IEEE ComSoc | IEEE COMPUTER SOCIETY | IEEE Signal Processing | VTS — IEEE Transactions on Machine Learning in Communications and Networking

Received 3 November 2023; revised 24 April 2024 and 14 July 2024; accepted 13 August 2024.
Date of publication 20 August 2024; date of current version 23 September 2024.

The associate editor coordinating the review of this article and approving it for publication was A. Eltawil.

Digital Object Identifier 10.1109/TMLCN.2024.3446743

Distinguishable IQ Feature Representation for Domain-Adaptation Learning of WiFi Device Fingerprints

ABDURRAHMAN ELMAGHBUB AND BECHIR HAMDAOUI (Senior Member, IEEE)

Oregon State University, Corvallis, OR 97330 USA

CORRESPONDING AUTHOR: A. ELMAGHBUB (elmaghba@oregonstate.edu)

This work was supported in part by NSF/Intel under Award 2003273.

ABSTRACT Deep learning (DL)-based RF fingerprinting (RFFP) technology has emerged as a powerful physical-layer security mechanism, enabling device identification and authentication based on unique device-specific signatures that can be extracted from the received RF signals. However, DL-based RFFP methods face major challenges concerning their ability to adapt to domain (e.g., day/time, location, channel, etc.) changes and variability. This work proposes a novel IQ data representation and feature design, termed Double-Sided Envelope Power Spectrum or `EPS`, that is proven to significantly overcome the domain adaptation challenges associated with WiFi transmitter fingerprinting. By accurately capturing device hardware impairments while suppressing irrelevant domain information, `EPS` offers improved feature selection for DL models in RFFP. Our experimental evaluation demonstrates the effectiveness of the integration of `EPS` representation with a Convolution Neural Network (CNN) model, termed `EPS-CNN`, achieving over 99% testing accuracy in same-day/channel/location evaluations and 93% accuracy in cross-day evaluations, outperforming the traditional IQ representation. Additionally, `EPS-CNN` excels in cross-location evaluations, achieving a 95% accuracy. The proposed representation significantly enhances the robustness and generalizability of DL-based RFFP methods, thereby presenting a transformative solution to IQ data-based device fingerprinting.

INDEX TERMS RF/device fingerprinting, domain adaptation, RF datasets, deep learning feature design, oscillators, RF data representation, envelope analysis, hardware impairments.

I. INTRODUCTION

DEEP learning (DL)-based RF fingerprinting (RFFP) emerges as a powerful physical-layer security mechanism [1], [2], [3], [4], [5], [6], enabling device identification and authentication through the extraction of unique device fingerprints embedded in the devices' transmitted RF signals. These fingerprints arise as a result of inherent hardware manufacturing imperfections of various RF circuitry components (e.g., local oscillators, mixers, power amplifiers) yielding RF signal distortions [2], [7] that collectively shape distinctive device signatures that can be extracted using DL models. Although DL has eliminated the need for data preprocessing and domain-knowledge to extract features from raw RF data, most of the DL-RFFP approaches rely on the assumption that the training and testing data are drawn from the same distribution, which falls short of the conditions of realistic RF scenarios [8], [9]. In other words, these approaches do not perform well in practical scenarios, in which the testing data is collected under a domain that is different from that used during training, where a *domain* here refers to a network condition (e.g., setting, environment) under which data is collected. This includes the collection time, the channel condition, the receiver hardware, the device location, and the protocol configuration, as well as other aspects. Therefore, any considerable change in the testing versus training settings yields a different domain, and as such, we define robustness to domain changes as the ability of a learning model to maintain its training domain performance when tested under new domains. This is often referred to as *domain adaptation* in the machine learning community.

Several experimental studies using LoRa and WiFi devices (e.g., [7], [10], [11], [12]) have revealed the sensitivity of DL-based RFFP approaches to domain changes, thereby limiting their practical adoption to security applications. Unraveling

such sensitivity issues is a complex task involving two black boxes: the deep learning model and the microelectronic circuitry of the RF devices. However, it is widely believed that the wireless channel, influenced by various confounding factors, plays a significant role in the failure of these approaches to adapt and generalize to different domains. To address the impact of the channel and enhance domain generalization, some studies have focused on removing channel dynamics from the raw signal through techniques like channel equalization [12], [13], [14], [15] or hardware impairment compensation [6]. However, these approaches have drawbacks. Channel equalization can inadvertently remove crucial device-specific features, resulting in discriminative information loss and poor RF fingerprinting performance [16]. Impairment compensation techniques, on the other hand, target specific channel impairments, limiting their generalizability across different environments and wireless channels.

Recent advancements in domain adaptation techniques have sparked interest for their potential to boost the robustness of RF fingerprinting systems by reducing domain-related biases in feature vectors. These methods, including adversarial learning, adversarial disentanglement learning, and cycle-consistent generative adversarial networks, show promise in distinguishing device-specific characteristics from those tied to the operational domain [17]. Despite their potential, these techniques face significant challenges that question their practicality in real RF fingerprinting contexts. For instance, [18] utilizes adversarial domain adaptation paired with a k-NN classifier on a modest dataset of 10 HackRF WiFi devices. Its two-day evaluation yields a classification accuracy of 64%, underscoring difficulties in reaching the accuracy needed for reliable RF fingerprinting. Moreover, the dependence on target domain data for effective training and fine-tuning presents an additional challenge. This reliance is problematic because access to target data may not always be available beforehand, and being constrained to a single-domain adaptation can lead to biases toward that specific target domain. Consequently, this necessitates retraining or fine-tuning the system each time there is a change in the target domain, such as a different day/location in our case studies. Similarly, [19] employs a combination of disentanglement representation and adversarial learning, tested on 50 WiFi devices. While it shows some success in short-term evaluations, its performance drastically declines in longer-term assessments, with an average accuracy plummeting to 15% over several days, which signals issues with maintaining effectiveness over time. Furthermore, [20] leverages a cycle-consistent generative adversarial network to create an environment translator aimed at separating hardware impairments from channel and environmental influences. Although this model achieved 83% accuracy in tests with a small dataset of 5 WiFi devices, its accuracy fell to 34% when applied to a larger group of 20 devices. These examples illustrate that while domain adaptation methods hold potential, their current applications in RF fingerprinting still encounter substantial hurdles that must be addressed to enhance their viability and effectiveness in real-world scenarios.

Most DL-based RFFP approaches use the time-domain IQ representation of the received RF signal as the input for the learning models. This is because of the ability of these models to extract relevant features from the raw IQ data without needing preprocessing or prior domain knowledge. However, recent studies [7], [21] have indicated that DL models relying solely on IQ values fail to adapt to domain changes, such as changes in the wireless channel and/or receiver hardware [22]. This phenomenon may arise because I/Q data representations often include a substantial amount of extraneous information that does not pertain directly to hardware fingerprinting, such as conveyed content and other domain-specific features. As a result, models are prone to overfitting to specific training environments like the communication channel, receiver hardware, and device placement, rather than effectively learning the unique hardware characteristics of each device [11]. Additionally, deep learning models that utilize neural network layers originally designed for image processing often struggle to extract relevant information from time-domain signal samples [23]. This mismatch in model design can lead to poor generalization when faced with new I/Q data that slightly varies in signal parameters from the training set. Our discussion does not seek to entirely dismiss the value of deep learning models from other domains. Instead, it emphasizes the need for a nuanced approach to adapting these models for RF fingerprinting. This adaptation is crucial to address the unique challenges posed by RF data, underscoring the importance of developing novel RF data representations that effectively capture the hardware impairments of devices and align more closely with the operational strengths of these models. Such tailored representations can enhance the feature selection process, enabling the models to focus on relevant and reliable features and reducing their reliance on misleading data.

To fulfill this need, this paper proposes a novel RF data representation that significantly enhances the robustness of DL-based RFFP methods to domain changes. Our motivation stems from the realization that raw IQ data representation contains a significant amount of device-irrelevant information. Consequently, extracting meaningful fingerprints from this raw IQ data becomes akin to finding a needle in a haystack filled with numerous deceptive needle-like objects. We overcome this limitation by proposing a novel RF data representation that vividly captures the device's hardware impairments while suppressing device-irrelevant information [24]. Specifically, the proposed data representation closely mirrors the impaired behavior of a key RF hardware component, the oscillator, whose impairment substantially contributes to the device's unique fingerprint [25], [26], [27]. To generate this representation, we extract the outer shape or envelope of the IQ signal, eliminate the resulting amplitude offset, and calculate the double-sided envelope's power

spectrum or simply EPS, yielding a novel data representation of the IQ signal that serves as an effective input for machine learning classifiers.

Through extensive evaluation on a testbed of 15 Pycom devices, running IEEE 802.11b WiFi protocol, we demonstrate the effectiveness of our proposed approach in real-world scenarios. Our experimental results show that when combined with standard CNN (Convolution Neural Network) models, the proposed EPS-based device fingerprinting framework achieves outstanding performance. Notably, it achieves a testing accuracy of over 99% in same-domain (day or location) scenarios, where training and testing are done on the same day/location. More importantly, in cross-location scenarios, the proposed framework maintains a testing accuracy of over 95%, whereas the same model trained with the conventional IQ representation achieves only 55%. Therefore, our EPS representation offers a transformative solution that significantly advances the RF fingerprinting field by substantially improving the accuracy and generalizability of DL-based RFFP approaches.

Our key contributions can thus be summarized as follows:

- We propose EPS, a novel RF signal representation input to DL-based RFFP approaches that substantially enhances the accuracy, robustness, and generalizability across various domains and hardware configurations.
- We demonstrate through extensive experimentation the distinguishability and reliability of the EPS representation across time, channel, location, and receiver domains, justifying its applicability to RF fingerprinting applications.
- We release massive 8TB IEEE 802.11b WiFi datasets of 15 Pycom devices that include both raw and processed files for more than 5000 packets for each device for four scenarios: Wired Setup, Wireless Setup, Different Locations Setup, and Random Deployment Setup.
- We extensively assess the performance of EPS when used as an input to a standard CNN, EPS-CNN, for classifying Pycom devices and showcase an exceptional cross-domain performance in real-world scenarios, achieving an average testing accuracy of 93% and 95% respectively for the cross-days and cross-location scenarios.
- We evaluate the effectiveness of integrating the EPS representation with other models, using the ResNet-18 model as a case study. This integration demonstrated a significant improvement, achieving a +50% increase in accuracy compared to the traditional ResNet-18 model which uses an IQ input.
- We demonstrate the impact of the carrier frequency instability during device hardware warm-up and stabilization on the performance of DL-based RFFP.

The rest of the paper is organized as follows. Sec. II presents the related works. Sec. III studies the impact of carrier frequency offset and inaccuracy on the behavior of IQ signals. Sec. IV presents the proposed IQ data representation approach, EPS. Sec. V presents the proposed EPS-based device fingerprinting framework. Sec. VI describes the testbed and the WiFi datasets used for the proposed framework evaluation, presented in Sec. VII. After that, Sec. VIII discusses the computational efficiency and inference latency of the EPS representation, and Sec. IX highlights the impact of hardware warm-up and stabilization on RF fingerprinting accuracy. Finally, the paper is concluded in Sec. X.

II. RELATED WORK

Prior works that aimed to address the domain generalizability challenges of DL-based RFFP can be broadly categorized into two approaches: data-centric and architecture-centric. For the data-centric approaches, various data augmentation techniques have been explored to expose the DL models to a wider range of wireless channel instances, thereby enhancing their robustness against channel variations. For instance, Soltani et al. [28] and Al-Shuwaili et al. [29] integrated data augmentation engines into the training process. These engines incorporated WLAN TGn and ITU-R channel models, respectively, along with an additive white noise model. While these techniques demonstrated a marginal improvement in testing accuracy, they do not offer a practical and scalable solution suitable for commercial deployment. Additionally, due to the intrinsic nature of wireless channels, it is challenging and impractical to devise a universal channel-augmenting model capable of significantly improving performance across a wide range of wireless channels. Other data-centric approaches mitigate the impact of the channel through channel equalization [12], [13], [14] or impairment compensation [6].

For the DL architecture-centric approaches, researchers have formulated the RFFP generalizability challenge as a domain adaptation problem [22], [30] and capitalized on the advancement in transfer learning to address it. The underlying assumption in these frameworks is that the source and target domains exhibit slightly different distributions. One notable domain adaptation framework, ADL-ID [19], integrates disentangled representation learning with adversarial learning to tackle the challenge of short-term temporal generalization in RFFP. ADL-ID involves segregating the feature vector into two distinct components: device-specific (fingerprints) and domain-specific features. During the inference stage, only the device-specific features, which encompass characteristics that remain invariant across the source and target domains, are utilized. Similarly, SignCRF [20] leverages a cycle-consistent generative adversarial network to construct an environment translator that effectively decouples hardware impairments from channel and environmental conditions. Adversarial domain adaptation techniques have also been employed in this context. For instance, RadioNet [18] adopts an adversarial learning scheme, utilizing a domain discriminator and a reversal gradient layer to minimize the domain-related information in the feature vector. The trained feature extractor is subsequently connected to a KNN classifier fine-tuned using the target data. Following a similar calibration approach, the Tweak approach, proposed by

Gaskin et al. [22], combines metric learning with lightweight calibration using the target data to enhance generalization across hardware, channel, and configuration dimensions.

While these domain adaptation methods provide valuable insights in enhancing RFFP generalizability, they exhibited limitations in providing satisfactory performance for medium-scale testbeds. Furthermore, they encounter challenges when faced with significant distribution gaps between the enrollment and deployment datasets or when confronted with unseen environmental conditions that differ from the target domain employed during adaptation.

III. UNDERSTANDING THE IMPACT OF CARRIER FREQUENCY INACCURACY ON IQ SIGNAL BEHAVIOR

Local oscillators are the transceiver hardware components that are responsible for producing oscillating signals needed for signal up-conversion at the sender side and for signal down-conversion at the receiver side. The inaccuracy and instability of the oscillating signal's frequency, typically caused by external factors like temperature, vibration, and electromagnetic interference, impact the overall system performance behavior. As such, in an effort to improve their robustness to these external factors, various types of different crystal oscillators have been developed over the years, including temperature-controlled crystal oscillators (TCXOs), which feature temperature compensation, and oven-controlled crystal oscillators (OCXOs), which place the crystal in a temperature-controlled environment to keep their temperatures at a constant level, thereby improving the accuracy of their oscillating frequencies [31].

A. THE CARRIER FREQUENCY OFFSET (CFO) IMPAIRMENT

This study concentrates on hardware impairments caused by oscillator frequency inaccuracies, specifically the Carrier Frequency Offset (CFO), which often results in signal distortion. This focus is driven by a comprehensive review of existing literature, which consistently identifies oscillator-caused impairments as notably distinguishable hardware imperfections. For instance, [25] ranks ''frequency error'' as the most effective metric for establishing radiometric identity. Similarly, [26] emphasized the distinctiveness of CFO imperfections, considering them the most identifiable. Moreover, findings from [27] underscore that CFO, linked to crystal imperfection, varies significantly across devices while remaining consistent over time, making it an ideal fingerprinting feature due to its immunity to software spoofing. This is echoed in the context of WiFi fingerprinting, where studies, including [32], [33], have identified CFO and IQ imbalance as the most separable features for WiFi fingerprinting. Moreover, [34], [35] further support this, indicating that oscillator imperfections can be accurately identified, even under challenging conditions such as low SNRs or short observation sequences.

Building on this understanding, it becomes clear that while CFO information is inherently present in the time-domain IQ representation of RF signals, the challenge arises when DL-based RFFP frameworks attempt to generalize this data across different domains. Recent findings, [7], [11], reveal significant shortcomings in these frameworks' ability to adapt when tested under varied conditions. For example, models trained on data from one day often exhibit a marked decrease in testing accuracy when evaluated on data collected from a different day [22]. These observations underscore the necessity for developing new RF signal representations that not only enhance feature selection processes but also bolster the domain adaptation capabilities of DL models. Our work responds to this need by proposing robust RF signal representations that effectively capture and utilize CFO impairments as distinctive features, thereby improving the adaptability of RFFP systems to domain variations. To effectively address this, it is crucial to thoroughly study and comprehend how carrier frequency offset and inaccuracies impact the behavior of received IQ signals.

B. THE IMPACT OF CARRIER FREQUENCY INACCURACY

To acquire a good understanding of the impact of the oscillating frequency inaccuracy on the IQ signal behavior, we leveraged our experimental testbed of 15 Pycom/IoT devices to observe, analyze, and compare the IQ signals collected from multiple different (but identical in hardware) off-the-shelf devices. This is done by having each of the 15 Pycom devices transmit multiple IEEE 802.11b WiFi packets after being powered on for more than 12 minutes to ensure hardware stabilization. We want to emphasize here the importance of waiting until the end of the warm-up/stabilization period of the devices' hardware before performing data collection to ensure robust and consistent measurements; we provide further explanation and illustration on this in Sec. IX. The transmitted signals are then captured and sampled at 45MSps using a USRP B210 receiver. More description and details on the testbed are provided later in Sec. VI.

We show in Fig. 1 the time behavior of both the I (in-phase) and Q (quadrature) signal components collected from Devices A, B, and C. Two key observations we draw from this experiment. First, observe the 'sinusoidal' behavior that the envelopes[1] of both the I and Q signals exhibit. More importantly, note that the number of 'humps' of the envelope changes across the devices: 12.5 for Device A, 19.5 for Device B, and 5 for Device C. Second, observe that the I and Q envelopes of a given device vary in the opposite direction—i.e., shifted by 180 degrees, though still exhibiting the same number of 'humps'. It is also worth mentioning that although shown for only three devices here, these reported sinusoidal behaviors of the IQ signals' envelopes are observed across all of the 15 tested Pycom devices, with each device exhibiting a slightly different number of humps.

[1]The envelope of an oscillating signal is the smooth boundary function that outlines the extremes of the signal (e.g., see [36], Appendix C).

The questions that arise now are: (i) what is the cause of the observed sinusoidal behavior of the IQ signal envelope? and (ii) why does the number of 'humps' differ from one device to another? We will show that the main cause behind such behavior is the CFO (carrier frequency offset) between the Pycom device's oscillating frequency and that of the USRP receiver that exists due to the instability and inaccuracy of the device's local oscillator. Specifically, we will next show that the number of humps in the sinusoidal envelope depends on the CFO value. This explains that the reason why different devices exhibit different numbers of humps is because each device presents a different CFO, which varies across devices due to the device's oscillator hardware imperfections incurred during manufacturing. Later in Sec. IX, we will also demonstrate that the CFO value (and hence the number of humps) of a given device keeps changing over time until the device hardware is stabilized; i.e., the CFO value keeps varying over time until the end of the hardware warm-up period. This is due to the instability that the carrier frequency exhibits when the oscillator hardware of the device is still warming up.

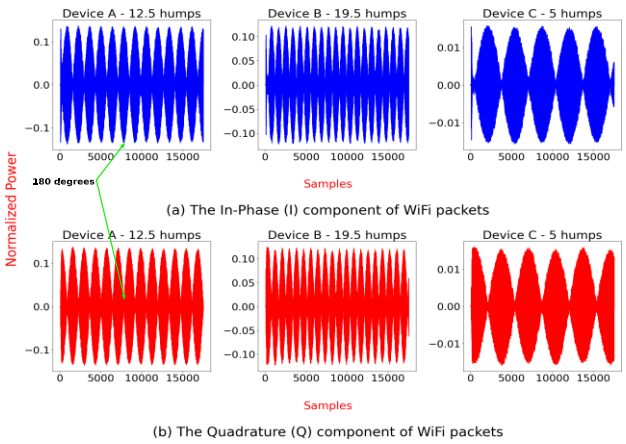


FIGURE 1. The time-domain IQ signal behavior across three different Pycom devices. The number of 'humps' are: 12.5 for Device A; 19.5 for Device B; and 5 for Device C.

C. THE CAUSE OF THE OBSERVED ENVELOPE BEHAVIOR

We now demonstrate and affirm that the CFO impairment is what is behind the sinusoidal behavior of the IQ signal's Envelope illustrated in Sec. III-B above.

1) ANALYTIC AFFIRMATION

In this subsection, we analytically examine the sinusoidal behavior of the IQ signal's envelope, focusing specifically on how the CFO impairment is responsible for this phenomenon. This analytical exploration will reveal the underlying mechanisms by which CFO influences the signal's characteristics. To illustrate, consider a baseband signal $a(t)$ exp $(j\varphi(t))$ modulated by an oscillating signal with a carrier frequency f_c and a CFO, $f.f$. The passband transmitted signal can be expressed as:

$$s(t) = a(t)\cos(2\pi(f_c + f.f)t + \varphi(t))$$

At the receiver side, the In-phase (I) component, $r_I(t)$, of the received signal can be recovered (demodulated) as:

$$r_I(t) = s(t)\cos(2\pi f_c t)$$

Due to the CFO (i.e., the carrier frequency at the receiver is slightly different from the carrier frequency used by the sender), the phase of the wave shifts over time, causing the amplitude of the received signal to vary sinusoidally over time. This can be seen by rewriting $r_I(t)$ as

$$r_I(t) = \frac{a(t)}{2}[\cos(2\pi(2f_c + f.f)t + \varphi(t)) + \cos(2\pi f.ft + \varphi(t))]$$

While the left term in the equation above represents the frequency sum of the two frequencies, typically filtered out by a bandpass filter, the right term represents the frequency difference which is, in this case, the carrier frequency mismatch or CFO, $f.f$, between the local oscillators of the transmitter and the receiver. When CFO $= f.f = 0$, the second term becomes 1 and the signal maintains a scaled version of its original amplitude, $\frac{a(t)}{2}$ (e.g. corresponding to Fig. 2a in the simulation case). However, a frequency mismatch will cause the received signal to be modulated at the frequency $f.f$, resulting in an amplitude of $\frac{a(t)}{2}\cos(2\pi f.ft + \varphi(t))$. This analysis clearly illustrates how the CFO, when nonzero, modulates the amplitude of the received signal, directly impacting its characteristics.

2) SIMULATED AFFIRMATION

To further substantiate our analysis, we now extend our affirmation into a practical scenario by simulating the impact of CFO within an actual WiFi system. Using MATLAB's WLAN toolbox, we crafted a model to simulate IEEE 802.11b WiFi DSSS waveforms with various CFO impairments, including 0 Hz, 50 Hz, 100 Hz, and 200 Hz. The CFO-impaired transmitted signal is first passed through an AWGN channel, and then down-converted and sampled by the receiver to generate IQ data samples. For each case, we collected 10 WiFi frames, with each frame having a size of 1000 bits. Then, we extracted the real (I) components of the signals and plotted them separately for CFO $= 0$ in Fig. 2a, CFO $= 50$Hz in Fig. 2b, CFO $= 100$Hz in Fig. 2c, and CFO $= 200$Hz in Fig. 2d. The simulated results clearly show the dependency between the CFO values and the number of observed 'humps' in the I signal's envelope, and that the CFO is what causes the observed Envelope shape. The same trends were observed for the Q signal components as well, but we did not include them here to limit redundancy.

We want to mention that we also experimented with varying other hardware impairments, including IQ imbalance, Phase Noise, and DC offset, but have not noticed any 'sinusoidal' behavior of the envelopes. This confirms that other transceiver hardware impairments, though do manifest

themselves in other types of distortions, do not yield the envelope behavior we observed with the CFO impairment.

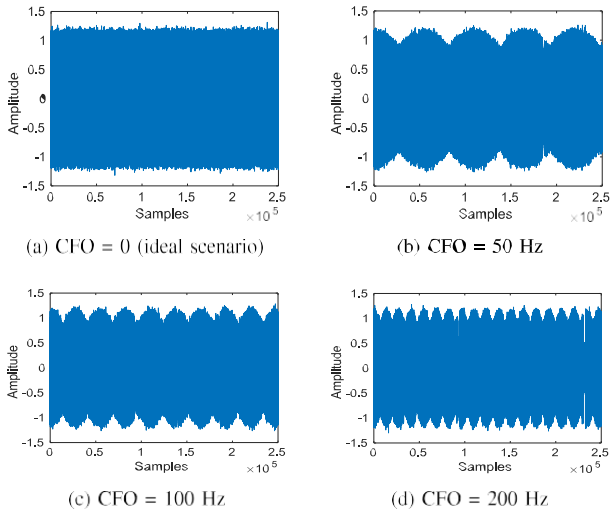


FIGURE 2. The I component of WiFi packets generated using a simulated WiFi system with CFO impairment blocks.

In conclusion, we confirm that these observed device-dependent, sinusoidal envelope behaviors of the IQ signals collected from the different off-the-shelf PyCom devices are indeed attributed to the CFO impairments. These CFOs exist because the carrier frequency generated by the local oscillator at the USRP receiver is (slightly) different from that generated by the local oscillator at the Pycom device. These demonstrations also confirm that devices with (even slightly) different oscillating frequencies yield different numbers of 'humps' in the received signals' envelope. And this work leverages such a difference in the number of humps across different devices to propose efficient IQ representations that are shown to significantly improve the accuracy and robustness of DL-based RF fingerprinting to domain changes.

IV. NOVEL IQ DATA REPRESENTATION FOR DISTINGUISHABLE NEURAL NETWORK FEATURES

In this section, we begin by presenting a novel IQ signal representation/feature extracted from the oscillator's envelope shape (observed and explained in the previous section) that substantially improves the robustness of device fingerprinting to domain changes and variations. We then evaluate the effectiveness of the proposed feature design vis-a-vis of its fingerprinting ability to (i) *distinguish between devices* and (ii) *adapt to domain changes* by maintaining high accuracy performance under varying domains.

A. CAPTURING THE OSCILLATOR'S ENVELOPE BEHAVIOR

In order to extract the CFO value resulting from the mismatch between the sender's and receiver's oscillating frequencies, which is embedded in the signal's envelope shape as observed and explained in Sec. III, we first create the analytic signal, $z(t)$, of the time-domain representation of the receiver packet, $r(t)$. The analytic signal $z(t)$ is a complex-valued signal, comprising the original signal, $r(t)$, as its real part and the Hilbert transform (HT) of $r(t)$ as its imaginary part, and can hence be written as $z(t) = r(t)+jHT(r(t))$ where $HT(r(t)) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{r(t-\tau)}{\tau}d\tau$. Formally, the envelope $e(t)$ of $r(t)$ is the magnitude of its analytic signal; i.e.,

$$e(t) \triangleq |z(t)| = \sqrt{r(t)^2 + HT(r(t))^2} \quad (1)$$

B. THE PROPOSED IQ DATA REPRESENTATION: THE DOUBLE-SIDED ENVELOPE'S POWER SPECTRUM (EPS)

After extracting the envelope of the IQ signal using the analytic signal presentation as described in Sec. IV-A, we remove the DC offset of the envelope and compute its normalized double-sided power spectrum, which results in one main sideband and its harmonics on each side. We propose this double-sided envelope's power spectrum, termed `EPS` for short, as the new IQ data representation to use as input to the deep learning models. As we show later, this improves the models' accuracy significantly and makes them highly robust to the domain adaptation challenges we described in Sec. I. Fig. 3 shows the three stages involved in extracting `EPS` from a WiFi frame sent by one of the Pycom devices. The figure at the top displays the time-domain I component values of the WiFi frame, which exhibits sinusoidal variations in amplitude due to the impairments of the crystal oscillator. The figure in the middle depicts the extracted envelope of the frame using the analytic signal representation. The figure at the bottom shows the double-sided envelope's power spectrum, `EPS`.

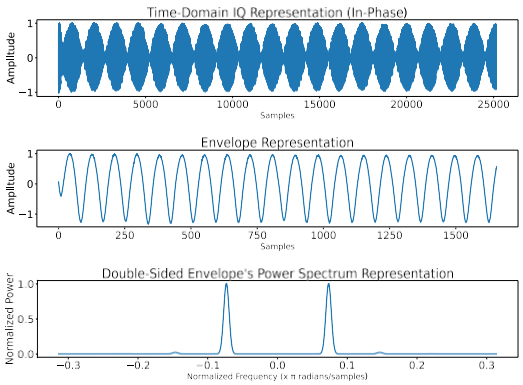


FIGURE 3. Extracting the `EPS` feature from a WiFi frame.

C. EPS DISTINGUISHABILITY ACROSS DIFFERENT DEVICES

In the context of RF fingerprinting, a signal representation that exhibits distinctive device-specific characteristics is critical. The proposed `EPS` feature possesses this property, as it captures the local oscillator's behavior, which is affected by the oscillator's unique hardware impairments. To validate this hypothesis, we conducted an experimental evaluation using our testbed consisting again of 15 Pycom devices, running the IEEE802.11b protocol and a USRP B210 receiver

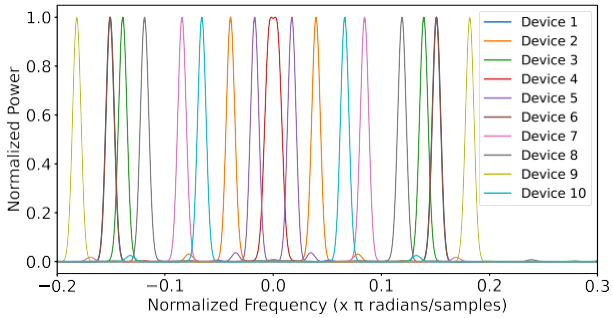


FIGURE 4. The EPS representation of 10 devices.

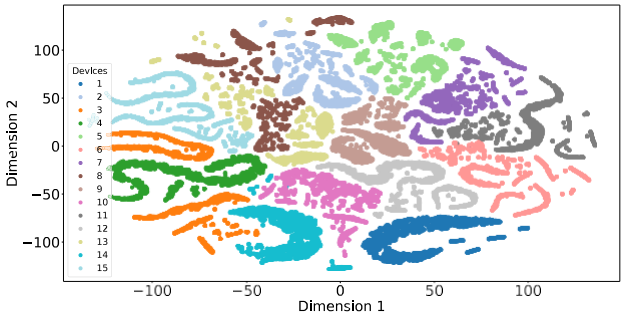


FIGURE 5. The t-SNE visualization of the complete testbed.

(more testbed details are provided later in Sec. VI). Our results depicted in Fig. 4 reveal that the EPS representation is indeed unique for each device, as evidenced by the discernible differences, across the 10 studied devices, in the shape and location of the main sideband and its harmonics. Moreover, this representation or feature contains thousands of samples, making it a high-dimensional data type that can benefit from non-linear dimensionality reduction techniques such as t-distributed stochastic neighbor embedding (t-SNE) [37]. By visualizing the representation of 3000 packets from each device using t-SNE, as shown in Fig. 5, we demonstrate that the EPS representation extracted from more than 40, 000 packets is well-separated and suitable as an effective input for device classification. This will be further validated through experimental results that are presented later in Sec. VII.

D. EPS ROBUSTNESS TO DOMAIN CHANGES

Exhibiting unique device-specific features is necessary but insufficient for a representation to serve as a good fingerprint for device classification. If a representation of a device varies randomly each time the signal is captured, it cannot offer a reliable fingerprint and therefore cannot be utilized as an input for the device classification system. Hence, after we showcased the distinguishability property of the EPS representation using our testbed, we now test its domain-adaptation ability by examining its robustness to maintaining device separability when there is a change across domains. And we do so by considering three domains: time, channel, and location.

1) TIME-AGNOSTIC FINGERPRINTING

The establishment of stability over time in any proposed representation is of paramount importance for the practical implementation of RF fingerprinting systems, particularly in dynamic real-world environments where temporal changes are expected. Factors such as the movement of people, and varying usage of surrounding WiFi access points, contribute to a variable RF landscape over time. Such environmental changes, compounded by shifts in office occupancy and ambient temperature throughout the day, underscore the importance of assessing the robustness of RF fingerprinting systems across temporal variations. Additionally, it's noteworthy that the power cycling of radios - turning devices off and then on - has been identified as a factor affecting RF fingerprints [38]. These conditions demonstrate that even within the same physical setting, time itself is a dimension where different variables can impact system performance, highlighting the need for time-agnostic properties in RF fingerprinting technologies. To ascertain the robustness and temporal reliability of our proposed Double-Sided Envelope Power Spectrum (EPS) representation, a comprehensive set of experiments was performed on our testbed, comprising 15 devices transmitting 802.11b packets. To capture the stability of the devices over time, we employed a wired connection as described in Sec. VI-C.1 and initiated the data-capturing process 20 minutes after the devices were activated, so as to ensure hardware settling and stabilization. Precisely, packet captures were obtained at specific intervals, namely 1 minute, 3 minutes, 8 minutes, 1 hour, 2 hours, 1 day, and 2 days, spanning three consecutive days. For each individual device, the EPS representation was extracted from the recorded packets at the aforementioned time points over the three-day period. Fig. 6 depicts the plotted results for four representative devices, clearly showcasing that for each device, all the EPS representations extracted at the different time intervals overlap. This demonstrates that the proposed EPS representation is time agnostic and remains unchanged over time. This uniformity in the EPS representation was consistently observed across all 15 devices (though shown only for 4 devices in the paper), thus providing compelling evidence of the stability and reliability of our proposed EPS representation over time. Furthermore, these findings underscore the efficacy of EPS in mitigating the sensitivity to temporal variations encountered in DL-based RF fingerprinting techniques.

2) CHANNEL-AGNOSTIC FINGERPRINTING

To investigate the impact of the wireless channel on the stability and consistency of EPS, we conducted the following experiment in an indoor environment. The devices were positioned at a fixed distance of 1 meter from the receiver, and packet captures were performed over a duration of three consecutive days, as detailed later in Sec. VI-C.2. The objective of this investigation was to compare the EPS representations of packets corresponding to each individual

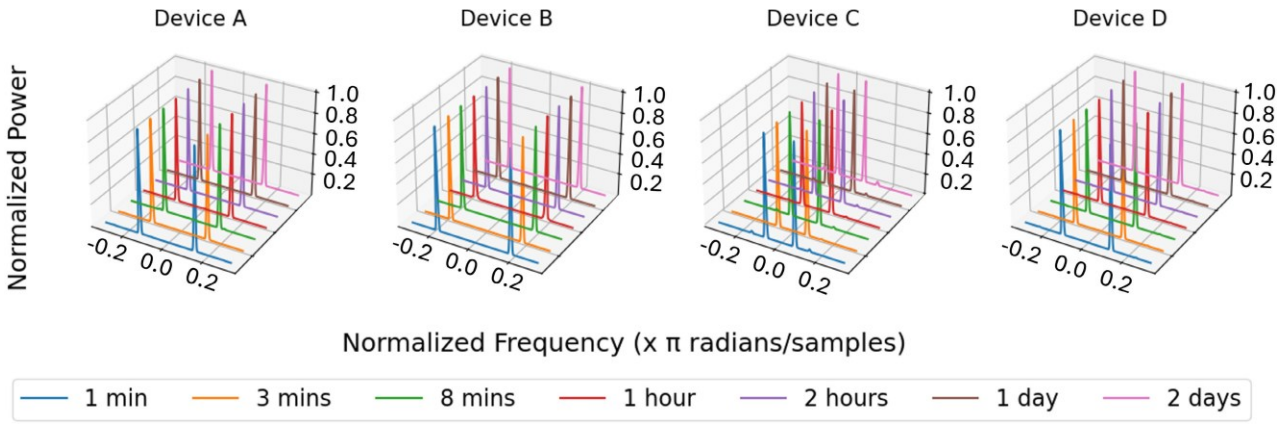


FIGURE 6. **Time-domain scenario showing the EPS representations of 4 devices extracted at 7 different time marks.**

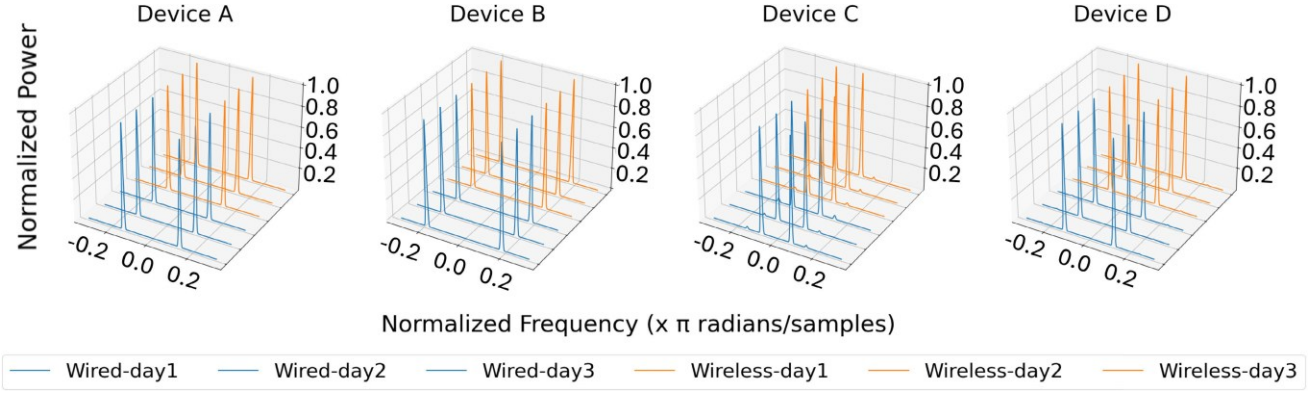


FIGURE 7. **Channel-domain scenario showing the EPS representations of 4 different devices extracted under both wired and wireless setups, each over three days.**

device across both the wired and wireless channels over time, thereby discerning the influence of channel variations over time. Fig. 7 presents the graphical representations of the EPS features obtained from four distinct devices under both wired and wireless channel conditions. Notably, the figures effectively demonstrate that the EPS representation of each device remains unaltered regardless of the underlying channel characteristics. This observation unequivocally establishes the inherent stability and reliability of our proposed EPS representation, even in the presence of wireless channel effects, over the course of three consecutive days. Importantly, this behavior was consistently observed across all 15 devices, thereby fortifying the empirical evidence supporting the robustness and efficacy of our proposed EPS. Consequently, these findings firmly substantiate the ability of the EPS representation to effectively mitigate the potential sensitivity to channel variations encountered in DL-based RF fingerprinting methods.

3) LOCATION-AGNOSTIC FINGERPRINTING

Changing the distance between the transmitting devices and the receiver after training can also lead to a drastic drop in performance. To evaluate the robustness of the EPS representation to such distance changes, we captured data at three different locations with the devices being placed 1m-away (Location A), 2m-away (Location B), and 3m-away (Location C) from the USRP receiver; this setup is shown in Fig. 13 and discussed in more detail in Sec. VI-C.3. Fig. 8 shows the EPS representation of WiFi packets from four devices over the three different locations. To extend the reliability test with regard to location and distance, we also considered another realistic scenario in which the devices were randomly deployed within a radius of 3m from the receiver as shown in Fig. 14 (refer to Sec. VI-C.4 for more details). The corresponding EPS representations are depicted as Location D (random) in Fig. 8. The plots in Fig. 8 manifest the stability of the EPS feature representation over the four studied location scenarios as the signal representations of the four locations completely overlap. To evaluate the EPS representation's consistency across locations under different environmental conditions, we conducted outdoor tests near the Kelley Engineering building on the Oregon State University campus. These experiments were carried out at distances of one, two, and three meters from the receiver

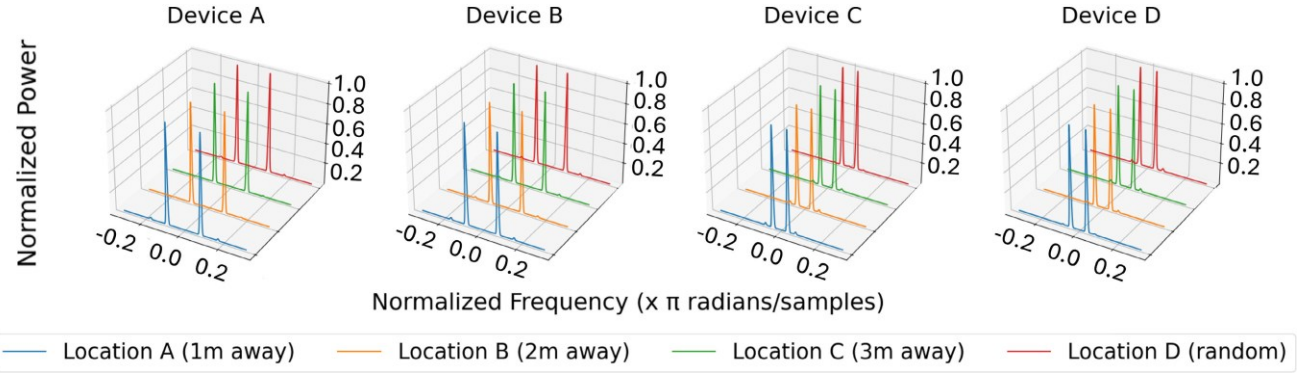


FIGURE 8. **Location-domain scenario showing the EPS representations of 4 devices extracted at 4 different locations. Devices in Locations A/B/C are all placed at fixed distances; devices in Location D are placed at random distances.**

and close to a main pedestrian path–on a busy day to simulate a dynamic environment. The resulting consistency of the EPS representations of four different devices, depicted in Fig. 9, confirms the robustness of our representation even in varied outdoor environments. Our findings confirm the stability of the EPS representation in indoor and outdoor scenarios in which the location, distance, and time of the training and testing sets are different, again making the proposed EPS a more reliable and robust input for DL-based RF fingerprinting methods.

4) RECEIVER-AGNOSTIC FINGERPRINTING

To investigate the robustness of the EPS representation against variations in receiver hardware, we designed and executed a controlled experiment using two USRP B210 receivers, each known for its reliable RF performance and flexibility in various signal processing tasks. The test involved data collection from four FiPy devices, randomly selected from our testbed. Data acquisition was carried out sequentially—first with one receiver and subsequently with the other—under indoor conditions. All transmitters were connected through a wired setup and powered via a USB hub, ensuring consistency in the power supply. Each receiver operated independently on its own internal clock, thus varying the only variable, the receiver unit. The EPS representations of all captured packets were generated and subsequently analyzed to determine the consistency of the EPS representation across the two different receivers. The analysis, visually summarized in Fig. 10, illustrates a remarkable stability of the EPS representation for the four tested devices across both receivers, as shown by the perfectly overlapping EPS plots. This visual evidence underscores the EPS representation's robustness to changes in receiver hardware within the confines of our experimental setup. Although the results are promising, they should be considered preliminary in establishing the EPS's universal receiver-agnostic properties. Further comprehensive testing with receivers from different manufacturers is essential to fully validate and confirm the receiver variability robustness of the EPS representation.

V. THE EPS-BASED FINGERPRINTING FRAMEWORK FOR DOMAIN-AGNOSTIC DEVICE CLASSIFICATION

Maintaining good performances of RF fingerprinting when faced with domain shifts due to changes in channel conditions and/or device location/distance has proven to be very challenging, hindering the widespread adoption of RF fingerprinting technology in real-world applications. Our proposed fingerprinting framework, based on the proposed EPS feature representation, has demonstrated stable behavior across various settings and increased resiliency to domain changes, thereby overcoming the aforementioned challenges.

In this section, we evaluate the effectiveness of the proposed EPS feature representation vis-a-vis of its ability to adapt to domain (channel, time, and location) changes when the EPS data is used as an input to a typical Convolutional Neural Network (CNN) device classifier [2], [14], [39], [40].

A. AN OVERVIEW OF THE PROPOSED EPS-CNN FRAMEWORK

At its high level, the proposed EPS-CNN framework, shown in Fig. 11, consists of an EPS generator, which takes the complex-valued IQ representation of a received frame, $r(t)$, as an input and then processes the I (In-phase) and Q (Quadrature) components separately. For each frame, the EPS generator first extracts the envelope of the signal, $e(t)$, and then generates the EPS representation of the two components: EPS(I) and EPS(Q). Refer to Sec. V-B for details about the EPS representation generation. The two EPS representations are then concatenated into a tensor (e.g., of size 2×4096) and passed to the CNN network that extracts the suitable features using the six convolution blocks followed by three fully connected layers and a Softmax layer. The CNN is also responsible for learning a classifier from the extracted features to accurately predict the corresponding device of the incoming frame (Device i).

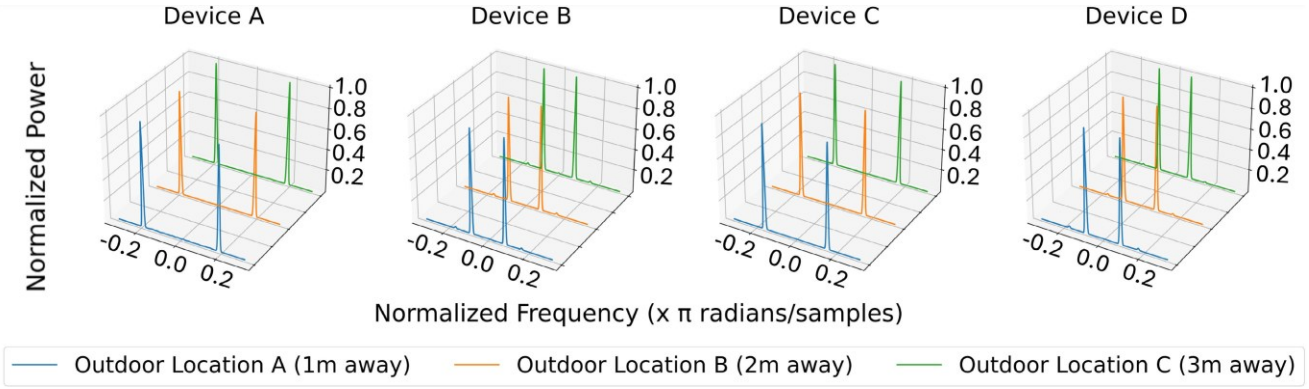


FIGURE 9. Outdoor Location-domain scenario showing the EPS representations of 4 devices extracted at 3 different locations.

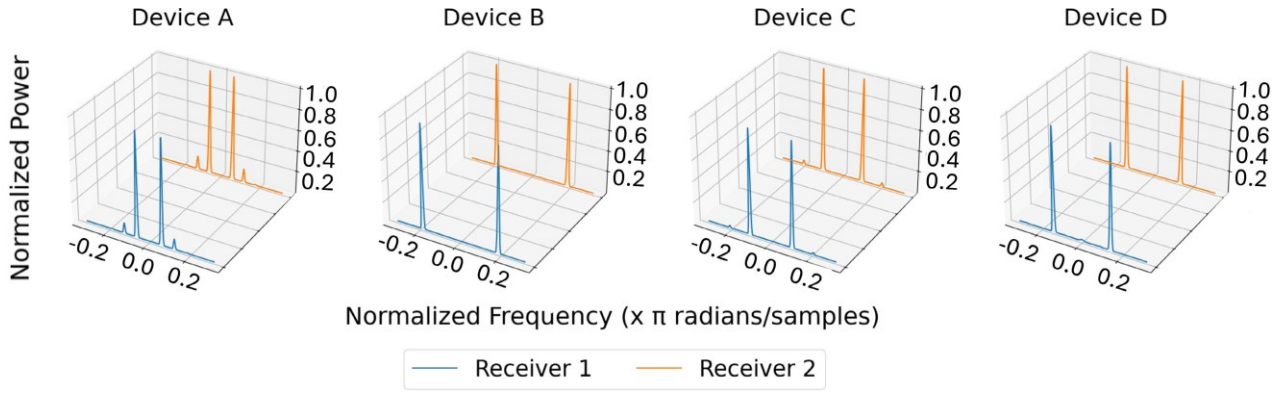


FIGURE 10. Receiver-domain scenario showing the EPS representations of 4 devices extracted using two different receivers.

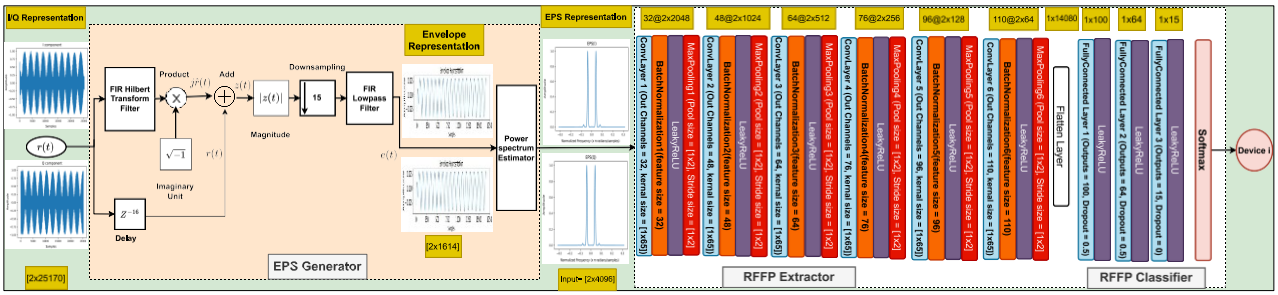


FIGURE 11. EPS-CNN framework overview.

B. EXTRACTION OF EPS REPRESENTATION

The implementation we used to extract the signal's envelope, $e(t)$, from the received signal, $r(t)$, is shown in Fig. 11; refer back to Sec. IV-A for the derived $e(t)$ expression. We construct the analytic signal by first passing the IQ values of the received frame through an FIR Hilbert transform filter based on the Parks-McClellan algorithm [41] implemented in MATLAB Signal Processing Toolbox. The output of the filter is then multiplied by $\sqrt{-1}$ (the imaginary unit) and added to the time-delayed original signal. It is important to introduce a delay in the input signal because the FIR filter implementation of the Hilbert transform introduces a delay equivalent to half the length of the filter. The signal's envelope, $e(t)$, is calculated by taking the absolute value of the analytic signal, which is characterized by a lower frequency compared to the original signal. Consequently, we first downsample the signal's envelope by a factor of 15, an empirically determined rate that optimally balances data rate reduction with the

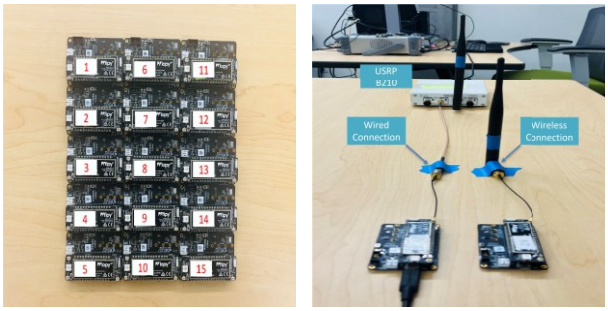
preservation of essential fingerprint features. Following this, the downsampled signal is passed through a lowpass filter to effectively eliminate ringing and smooth the envelope. Once the envelope is extracted, we center the envelope's amplitude around zero before generating the corresponding normalized double-sided envelope's power spectrum, i.e., EPS, representation using the power spectrum estimator. The decision to utilize a double-sided representation over a single-sided one allows for capturing the full spectral information inherent in these signals. This comprehensive approach enhances the discriminative capability of our classifier and consistently improves classification accuracy by 2-3% across all testing scenarios, demonstrating the value of including a more extensive feature set derived from the complete spectrum of complex-valued signals.

C. CNN ARCHITECTURE

We train our CNN, using the PyTorch library, on an NVIDIA Cuda-enabled NVIDIA GeForce RTX 2080 Ti GPU system for 30 epochs. The input to the model is the EPS representation with a dimension of 2×4096, representing the EPS representation of the In-phase and quadrature components. The first layer applies a 2D convolution with a filter size of 1×65, followed by batch normalization, and LeakyReLU activation. The kernel size was empirically determined to optimize performance, ensuring symmetric padding and preserving the spatial dimensions of the input volume post-convolution. We systematically tested a range of kernel sizes spanning [2, 5, 7, 9, 17, 33, 65, 129, 257, 513, 1025] samples, and found that a kernel size of 1×65 consistently yielded the best performance. Then, a max-pooling layer with kernel size 1×2 and stride 1×2 is applied to reduce the dimensions of the feature maps. The same pattern is repeated five more times, with increasing numbers of output channels (32, 48, 64, 76, 96, and 110) and decreasing feature map sizes, until a final feature map size of 2×64 is obtained. Then, two fully connected layers with output sizes equal to 100 and 64 are applied, each followed by a dropout layer and LeakyReLU activation. Finally, the output of the second fully-connected layer is passed to another fully-connected layer that maps it to the 15 output classes, which is passed to a softmax layer to produce the predicted label. We experimentally chose a learning rate of 3e-4 which decays by decaying factor after every 3000 steps. Finally, we use the stochastic gradient descent optimizer with momentum and a weight decay parameter that adds an L2 regularization on the weights to avoid overfitting. The code used in this paper is publicly available for researchers to use and can be downloaded at https://github.com/NetSTAR-Lab.

VI. TESTBED, DATASETS AND EXPERIMENTAL SCENARIOS

In this section, we describe the testbed setup, the experimental scenarios, and the collected WiFi datasets used for evaluating the effectiveness and robustness of the proposed techniques. The WiFi datasets, their description and their download information can be found at http://research.engr.oregonstate.edu/hamdaoui/datasets.



(a) 15 Pycom transmitting devices. (b) Wired-WiFi vs. wireless-WiFi.

FIGURE 12. IoT Testbed consisting of 15 Pycom transmitting devices and a USRP B210 receiving device.

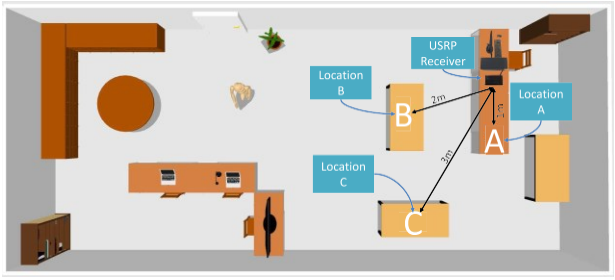


FIGURE 13. Different-Location setup.

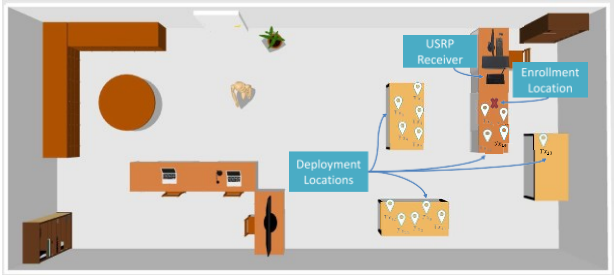


FIGURE 14. Random-Location setup.

A. TESTBED SETUP

The testbed used for our evaluation, depicted in Fig. 12, consists of 15 Pycom devices, including 10 FiPy boards and 5 LoPy boards. Both the FiPy and LoPy boards are equipped with the ESP32 chip, which supports WiFi and Bluetooth. Additionally, both boards feature the Semtech SX1276 chip, enabling LoRa and Sigfox communications. The FiPy boards also include a Sequans Monarch LTE module for cellular connectivity, further enhancing their capability to support a wide range of network protocols including LTE. The data acquisition was performed using an Ettus USRP B210 receiver, which was synchronized with an external OCXO for improved sampling accuracy

and stability. All devices were powered via USB from an HP laptop and configured to transmit IEEE802.11b WiFi packets using the high-rate direct-sequence spread-spectrum (HR/DSSS) physical layer in the 2GHz spectrum. The transmitting devices transmitted at a rate of 1Mbps with a carrier frequency of 2.412GHz and a bandwidth of 20MHz, while connected to the same 1/2 Wave Whip antenna.

B. DATASET COLLECTION

We initiated the data-capturing process 12 minutes after devices were activated, so as to ensure hardware stabilization; see Sec. IX for further details on the impact of device stabilization and warm-up time. Each device was configured to operate over WiFi Channel 1 with a center frequency of 2412MHz and a bandwidth of 20MHz. The transmitters were programmed to transmit identical IEEE 802.11b frames with a duration of 559us back to back, separated by a small gap. We captured the first two minutes of transmissions using the USRP B210 at a sample rate of 45MSps, generating more than 5000 identical packets from each device at each capturing event. The captured signals were then digitally down-converted to the baseband and stored as IQ samples on our computer. To avoid any data dependency on the identity of the WiFi transmitter, all transmitters were configured to broadcast the same packets, which include the same spoofed MAC address and a payload of zero bytes. Finally, WiFi packets were extracted from the raw IQ samples and stored in HDF5 formats to maintain the integrity and accuracy of the captured signals.

C. EXPERIMENTAL SCENARIOS

Our WiFi dataset contains more than 8TB of WiFi transmissions from 15 Pycom devices captured in four different setups/scenarios: Wired Setup, Wireless Setup, Different-Locations Wireless, and Random-Location Wireless. Notably, all data collection scenarios were conducted in an open/shared lab environment, simulating typical office conditions where environmental factors are uncontrolled. This approach ensures that our findings are applicable to real-world situations, reflecting the complexities of everyday wireless communication environments.

1) WIRED SETUP

To rule out the impact of the wireless channel, we connected our transmitters directly to the USRP receiver via SMA cabling, and collected data over three days, generating more than 5000 WiFi frames per device every day.

2) INDOOR WIRELESS SETUP

Instead of wiring the transmitters to the USRP receiver as done in the Wired Setup, we placed them at a fixed location, 1m away from the USRP receiver which uses a VERT900 antenna to capture the signal. We repeated this experiment over three days to assess the generalizability of the proposed technique over time. This setup generated more than 5000 WiFi frames per device every day.

3) DIFFERENT-LOCATIONS WIRELESS SETUP

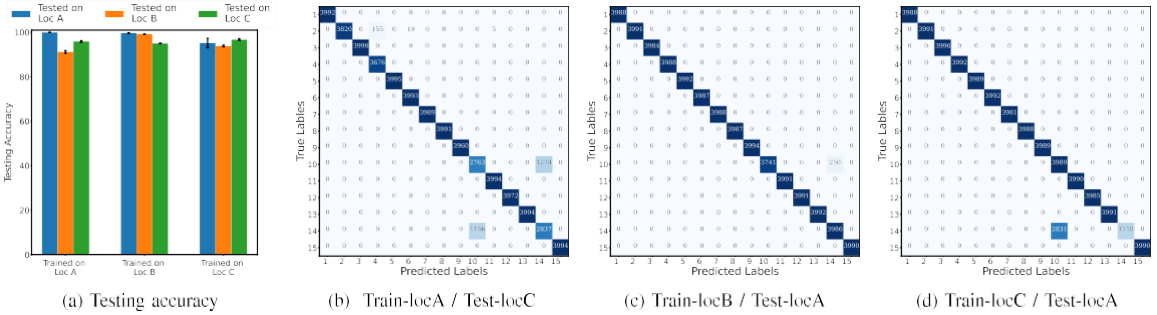
The location from where the transmitter sends its data impacts the characteristics of the received signal, as signals transmitted from different distances/locations usually experience different channel conditions, which is considered in this work as another varying domain. For each transmitter, we then collected data at three different locations, A, B, and C, which are 1m, 2m, and 3m away from the USRP receiver, respectively, as shown in Fig. 13. This was carried out in one day and generated more than 5000 WiFi frames per device at each location.

4) RANDOM-LOCATION WIRELESS SETUP

From a practical viewpoint, when the fingerprinting framework is used for device authentication, the messages that are sent by the devices and are to be used for authentication are likely to come from different random locations, and these random locations are also likely to be different from the locations used in the enrollment (training) stage. Therefore, we considered collecting datasets for two random-location scenarios on two different days, each consisting of an enrolment phase (data used for training) and a deployment phase (data used for testing). In both enrolment phases, all the transmitters transmitted from the same location, 1m away from the receiver, and in both deployment phases, the transmitters were located randomly within a radius of 3m away from the receiver as shown in the floor plan in Fig. 14. The enrollment datasets were collected in the morning while the random deployment datasets were collected on the night of that same day, generating more than 5000 WiFi frames per device for each dataset.

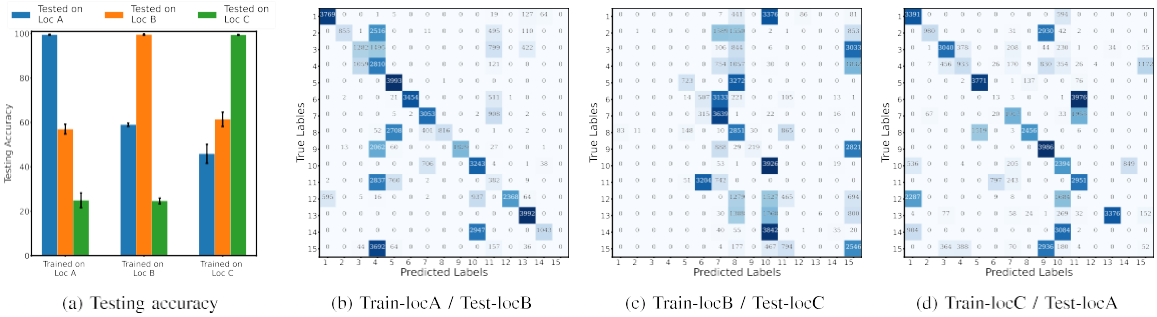
VII. DEVICE IDENTIFICATION RESULTS

To assess the effectiveness of our proposed `EPS` feature representation in improving the performance of DL-Based RFFP methods across domains, we considered two performance metrics: same-domain accuracy and cross-domain accuracy. Same-domain accuracy measures the ability of the DL models to identify devices accurately when the testing data/packets (unseen in the training phase) are drawn from the same training domain. On the other hand, cross-domain accuracy evaluates the models' ability to generalize across different domains, such as different locations, channels or days. For temporal domain adaptation, we initially train the model using data from one day and subsequently test it on data from the other two days to verify consistent performance over time. Similarly, in spatial domain adaptation, training occurs with data from a specific location, then testing on data from various locations with different distances and orientations to the receiver to rigorously evaluate the model's adaptability to channel characteristic changes. We evaluated the performance of a standard CNN framework when fed with our proposed `EPS` representation as an input (referred to as `EPS-CNN`) and compared it with the same CNN framework but when fed with a typical IQ representation as an input



(a) Testing accuracy (b) Train-locA / Test-locC (c) Train-locB / Test-locA (d) Train-locC / Test-locA

FIGURE 15. EPS-CNN**'s performance: (a) Testing accuracy; (b)-(d) Confusion matrices for different Train-Location/Test-Location combinations. LocA, LocB, and LocC correspond to when the transmitters are 1, 2, and 3m away from the receiver.**



(a) Testing accuracy (b) Train-locA / Test-locB (c) Train-locB / Test-locC (d) Train-locC / Test-locA

FIGURE 16. IQ-CNN's performance: (a) Testing accuracy; (b)-(d) Confusion matrices for different Train-Location/Test-Location combinations. LocA, LocB, and LocC correspond to when the transmitters are 1, 2, and 3m away from the receiver.

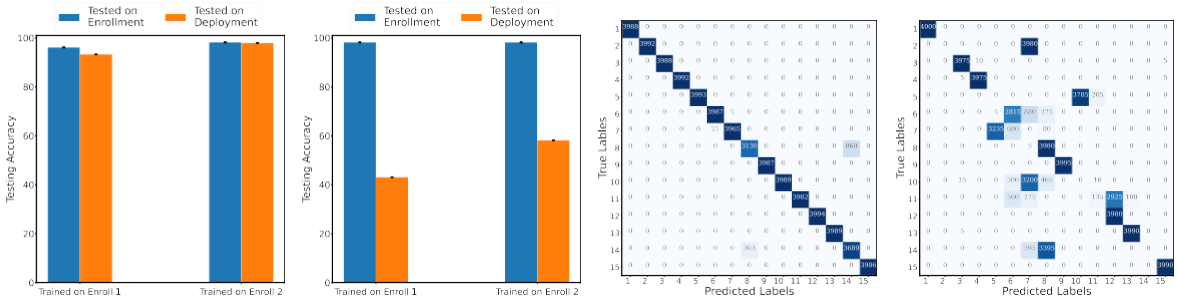
TABLE 1. Testing accuracy of EPS-CNN**and IQ-CNN on the fixed placement scenario.**

Model	Train on LocA			Train on LocB			Train on LocC		
	Test on LocA	Test on LocB	Test on LocC	Test on LocA	Test on LocB	Test on LocC	Test on LocA	Test on LocB	Test on LocC
EPS-CNN	100%	91.3%	95.5%	99.7%	99.6%	95.04%	95.3%	93.9%	96.7%
IQ-CNN	99.53%	57.01%	24.96%	59%	99.57%	24.7%	45.9%	61.4%	99.46%

(referred to as IQ-CNN). By using the traditional CNN as a benchmark, we provide a clear and direct comparison showing the improvement brought by the EPS representation. We employed the 5-fold cross-validation method, where each device's data is divided into five non-overlapping, equally-sized partitions. In each fold of cross-validation, we used four partitions for training (3200 packets) and the remaining partition for testing (800 packets). We then averaged the results obtained from each fold to produce a final estimate of the model's performance. For the EPS input, we represented each packet by a 2×4096 tensor, which encapsulates the EPS representation of both the I and Q components. In contrast, for IQ input, we represented each packet using a 2×8192 tensor, comprising the first 8192 samples of both the I and Q components as this window size provides the best performance for the IQ representation input.

A. ADAPTATION TO LOCATION CHANGES: FIXED PLACEMENT

First, we begin by evaluating the proposed EPS-CNN framework using the WiFi dataset captured in the three different locations, as described in Sec. VI-C.3, to assess its robustness to changes in device locations. The results are shown in Fig. 15 for EPS-CNN and Fig. 16 for IQ-CNN, where again LocA, LocB, and LocC correspond to when the transmitters are placed 1m, 2m and 3m away from the USRP receiver. The results shown in Fig. 15 demonstrate that our EPS-CNN framework is highly effective in device fingerprinting, achieving exceptional same-domain testing accuracies at all locations. Specifically, the average testing accuracies at Locations A, B, and C are 100%, 99.6%, and 96.7%, respectively, as shown in Fig. 15a. Even more impressive is the performance of our EPS-CNN framework in cross-domain evaluation, where the model is trained on one location and tested on datasets captured in different locations. The results show that EPS-CNN maintains high performances, with average testing accuracies of 91.3% and 95.5% when trained on Loc A and tested on Loc B and Loc C, respectively. Similarly, EPS-CNN achieves a testing accuracy of 99.7% and 95.04% when trained on Loc B and tested on Loc A and Loc C, and an accuracy of 95.3% and 93.9% when trained on Loc C and tested on Loc A and Loc B. To the best



(a) EPS-CNN (b) IQ-CNN (c) EPS-CNN (d) IQ-CNN

FIGURE 17. EPS-CNN/**IQ-CNN's performance under random deployments: (a)-(b): testing accuracy; (c)-(d): confusion matrices obtained under random-location setup 2; that is, when training is done under enrolment setup 2 and testing is done under the corresponding deployment setup.**

of our knowledge, this is the highest performance achieved by a DL-based device fingerprinting method when the learning models are tested and trained on different domains. The cross-location confusion matrices are shown in Figs. 15b, 15c, 15c, further highlighting the effectiveness of EPS-CNN .

In contrast, the conventional IQ-CNN framework, which uses raw IQ representation instead of the EPS representation and whose results are shown in Fig. 16, struggles to maintain its performance in cross-domain evaluation. Although IQ-CNN performs well in the same-domain evaluation, it performs poorly when tested on a dataset captured in a different location. Specifically, the average testing accuracy when IQ-CNN is trained on location A and tested on locations B and C is 57.01% and 24.96%. This significant drop in performance is also seen in the other locations, as the testing accuracy on locations A and C when the model is trained on location B is 59% and 24.7%, whereas the average testing accuracy when the model is trained on location C and tested on locations A and B is 45.9% and 61.4%, respectively. Note the significant accuracy difference between that achieved by the proposed EPS-CNN and that achieved by the conventional IQ-CNN (i.e., a drop from 90_{+}% to as low as 25%). The confusion matrices in Figs. 16b, 16c, 16d show the struggle of the trained model to correctly classify the devices when the corresponding packets are captured in a different location. The results clearly demonstrate the superiority of the EPS-based deep learning framework in device fingerprinting, particularly in cross-domain evaluations. Table. 1 summarizes the testing accuracies of all setups of the fixed placement scenario.

B. ADAPTATION TO LOCATION CHANGES: RANDOM PLACEMENT

We also considered evaluating the effectiveness of the proposed EPS-CNN framework under two random-location setups, as described in Sec. VI-C.4. For each setup, during training (referred to as enrolment), all devices transmit from a fixed location, 1m away from the receiver; and during testing (referred to as deployment), the devices transmit from random locations all within 3m from the receiver; refer to Fig. 14

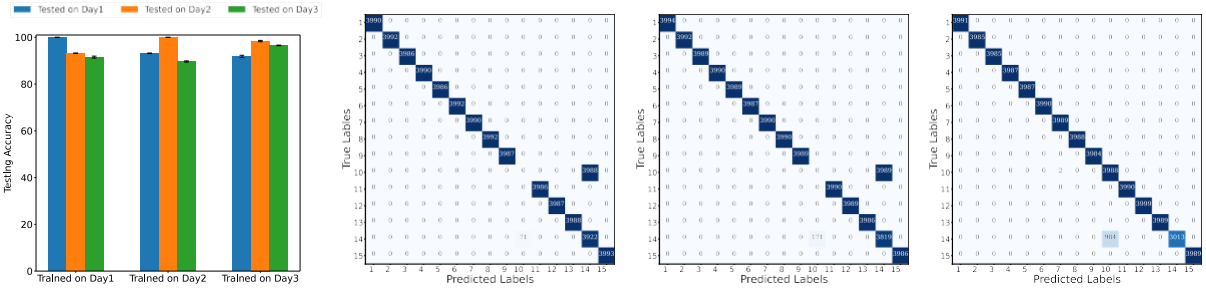
TABLE 2. Testing accuracy of EPS-CNN and IQ-CNN on the random placement scenario.

Model	Train on Enrol1		Train on Enrol2	
	Test on Enrol1	Test on Deploy1	Test on Enrol2	Test on Deploy2
EPS-CNN	96.7%	93.1%	98.3%	98.1%
ine IQ-CNN	99.53%	40.2%	99.51%	58.2%

for visualization of this random deployment scenario. The EPS-CNN framework exhibits strong performance in both same-domain and cross-domain evaluations under random-location setups. Fig. 17a shows that EPS-CNN achieves high average same-domain testing accuracies of 96.7% and 98.3% respectively under random-location setups 1 and 2. Furthermore, the figure demonstrates the robustness of the framework in cross-domain with testing accuracies of 93.1% and 98.1% under random-location setups 1 and 2, respectively. Notably, Fig. 17c, showing the confusion matrix under random-location setup 2, showcases the framework's exceptional accuracy for most devices. In contrast, IQ-CNN experiences significant performance degradation on both random-location setups, with cross-domain testing accuracies of only 40.2% and 58.2% on setups 1 and 2, respectively. Through the confusion matrix, Fig. 17d provides a clear depiction of the IQ-CNN framework's struggle in recognizing devices when randomly deployed around the receiver. Table. 2 summarizes the testing accuracies of all setups of the random placement scenario.

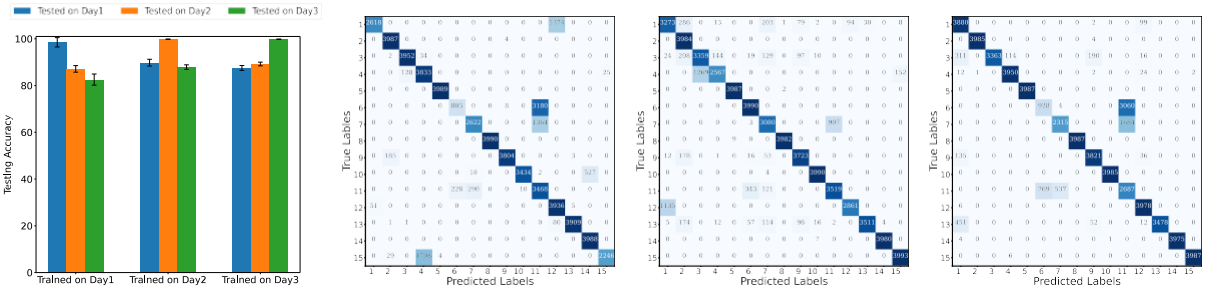
C. ADAPTATION TO TIME CHANGES: DIFFERENT DAYS

The effectiveness of the proposed EPS-CNN framework was also evaluated on a cross-days scenario using the indoor wireless WiFi dataset as described in Sec. VI-C.2. Fig. 18a presents the average testing accuracy of the proposed EPS-CNN framework when trained on one day and tested on one of the other three days. The same-domain testing accuracies (both training data and testing data are collected the same day) were found to be 100%, 100%, and 96.5% for day 1, day 2, and day 3, respectively. These research findings demonstrate the distinguishability of the EPS feature representation, as the learning model was able



(a) Testing accuracy (b) Train-day1 / Test-day2 (c) Train-day2 / Test-day1 (d) Train-day3 / Test-day2

FIGURE 18. EPS-CNN's performance across three different days: (a) Testing accuracy; (b)-(d) confusion matrices for different Train-day/Test-day combinations.



(a) Testing accuracy (b) Train-day1 / Test-day3 (c) Train-day2 / Test-day1 (d) Train-day3 / Test-day1

FIGURE 19. IQ-CNN's performance across three different days: (a) Testing accuracy; (b)-(d) confusion matrices for different Train-day/Test-day combinations.

TABLE 3. Testing accuracy of EPS-CNN and IQ-CNN on the cross-day scenario.

Model	Train on Day1			Train on Day2			Train on Day3		
	Test on Day1	Test on Day2	Test on Day3	Test on Day1	Test on Day2	Test on Day3	Test on Day1	Test on Day2	Test on Day3
EPS-CNN	100%	93.2%	91.5%	93.2%	100%	89.7%	91.8%	98.4%	96.5 %
IQ-CNN	98.5%	87.5%	89.2%	89.8%	99.83%	87.9%	87.5%	89.2%	99.82%

to extract unique features from each device, achieving high performance on the same-domain performance metric.

More interestingly, for the cross-day evaluation, Fig. 18a shows that the proposed EPS-CNN framework maintains remarkable performance accuracy when tested on a different day. Specifically, when the learning model is trained on day 1 data, the average cross-domain testing accuracy is 93.2% when the model is tested on day 2 data and 91.5% when tested on day 3 data. This achievable performance is consistent across other days, with an accuracy of 93.2% or 89.7% when training on day 2 data but testing on day 1 or day 3 data, respectively. Similarly, when the model is trained on day 3 and tested on day 1 data or day 2 data, the testing accuracies are 91.8% or 98.4%, respectively. The aggregate confusion matrices of the cross-day testing over the three days, shown in Figs. 18b, 18c, 18d, further indicate that most of the devices achieved perfect classification accuracies across the three tested days, with only one or two devices causing a small drop in performance.

In comparison, the performance of the IQ-CNN framework in cross-domain testing, shown in Fig. 19, is inferior to that of the proposed EPS-CNN framework. Our results from Fig. 19 indicate that when the deep learning model is trained on day 1 data, the average cross-domain testing accuracy is 87.5% when tested on day 2 data and 89.2% when tested on day 3 data. When the model is trained on day 2 data, this average cross-domain testing accuracy is 89.8% when tested on day 1 data or 87.9% when tested on day 3 data. The testing accuracy when the model is tested on day 1 or day 2 data but trained on day 3 data is 87.5% or 89.2%, respectively. The aggregate confusion matrices of the cross-domain testing over the three tested days are also shown in Figs. 19c, 19d, 19e. Although both EPS-CNN and IQ-CNN frameworks achieved close-to-perfect performance in the same-domain testing accuracies, the proposed EPS-CNN outperforms the conventional IQ-CNN framework in the cross-domain performance metric on the three tested days. Our results show that the deep learning models when fed with our proposed EPS features are highly effective in addressing and mitigating the cross-day sensitivity of deep learning-based RF fingerprinting.

The relatively good performance of conventional IQ-CNN fingerprinting in the cross-day metric suggests that the indoor wireless channel in this scenario did not change

significantly over the three tested days of the experiment. This leads us to postulate that time itself is not a factor or domain that significantly affects the learning model's performance. Instead, time is simply a space in which various events can occur, leading to changes in the environment that can affect performance. Hence, we hypothesize that in a stable environment, cross-time evaluations may not be sufficient, and further tests are necessary to assess the model's performance under varying channel conditions, due to changing locations and distances. Our proposed EPS-CNN framework indeed maintains high performances even under varying locations as shown in our results presented earlier in Sec. VII-A and Sec. VII-B. Table. 3 summarizes the testing accuracies of all setups of the cross-day scenario.

D. ADAPTATION TO MODEL CHANGES: ResNet MODEL

Our previous evaluations focused on the performance of EPS-CNN, which combines the EPS representation with a CNN model, across various domain changes. To further assess the benefits of our EPS representation, we explored its integration with ResNet, specifically a tailored variant of ResNet-18 [42]. This adapted architecture, named EPS-ResNet, leverages the EPS representation and is benchmarked against the traditional IQ representation, known as IQ-ResNet. This comparison aims to highlight EPS-ResNet's enhanced adaptability and superior performance in RF fingerprinting tasks, particularly across different location scenarios.

1) ResNet ARCHITECTURE

In our implementation, this architecture begins with an initial convolutional layer that uses 64 filters with a kernel size of 64 and a stride of 2, followed by batch normalization and a ReLU activation function. This layer is followed by a max pooling operation with a pool size of (1, 2) and a stride of 2. The core of the ResNet architecture consists of four stages, each comprising a series of convolutional blocks. First Stage: Two ResNet blocks with 64 filters each, where the first block adjusts for the initial feature map size. Second Stage: Two ResNet blocks with 128 filters, with the first block applying a stride of 2 to reduce the dimensionality. Third Stage: Similarly, this stage has two blocks with 256 filters, again with the first block applying a stride of 2. Fourth Stage: The final stage includes two blocks with 512 filters, with the first block reducing dimensions with a stride of 2. Each ResNet block in these stages consists of two convolutional layers with batch normalization. A shortcut connection links the input to the output of these layers, which helps in mitigating the vanishing gradient problem by allowing gradients to flow through the network. The network concludes with a global average pooling layer that helps to reduce the dimensions and is connected to a dense layer with softmax activation tailored to the number of devices in the dataset.

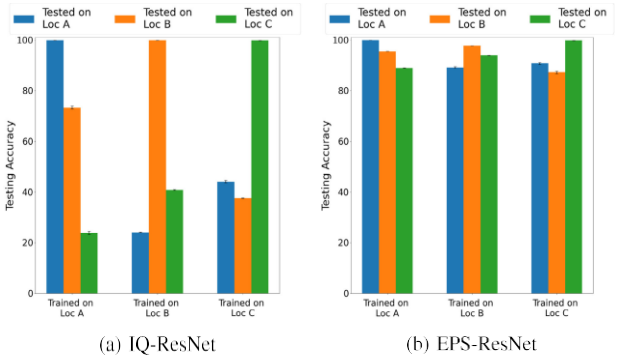


FIGURE 20. Testing accuracy of IQ-ResNet Vs. EPS-ResNet in Cross-Location scenario.

2) ResNet'S PERFORMANCE EVALUATION

Fig. 20 illustrates the testing accuracies for both same-domain and cross-domain scenarios of EPS-ResNet compared to IQ-ResNet. The data reveals that while IQ-ResNet achieves nearly perfect same-domain accuracies across three locations, its performance significantly drops in cross-domain tests; specifically, accuracies drop to 24% and 73% when trained on location A and tested on locations B and C respectively, as shown in Fig. 20a. Conversely, EPS-ResNet demonstrates a substantial improvement in bridging the gap between same-domain and cross-domain performances, achieving an average accuracy improvement of 50% across all tested locations. For instance, it maintains high testing accuracies of 89% and 91% when trained on location A and tested on locations B and C respectively. This pattern holds true across all tested locations, achieving an average cross-domain accuracy of 91%, as shown in Fig. 20b. These findings underscore the efficacy of the EPS representation in enhancing the adaptability of ResNet to different domains and its superior performance over the conventional IQ representation. Table. 4 provides a comparative analysis of IQ-ResNet, EPS-ResNet, and EPS-CNN under cross-location testing, highlighting that despite EPS-ResNet's strong performance, the EPS-CNN framework still excels in domain adaptation.

VIII. COMPUTATIONAL EFFICIENCY AND INFERENCE LATENCY OF EPS REPRESENTATION EXTRACTION

In this section, we address the computational efficiency and inference latency associated with our EPS representation extraction process, as these factors are crucial in RF fingerprinting systems, particularly in comparison to cryptographic authentication methods.

A. COMPUTATIONAL EFFICIENCY AND POWER CONSUMPTION

The EPS extraction is engineered for computational efficiency, requiring approximately 2,518,678 arithmetic operations. This count, while significant, contributes minimally to the overall computational load due to the reduction in data

TABLE 4. Comparison of testing accuracy of IQ-ResNet, EPS-ResNet, and EPS-CNNon the cross-location scenario.

Model	Train on LocA			Train on LocB			Train on LocC		
	Test on LocA	Test on LocB	Test on LocC	Test on LocA	Test on LocB	Test on LocC	Test on LocA	Test on LocB	Test on LocC
IQ-ResNet	100%	23.87%	73.30%	24.09%	99.87%	40.82%	44.12%	37.61%	99.74 %
EPS-ResNet	100%	89.13%	90.74%	95.51%	97.76%	87.27%	88.98%	93.89%	99.74%
EPS-CNN	100%	91.3%	95.5%	99.7%	99.6%	95.04%	95.3%	93.9%	96.7%

size by a factor of two. This reduction not only streamlines processing but also decreases the computational demands on the neural network. Specifically, the EPS-CNN model requires only 3,155,189,619 Floating Point Operations (FLOPs), compared to 6,313,136,499 FLOPs needed by the IQ-CNN model. Given that the extraction is performed on the receiver side, where power constraints are generally less severe than on IoT transmitters, this setup is well-suited to environments with ample computational resources.

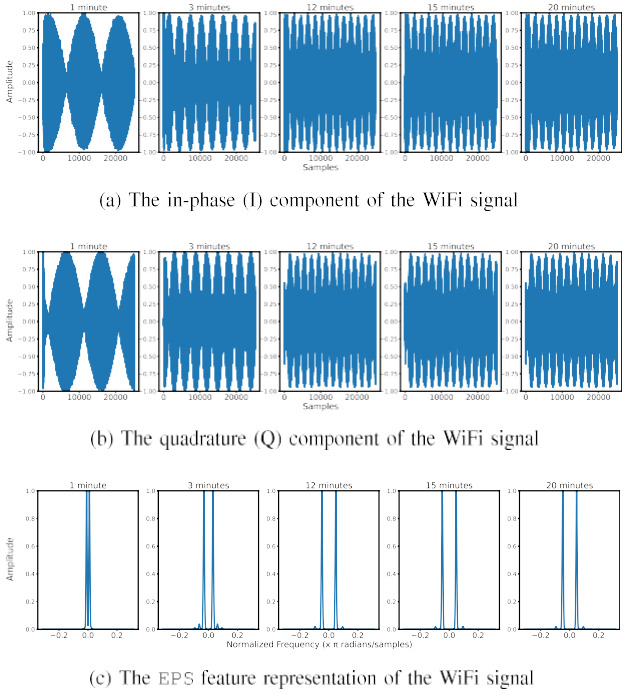
B. INFERENCE LATENCY

Despite the EPS extraction introducing an initial delay, the reduction in input size significantly counteracts this increase in latency. In a theoretical scenario where computing capabilities reach 1 Tera Operations Per Second (TOPS), the EPS extraction adds a mere 2.52 microseconds of delay. This minimal increase is balanced by the quicker processing enabled by smaller input sizes, resulting in inference latencies of approximately 3.16 milliseconds for EPS-CNN and 6.31 milliseconds for IQ-CNN.

These findings underscore that the EPS extraction method, by reducing data complexity and optimizing the use of available computational resources, aligns with the critical requirements of power efficiency and minimal latency in effective RF fingerprinting systems. This approach ensures that our method remains competitive, even against traditional models, offering a balanced solution to the evolving challenges in RF authentication.

IX. ON THE IMPACT OF TRANSCEIVER HARDWARE WARM-UP AND STABILIZATION PERIOD

In alignment with best practices for data collection in RF fingerprinting, we have ensured that both the training and testing datasets were gathered during the stable phase of the transmitters, specifically after the warm-up period had concluded. Our next step is to explore how the hardware stabilization during the warm-up duration impacts the performance of device fingerprinting. Specifically, we aim to assess how DL-based RF fingerprinting (RFFP) frameworks, which have been trained on data collected during the stable phase, perform when encountering inference data transmitted from devices still within their warm-up period. Despite the rich amount of literature available on this RF fingerprinting topic, the impact of hardware stabilization and warm-up time has not been carefully considered [43]. And for completeness, it is our goal here to shed some light on what could go wrong had such stabilization aspects not been carefully accounted for. More specifically, our objective in this section is to investigate and study the impact of the transceiver hardware warm-up on (i) the observed Envelope behavior of the time-domain IQ signals, (ii) the EPS features, and (iii) the overall EPS-based device fingerprinting performance.



(a) The in-phase (I) component of the WiFi signal

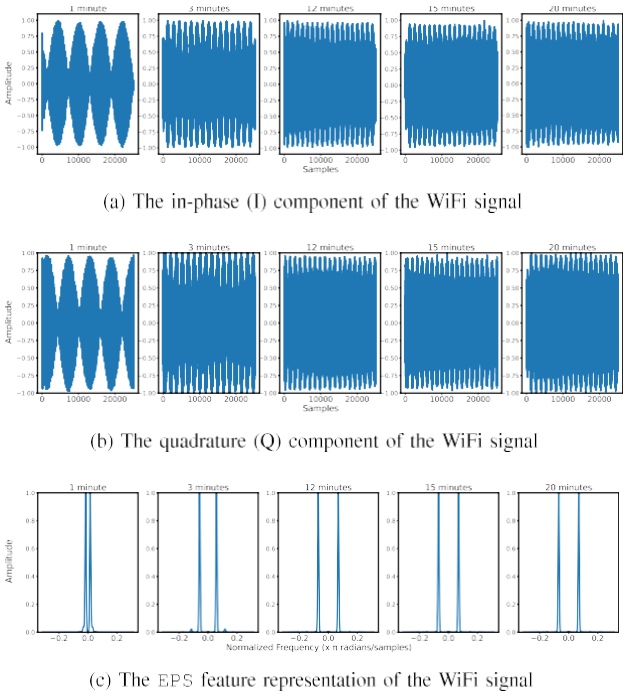
(b) The quadrature (Q) component of the WiFi signal

(c) The EPS feature representation of the WiFi signal

FIGURE 21. Representations of the RF signal captured from Device A and observed at different times during the warm-up period of the device.

A. BEHAVIOR OF RECEIVED RF SIGNALS DURING THE WARM-UP AND STABILIZATION PERIOD OF THE TRANSMITTING DEVICE

We begin by studying the behavior of the I, Q and EPS representations during the hardware warm-up time. For this, we closely monitored the IQ signal behavior of two off-the-shelf (FiPy) devices from our testbed during the initial 20 minutes following device activation. This involved capturing 802.11b WiFi packets transmitted by the devices using the USRP B210 at a sampling rate of 45MSps. The USRP receiver was clocked using an external 10 MHz OCXO (oven-controlled, high-performing crystal oscillator) reference signal to ensure measurement accuracy and stability.



(a) The in-phase (I) component of the WiFi signal

(b) The quadrature (Q) component of the WiFi signal

(c) The EPS feature representation of the WiFi signal

FIGURE 22. Representations of the RF signal captured from Device B and observed at different times during the warm-up period of the device.

We show in Fig. 21 the I, Q and EPS representations of the WiFi signal captured on Device A at different times during the device warm-up period; i.e., the figure on the far-left corresponds to the signal captured one minute from when the device was powered on, the figure on the far-right corresponds to the signal captured 20 minutes from when the device was powered on, and so on. Four important observations can be drawn from this figure. First, the results confirm the presence of a CFO impairment, which is manifested in the observed sinusoidal shape of IQ signal's Envelope, as was illustrated and explained in Section III. Second, observe that the I shape (Fig. 21a), Q shape (Fig. 21b) and the EPS shape (Fig. 21c) all change over time as the device hardware warms up, with the frequency of humps in the Envelopes of the I and Q signals increasing over time until hardware stabilization. This increase indicates a varying CFO value during hardware warm-up time that is resulting from the instability of the local crystal oscillators; this finding is well aligned with the Envelope behavior observed and reported in Sec. III-B and explained in Sec. III-C. Third, the I, Q and EPS shapes all seem to converge and stabilize after some time (i.e., around 12 minutes in the figure). Note that these shapes observed at minutes 12, 15 and 20 resemble one another, meaning that the shapes converge at around 12 minutes from device activation, which indicate that the local oscillator has reached a stable operating point by minute 12. Fourth, observe that the I and Q components vary on the opposite direction; i.e., shifted by 180 degrees, at any stage during the warm-up period; this also is well aligned with what was observed and reported in Sec. III.

To assess the consistency of these trends across different devices, we also monitored these IQ data representations based on signals captured from several other devices also at different times during the device warm-up period. Our experimental results using other devices (we only show one more device, Device B, here in Fig. 22) confirm that the reported trends are also observed across all other devices, although each device exhibits slightly different initial and stable shapes.

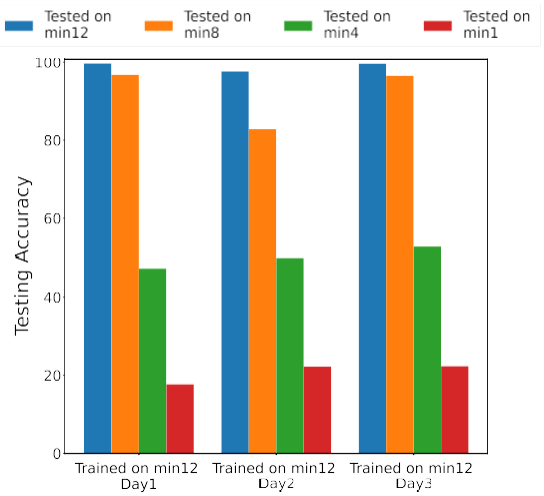


FIGURE 23. Classification accuracy when training data is collected at minute 12 (after device stabilization) but testing data is collected at 1, 4, 8 and 12 minutes from device activation on the same day.

B. SENSITIVITY OF RF DATA-DRIVEN DEVICE FINGERPRINTING TO TRANSCEIVER HARDWARE WARM-UP AND STABILIZATION

We now show the effect of hardware warm-up that is manifested in the observed IQ signal Envelope behavior on the device fingerprinting accuracy. For this, we run experiments whereby the proposed EPS-CNN framework is trained with data collected after stabilization (i.e., after 12 minutes from device activation) and tested with same-day data but collected at various different times during warm-up period (i.e., during the initial 12 minutes from the activation of devices). To mask the impact of the wireless channel, we considered in this experiment the wired setting described in Sec. VI-C.1.

Fig. 23 shows the testing accuracy of EPS-CNN when testing data is collected before the device hardware is stabilized; that is, at 1, 4, and 8 minutes from when the devices are powered on. For ease of comparison, the figure also includes the case when the test data is collected after device stabilization; i.e., using data collected at minute 12. The figure clearly shows the dependency of the achieved accuracy on the time at which testing data is collected during

warm-up period. Note that the closer to the stabilization time the testing data collection takes place, the higher the testing accuracy. The figure also confirms that this observed trend is consistent across different days.

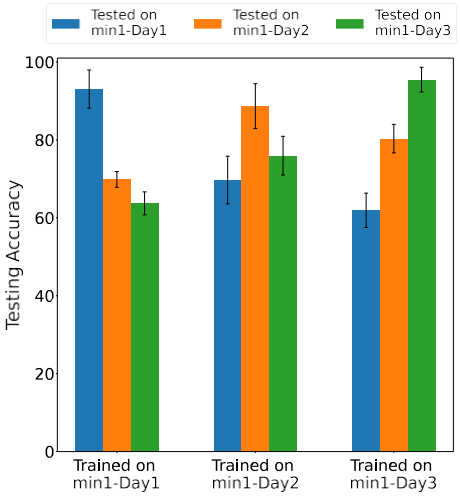


FIGURE 24. Classification accuracy when training data is collected at minute 1 and testing data is collected also at minute 1 from device activation but on a different day.

One key observation that is worthy of note is that when both training and testing are done on data collected at about the same time from device activation even during warm-up time, the testing accuracy that the learning models achieve is not as low as what they achieved when testing and training data were done at different times during stabilization. For instance, we show in Fig. 24 the accuracy when the model is trained on data captured within the first minute after activation of one day and tested on data collected in the same time (i.e., within the first minute from activation) but of another day. The figure demonstrates that when the model is trained on minute 1 captures of day 1, an average testing accuracy of 70% (resp. 62%) is achieved when the model is tested on minute 1 captures of day 2 (resp. day 3), which is considerably higher than the testing accuracy when the model is tested on stabilized data (after 12 minutes) of the same day. These research findings indicate a systematic drift in the characteristics of the received IQ signals during the stabilization and warm-up period, with consistent behavior observed across different days, and highlight the challenges faced by the deep learning models in recognizing devices during the hardware warm-up period. These results thus underscore the importance of considering the stabilization aspects of the oscillator hardware (as well as the other transceiver hardware components) when developing hardware-impairment-driven RF fingerprinting techniques for robust device identification and classification. It is worth noting that although the performance of DL-based RF fingerprinting frameworks like the EPS-CNN is impacted by data collected during the warm-up period, these frameworks can be effectively integrated with mechanisms designed to address these initial fluctuations. This integration is vital since the warm-up period often represents a minor fraction of a device's operational lifespan, during which EPS-based frameworks otherwise perform exceptionally well. This approach ensures that the fingerprinting system remains effective and reliable throughout the majority of the device's usage.

X. CONCLUSION

In conclusion, this paper addresses the limitations of conventional RF signal representations in deep learning-based RF fingerprinting methods. We propose the Double-Sided Envelope Power Spectrum (EPS) as a novel RF signal representation that effectively captures device hardware impairments while eliminating irrelevant information. Experimental results demonstrate the superior performance of the EPS representation in terms of accuracy, robustness, and generalizability across various domains. By leveraging EPS, DL-based RFFP methods can achieve unprecedented testing accuracy in same-domain evaluations and maintain high performance in cross-domain scenarios. The proposed representation offers a transformative solution for enhancing the security and privacy of wireless networks by advancing the accuracy and reliability of device identification through RF fingerprinting. Finally, we release large WiFi 802.11b datasets containing captures for different scenarios to allow others to further investigate these fingerprinting issues.

ACKNOWLEDGMENT

The authors would like to thank Nora Basha for her help with the Matlab coding.

REFERENCES

[1] B. Hamdaoui, A. Elmaghbub, and S. Mejri, ''Deep neural network feature designs for RF data-driven wireless device classification,'' *IEEE Netw.*, vol. 35, no. 3, pp. 191–197, May 2021.

[2] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, ''ORACLE: Optimized radio classification through convolutional neural networks,'' in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 370–378.

[3] T. Jian et al., ''Deep learning for RF fingerprinting: A massive experimental study,'' *IEEE Internet Things Mag.*, vol. 3, no. 1, pp. 50–57, Mar. 2020.

[4] L. Ding, S. Wang, F. Wang, and W. Zhang, ''Specific emitter identification via convolutional neural networks,'' *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2591–2594, Dec. 2018.

[5] N. Basha, B. Hamdaoui, and K. Sivanesan, ''Leveraging MIMO transmit diversity for channel-agnostic device identification,'' in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 2254–2259.

[6] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, ''Radio frequency fingerprint identification for LoRa using deep learning,'' *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2604–2616, Aug. 2021.

[7] A. Elmaghbub and B. Hamdaoui, ''LoRa device fingerprinting in the wild: Disclosing RF data-driven fingerprint sensitivity to deployment variability,'' *IEEE Access*, vol. 9, pp. 142893–142909, 2021.

[8] B. Hamdaoui and A. Elmaghbub, ''Uncovering the portability limitation of deep learning-based wireless device fingerprints,'' 2022, *arXiv:2211.07687*.

[9] H. Fu, L. Peng, M. Liu, and A. Hu, ''Deep learning-based RF fingerprint identification with channel effects mitigation,'' *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1668–1681, 2023.

[10] B. Hamdaoui and A. Elmaghbub, ''Deep-learning-based device fingerprinting for increased LoRa-IoT security: Sensitivity to network deployment changes,'' *IEEE Netw.*, vol. 36, no. 3, pp. 204–210, May 2022.

[11] A. Al-Shawabka et al., ''Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting,'' in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 646–655.
[12] 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. 22808–22818, 2022.
[13] N. Basha, B. Hamdaoui, K. Sivanesan, and M. Guizani, ''Channel-resilient deep-learning-driven device fingerprinting through multiple data streams,'' *IEEE Open J. Commun. Soc.*, vol. 4, pp. 118–133, 2023.
[14] F. Restuccia et al., ''DeepRadioID: Real-time channel-resilient optimization of deep learning-based radio fingerprinting algorithms,'' in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jul. 2019, pp. 51–60.
[15] B. Hamdaoui, N. Basha, and K. Sivanesan, ''Deep learning-enabled zero-touch device identification: Mitigating the impact of channel variability through MIMO diversity,'' *IEEE Commun. Mag.*, vol. 61, no. 6, pp. 80–85, Jun. 2023.
[16] S. Rajendran and Z. Sun, ''RF impairment model-based IoT physical-layer identification for enhanced domain generalization,'' *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1285–1299, 2022.
[17] A. Celik and A. M. Eltawil, ''At the dawn of generative AI era: A tutorial-cum-survey on new frontiers in 6G wireless intelligence,'' *IEEE Open J. Commun. Soc.*, vol. 5, pp. 2433–2489, 2024.
[18] H. Li, K. Gupta, C. Wang, N. Ghose, and B. Wang, ''RadioNet: Robust deep-learning based radio fingerprinting,'' in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2022, pp. 190–198.
[19] A. Elmaghbub, B. Hamdaoui, and W.-K. Wong, ''ADL-ID: Adversarial disentanglement learning for wireless device fingerprinting temporal domain adaptation,'' 2023, *arXiv:2301.12360*.
[20] A. Al-Shawabka, P. Pietraski, S. B Pattar, P. Johari, and T. Melodia, ''SignCRF: Scalable channel-agnostic data-driven radio authentication system,'' 2023, *arXiv:2303.12811*.
[21] J. A. Snoap, J. A. Latshaw, D. C. Popescu, and C. M. Spooner, ''Robust classification of digitally modulated signals using capsule networks and cyclic cumulant features,'' 2022, *arXiv:2211.00232*.
[22] J. Gaskin, B. Hamdaoui, and W.-K. Wong, ''Tweak: Towards portable deep learning models for domain-agnostic LoRa device authentication,'' 2022, *arXiv:2209.00786*.
[23] J. A. Snoap, D. C. Popescu, and C. M. Spooner, ''Novel nonlinear neural-network layers for high performance and generalization in modulation-recognition applications,'' in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2023, pp. 562–567.
[24] A. Elmaghbub and B. Hamdaoui, ''A needle in a haystack: Distinguishable deep neural network features for domain-agnostic device fingerprinting,'' in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2023, pp. 1–9.
[25] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, ''Wireless device identification with radiometric signatures,'' in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, Sep. 2008, pp. 116–127.
[26] H. Givehchian, N. Bhaskar, A. Redding, H. Zhao, A. Schulman, and D. Bharadia, ''Practical obfuscation of BLE physical-layer fingerprints on mobile devices,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, Oct. 2024, p. 73.
[27] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong, ''Accurate and efficient wireless device fingerprinting using channel state information,'' in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jun. 2018, pp. 1700–1708.
[28] N. Soltani, K. Sankhe, J. Dy, S. Ioannidis, and K. Chowdhury, ''More is better: Data augmentation for channel-resilient RF fingerprinting,'' *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 66–72, Oct. 2020.
[29] 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 *Proc. 22nd Int. Symp. Theory Algorithmic Found. Protocol Design Mobile Netw. Mobile Comput.*, 2021, pp. 251–260.
[30] I. Redko, E. Morvant, A. Habrard, M. Sebban, and Y. Bennani, ''A survey on domain adaptation theory: Learning bounds and theoretical guarantees,'' 2020, *arXiv:2004.11829*.
[31] H. Zhou, C. Nicholls, T. Kunz, and H. Schwartz, ''Frequency accuracy & stability dependencies of crystal oscillators,'' Dept. Syst. Comput. Eng., Carleton Univ., Ottawa, ON, Canada, Tech. Rep. SCE-08-12, 2008.
[32] H. Givehchian et al., ''Evaluating physical-layer BLE location tracking attacks on mobile devices,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, Sep. 2022, pp. 1690–1704.
[33] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, ''Fingerprinting Wi-Fi devices using software defined radios,'' in *Proc. 9th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jul. 2016, pp. 3–14.
[34] A. C. Polak and D. L. Goeckel, ''Wireless device identification based on RF oscillator imperfections,'' *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2492–2501, Dec. 2015.
[35] X. Gu et al., ''TEA-RFFI: Temperature adjusted radio frequency fingerprint-based smartphone identification,'' *Comput. Netw.*, vol. 238, Jan. 2024, Art. no. 110115.
[36] C. R. Johnson Jr., W. A. Sethares, and A. G. Klein, *Software Receiver Design: Build Your Own Digital Communication System in Five Easy Steps*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[37] G. E. Hinton and S. Roweis, ''Stochastic neighbor embedding,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 15, 2002, pp. 1–15.
[38] A. Saeif, S. Savio, and O. Gabriele, ''The day-after-tomorrow: On the performance of radio fingerprinting over time,'' in *Proc. Annu. Comput. Secur. Appl. Conf.*, Dec. 2023, pp. 439–450.
[39] A. Elmaghbub, B. Hamdaoui, and A. Natarajan, ''WideScan: Exploiting out-of-band distortion for device classification using deep learning,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.
[40] X. Liu, D. Yang, and A. E. Gamal, ''Deep neural network architectures for modulation classification,'' in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 915–919.
[41] T. Parks and J. McClellan, ''Chebyshev approximation for nonrecursive digital filters with linear phase,'' *IEEE Trans. Circuit Theory*, vol. CT-19, no. 2, pp. 189–194, Mar. 1972.
[42] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[43] X. Gu et al., ''TeRFF: Temperature-aware radio frequency fingerprinting for smartphones,'' in *Proc. 19th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Sep. 2022, pp. 127–135.

ABDURRAHMAN ELMAGHBUB received the B.S. (summa cum laude) and M.S. degrees in electrical and computer engineering from Oregon State University, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering and Computer Science. His research interests include wireless communication and networking with a current focus on applying deep learning to wireless device classification.

BECHIR HAMDAOUI (Senior Member, IEEE) received the M.S. degree in electrical and computer engineering, the M.S. degree in computer science, and the Ph.D. degree in electrical and computer engineering from the University of Wisconsin–Madison, Madison, WI, USA, in 2002, 2004, and 2005, respectively. He is currently a Professor with the School of Electrical Engineering and Computer Science and the Founding Director of the NetSTAR Laboratory, Oregon State University. His research interests include theoretical and experimental research that enhances the cybersecurity and resiliency of future intelligent networked systems, including connected and autonomous vehicles, 5G/6G wireless, networked drones, smart cities, and cloud data centers. He and his team have won several awards, including the ISSIP 2020 Distinguished Recognition Award, the 2009 NSF CAREER Award, the ICC 2017 Best Paper Award, and the 2016 EECS Outstanding Research Award. He chaired and organized many IEEE/ACM conference symposia and workshop programs. He serves/served as an associate editor for several IEEE journals and magazines. He served as a Distinguished Lecturer for the IEEE Communication Society in 2016 and 2017 and the Chair and Co-Chair of the IEEE Communications Society's Wireless Technical Committee (WTC) from January 2019 to December 2022.