EXPERIMENTAL RESULTS FOR OOD DETECTION IN COMPLEX ENVIRONMENTS

	FPR95↓	FPR80↓	AUROC↑
Case 1	0.2983	0.0207	0.9568
Case 2	0.6392	0.0940	0.9031

As shown in Table III, experimental results demonstrate that with $K = 5$, the method achieves favorable OOD detection performance in Case 1. The AUROC reaches 0.9568,

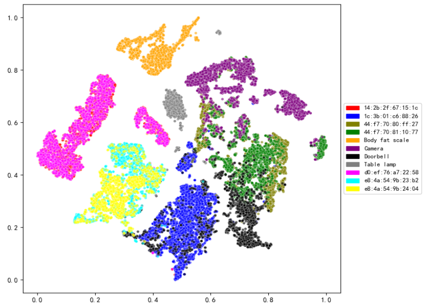
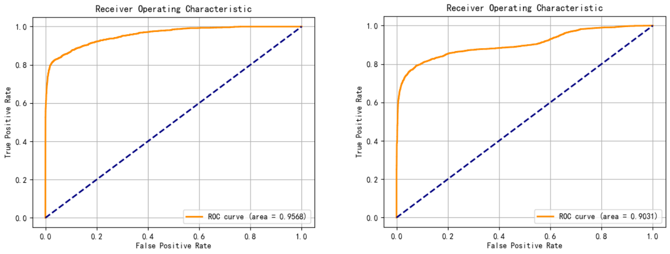


Fig. 4. Scatter plot of Case 1 under complex settings, showing four ID categories labeled by product names and four OOD categories identified by MAC addresses.

as illustrated by the ROC curve in Fig. 5(a), with FPR95 at 0.2983 and FPR80 significantly reduced to 0.0207. In Case 2, although the FPR95 rises to 0.6392, the AUROC remains above 0.9, and the FPR80 drops to 0.0940, as shown in the ROC curve in Fig. 5(b). These results indicate that the method maintains strong discriminative capability across different category distributions.



(a) Case 1

(b) Case 2

Fig. 5. ROC curve and AUC value experimental result plots in complex environments

Combining the experimental results in Table II and Table III, all AUROC values exceed 0.9, demonstrating the effectiveness of our method in IoT device OOD detection. By capturing 802.11 protocol traffic, extracting temporal features, and applying a KNN-based distance metric, the method accurately distinguishes ID from OOD samples. It shows robust performance when detecting devices with clear behavioral differences, while temporal feature similarities among certain devices can cause overlaps in the feature space, leading to a moderate decline in detection performance for behaviorally similar OOD samples.

VI. Conclusion

This paper presents an effective OOD detection method for IoT devices, leveraging features extracted solely from the 802.11 protocol. By capturing temporal features from Wi-Fi data frames and employing a bidirectional LSTM model combined with a KNN distance metric, the proposed approach effectively distinguishes between ID and OOD devices without relying on upper-layer protocol information. Experimental evaluations on 12 commercial IoT devices across 8 categories demonstrate that the method achieves high detection performance, with AUROC values exceeding 0.9 in various scenarios. The results indicate that the approach maintains robust detection capabilities even in complex environments with a high diversity of unknown devices. These findings underscore the potential of utilizing 802.11 protocol features for enhancing the security and reliability of IoT networks.

References

[1] Enea, "Wi-fi based iot - insights and market trends," https://www.enea.com/insights/wi-fi-based-iot/, 2023.

[2] R. A. R. Ait Mouha *et al.*, "Internet of things (iot)," *Journal of Data Analysis and Information Processing*, vol. 9, no. 02, p. 77, 2021.

[3] H. Yan, X. Zhang, J. Huang, Y. Feng, M. Li, A. Wang, W. Ou, H. Wang, and Z. Liu, "Wi-sfdagr: Wifi-based cross-domain gesture recognition via source-free domain adaptation," *IEEE Internet of Things Journal*, 2025.

[4] X. Zhang, J. Huang, H. Yan, Y. Feng, P. Zhao, G. Zhuang, Z. Liu, and B. Liu, "Wiopen: A robust wi-fi-based open-set gesture recognition framework," *IEEE Transactions on Human-Machine Systems*, 2025.

[5] X. Zhang, J. Zhang, Z. Ma, J. Huang, M. Li, H. Yan, P. Zhao, Z. Zhang, Q. Guo, T. Zhang *et al.*, "Camlopa: A hidden wireless camera localization framework via signal propagation path analysis," *arXiv preprint arXiv:2409.15169*, 2024.

[6] J. Huang, B. Liu, C. Miao, X. Zhang, J. Liu, L. Su, Z. Liu, and Y. Gu, "Phyfinatt: An undetectable attack framework against phy layer fingerprint-based wifi authentication," *IEEE Transactions on Mobile Computing*, vol. 23, no. 7, pp. 7753–7770, 2023.

[7] X. Ma, J. Qu, J. Li, J. C. Lui, Z. Li, and X. Guan, "Pinpointing hidden iot devices via spatial-temporal traffic fingerprinting," in *IEEE INFOCOm 2020-IEEE conference on computer communications*. IEEE, 2020, pp. 894–903.

[8] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Z. Yang, "Automatic device classification from network traffic streams of internet of things," in *2018 IEEE 43rd conference on local computer networks (LCN)*. IEEE, 2018, pp. 1–9.

[9] B. I. Reddy and V. Srikanth, "Review on wireless security protocols (wep, wpa, wpa2 & wpa3)," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 5, no. 4, pp. 28–35, 2019.

[10] R. A. Sharma, E. Soltanaghaei, A. Rowe, and V. Sekar, "Lumos: Identifying and localizing diverse hidden {IoT} devices in an unfamiliar environment," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1095–1112.

[11] Q. Zou, Q. Li, R. Li, Y. Huang, G. Tyson, J. Xiao, and Y. Jiang, "Iotbeholder: A privacy snooping attack on user habitual behaviors from smart home wi-fi traffic," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 7, no. 1, pp. 1–26, 2023.

[12] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[13] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized iot devices using machine learning techniques," *arXiv preprint arXiv:1709.04647*, 2017.

[14] H. Guo and J. Heidemann, "Ip-based iot device detection," in *Proceedings of the 2018 workshop on IoT security and privacy*, 2018, pp. 36–42.

[15] Y. Wan, K. Xu, F. Wang, and G. Xue, "Iotathena: Unveiling iot device activities from network traffic," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 651–664, 2021.

[16] M. Alyami, I. Alharbi, C. Zou, Y. Solihin, and K. Ackerman, "Wifi-based iot devices profiling attack based on eavesdropping of encrypted wifi traffic," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 385–392.

[17] Y. Mizuno, Y. Ohishi, and N. Ishikawa, "A simple metric that correlates with public wi-fi throughput," *Electronics Letters*, vol. 59, no. 8, p. e12795, 2023.

[18] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *International Conference on Machine Learning*. PMLR, 2022, pp. 20 827–20 840.