Cross-domain, Scalable, and Interpretable RF Device Fingerprinting

Tianya Zhao*, Xuyu Wang* §, Shiwen Mao†
*Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL 33199, US
†Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, US
Emails: tzhao010@fiu.edu, xuywang@fiu.edu, smao@ieee.org

IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621423

***Abstract*—In this paper, we propose a cross-domain, scalable, and interpretable radio frequency (RF) fingerprinting system using a modified prototypical network (PTN) and an explanation-guided data augmentation across various domains and datasets with only a few samples. Specifically, a convolutional neural network is employed as the feature extractor of the PTN to extract RF fingerprint features. The predictions are made by comparing the similarity between prototypes and feature embedding vectors. To further improve the system performance, we design a customized loss function and deploy an eXplainable Artificial Intelligence (XAI) method to guide data augmentation during fine-tuning. To evaluate the effectiveness of our system in addressing domain shift and scalability problems, we conducted extensive experiments in both cross-domain and novel-device scenarios. Our study shows that our approach achieves exceptional performance in the cross-domain case, exhibiting an accuracy improvement of approximately 80% compared to convolutional neural networks in the best case. Furthermore, our approach demonstrates promising results in the novel-device case across different datasets. Our customized loss function and XAI-guided data augmentation can further improve authentication accuracy to a certain degree.**

***Index Terms*—Radio frequency fingerprinting, cross-domain identification, few-shot learning, explainable artificial intelligence.**

I. Introduction

In recent years, the proliferation of the Internet of Things (IoT) has contributed to the widespread integration of wireless technology into daily life. While the IoT-based applications are promising, there are also existing security issues, such as device identification and authentication [1], [2]. To mitigate these security issues, various approaches have been devised and put into practice. Although conventional cryptographic authentication methods based on Internet Protocol (IP) and Media Access Control (MAC) addresses have been widely employed [3], they suffer from inherent vulnerabilities, such as susceptibility to spoofing and tampering [4]. Moreover, these methods may not be suitable for ultra-low-power devices or outdated hardware that is no longer actively maintained [5]. To address these issues, radio frequency (RF) fingerprinting has emerged as a promising device identification solution that leverages the intrinsic characteristics of RF devices to improve safety and security in a variety of settings [6].

RF fingerprints are attributed to inherent physical imperfections in the analog circuity of RF emitters during the manufacturing process, which affects the transmitted signals but does not affect the performance of devices. Therefore, RF fingerprint serves as a unique property for each device, including ultra-low-power devices and old equipment. In comparison to conventional cryptographically secure authentication methods, the distinctive nature of the RF fingerprint makes it resistant to tampering and spoofing, thereby ensuring the security of the device [7]. This property makes RF fingerprint particularly suitable for high-level security demanding scenarios. Furthermore, RF fingerprint-based identification does not require additional power consumption as it is intrinsically linked to the transmitted signals. Due to the benefits of RF fingerprint, numerous studies have studied RF fingerprinting for device identification, including UWB [8], LoRa [9]–[11], RFID [12], ZigBee [13], and WiFi [14]–[17].

RF fingerprinting generally includes fingerprint feature extraction and multi-class identification. Effective feature extraction is essential for accurately classifying different RF fingerprints. Knox *et al.* present an RF fingerprint authentication method based on automatic gain control circuitry to distinguish between different transmitters [18]. Huang *et al.* extract the permutation entropy as the fingerprint to identify the unique transmitter [19]. However, the above traditional RF fingerprint extraction methods are hand-crafted, inefficient, and require a thorough understanding of communication technologies and protocols. In contrast, deep neural networks (DNNs) have gained considerable popularity in RF fingerprinting. This is primarily attributed to their powerful capability of feature extraction and classification. By directly using raw or simple-processed in-phase/quadrature (IQ) samples as input, DNNs can automatically extract meaningful features and classify various devices.

In an ideal scenario, deep learning-based fingerprinting systems can automatically extract fingerprint features and accurately classify devices. However, RF fingerprints are embedded in transmitted wireless signals, and these signals can undergo variations in decay and reflection across different environments. This can cause a problem for DNNs, known as domain shift. While DNNs can be very accurate in familiar domains, they struggle to adapt to new domains. Moreover, DNNs typically have a fixed output layer size, which limits their scalability. This means that they can only classify a specific range of devices. This can be a problem in real-world scenarios where device additions or removals

§The corresponding author is Xuyu Wang (xuywang@fiu.edu).

979-8-3503-8350-8/24/$31.00 ©2024 IEEE

are common. If a new device is added, the DNN needs to be retrained from scratch with a new dataset. This can be a time-consuming and expensive process. Fortunately, there are a number of approaches that have been proposed to address these issues, including transfer learning, adversarial domain adaptation (ADA), and few-shot learning (FSL). In this paper, we focus on FSL, as it does not require a large number of samples and is relatively convenient to deploy. This alleviates the burden of re-collecting data and re-training models, making RF fingerprinting more practical in real-world situations.

Challenges. Designing a scalable and domain-robust RF fingerprinting system based on FSL is still a challenging task. There are several key challenges that need to be addressed. First, while FSL demonstrates the capability to classify new data with only a few samples, obtaining stable RF fingerprints from a limited number of wireless signals poses difficulties due to substantial variations across different domains. Second, RF fingerprints are more subtle than domain factors and transmitted signals. Additionally, emitter imperfections may be similar between devices of the same manufacturer. Therefore, the model's feature extraction capability must be highly precise in order to effectively identify devices across diverse domains. Third, in complicated tasks, fine-tuning is commonly employed to enhance model accuracy. However, when working with a limited number of samples, there is a risk of overfitting. To address this, regularization techniques are typically introduced through data augmentation during the fine-tuning process. Nevertheless, implementing an effective data augmentation method specifically for wireless signals within FSL frameworks remains challenging.

Our solution. To address these challenges, we carefully redesign the classical prototypical network (PTN) [20] to extract relatively stable fingerprint features across various domains and accurately identify new devices with only a small number of samples. Specifically, we design a similarity-based loss function for training and fine-tuning the PTN to optimize the feature extractor to generate unique RF fingerprints that are robust in different domains. The well-trained extractor can effectively extract unique fingerprint features for each device. We then compute a prototype for each device by averaging the extracted feature vectors. This prototype serves as a stable fingerprint for that device. During the authentication phase, the input device is assigned to the class whose prototype is most similar to the feature vector of the input device, thereby determining its identity. In complicated tasks, fine-tuning is necessary but can lead to overfitting with only a few samples. To address this issue, we leverage a classic eXplainable Artificial Intelligence (XAI) technique called Local Interpretable Model-agnostic Explanations (LIME) [21] to design a data augmentation technique for RF fingerprinting. To evaluate the effectiveness of our proposed RF fingerprinting system, we conduct comprehensive assessments across various datasets, devices, and domains. The main contributions of this paper are as follows.

- To the best of our knowledge, this is the first time to discuss RF fingerprinting in the cross-dataset case. This is a more challenging and practical scenario as it encompasses both domain shift and scalability challenges.
- We devise a data augmentation technique based on an XAI method and a customized loss function that aids in improving the accuracy of the system.
- We experimentally demonstrate the effectiveness of our proposed RF fingerprinting system in both in-dataset and cross-dataset scenarios using three public datasets. The results showed that our system can improve accuracy by up to 80% in the best case for in-dataset scenarios, and achieve a mean accuracy of 76% for the more challenging cross-dataset scenario.

The rest of the paper is organized as follows. Section II reviews related work. Section III introduces the background and motivations. Section IV introduces our system design. In Section V, we evaluate our experiments comprehensively. Section VI concludes this paper.

II. Related Work

In recent years, deep learning techniques have been widely applied in the field of RF fingerprinting. In [22], convolutional neural network (CNN) consistently outperformed other networks such as long-short term memory (LSTM) and multilayer perceptrons (MLPs). [9] explored the various neural networks with different signal representations (IQ, amplitude-phase, and spectrogram) and employed the DeepLoRa augmentation technique to enhance the performance in cross-day scenarios. [23] showed the advantages of complex-valued neural networks for RF fingerprinting. Shen *et al.* employed a spectrogram-based approach and incorporated the estimated carrier frequency offset (CFO) into their CNN model for LoRa device fingerprinting [10]. Jafari *et al.* proposed traditional neural networks on RF traces collected from six ZigBee devices at various signal-to-noise ratio (SNR) levels [24]. Pan *et al.* introduced the RF-DNA structure, a complex arrangement of millions of Dual Natural Attributes (DNA) in a helical configuration for RFIDs [12].

To enhance the robustness of RF fingerprinting, Chen *et al.* proposed an identification system that combines software defined radio (SDR) and transfer learning technology [25]. Yu *et al.* proposed a multi-sampling CNN and an SNR adaptive region of interest (ROI) selection algorithm to extract RF fingerprinting for the purpose of classifying ZigBee devices [26]. RadioNet employed adversarial domain adaptation and introduced a novel metric (device rank) to enhance the effectiveness of radio fingerprinting in cross-day scenarios [27]. In [28], semi-supervised deep learning and RF fingerprinting with meta pseudo time-frequency labels have been deployed to improve identification performance in small-scale labeled datasets. Yang *et al.* proposed a solution to the security issues by presenting a method of generating RF fingerprinting recognition using generative adversarial networks (GAN) [29].

FSL has been widely used in RF fingerprinting and related fields to solve domain-shift problems. [30] employed metric learning to address domain shift and scalability issues in LoRa RF fingerprinting. Jin *et al.* proposed a Wi-Fi-based human identification system by using FSL and generative adversarial networks [31]. FewSense employed FSL to enhance the performance of a cross-domain Wi-Fi sensing system [32]. Wi-Learner improved generalization ability on cross-domain Wi-Fi-based gesture recognition by using one-shot learning [33].

Our work differs from related work in several key aspects. First, our datasets are generated from different groups, leading to a more diverse division of devices and domains. Second, we utilize the principles of few-shot learning and employ a modified PTN to extract domain-invariant RF fingerprint features, yielding promising results in various scenarios. Third, we customize the loss function and propose an XAI-aided data augmentation to improve cross-dataset accuracy.

III. Background and Motivation

A. Problem Scope

RF fingerprint-based device authentication systems have gained increased attention due to their uniqueness and robustness in countering spoofing and attacks. By using deep learning, RF fingerprints can be better extracted and identified. However, traditional deep learning approaches have a fixed output size for a specific task, which poses a challenge for RF fingerprinting systems that require frequent addition or removal of devices. Meanwhile, incorporating new devices also brings unseen domains to the system. The domain shift issue becomes a critical concern in RF fingerprinting systems, despite the powerful computing and mapping capabilities are offered by deep learning models. It will significantly reduce the accuracy in unseen domains. The primary reason behind this is the sensitivity of wireless signals to environmental changes, resulting in variations in scattering and reflection patterns. Consequently, deep learning models can be effectively trained on known domains but may struggle to generalize to new devices and environments. Therefore, this paper aims to address these challenges and is based on the following objective and underlying assumption.

- The **objective** is to enhance the scalability and domain robustness of the RF fingerprinting system, enabling it to perform well across unseen domains and devices using only a limited number of samples.
- We make an **assumption** that our system has a base dataset $\mathcal{E}_{\text{base}}$ consisting of a group of known devices within a particular range of domains. The feature extractor is trained using this dataset and needs to extract stable fingerprint features across various domains. Although the metric-based approach has the ability to detect unknown devices, this paper concentrates on addressing the domain shift and the scalability challenges.

B. Motivation

1) Physical Layer Identification: Traditional IP or MAC address-based identification schemes still face many security issues. Moreover, some IoT devices lack sufficient computational power, making it impractical to deploy cryptographic authentication schemes. To overcome these challenges, the physical layer-based security paradigm has been proposed. This paradigm leverages unique, permanent, and unavoidable physical imperfections generated during the manufacturing process. These imperfections can be utilized as unique fingerprints, enabling them to be used for authentication purposes.

2) Domain Shift Problem: Although the deep learning-based RF fingerprinting system is promising due to its uniqueness, the domain shift problem still exists because the fingerprint is transmitted via wireless signals. This implies that even though the fingerprint itself is stable and distinct, environmental changes can greatly affect signal propagation, resulting in a lack of robustness in identifying RF fingerprints. Fig. 1 and Fig. 2 present the accuracy drops as the unknown domains increase, and the data distribution varies among different datasets (i.e., CORES [34], WiSig [15], and ORACLE [35]). Thus, a robust RF fingerprinting system is needed, which can identify devices in new and diverse domains.

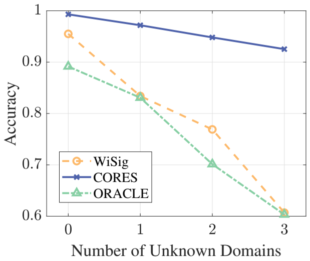

Fig. 1. Classification accuracy decreases as the number of unknown domains increases.

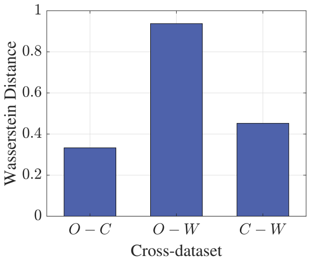

Fig. 2. The dissimilarity between different datasets, where C, W, and O represent CORES, WiSig, ORACLE.

3) Scalability Problem: In real-world scenarios, it is common to introduce new devices or remove existing ones from an RF fingerprinting system. However, traditional deep learning methods such as CNN and LSTM encounter scalability limitations because their fixed output layers constrain their ability to handle a varying number of classes once trained. Adapting these models to new settings requires extensive retraining with a large volume of training samples, which is time-consuming.

4) Small Sample Problem: During the training phase of the RF fingerprinting system, a vast dataset can be collected to train a deep learning model offline. However, when it comes to implementing the system in new domains or with new devices, collecting a large dataset becomes impractical and infeasible, which can pose challenges related to domain shift and scalability. This limitation has the potential to impact the system's ability to accurately identify devices. Therefore, it is necessary to ensure the efficacy of the RF fingerprinting system even with only a limited number of new samples.

C. Few-shot Learning

Few-shot learning aims to achieve generalization to new classes and new domains that are not seen in the training set, based on only a limited number of examples of each new class. This distinguishes it from most traditional deep learning techniques that require large quantities of labeled data for training. As a result, FSL is particularly valuable in scenarios where labeled data is scarce or costly to obtain and where the model needs to adapt to new, unseen tasks with minimal examples quickly. In contrast to conventional machine learning, FSL adopts a different approach to partitioning datasets. In this paper, we have a base dataset $\mathcal{E}_{\text{base}}$ to train a feature extractor f_θ. Then, we construct a support set $\mathcal{E}_{\text{support}}$ comprising a small number of labeled samples and the query set $\mathcal{E}_{\text{query}}$ contains data that we need to infer the labels. The N-way K-shot learning scheme is a general approach in FSL. The N and K refer to the number of classes the model is trained on and the volume of labeled examples per class, respectively.

IV. RF Fingerprinting System

The overview of the proposed RF fingerprinting system is shown in Fig. 3. This section will introduce it in detail.

A. Problem Definition

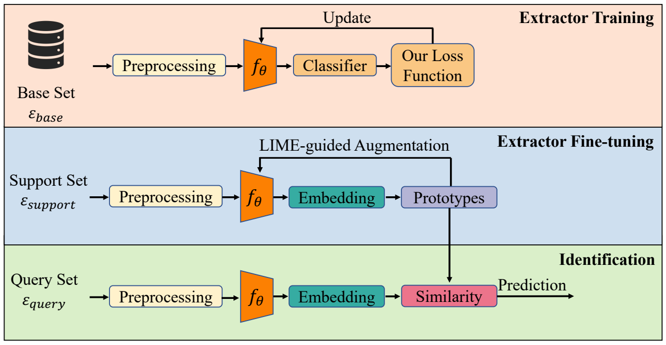

Fig. 3. Overview of RF fingerprinting system.

In this paper, we employ three different public datasets and partition them into various domains according to different days and distances. We denote our input time-domain IQ data, which has a shape of 2×256, as $\mathcal{X} = \{X, E\}$. It encompasses a feature space represented by $X = \{x_1, x_2, \ldots, x_n\}$ and complex environmental factors denoted as $E \sim (T, D)$. Due to the changes in environments E, the input signal $\mathcal{X}$ changes with varying time T and propagation distance D.

As illustrated in Section III, we have a base set of M labeled samples $B = \{(\mathcal{X}_1^b, \mathcal{Y}_1^b), \ldots, (\mathcal{X}_M^b, \mathcal{Y}_M^b)\}$, where $\mathcal{Y}^b$ represents the known devices in the base set $\mathcal{E}_{\text{base}}$. The domains of the base set are defined as the source domains $\mathcal{D}_s$. In our approach, we employ a standard supervised learning method to train both the feature extractor f_θ and a classifier $C(\cdot)$ using data from the base set. We define the target domains denoted as $\mathcal{D}_t$, which comprises domains not seen in the base set $\mathcal{E}_{\text{base}}$. Both the support set $\mathcal{E}_{\text{support}}$ and the query set $\mathcal{E}_{\text{query}}$ are presented in the target domains. In the N-way K-shot scheme, K labeled samples for N devices are gathered to create the support set $\mathcal{E}_{\text{support}}$, which is subsequently used to generate prototypes $\mathbf{c}$ and fine-tune the feature extractor f_θ. Following this, we compare the similarity d of each prototype $\mathbf{c}$ to predict the labels $\mathcal{Y}^q$ of the query set $\mathcal{E}_{\text{query}}$.

To address the domain shift and scalability issue, we formulate two different cases: the *cross-domain case* and the *novel-device case*. In the *cross-domain case*, the target devices remain the same as those in the base set but belong to different domains. On the other hand, the *novel-device case* introduces new devices, encompassing two scenarios: the *in-dataset case* and the *cross-dataset case*. The *in-dataset case* involves novel devices that originate from the same dataset. The *cross-dataset case* involves novel devices from a different dataset, which poses greater challenges.

B. Stable Fingerprint Extraction

The first challenge in building a domain-robust and scalable RF fingerprinting system is to extract stable fingerprint representations from different devices and domains. A stable RF fingerprint is one that changes very little or even remains unchanged over multiple fingerprint extractions. This stability is essential for accurate device classification, as it allows the system to distinguish between devices even if the environmental conditions changed. In this section, we will discuss how to extract stable RF fingerprints. We start by introducing the signal pre-processing of the input IQ samples. Then we describe our feature extractor that is feasible for scalable tasks. Last, we integrate PTN to generate stable RF fingerprints.

1) Signal Pre-processing: As shown in Fig. 4, a preamble is transmitted prior to the start of the main data transmission to help the receiver detect the beginning of the data and synchronize its clock with the transmitter's clock. The frame preamble's structure is typically standardized for a specific wireless communication system. This structure is particularly well-suited for the RF fingerprinting task, as it usually consists of a fixed pattern of recognizable symbols. This stable pattern, which can be easily distinguished from the data, is useful in extracting stable fingerprint features. Furthermore, it also helps reduce privacy concerns when identifying devices since it avoids including data information. Consequently, by isolating the identical structure of the frame preamble, it becomes possible to extract data-agnostic and stable RF fingerprints, as illustrated in Fig. 5.

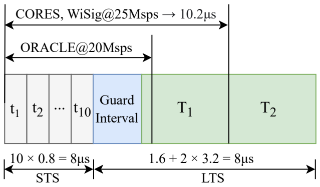

Fig. 4. 802.11a/g frame preamble structure.

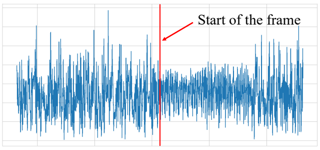

Fig. 5. Detect the start of the preamble.

2) Feature Extraction: CNNs have demonstrated their ability to extract useful features from input data, which makes them well-suited as feature extractors for various models. In this work, we deploy a simple CNN that closely resembles the CNN architecture proposed in [35], to serve as the feature extraction function f_θ, which generates embedding vectors for calculating prototypes. The architecture of our CNN is simple, consisting of only two convolutional blocks and two fully-connected layers. Concretely, the input time-domain IQ data is a two-dimensional vector with the shape of 2×256. The first convolutional block contains a convolution layer with 256 filters of size 1×7 with stride 1, a ReLu activation layer, and a batch normalization layer. The second convolutional block has a similar structure as the previous one, except that the convolution layer is different. The second convolutional layer has 2×7 convolution kernels with stride 1 in 80 output channels. Both of the two layers have a 0×3 padding. Then, the following linear layers consist of 1024 and 256 neurons, respectively. These layers are accompanied by a ReLU activation function and one-dimensional batch normalization. To ensure the embedding vectors reside on a hypersphere with a constant radius, an L_2-norm layer is added prior to the final classification layer as

$$f_\theta(\mathcal{X}_i) = \frac{f'_\theta(\mathcal{X}_i)}{\|f'_\theta(\mathcal{X}_i)\|_2}, \tag{1}$$

where $f'_\theta(\mathcal{X}_i)$ is the output before the L_2-norm $\|\cdot\|_2$, and $f_\theta(\mathcal{X}_i)$ denotes the final feature embeddings. After the feature extraction block, there is a final classifier $C(\cdot)$ that can be adjusted according to the number of known devices. The feature extractor is trained using a traditional supervised learning scheme, employing our base set $\mathcal{E}_{\text{base}}$ as the training data.

3) Stable Fingerprints: Maintaining the stability of device fingerprints is crucial when identifying devices across diverse domains. To achieve this, we employ a modified PTN which involves calculating the mean value of the generated feature vectors, resulting in stable and reliable representative vectors. These representative vectors, known as prototypes, encapsulate the fundamental characteristics of a specific class of devices. By computing prototypes for each device, we obtain generalized representations that remain relatively invariant across different domains. These stable representations play a crucial role in the classification process, enabling accurate and reliable device identification.

After we employ the feature extractor f_θ to obtain the embedding vectors from the input IQ samples as illustrated in Section IV-B2, the prototype for each device is determined by averaging all the embedding vectors belonging to the same class. The computation of prototypes can be expressed as follows:

$$\mathbf{c}_i = \frac{1}{n} \sum_{\mathcal{X}_i \in \mathcal{E}_{\text{data}}}^{n} f_\theta(\mathcal{X}_i), \tag{2}$$

where $\mathbf{c}_i$ denotes the prototypes of device $\mathcal{Y}_i$, and n denotes the number of samples per class in the dataset.

C. Precise Fingerprint Extraction

By implementing the above processes, we can extract stable RF fingerprints for different devices. However, it is still a challenge to ensure that these stable RF fingerprints can be used to effectively distinguish devices across various domains.

1) Similarity Metric: Once the prototypes of each device have been determined, the model can leverage them to generate predictions for new samples. This is achieved by computing the similarity between the feature embedding of the new sample and the prototypes associated with each class. In our experimental setup, we quantify the similarity scores by using cosine similarity as below:

$$\mathbf{D} = d(\mathbf{c}, f_\theta(\mathcal{X}_i)) = \frac{\mathbf{c} \cdot f_\theta(\mathcal{X}_i)}{\|\mathbf{c}\|_2 \, \|f_\theta(\mathcal{X}_i)\|_2}, \tag{3}$$

where $\mathbf{D}$ is the similarity matrix between the input sample $\mathcal{X}_i$ and prototypes of all known devices. The cosine similarity $d(\mathbf{c}, f_\theta(\mathcal{X}_i))$ between a feature embedding and its corresponding prototype ranges from -1 to 1. A value closer to 1 indicates higher similarity. The prediction result will be determined based on the class with the highest similarity score.

In the original PTN [20], authors deploy Euclidean distance as the similarity metric. While both metrics have their advantages, we choose cosine similarity for the following reasons. First, cosine similarity is scale-invariant whose value has a fixed range from -1 to 1, while Euclidean distance is variant. Given our focus on addressing domain shift and scalability issues, we prefer a fixed metric to assess similarity. Euclidean distance may change drastically when introducing novel devices and domains, which is not meet our expectations. Second, cosine similarity focuses on the orientation of embedding vectors. By considering the angular separation rather than the magnitude, cosine similarity allows for a more robust comparison of embedding vectors.

2) Accurate Classification: To ensure that the RF fingerprint features are extracted accurately, the first step is to align the fingerprint feature with its corresponding device class. During the training phase, we only use base set B to train the model. The feature extractor f_θ generates feature embeddings, while the classifier $C(\cdot)$ produces logits. To measure the difference between the predicted outputs and the actual labels, we employ the classic multi-class cross-entropy loss function as follows:

$$\mathcal{L}_{CE} = -\sum_i \mathcal{Y}_i \cdot log(C(f_\theta(\mathcal{X}_i))). \tag{4}$$

By using cross-entropy loss, we can ensure the extracted RF fingerprints can be correctly classified.

3) High Similarity: Given that our system authenticates devices by comparing feature vectors with prototypes rather than relying solely on the output logits from the classifier, using only cross-entropy loss for model training can potentially introduce biases. To address this concern, we propose a similarity loss, which aims to generate an RF fingerprint that exhibits high similarity to the corresponding prototype.

Since we already have a cross-entropy loss function to facilitate the accurate classification of input data, the similarity loss function mainly focuses on optimizing our model from a similarity perspective. First, the cosine similarity between the feature vector of the true label and its corresponding prototype should be maximized. This high similarity assists the model in making accurate classifications. Second, the maximum similarity score in the similarity matrix $\mathbf{D}$ should be as high as possible. Hence, we compute similarity loss as

$$\mathcal{L}_S = \alpha \cdot (1 - d_{true}) + \beta \cdot (1 - d_1), \tag{5}$$

where d_{true} denotes the similarity score of true labels, and d_1 represents the highest similarity score. By employing similarity loss, the model is optimized to output feature embeddings with higher similarity to the corresponding prototype. It is important to mention that we avoid directly calculating the absolute error between the maximum similarity and the similarity of the true labels. This is because it may lead to the undesired situation that the two similarities may be the same but relatively low, even if the classification is correct.

4) High Discriminability: Deploying the above similarity loss can yield feature vectors that closely resemble the target prototype. However, this may also generate interfering features that exhibit high similarity to the prototype. To overcome this issue, we propose a discriminability loss function which is inspired by the concept of triplet loss. The purpose of this discriminability loss is to encourage feature embeddings to exhibit a low similarity to other prototypes. By incorporating this discriminability loss, we aim to enhance the distinctiveness of the feature vector in relation to its associated prototype. Therefore, we have

$$\mathcal{L}_D = max(0, \epsilon + d_2 - d_1), \tag{6}$$

where d_2 denotes the second highest similarity score, and ϵ represents the discriminability level.

Overall, the loss function of our system is a combination of cross-entropy loss, similarity loss, and discriminability loss as

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{CE} + \lambda_2 \cdot \mathcal{L}_S + \lambda_3 \cdot \mathcal{L}_D, \tag{7}$$

where λ_1, λ_2, and λ_3 are coefficients that control the significance of the three loss components. By implementing this customized loss function, our RF fingerprinting system is able to extract more precise fingerprints.

D. Few-shots Fine-tuning

In certain challenging scenarios, fine-tuning becomes essential to improve the system performance. However, it is important to note that our system operates on a few samples from the support set. This limited available samples may hinder the ability of the model to effectively generalize to the challenging task. Data augmentation is a widely used technique that helps machine learning models improve their generalization ability and make accurate predictions on previously unseen data. However, due to the small size of the support set and the unintuitive nature of the input IQ samples, using inappropriate data augmentation methods may degrade the model's performance. To address this challenge, we employ LIME to guide the data augmentation process.

Using the inputs provided by the support set $\mathcal{E}_{\text{support}}$, we can generate predictions by our system. Since our system primarily focuses on extracting domain-invariant features to output results, we employ LIME to identify the specific areas of focus. We first partition the support time-domain IQ data $\mathcal{X}_i^s$ into 16 smaller segments. From these segments, we randomly select subsets to create perturbed IQ samples $\mathcal{X}_i^p$. We then feed these perturbed samples into our system to generate corresponding perturbed predictions $\mathcal{Y}_i^p$. Next, we compute the cosine distances between the perturbed data and the original data, which serves as the weights of the perturbed samples. Subsequently, we train a linear regression model (i.e., an explainable model) using the perturbed samples, associated weights, and perturbed predictions. The resulting coefficients obtained from this linear model indicate the level of attention our system assigns to different sections of the data. Larger coefficients signify a higher degree of focus on specific areas.

Algorithm 1 Feature extractor fine-tuning with LIME-guided augmentation

```
INPUT: Support set $\mathcal{E}_{\text{support}} = \{(\mathcal{X}_i^s, \mathcal{Y}_i^s)^K, i = 1, \dots, N\}$,
    feature extractor $f_\theta$, classifier $C$, learning rate $lr$, hyper-
    parameters $\alpha, \beta, \epsilon, \lambda_1, \lambda_2, \lambda_3$
OUTPUT: fine-tuned feature extractor $f_\theta$
  Step 1: Fine-tune with support set
  for number of epoch do
    for $(\mathcal{X}_i^s, \mathcal{Y}_i^s)^K \in \mathcal{E}_{\text{support}}$ do
      $c_i \leftarrow \frac{1}{K} \sum^K f_\theta(\mathcal{X}_i)$
      $\mathbf{D} \leftarrow CosineSimilarity(\mathbf{c}, f_\theta(\mathcal{X}_i^s))$
      $(d_i, d_1, d_2) \leftarrow (\mathbf{D}_i, max(\mathbf{D}), secondmax(\mathbf{D}))$
      $\mathcal{L}_{CE} \leftarrow CrossEntropy(C(f_\theta(\mathcal{X}_i^s)), \mathcal{Y}_i^s)$
      $\mathcal{L}_S \leftarrow \alpha \cdot (1 - d_i) + \beta \cdot (1 - d_1)$
      $\mathcal{L}_D = max(0, \epsilon + d_2 - d_1)$
      $\mathcal{L} = \lambda_1 \cdot \mathcal{L}_C E + \lambda_2 \cdot \mathcal{L}_S + \lambda_3 \cdot \mathcal{L}_D$
    end for
    $\theta \leftarrow \theta - lr \cdot \nabla_\theta \mathcal{L}$
  end for
  Step 2: LIME-guided augmentation
  for $(\mathcal{X}_i^s, \mathcal{Y}_i^s)^K \in \mathcal{E}_{\text{support}}$ do
    $\mathcal{X}_i^p \leftarrow Segment(\mathcal{X}_i^s)$
    $\mathcal{Y}_i^p \leftarrow argmax(CosineSimilarity(\mathbf{c}, f_\theta(\mathcal{X}_i^p))$
    $Regions \leftarrow LinearRegression(\mathcal{X}_i^p, \mathcal{Y}_i^p, d(\mathcal{X}_i^p, \mathcal{X}_i^s))$
    $\mathcal{X}_i^a \leftarrow \mathcal{X}_i^s[Regions]$
    $\mathcal{E}_{\text{augment}} \leftarrow \mathcal{E}_{\text{support}} + \{(\mathcal{X}_i^a, \mathcal{Y}_i^s)^K, i = 1, \dots, N\}$
  end for
  Step 3: Fine-tune with augmented set
  for number of epoch do
    $f_\theta \leftarrow FineTune(\mathcal{E}_{\text{augment}})$
  end for
  return $f_\theta$
```

In this study, we augment the support data by preserving values in the segments of the top 10 largest coefficients and setting all other values to zero. This approach aims to compel our feature extractor to extract robust fingerprint features from important segments and ignore interference from other segments. Meanwhile, considering the limited volume of data in the support set, this procedure does not substantially increase time complexity.

E. Summary

In this section, we will introduce how to integrate these blocks to train our system and make predictions. First, our customized loss function is used to optimize the feature extractor on the base set using a classic supervised learning scheme. This ensures that the feature extractor extract features that are discriminative for the different device classes. Then, we propose a LIME-guided augmentation with fine-tuning to improve the performance on the challenging tasks. The pseudocode of implementing LIME-guided augmentation in fine-tuning is described in Algorithm 1. The first step is fine-tuning the model with a small number of iterations. This allows the extractor warm up to the features of the new samples. Next, LIME is deployed to guide data augmentation for the fine-tuned model. Last, the model is fine-tuned again using the augmented dataset. The whole process is not time-consuming because the amount of data and the number of iterations are small.

V. Experimental Evaluation

A. Experiment Setup

In all experiments, the learning rate was set to 0.0001. K_{shot}, N_{query}, and max epochs were set to 5, 20, 50, respectively. The value of N_{way} was set to the size of all label sets for the datasets involved in a given experiment. The coefficients $\lambda_1, \lambda_2, \lambda_3$ for the loss function $\mathcal{L}$ were set to 1.0, 0.8, and 0.8. For the similarity loss $\mathcal{L}_S$, the alpha and beta coefficients were 1.2 and 1.0, respectively. The experiments were conducted on a server with an Intel Xeon E5-2650L v4 CPU and 8 NVIDIA GeForce GTX 1080Ti GPU.

For the in-dataset case, we selected data with partial domains as the base set, and the remaining data were allocated to the support and query sets. For the cross-dataset scenario, we designated devices with all domains as the base set, while new devices with all domains constituted the target domains.

B. Datasets

We leverage three public datasets in this paper. Table I shows brief information on these datasets.

1) ORACLE: The original ORACLE dataset [35] is captured with 16 USRP X310 transmitters and a USRP B210 receiver at 6-foot increments from 2 to 62 feet. The dataset is divided into ORACLE.1 and ORACLE.2 based on time. We also include ORACLE.F1 and ORACLE.F2, which are generated by frame isolation as mentioned in Section IV-B1. We use distance as the domain partition criterion. Due to an inadequate number of frames, we exclude distances of 2, 56, and 62 feet. As a result, there are a total of eight domains in each ORACLE dataset. We randomly select 10 devices in 4 domains as the base set B, the remaining data are used to discuss the domain shift and scalability issues.

2) CORES: The original dataset [34] consists of 163 consumer Wi-Fi cards arranged in a grid at the Orbit Testbed [36]. This dataset was collected by the UCLA CORES lab and is hereafter referred to as CORES. In this work, we use the 58 devices in all five days of this dataset, where each day represents a distinct domain. The base set of CORES comprises a total of 30 devices across 2 domains.

TABLE I
Dataset summaries.

Dataset Name	Emitter Models	Examples	Domains
ORACLE.1	16 USRP X310	1,280,000	8
ORACLE.2	16 USRP X310	1,280,000	8
ORACLE.F1	16 USRP X310	256,000	8
ORACLE.F2	16 USRP X310	256,000	8
CORES	58 COTS Wi-Fi Cards	250,681	5
WiSig	130 COTS Wi-Fi Cards	270,616	4

3) WiSig: Conducted by the same team as the CORES, the WiSig dataset [15] is collected by 41 unspecified USRP receivers to capture wireless signals from 174 COTS Wi-Fi cards. Being much larger than previous datasets, we use data from only one receiver (labeled ”node3-19”) for simplicity. We use the 130 emitters present on all four days. The base set is constructed using 100 devices across two domains.

C. Evaluation on the Cross-domain Case

In the cross-domain case, our primary focus is to address domain shift. We train our model using source domains and then test its performance on unknown domains. To demonstrate the robustness of our system with respect to domain shift, we also evaluate several classic methods in this setting. The results of various methods in the cross-domain scenario are presented in Table II. While the performance varies across models, they all exhibit satisfactory results in source domains. However, when it comes to target domains, their accuracy significantly decreases. In particular, when considering the ORACLE dataset, even the K-nearest neighbor (KNN) [37], CNN [35] and LSTM [11] models that perform well in source domains (above 90%), struggle to achieve acceptable performance in target domains, with only about 7% accuracy. This can be attributed to the excessive impact of distance on the signal strength, resulting in the failure classification. The accuracy still remains inadequate even with ADA [38] and ADA+KNN [27] methods, likely because the extracted domain-invariant features are not related to fingerprints. Besides, all models demonstrate superior performance on CORES and WiSig datasets compared to the ORACLE datasets. As previously mentioned in Section V-B, these two datasets were collected by the same team and partitioned according to different days. This suggests that domain shifts that occur at different distances can be more disruptive than domain shifts that occur in the same environment over time.

Our method consistently outperforms other models across all datasets, even in cases where other models perform well. For example, on the WiSig and CORES datasets, our method achieves exceptionally high accuracy rates of 95% and 99%, respectively. On the ORACLE dataset without signal preprocessing, our method effectively enhances accuracy by approximately 70%. Furthermore, for the framed ORACLE datasets, our method achieves accuracy improvements exceeding 80%. These results clearly highlight the consistent and substantial enhancements delivered by our approach.

TABLE II
THE PERFORMANCE OF OUR PROPOSED SYSTEM AND BASELINE METHODS IN THE CROSS-DOMAIN CASE.

	WiSig		CORES		ORACLE.1		ORACLE.2		ORACLE.F1		ORACLE.F2	
	Source	Target	Source	Target	Source	Target	Source	Target	Source	Target	Source	Target
KNN [37]	0.9158	0.7268	0.9909	0.7912	0.3407	0.0628	0.3421	0.0605	0.8604	0.1089	0.8604	0.1090
CNN [35]	**0.9817**	0.7658	0.9997	0.7521	**0.9091**	0.0680	**0.9131**	0.0710	0.9529	0.0578	0.9550	0.0639
LSTM [11]	0.9507	0.7347	0.9924	0.6419	0.6199	0.0625	0.6270	0.0623	0.8755	0.1243	0.8757	0.1267
ADA [38]	0.8476	0.5203	0.9028	0.6255	0.4711	0.0752	0.4071	0.0765	0.6090	0.0606	0.5910	0.0536
ADA+KNN [27]	0.9099	0.6922	0.9885	0.7751	0.5491	0.0676	0.5149	0.0682	0.6537	0.1072	0.6766	0.1059
Proposed	0.9774	**0.9512**	**0.9999**	**0.9917**	0.8678	**0.7809**	0.8880	**0.7910**	**0.9809**	**0.8770**	**0.9825**	**0.8630**

D. Novel-device Case

As mentioned in Section IV-A, traditional deep learning techniques are not convenient for handling scenarios that involve introducing new devices. To investigate the scalability of our system, we conduct evaluations in both the in-dataset case and the cross-dataset case. In the in-dataset case, our system mainly focuses on addressing scalability concerns since the domain information remains similar. In contrast, the cross-dataset case presents a more challenging scenario where both the device type and domain information undergo changes, introducing difficulties associated with domain shift and scalability.

TABLE III
TARGET ACCURACY OF THE IN-DATASET CASE. THE ABBREVIATIONS FT, GN, AND LIME STAND FOR FINE-TUNING, GAUSSIAN NOISE AUGMENTATION, AND LIME-GUIDED AUGMENTATION RESPECTIVELY.

	Baseline	+ FT	+ GN	+ LIME
ORACLE.F1	0.9233	0.9344	0.9079	0.9173
ORACLE.F2	0.9179	0.9219	0.8958	0.9208
ORACLE.1	0.8546	0.8698	0.8229	0.8708
ORACLE.2	0.8662	0.8781	0.8489	0.8958
WiSig	0.9515	0.9608	0.9621	0.9666
CORES	0.9668	0.9746	0.9804	0.9843

1) In-dataset case: Table III presents the results of novel device authentication under the in-dataset case. This table demonstrates that our system has the capability to deliver outstanding performance concerning the scalability challenge. To substantiate the effectiveness of our proposed LIME-guided data augmentation, we implemented Gaussian noise (GN) as an alternative augmentation method for contrast.

For the ORACLE dataset, the pre-processed datasets (ORACLE.F1 and ORACLE.F2) achieve higher accuracy, aligning with the trend observed in the cross-domain case. This confirms the effectiveness of pre-processing. Interestingly, while fine-tuning enhances the accuracy across all datasets, data augmentation appears to negatively affect accuracy in the framed ORACLE datasets. However, this reduction in accuracy is acceptable given the high overall accuracy. Among all scenarios, our system achieves the highest accuracy of 0.9843 on the CORES dataset. The most substantial improvement through fine-tuning and LIME-guided augmentation is observed in ORACLE.2, where accuracy increases by almost 3%.

2) Cross-dataset case: Fig. 6 shows the performance of our system on the *cross-dataset case*, where new devices with different manufacturers and domains present a greater challenge compared to the previous case. When the ORACLE dataset is used as the base set, our system performs well on other datasets. However, when ORACLE is employed as the target dataset, the accuracy decreases significantly. This is because ORACLE divides domains by distance, which may lead to greater differences between domains, thus making it more difficult for the system to adapt.

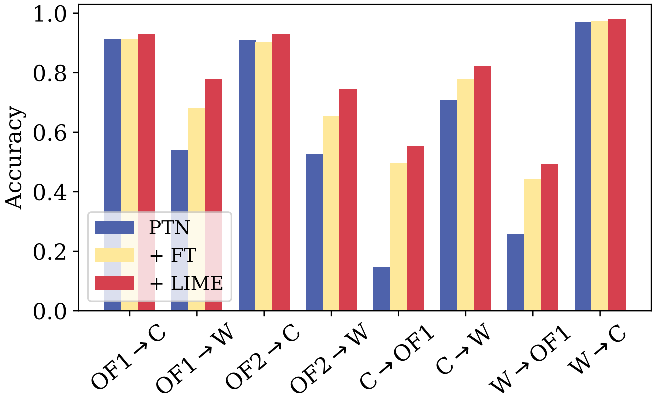

Fig. 6. Identification performance in the cross-dataset case. The abbreviations C, W, OF1, and OF2 stand for CORES, WiSig, ORACLE.F1, and ORACLE.F2, respectively. OF1→C denotes that our system is trained on ORACLE and tested on novel devices from the CORES dataset.

We address this by leveraging fine-tuning and LIME-guided augmentation techniques, which significantly improve the lower accuracies and further boost the already good performance. Specifically, the worst-case scenario, CORES→ORACLE.F1, initially achieves an accuracy of only 14.57%. However, after implementing fine-tuning and LIME-guided augmentation, the accuracy experiences a remarkable improvement, reaching 55.31%. On the other hand, the best-case scenario, WiSig→CORES, already achieves a high accuracy of 96.81% using PTN alone. With the integration of fine-tuning and LIME-guided augmentation, the accuracy further increases to 98.05%.

E. Evaluation on Similarity Metrics and Loss Functions

Table IV presents the results from different similarity metrics and loss functions. The original PTN uses Euclidean distance to determine the similarity. However, it appears that the cosine similarity provides more effective measurements to our system when it comes to RF fingerprinting. This is probably because cosine similarity is more effective at measuring the similarity between fingerprint feature vectors within a multi-dimensional space. In comparison to the classic cross-entropy loss, our customized loss function also contributes to improving RF fingerprinting classification accuracy. The impact of our loss function on accuracy is more pronounced in the cross-dataset case as compared to the cross-domain scenario. For example, the accuracy increases from 84.52% to 91.13% in the OF1→C case. These findings demonstrate the effectiveness of our customized loss function in RF fingerprint extraction and classification.

TABLE IV
TARGET ACCURACY IN CROSS-DOMAIN AND CROSS-DATASET SCENARIOS WITH DIFFERENT SIMILARITY METRICS AND LOSS FUNCTIONS.

	Cross-Entropy Loss		Customized Loss		
	Euclidean	Cosine	Baseline	GN	LIME
OF1	0.7809	0.8172	0.8288	0.8210	0.8770
OF2	0.7672	0.8492	0.8396	0.8480	0.8630
O1	0.7136	0.7516	0.7634	0.7090	0.7809
O2	0.7225	0.7699	0.7712	0.6910	0.7910
W	0.8753	0.9355	0.9464	0.9503	0.9512
C	0.9771	0.9850	0.9827	0.9928	0.9971
OF1→W	0.4963	0.5021	0.5406	0.7650	0.7792
OF1→C	0.8175	0.8452	0.9113	0.9162	0.9282

F. Evaluation on LIME-guided Data Augmentation

Table V shows the results of using different data augmentation techniques in the cross-dataset scenario. In conjunction with the previous findings, our proposed LIME-guided data augmentation demonstrates an enhanced performance for our RF fingerprinting system. We can see that the CORES→ORACLE.F1 case shows the most significant improvement, with an accuracy increase of approximately 41% compared to the baseline PTN. LIME is deployed to understand the important regions of the input IQ samples, allowing us to retain these parts as the augmented data. Consequently, our system becomes better at extracting features from these regions, making it more resilient to irrelevant disturbances. For instance, Gaussian noise data augmentation is ineffective for the ORACLE dataset in most cases, resulting in a decline in model accuracy. However, using LIME-guided data augmentation, accuracy improvements are observed across all cases. In particular, in the cross-domain scenario of ORACLE datasets, where Gaussian noise significantly decreases accuracy, LIME-guided data augmentation continues to improve model performance.

G. Evaluation on Hyperparameter

In the N-way K-shot scheme of FSL, it is generally observed that higher accuracy is achieved with an increase in the number of shots. Fig. 7 shows the trend of accuracy improvement as the number of shots increases. In cases where the accuracy of a 1-shot approach is insufficient, such as W→OF1, a significant improvement of approximately 35% can be achieved by using 20 shots. However, it is important to note that this increase is not linear. Improvements in accuracy are most significant from 1-shot to 5-shots, after which accuracy tends to improve more slowly. Increasing the number of shots in FSL can improve accuracy, but it also requires more data and computational resources. This can be a challenge, as it contradicts the original intent of our system, which is to generalize from a small amount of data. Therefore, the choice of the number of shots is a trade-off between accuracy gains and resource requirements. On the other hand, Fig. 8 shows that our system is relatively stable as the number of queries varies.

TABLE V
TARGET ACCURACY IN CROSS-DATASET SCENARIOS WITH DIFFERENT AUGMENTATION TECHNIQUES.

	OF2→W	OF2→C	W→OF1	W→C	C→OF1	C→W
GN	0.7297	0.9155	0.4625	0.9762	0.5113	0.8104
LIME	0.7442	0.9298	0.4938	0.9805	0.5531	0.8218

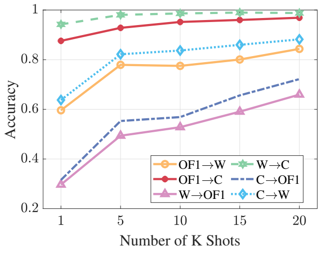

Fig. 7. The performance of our system varies with the number of training shots (K).

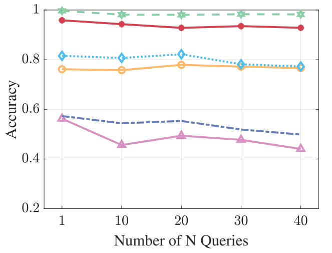

Fig. 8. The performance of our system varies with the number of testing queries.

VI. CONCLUSION

This paper presented a novel approach for building a robust RF fingerprinting system to effectively address the challenges of domain shift and scalability. To overcome these challenges, our system employed a modified PTN to enable adaptation to new domains and devices with only a few samples. To further enhance performance, we designed a customized loss function and developed a LIME-guided data augmentation technique. We extensively evaluated the capabilities of our system across various scenarios and datasets. Our results demonstrated that our approach outperformed other methods in addressing domain shift issues. To the best of our knowledge, this study is the first to comprehensively address these challenges across different datasets and achieve outstanding performance.

ACKNOWLEDGMENTS

This work is supported in part by the NSF (CNS-2321763, CNS-2319343, CNS-2317190, IIS-2306791, IIS-2306789, CNS-2107190, and CNS-2319342).

References

[1] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (iot) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, 2019.

[2] E. Perenda, S. Rajendran, G. Bovet, S. Pollin, and M. Zheleva, "Learning the unknown: Improving modulation classification performance in unseen scenarios," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[3] M. Lehtonen, D. Ostojic, A. Ilic, and F. Michahelles, "Securing RFID systems by detecting tag cloning," in *International Conference on Pervasive Computing*. Springer, 2009, pp. 291–308.

[4] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[5] D. Formby, P. Srinivasan, A. M. Leonard, J. D. Rogers, and R. A. Beyah, "Who's in control of your control system? device fingerprinting for cyber-physical systems." in *NDSS*, 2016.

[6] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," *Canadian Journal of Electrical and Computer Engineering*, vol. 32, no. 1, pp. 27–33, 2007.

[7] Q. Tian, Y. Lin, X. Guo, J. Wen, Y. Fang, J. Rodriguez, and S. Mumtaz, "New security mechanisms of high-reliability iot communication based on radio frequency fingerprint," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7980–7987, 2019.

[8] M. Kheir, H. Kreft, and R. Knöchel, "UWB on-chip fingerprinting and identification using carbon nanotubes," in *2014 IEEE International Conference on Ultra-WideBand (ICUWB)*. IEEE, 2014, pp. 462–466.

[9] A. Al-Shawabka, P. Pietraski, S. B. Pattar, F. Restuccia, and T. Melodia, "DeepLoRa: Fingerprinting LoRa devices at scale through deep learning and data augmentation," in *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2021, pp. 251–260.

[10] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRa using spectrogram and CNN," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[11] ——, "Radio frequency fingerprint identification for LoRa using deep learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2604–2616, 2021.

[12] Q. Pan, Z. An, X. Yang, X. Zhao, and L. Yang, "RF-DNA: large-scale physical-layer identifications of rfids via dual natural attributes," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 419–431.

[13] H. J. Patel, M. A. Temple, and R. O. Baldwin, "Improving ZigBee device network authentication using ensemble decision tree classifiers with radio frequency distinct native attribute fingerprinting," *IEEE Transactions on Reliability*, vol. 64, no. 1, pp. 221–233, 2014.

[14] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, 2008, pp. 116–127.

[15] S. Hanna, S. Karunaratne, and D. Cabric, "WiSig: A large-scale wifi signal dataset for receiver and channel agnostic rf fingerprinting," *IEEE Access*, vol. 10, pp. 22 808–22 818, 2022.

[16] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia, "Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 646–655.

[17] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong, "Accurate and efficient wireless device fingerprinting using channel state information," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1700–1708.

[18] D. A. Knox and T. Kunz, "AGC-based RF fingerprints in wireless sensor networks for authentication," in *2010 IEEE International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2010, pp. 1–6.

[19] G. Huang, Y. Yuan, X. Wang, and Z. Huang, "Specific emitter identification based on nonlinear dynamical characteristics," *Canadian Journal of Electrical and Computer Engineering*, vol. 39, no. 1, pp. 34–41, 2016.

[20] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[22] P. Robyns, E. Marin, W. Lamotte, P. Quax, D. Singelée, and B. Preneel, "Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017, pp. 58–63.

[23] I. Agadakos, N. Agadakos, J. Polakis, and M. R. Amer, "Chameleons' oblivion: Complex-valued deep neural networks for protocol-agnostic rf device fingerprinting," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020, pp. 322–338.

[24] 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)*. IEEE, 2018, pp. 1–9.

[25] L. Chen, C. Zhao, Y. Zheng, and Y. Wang, "Radio frequency fingerprint identification based on transfer learning," in *2021 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2021, pp. 81–85.

[26] J. Yu, A. Hu, G. Li, and L. Peng, "A robust RF fingerprinting approach using multisampling convolutional neural network," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6786–6799, 2019.

[27] H. Li, K. Gupta, C. Wang, N. Ghose, and B. Wang, "RadioNet: Robust deep-learning based radio fingerprinting," in *2022 IEEE Conference on Communications and Network Security (CNS)*, 2022, pp. 190–198.

[28] Z. Ren, P. Ren, and T. Zhang, "Deep RF device fingerprinting by semi-supervised learning with meta pseudo time-frequency labels," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 2369–2374.

[29] Y. Yang and T. Yan, "Radio frequency fingerprint recognition method based on generative adversarial net," in *2021 13th International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2021, pp. 361–364.

[30] G. Shen, J. Zhang, A. Marshall, and J. R. Cavallaro, "Towards scalable and channel-robust radio frequency fingerprint identification for lora," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 774–787, 2022.

[31] J. Zhang, Z. Chen, C. Luo, B. Wei, S. S. Kanhere, and J. Li, "Metaganfi: Cross-domain unseen individual identification using wifi signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–21, 2022.

[32] G. Yin, J. Zhang, G. Shen, and Y. Chen, "Fewsense, towards a scalable and cross-domain wi-fi sensing system using few-shot learning," *IEEE Transactions on Mobile Computing*, 2022.

[33] C. Feng, N. Wang, Y. Jiang, X. Zheng, K. Li, Z. Wang, and X. Chen, "Wi-learner: Towards one-shot learning for cross-domain wi-fi based gesture recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–27, 2022.

[34] S. Hanna, S. Karunaratne, and D. Cabric, "Open set wireless transmitter authorization: Deep learning approaches and dataset considerations," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 59–72, 2020.

[35] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized radio classification through convolutional neural networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 370–378.

[36] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE Wireless Communications and Networking Conference, 2005*, vol. 3. IEEE, 2005, pp. 1664–1669.

[37] Y. Li, Y. Lin, Z. Dou, and Y. Chen, "Research on RF fingerprint feature selection method," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[38] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.