No Radio Left Behind: Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments

Kunal Sankhe, Mauro Belgiovine, Fan Zhou, Luca Angioloni, Francesco Restuccia, Salvatore D'Oro,
Tommaso Melodia, Stratis Ioannidis, and Kaushik Chowdhury
Northeastern University, Boston, MA, USA

***Abstract*—Due to the unprecedented scale of the Internet of Things, designing scalable, accurate, energy-efficient and tamper-proof authentication mechanisms has now become more important than ever. To this end, in this paper we present ORACLE, a novel system based on convolutional neural networks (CNNs) to "fingerprint" (*i.e.*, identify) a unique radio from a large pool of devices by deep-learning the fine-grained hardware impairments imposed by radio circuitry on physical-layer I/Q samples. First, we show how hardware-specific imperfections are learned by the CNN framework. Then, we extensively evaluate the performance of ORACLE on several first-of-its-kind large-scale datasets of WiFi- transmissions collected "in the wild", as well as a dataset of nominally-identical (*i.e.*, equal baseband signals) WiFi devices, reaching 80-90% accuracy is many cases with the error gap arising due to channel-induced effects. Finally, we show through an experimental testbed, how this accuracy can reach over 99% by intentionally inserting and learning the effect of controlled impairments at the transmitter side, to completely remove the impact of the wireless channel. Furthermore, to scale this approach for classifying potential thousands of radios, we propose an impairment hopping spread spectrum (IHOP) technique that is resilient to spoofing attacks.**

I. Introduction

Sensing the wireless spectrum and identifying active radios within the bands of interest directly impacts spectrum usage. This paper takes the first step in distinguishing radios in a shared spectrum environment by using machine learning to detect characteristic reference signatures embedded in their transmitted electromagentic waves, a process known as *RF fingerprinting*. Our goal is to achieve this with information that can be leveraged at the radio hardware front-end. We separately consider situations where the channel is unchanging between training and validation (idealized) and when the channel is dynamic (practical). The key innovation in our approach, termed ORACLE (*Optimized Radio clAssification through ConvolutionaL neural nEtworks*), is that it learns the unique modifications present within the in-phase (I) and quadrature-phase (Q) samples that are introduced in the signal as it passes through the transmitter chain. ORACLE uses Convolutional Neural Networks (CNNs) to learn and then identify individual radios through device-specific variations contributed by the inherent randomness in the manufacturing process. These so called *imperfections* are present within the analog components (digital-to-analog converters, band-pass filters, frequency mixers and power amplifiers) that compose a typical transmission chain, differentiating radio devices even if their manufacturer and make/model are identical. ORACLE can transform many emerging areas, such as the Internet of Things (IoT), which will result in billions of devices deployed worldwide [1]. For this reason, one of IoT's most crucial issues is designing scalable, reliable and energy-efficient authentication mechanisms. Most of the existing authentication mechanisms are not directly applicable to the IoT since they are based on energy-expensive cryptography-based algorithms and protocols [2]. ORACLE proposes a way forward for achieving such authentication at a device level, which cannot be be tampered with software-based code insertion.

A. Signatures contained within IQ samples

Radio fingerprinting leverages tiny imperfections of off-the-shelf wireless circuitry that make a number of wireless devices operating on the same baseband signal necessarily transmit two slightly different waveforms [3]. These so called *imperfections* are present within the analog components (digital-to-analog converters, band-pass filters, frequency mixers and power amplifiers) that compose a typical transmission chain, differentiating radio devices even if their manufacturer/make/model are identical. Physical (PHY) layer, medium access control (MAC) layer, and upper layers have also been utilized for fingerprinting [4]. However, simple unique identifiers such as IP addresses, MAC addresses, international mobile station equipment identity (IMEI) numbers can easily be spoofed. Location-based features such as radio signal strength (RSS), angle of arrival (AoA) and channel state information (CSI) are susceptible to mobility and environmental changes. ORACLE, instead, focuses on those transmitter features that are inherent to a device's hardware makeup, which are unchanging and cannot be easily replicated by malicious agents.

Fig. 1 indicates an example scenario of these so called transmitter signatures (rigorously studied in Sec. III) for 16-QAM constellation. The red circles indicate the ideal constellation points formed by the I (x-axis) and Q (y-axis) samples, and the black crosses indicate actual constellation points that are shifted due to a specific type of hardware imperfection. Practical transmitters have a combination of these shifts that form their unique signatures, though we show only three plots caused by IQ imbalance, nonlinear distortion and phase noise in the figure. ORACLE aims to learn and intentionally modify some of these features on the transmitter through USRP Hardware Driver (UHD) software API commands, thereby enhancing identifiability/classifier efficiency. We note that ORACLE can be easily used in conjunction with other existing and higher layer classification approaches.

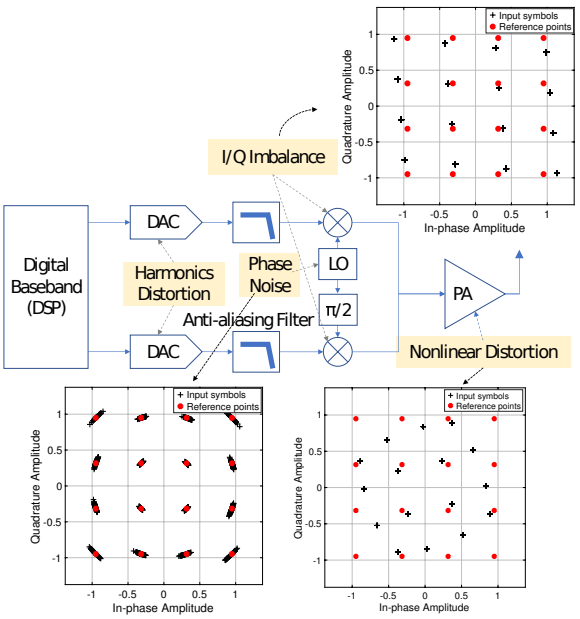


Figure 1: Typical transceiver chain with various sources of RF impairments.

B. Machine learning for RF fingerprinting in ORACLE

Machine learning (ML) techniques have shown great promise in image and speech identification problems, and are steadily gaining traction in applications within the wireless domain. ORACLE is solely built on a convolutional neural network architecture that has not only seen success in the above areas, but has also been previously used for modulation [5] and protocol identification [6]. ORACLE adopts a stagewise approach towards achieving practical classification. We attain this in the first step by demonstrating 99% accuracy on an externally obtained data set of 100+ COTS WiFi radios (not all of which are bit-similar), as well as on our testbed of 16 bit-similar USRP X310 radios that we configure to be exactly similar in terms of waveforms generated (same 802.11a PHY frame, modulation/protocol/mac ID).

C. The ORACLE approach

For radios operating in a channel-invariant environment (henceforth referred to as a *static* channel), ORACLE can identify radios by using both raw IQ samples as well as by estimating the channel from training datasets and removing, wherever possible, channel dependent shifts in a process that we call as *undermodulation* for brevity. Though this feature engineering step of *undermodulation* does increase the resilience of the CNN for certain test scenarios reaching 80-95%, the variation of the wireless channel (henceforth referred to as a *dynamic* channel) generally has a dominant impact on the transformation of the IQ samples in the complex plane. Here we make an interesting observation: training with undermodulated symbols makes low-end SDRs (such as the Ettus N210 USRP) robust to channel variations. However, high-performance SDRs (such as the X310 USRP) that are manufactured with components with lower variability need an additional step. For such high-end bit-similar devices, ORACLE has a principled method for intentionally introducing impairments to increase differentiability while minimizing the bit error rate (BER) for each transmitter. The key insight here is that controlled addition of impairments in a bit-similar radio generates a unique pattern in the demodulated signal at the receiver, which is independent of channel variations.

In summary, the main contributions of this paper are:

- We study the different causes of transmitter-side reference signatures, and visualize their impact on the IQ constellation space. We identify specific features that are amenable to fine tuning by the receiver feedback using software APIs.
- Using an SDR testbed and an external database of signals collected from 50-500 mixed WiFi device transmitters, we propose the design of ORACLE, which includes a robust CNN architecture returning 99% device classification accuracy on quasi-static channel conditions using only raw samples and 83.5% of accuracy on dynamic channel conditions using I/Q samples with impairment obtained with the *undercomplete* demodulation approach.
- We propose and implement a scalable and secure identification technique, called as impairment hopping spread spectrum (IHOP), which identifies a radio through a random pseudo-noise (PN) binary sequence, termed as '*identification key*'. The transmitter conveys this key by switching between pairwise impairments. With experimental evaluations, IHOP achieves identification accuracy of $> 99\%$, while ensuring the BER constraint for each radio. The identification key and the pairwise impairments change after every successful identification, thus making it difficult for an adversary from performing a spoofing attack.

II. Related Work

Traditional techniques for radio fingerprinting [7], [8], [9], [10], [11], [12], [13] rely on complex, hand-made, ad-hoc features that are tailored to address specific classes of wireless devices and protocols. In this paper, we approach the problem from a different perspective and use techniques based on *deep learning* [14] to design *general-purpose and high-accuracy* radio fingerprinting algorithms. Deep learning models go beyond legacy "shallow" neural networks and can autonomously extract extremely complex features without the need of application-specific and computational-expensive feature extraction/selection algorithms [15], [16], [17]. Furthermore, different models can be trained on the exact same input data, providing a common test bench for researchers to test the performance of their algorithms.

While there exists a vast literature on the theory and applications of machine learning, we only review works that are directly relevant to the problem of RF fingerprinting. Given the ground truth to facilitate model creations, we follow the supervised learning paradigm, where a large collection of labeled samples are applied for training, prior to network deployment.

The vast majority of existing work has applied carefully-tailored feature extraction techniques at the physical layer to fingerprint wireless devices [7], [8], [9], [10], [11], [12], [13]. In particular, Brik *et al.* [8] considered a combination of frequency offset, transients, and constellation errors to

fingerprint 130 IEEE 802.11b cards with an accuracy of 99%, similarly to our approach. However, conversely from ours, the experiments in [8] were performed in an RF-insulated environment (*i.e.*, without any channel effect), thus their effectiveness in real-world environments has yet to be established. Vo *et al.* [10] proposed a series of algorithms with features based on frequency offsets, transients and the WiFi scrambling seed. The algorithms were validated with data collected from a series of COTS WiFi cards in a non-controlled RF environment, achieving accuracy between 44 and 50% on 93 devices. Recently, Peng *et al.* [11] proposed fingerprinting algorithms for ZigBee devices, showing that their features achieve almost 95% accuracy on a 54-radio testbed.

The key drawback of feature-based fingerprinting techniques is that they are inherently tailored for a specific technology only, which ultimately limits their applicability. Moreover, existing work has not considered the problem of optimizing the algorithm's accuracy in real-time. On the other hand, deep learning is a more general method that offers a powerful framework for learning complex functions, leveraging large datasets. For this reason, research has started developing deep learning models to address physical-layer classification problems such as modulation recognition, and OFDM parameter identification [15], [18], [19], [20], [17]. Wang *et al.* [21] present a general deep learning framework for RF sensing in the IoT, along with several experimental case studies such as location fingerpinting [22] and healthcare sensing [23] applications. Ferdowsi and Saad [24] use an LSTM to extract stochastic features from IoT signals and dynamically watermark these features inside the original signal, in order to avoid eavesdropping attacks in IoT devices deployments. O'Shea and Corgan [5] and O'Shea and Hoydis [25] apply deep learning at the physical layer, specifically focusing on modulation recognition using IQ samples and convolutional neural networks. They classify 11 different modulation schemes. However, this approach does not identify a device like ORACLE, but only the modulation type used by the transmitter. In our previous work [26], [27], we have explored CNNs to fingerprint 16 bit-similar USRP X310 devices (*i.e.*, same hardware, protocol, physical address, MAC ID) using only IQ samples at the physical layer, showing that by using artificially-introduced hardware impairments the accuracy can be improved to 99%. This paper builds on this work, but explores many additional experimental scenarios and the impact of feature engineering on large populations of devices (50-500).

To the best of our knowledge, ORACLE is the first work that allows training a CNN for bit-similar device identification such that the same classifier may operate in unknown/dynamic channel conditions without the need for new trials.

III. A CLOSER LOOK AT DEVICE SIGNATURES

In this section, we first study RF hardware impairments that cause variations in IQ samples, resulting in a unique *signature* for each device. We focus on IQ imbalance and DC offset, the two impairments that (i) are independent of the environment, and (ii) do not apply only in context of a specific transmitter-receiver pair (as opposed to, say, relative phase offset). Then, we present a method of introducing controlled impairments using GNU Radio UHD API at the receiver. Subsequently, we explain the experimental testbed setup for trace data collection.

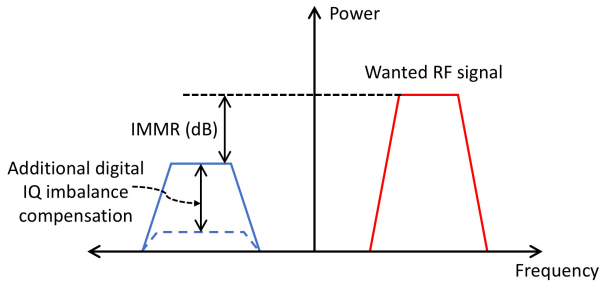


Figure 2: Effect of IQ imbalance quantified through IMRR.

A. RF impairments

Using the MATLAB Communications System Toolbox, we simulate a typical wireless communications processing chain (see Fig. 1, with the shifts in the received complex valued IQ samples), and then modify the ideal operational blocks to introduce RF impairments, typically seen in actual hardware implementations. This allows us to individually study the IQ imbalance, DC offset, phase noise, carrier frequency offset and nonlinear distortions of power amplifier. In this paper, we focus on the two impairments (IQ imbalance and DC offset) owing to space constraints, though our approach can be trivially extended for others as well.

•**IQ imbalance:** Quadrature mixers are often impaired by gain and phase mismatches between the parallel sections of the RF chain dealing with the I and Q signal paths. The mismatch in their gains causes amplitude imbalance, whereas phase deviation from 90° in the quadrature signal results in phase imbalance. IQ imbalance varies only with frequency due to frequency-dependent low pass filters, and thus, it carries a unique signature of a transmitter for that frequency.

•**DC offset:** This is caused within the quadrature mixers due to the finite isolation between Local Oscillator (LO) and RF ports of a mixer, and a direct feedthrough from the LO signal often gets coupled to the output.

B. Software-based control of impairments

We first explain the use of self-calibrations utilities provided by Ettus to set IQ imbalance and DC offset in the transmitter chain using GNU Radio functions.

• ***IQ imbalance compensation*:** Let $s(t) \in \mathbb{C}$ be the transmitted baseband complex signal at time t before being distorted by IQ imbalance. Then, the distorted baseband signal in the time domain is:

$$s_d(t) = \mu_t s(t) + v_t s^*(t), \tag{1}$$

where the distortion parameters μ_t and v_t are related to amplitude and phase imbalances in the I and Q paths of the quadrature mixer in the transmitter chain.

In a simplified model, we define these distortions parameters as $\mu_t = \cos\left(\frac{\theta_t}{2}\right) + j\alpha_t \sin\left(\frac{\theta_t}{2}\right)$ and $v_t = \alpha_t \cos\left(\frac{\theta_t}{2}\right) - j \sin\left(\frac{\theta_t}{2}\right)$,

Table I: A snapshot of IMRR levels of IQ imbalance recorded using `uhd_cal_tx_iq_balance` utility

Correction factor	Power of main tone	Power of image tone	IMRR (dB)
$-0.272 - 0.636$	-49.036	-66.138	-17.102
$-0.636 - 0.636$	-48.852	-66.306	-17.454
$-0.454 - 0.0909$	-49.091	-67.326	-18.235

where α_t and θ_t are the amplitude and phase imbalance between the I and Q signal paths at the transmitter, respectively. The phase imbalance θ_t is any phase deviation from the ideal 90°. The amplitude imbalance is defined as $\alpha_t = \frac{\alpha_I - \alpha_Q}{\alpha_I + \alpha_Q}$, where α_I and α_Q are the respective gain amplitudes on the I and Q paths.

IQ imbalance causes interference in the signal by generating its image at a mirror frequency. It is quantified by measuring the power of the image with respect to the desired signal, also called as Image Rejection Ratio (IMRR), as shown in Fig. 2. The IMRR is calculated by sending a complex sinusoidal e^{jwt}, and by taking ratio of the power of the signal at the image frequency $(-w)$ and desired frequency (w). Thus, IMRR at desired center frequency w is defined as:

$$IMRR(w) = \frac{\gamma_t^2(w) + 1 - 2\gamma_t(w)\cos\big(\theta_t(w)\big)}{\gamma_t^2(w) + 1 + 2\gamma_t(w)\cos\big(\theta_t(w)\big)}, \qquad (2)$$

where $\gamma_t(w) = \alpha_t(w) + 1$; $\alpha_t(w)$, $\theta_t(w)$ are amplitude imbalance and phase difference respectively measured at center frequency w.

While many theoretical time and frequency domain methods allow compensation for the IQ imbalance, we use the Ettus provided UHD calibration utility `uhd_cal_tx_iq_balance`. It performs a calibration sweep over a range of frequencies checking the transmission path signal leakage into the receive path.

At runtime, the UHD software automatically applies the correction, typically a single complex factor, to the transmit chain of the RF daughterboard. For a given value of correction factor, a single frequency tone is transmitted, and the power of the desired tone and the image tone are measured to compute IMRR. We modified this utility to record the correction factors and the corresponding IMRR. Table I shows a snapshot of the recorded IMRR levels for USRP X310 radio at a center frequency of 2.45 GHz.

• ***DC offset compensation*:** DC offset results in a large spike in the center of the spectrum. By measuring the power of the main tone at the DC frequency, we can measure the amount of DC offset. A UHD calibration utility `uhd_cal_tx_dc_offset` uses a single complex factor to correct DC offset level. It finds the best correction factor that minimizes the power of the DC tone. Again, by modifying the utility, we record the levels of DC offset level for the correction factor.

We use the open-source GNU Radio companion (GRC) to transmit standard-compliant IEEE 802.11a WiFi packets through the SDR. Using `set_iq_balance` and `set_dc_offset` functions in GRC, these two separate complex correction factors can be set to intentionally introduce required level of impairments in the radio.

C. Trace Data collection

We use two different data sets for our study.

1) External data set: We use an external dataset (about 4TB) consisting of about 103 million *signals* collected from over 50 thousands various kinds of WiFi devices (phones/laptops/tablets/drone) made available by the US Defence Advanced Research Projects Agency (DARPA). Unfortunately, this dataset is not openly available and is currently protected by a non-disclosure agreement (NDA). Here we use signal to refer a sequence of consecutive I/Q samples. Most of the WiFi signals are recorded in the wild with a Tektronix RSA at 200 million samples per second (MSPs). Furthermore, to test the generality of the classifier, the external dataset includes a small subsets recorded with USRP under multiple antenna polarizations, indoor/outdoor channels and various signal densities. In the dataset, about 20 thousands devices have more than 100 signals and over 1 thousand devices has more than 10 thousands signals. In this paper, we mainly show the results of ORACLE with four subsets with different number of devices (50, 100, 250, 500) that randomly sampled from the external dataset. In all of the datsets we used, every device has 141 training signals and 35 testing signals, and each signal consists an average of about 15 thousands I/Q samples unless otherwise specified.

2) Laboratory-generated data: For a more controlled study of our architecture, we also collect IQ samples from an experimental setup of USRP SDRs with a fixed USRP B210 as the receiver. All transmitters are bit-similar USRP X310 radios that emit IEEE 802.11a standards compliant frames generated via a MATLAB WLAN System toolbox. The data frames generated contain random payload but have the same address fields, and are then streamed to the selected SDR for over-the-air wireless transmission. The receiver SDR samples the incoming signals at 5 MS/s sampling rate at center frequency of 2.45 GHz for WiFi. The collected complex IQ samples are partitioned into subsequences. For our experimental study, we set a fixed subsequence length of 128, i.e., the length of contiguous samples that will be used at a time for training and classification. Overall, we collect over 20 million samples for each radio, subsequently divided into training, validation and test set. The authors commit to release this dataset immediately after acceptance of the paper.

IV. ORACLE CNN ARCHITECTURE

A. CNN model

Our proposed architecture, as shown in Fig. 3, consists of a deep Convolutional Neural Network (CNN). Our approach is partly inspired from AlexNet [28], which is a deep CNN architecture specifically designed to classify 1.2 million high-resolution images available in the ImageNet dataset into 1000 different classes. Unlike AlexNet, which is made up of 8 layers (5 convolution and 3 fully connected), our architecture extends to 15 layers. We use one-dimensional (1D) convolutions to capture the local temporal relations within I/Q symbols, which carry subtle identifying information of transmitting radios. Since we rely on only those hardware impairments that do not vary over time, their effect on the transmitted signal can

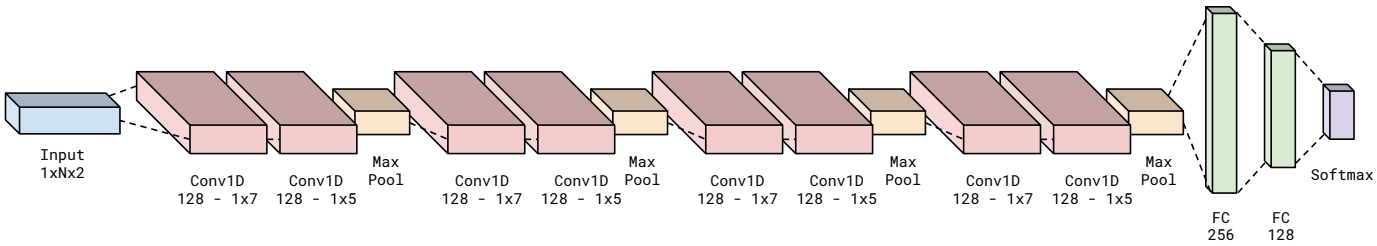


Figure 3: Our proposed CNN architecture with 8 convolution 1D and 2 fully connected layers.

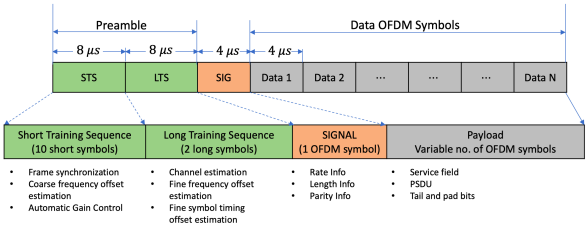


Figure 4: IEEE 802.11a OFDM Frame Structure

be identified in different local portions of the entire received waveforms. 1D CNNs are particularly effective at these kind of tasks, i.e., identifying features from fixed-length segments of the complete dataset when the location of such features within the segment are not highly correlated. ORACLE operates 1D convolutions along the time axis and uses I and Q data as two distinct channels of the 1D sequence.

A main building block of the proposed CNN model consists of two 1D-convolution layers, each has 128 filters of size 7 for the first layer and 5 for the second one. These two convolutional layers are followed by a Max Pooling layer, used to provide (a) shift invariance and (b) reduce the dimensionality of the output feature maps of the preceding convolution layer, while retaining the most important information. We then stack 4 of such building blocks, followed by a set of 2 Fully Connected (FC) layers, composed of 256 and 128 neurons respectively, and a Softmax classifier layer. In order to overcome overfitting, we set the dropout rate to 50% at the FC layers. We train the neural network using Adam optimizer with a learning rate $lr = 0.0001$. The number of epochs is determined by *early stopping* criterion (training is stopped if validation accuracy didn't increase in last 10 epochs) and batchsize is 1024. We choose a *sliding window* approach to partition the training signals into overlapping sequence of samples, referred as *slices*, to enhance the shift invariance of the features learned by the CNN. Since ORACLE's CNN model is trained with complex IQ symbols, it is suitable for any radio fingerprinting applications and is independent of underlying PHY layer protocols and modulation schemes.

B. Feature engineering: undercomplete demodulation

We propose an *undercomplete* demodulation that aims to remove only an effect of the channel from raw IQ samples, without compensating the device's imperfections. In particular, we improve the classifier's accuracy by compensating the effect of the channel by channel estimation and equalization, however, leaving the *frequency and sampling offsets* in the I/Q samples. Thus, we (i) first estimate and compensate the carrier and sampling frequency offsets (ii) estimate the channel using pilot training sequence (iii) reapply the offsets computed in step (i) to obtain our final sequence of demodulated symbols that is fed to the CNN.

Fig. 4 shows IEEE 802.11a OFDM frame structure. It consists of two training sequences, namely Short Training Sequence (STS) and Long Training Sequence (LTS), each with duration of 8 μsec. The STS is primarily used for frame synchronization, Automatic Gain Control (AGC), and coarse frequency offset estimation. The LTS is mainly used for channel estimation, fine frequency and symbol timing offset estimation. Using algorithms proposed in [29], we first use both STS and LTS to estimate coarse and fine carrier and sampling frequency offsets. Subsequently, we use LTS to find channel estimates using least-square (LS) method. After equalization using estimated channel, we reintroduce those offsets to obtain *undercomplete demodulated* symbols. The outcome here is that we are now able to retrieve IQ samples that are not impacted by the wireless channel but still capture the inherent hardware imperfections of the radio devices.

V. CNN Performance and Impact of the Wireless Channel

First, we verify the performance of ORACLE's CNN architecture on large-scale wireless systems and later on experimental testbed developed in laboratory. Our goal here is to show the generality of our CNN architecture, and identify its benefits and limitations in scenarios where the wireless channel shows variations.

A. External data set

We consider 500-device dataset of IEEE 802.11a/g (WiFi) transmissions obtained through the DARPA RFMLS program [30]. For data collection, raw IQ samples were collected "in the wild" (i.e., no controlled environment) with a Tektronix RSA operating at 200 MSPS.

Fig. 5 shows the performance of the CNN using raw samples or the featured engineered undercomplete modulated data as input. It shows that the ORACLE's performance decreases significantly when raw IQ samples are fed to the CNN. This is primarily due to the following: (i) WiFi signals collected in the wild may suffer from adjacent channel interference, packet collisions, and even other interference from other devices operating in the same channel. Thus, the signals are not obtained in pristine environments - in fact other devices from the same pool of radios may be operating at concurrent times; (ii) The wireless channel changes over the duration of a day, and the obvious impact of scaling up the number of devices

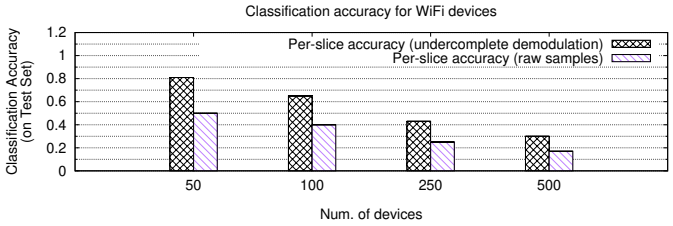


Figure 5: Classification accuracy comparison for WiFi devices using raw samples and undercomplete demodulation. All signals were collected in the same day.

is also visible in the plot. The learning enabled through the CNN, regrettably, is not fully divorced from the channel effects which dominate classification accuracy, impacting highly those radios that have signals with low SNR. The undercomplete demodulation partially addresses this problem giving about 85% accuracy for 50 devices. Again this method too does not scale well with increasing number of devices.

The impact of the channel becomes even more acute when the training and testing occur on different days. On a dataset of IQ samples collected in separate days for training and testing the model, the classification accuracy for 50 COTS WiFi devices drops to 46%.

B. Testbed Results

Our preliminary evaluation aims to demonstrate the accuracy of ORACLE's CNN architecture under different conditions by using several signal datasets coming from different radio sources, in order to show the generality of this approach and identifying its benefits and limitations in such scenarios.

1) Accuracy in quasi-static channel conditions: First, we show the performance of the CNN using only raw samples as input. This approach is particularly appealing because it doesn't require any previous knowledge of the modulation scheme and protocol used, making it possible to also remove specific pre-processing steps necessary to demodulate the wireless transmissions. Using 16, high-end X310 USRP SDRs, with the same B210 radio as a receiver, we performed data collection for one radio at a time, placed in the same position of a wide empty room, in line-of-sight with the receiver. We refer to this setup as a *controlled* environment, given the ideal conditions of trace collection. Our training set for this experiment consists, per radio, of $200K$ windowed training examples and $10K$ examples for validation. We use another $50K$ examples for each device to test the performance of our trained model. It takes $\approx 30min$ with our current setup to train the model for 16 radios. For this setup, we obtained 98.6% accuracy on the test set, shown in Fig. 6a. It is important to note that in such controlled environment the channel conditions can be considered *quasi-static*, meaning that we didn't observe drastic changes in subcarriers' gain for the training and test samples.

2) Limitations of raw IQ samples in dynamic channels: Multipath reflection and fading have considerable impact on received IQ samples, at times distorting the samples wherein the classifier no longer correctly identifies the radios. Typically, the effect of the channel is compensated by channel estimation and equalization techniques to correctly retrieve over-the-air transmitted data. Thus, as we show next, classification performance degrades severely when either (i) classifiers are trained on raw IQ samples under a given channel and then tested on IQ samples obtained under different channels, or (ii) transmitters experience very similar channel conditions.

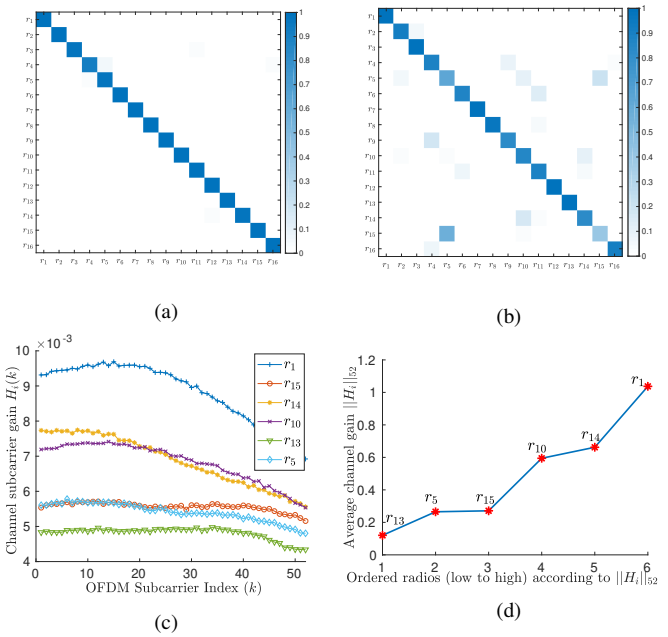
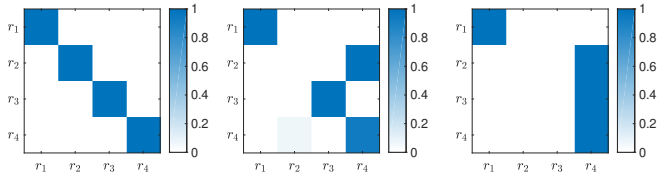


Figure 6: Confusion matrix relative to two experiments with same devices and different locations: (a) overall accuracy is 98.60%; (b) overall accuracy is 87.13%. (c) Estimated channel gain $\tilde{H}_i(k)$ for k^{th} subcarrier for each radio $r_i \in R$ (d) Magnitude of estimated channel $||\tilde{H}_i||_{52}$ for all radios $r_i \in R$ (ordered from lower to higher).

As mentioned before, Fig. 6a shows the classification accuracy with our testbed composed of 16x X310 radios, with near-perfect results for all the devices using raw samples. However, Fig. 6b shows the same setup in a different location where several outliers exist, as the confusion matrix shows, e.g., see radio pairs (5,15), (10, 14). The reason is that *the similarity in the wireless channel experienced by certain transmitter pairs dominates subtle hardware variations*. Given a set of R radios, $\tilde{H}_i(k)$ represents the average channel gain in k^{th} subcarrier of each radio $r_i \in R$, estimated over WiFi packets belonging to the training dataset.

Fig. 6c and 6d reveal how received samples from transmitters with smaller differences in channel estimation are more likely to be misclassified by ORACLE during testing. This shows that wireless channel state affects the distribution of complex symbols captured by the receiver in a non-negligible manner, and therefore *becomes a discriminating factor when the classifier is trained with raw IQ samples*. If we try to use a pre-trained model and use it to classify samples collected from same devices but at different times or locations, the classification result is unpredictable. See Fig. 7a, 7b and 7c for the classification results showing the time and location dependence of the trained classifier.



(a) (b) (c)

Figure 7: (a) Classification accuracy for 4 devices tested at time t_1 and location l_1; (b) time t_2 and same location l_1; (c) time t_3 and different location l_3.

VI. ORACLE with Feedback for Dynamic Channels

This section describes the enhancements in ORACLE that allow it to robustly classify transmitters in unseen environments. The two main assumptions here are: (i) instead of raw IQ samples, ORACLE works with demodulated symbols without feature engineering proposed in Sec. IV-B, and (ii) in a pre-deployment phase, the receiver provides feedback to the transmitter to incorporate controlled impairments. We first present a method of introducing controlled impairments using GNU Radio UHD API at the receiver. We then discuss the role of impairments in generating unique pattern in demodulated data, and show that the impairments (i) are independent of the environment, and (ii) do not apply only in context of a specific transmitter-receiver pair (as opposed to, say, relative phase offset). Finally, we introduce the ORACLE training process after allocating the unique patterns to bit-similar radios.

A. Impact of impairments on undercomplete demodulation

ORACLE modifies the transmitter chain of the SDRs such that their respective demodulated symbols acquire unique characteristics that make the CNN robust to channel changes, i.e., it makes the transmitter hardware *dominate* channel induced variations. We first validate the hypothesis that a given combination of impairments results in repeatability in the outcome of the classification. To demonstrate this, consider demodulated symbols received from two X310 radios, over cable and air channels, as shown in Fig. 8, for three different levels of IQ imbalance. The first row shows slight differences in the demodulated samples when the channel is completely changed (i.e., air to cable) for the same transmitter. In the second row, when the same channel is maintained, but the transmitters themselves are different, adding the same level of IQ imbalance results in virtually the same pattern in each case, ensuring repeatability and robustness.

We also quantitatively analyze the property of the channel- and device- invariance of the patterns with Earth Mover's Distance (EMD), a widely used metric to measure similarities between two multi-dimensional distributions. More precisely, suppose we have two sets of points in $\mathbb{R}^2$. Let $A \subset \mathbb{R}^2$ and $B \subset \mathbb{R}^2$ be two subsets of equal size, i.e., $|A| = |B|$. Let F be the set of all possible bijections ($1-1$ and onto mappings) from A to B. The EMD between A and B is given by:

$$EMD\,(A, B) = \min_{f \in F} \sum_{x \in A} \|x - f(x)\| \qquad (3)$$

where $f(x) \in B$. In other words, EMD is given by the smallest possible sum of Euclidean distances between points in A and

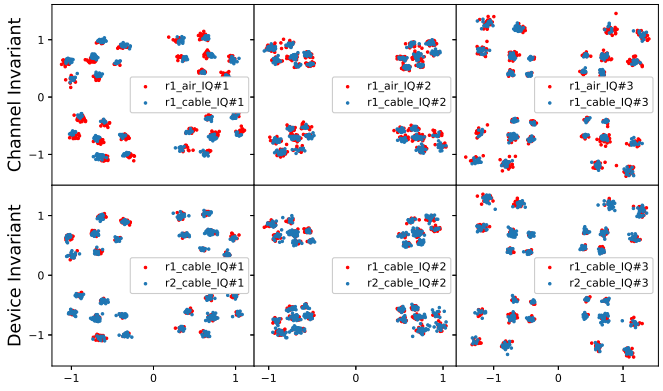
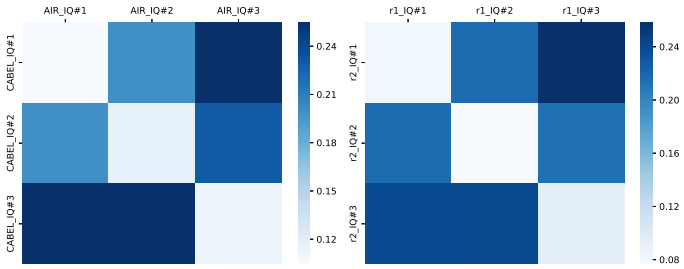


Figure 8: Patterns generated by 3 impairments on 2 devices under 2 channel conditions. First and second row show the channel- and device- invariance of the patterns respectively.



(a) (b)

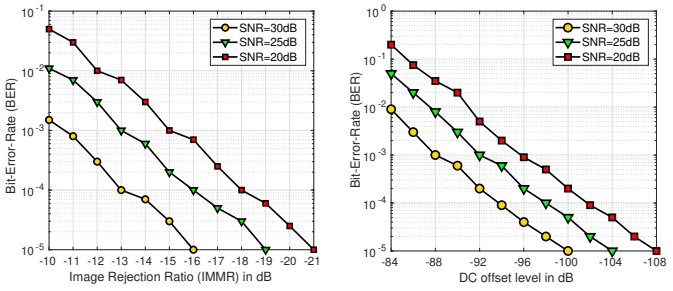
Figure 9: The EMD matrix of patterns generated (a) under different channel conditions; (b) on different devices.

B, over all possible valid bijections $f : A \to B$. Smaller EMD indicates more similarities between two patterns and vice versa. Fig. 9 (a) and (b) show the EMD matrix of patterns generated on different channel conditions and devices respectively with the same set of impairments in Fig. 8. We see that computed EMD on the matrix diagonal, which represents the patterns generated by the same impairments, are much lower than the EMD of patterns generated by different impairments. We further evaluate the EMD for the demodulated signal collected under 3 different channel conditions, 4 devices across 32 different levels of impairments. We see that the average EMD remains around 0.1 and 0.2 for patterns generated by the same and different level of impairments, respectively, despite of the variations caused by channel conditions. This result matches closely with Fig. 9 and verifies our intuition.

B. Identifying feasible impairments

The naive approach of introducing random combinations of impairments before training the CNN has three problems:

1) *Scalability:* If a new transmitter is introduced in the network, then we have to re-train the entire CNN, which is a time- and computation-heavy process.
2) *Accuracy:* It is possible that demodulated samples originating from two different transmitters (previously, easily differentiable) now appear clustered together owing to the modification in their placement on the IQ plane. This may reduce the performance of the classifier.



(a) (b)

Figure 10: (a) BER vs. IMRR value of IQ imbalance; (b) BER vs. DC offset level for different SNRs.

3) *Communication impact:* Adding impairments naturally increases the BER. Hence judicious and controlled addition is needed to limit any adverse impact on BER.

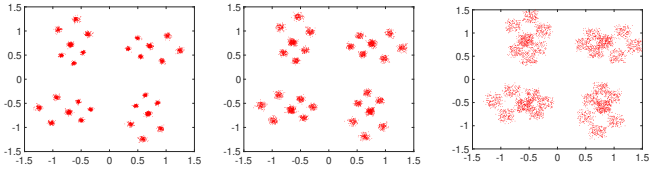
To solve these issues, ORACLE automatically selects feasible impairments that produce IQ sample constellation points that are significantly different from each other, while minimizing the influence on the BER for the transmitter. This step allows ORACLE to pre-train on virtual radios transmitter chains (constructed in GNU Radio) as the impairments dominate other variations introduced by its own hardware and the wireless channel. Thus, ORACLE learns the impairment patterns, which we have shown in Fig. 8 to be both device and channel agnostic, i.e., two different radios will result in a similar demodulated IQ pattern at the receiver under the same impairment. This approach greatly increases the flexibility of ORACLE: if a new transmitter is added, we simply assign it one of the feasible and uncommitted impairments, without any need to re-train the CNN.

We use a generic X310 USRP radio that operates in a loop while automatically adding IQ imbalance and DC offset to its hardware through utilities `uhd_cal_tx_iq_balance` and `uhd_cal_tx_dc_offset` respectively. Then the transmitter sends a stream of known data over cable to the B210 USRP receiver that checks the BER. For our experiment, we consider 80 different levels of IQ imbalance with IMRR value ranging from -9 dB to -44 dB and 120 levels of DC offset ranging from -82 dB to -140 dB. The BER plots are shown in Fig. 10a and Fig. 10b for different SNR levels, which we concisely refer to as an impairment map M, and use it later in Sec. VII. The bounds on the impairments depend on the SNR that the radios operate in. For e.g., our lab has a noise floor of -70 dBm, for which we assume an average 30 dB SNR level with the constraint on BER of 10^{-4}. Accordingly, we choose upper bound -13 dB on IMRR for IQ imbalance and -94 dB for DC offset level.

Next we explain how to identify the feasible set S of impairment combinations that satisfy the BER constraint, as shown in Algorithm 1. C_{IQ} is the set of different levels of IQ imbalance $c_1, c_2, \cdots, c_i$, ordered by their corresponding BER, i.e., $BER_{c_x} < BER_{c_{x+1}}$. Therefore, c_i is the maximum IQ imbalance we can add without exceeding the BER constraint. Note that the BER constraint of 10^{-4} is evaluated under *ideal* SNR level (40 dB). Starting from c_1, we progressively add subsequent impairments to S if the difference in EMD between

Result: S set of feasible impairments
S = $\emptyset$;
$C_{IQ} = \{c_1, c_2, ..., c_i\}$;
$C_{DC} = \{d_1, d_2, ..., d_j\}$;
Add c_1 to S;
$x = 2$; $y = 1$;
while $|S| < N$ ***and not***$(x > i$ ***and*** $y > j)$ **do**
  if $x \leq i$ ***and*** *For every* $s_z \in S$, *it is True that* $EMD(P(c_x), P(s_z)) > T$ **then**
    Add c_x to S;
    $x = x + 1$;
  end
  if $x > i$ ***and*** $y \leq j$ ***and*** *For every* $s_z \in S$, *it is True that* $EMD(P(d_y), P(s_z)) > T$ **then**
    Add d_y to S;
    $y = y + 1$;
  end
end

Algorithm 1: Greedily identify the feasible set of impairments S. N is the number of radios and $P(z)$ represents the function that generates the pattern of I/Q samples with the specified z impairment, used to compute the EMD distance. C_{IQ} and C_{DC} are sets of impairment configurations for IQ imbalance and DC offset, respectively. $T = 0.15$ is the EMD distance threshold.



(a) (b) (c)

Figure 11: Pattern generated with (a) original (demodulated) data; (b) data after adding -17 dB noise, EMD with (a): 0.07; (c) data after adding -9 dB noise , EMD with (a): 0.18.

the pattern generated by c_i and that of any existing c_k in S is larger than a threshold T. As we have seen in Sec. VI-A, $T = 0.15$ allows for an acceptable buffer in evaluating how close a given IQ pattern is to another. After we have reached c_i, we configure the radio with a different type of impairment (i.e. DC offset) until $\|S\| > N$, where N is the number of bit-similar radios to be identified.

C. CNN classifier using transmitter-side impairments

In this section, we discuss to train the classifier for the patterns (see Sec. VI-B). We reuse the same CNN architecture and the input data format as described in Sec. IV. Note all IQ samples for training are collected over the cable, i.e, we remove the influence of wireless channel so that CNN can capture the pattern generated solely by hardware impairments.

ORACLE deliberately introduces random noise by modifying the original data to increase the number and variability of the initial dataset before input to the classifier, a technique commonly used in deep learning. Since low SNR of the received samples results in scattering around the ideal constellation point location within the IQ plane, the noise is

modeled as a Gaussian variable. We note that noise may result in an altered demodulated IQ sample pattern that is different from the original one, as shown in Fig. 11. To finely control the possible variations, we maintain the EMD under 0.1 *after* adding noise, since two sample patterns up to this level are still similar to each other (see Sec. VI-A). Thus, adding noise power less than $\sigma_n^2 = -13$ dB ensures that the EMD between original and altered patterns is below this threshold.

VII. ORACLE for Scalable and Secure Radio Identification

In this section, we discuss how ORACLE identifies radios using judicious allocation of the impairments. A radio identification using greedy heuristic algorithm proposed in [27] assigns an unique impairment to each transmitter radio with an objective to minimize the sum total BER experienced by all the radios in the network. However, this approach has two limitations:

1) *Scalability:* If we allocate the unique impairment to each transmitter radio, the number of radios to be identified is limited by the number of feasible impairments derived in Sec VI-B. However, the number of impairments does not scale up due to BER constraint of the radios and thus it prevents the application of ORACLE for a large number of radios.
2) *Security:* An adversary radio can learn the unique impairment used by a legitimate radio either through eavesdropping or brute-forcing technique. By simply introducing the learned impairment during its own transmission, it can easily imitate the legitimate radio. Thus, this approach is susceptible for spoofing attack.

To solve these issues, we propose *'Impairment HOPping spread spectrum' (IHOP)*, a scalable and secure radio identification technique. IHOP is partially motivated from frequency hopping spread spectrum (FHSS) with several distinctions. In FHSS, the transmitter switches its carrier frequency among available frequencies using pseudo random binary hopping sequence known to both transmitter and receiver. For K number of available frequencies, the transmitter uses a hopping sequence of length $\log_2(K)$ to switch its carrier frequency. Similar to FHSS, simply hopping across different impairments has several disadvantages. First, a limited number of feasible impairments results in a shorter length of the hopping sequence. There is a possibility that multiple radios may introduce the same impairment at the same time if the pool of radios is large. This in turn will result in an increasing number of mis-classifications. Moreover, radios in a live deployment may not have high enough SNR to use any given impairment level, thus limiting the application of this approach.

Our approach is based on a novel design that overcomes the above limitations: ORACLE does not bind a given radio to a unique impairment. Rather, it identifies a radio through a random pseudo-noise (PN) binary sequence, henceforth referred to as '*identification key*'. The transmitter conveys this key switching between pairwise impairments (instead of many impairments as in FHSS). The identification key and the pairwise impairments change after every successful identification, thus making it difficult for an adversary from learning the pattern of sequence and thwarts its ability to perform a spoofing attack.

Fig. 12 provides a walk-through of the main operations involved in the process of IHOP based radio identification.

Phase 1: Identification through a random PN binary sequence. The transmitter and receiver both share exactly identical PN binary sequence generator that is implemented using a linear-feedback shift register (LFSR) to produce a sequence of pseudorandom binary numbers. Over a secure feedback channel, the receiver shares coefficients of LFSR through a generator polynomial with each transmitter radio. Assume the transmitter and receiver generate the sequence: 0010101, a random, yet exactly identical binary sequence. The receiver uses this sequence as an 'identification key' to uniquely identify the transmitter radio.

Phase 2: Selection of I_k and I_l Using the steps followed in Sec. VI-B, the receiver identifies a feasible set of impairments given its perceived SNR during operation, generates the pairwise impairments table and returns this to the transmitter over a secure channel. Towards this aim, the receiver first determines the upper bound on the impairment level, say I_S for each transmitter radio using its estimated SNR while just satisfying the BER constraint as shown in Sec. VI-B and Fig. 10. After determining I_S, the receiver first randomizes the ordered set of allowed impairments, $[I_1, I_2, ..., I_S]$, where $BER_{I_1} < BER_{I_2} < \cdots < BER_{I_S}$. It then finds different pairwise permutations of impairments from this set. For e.g., assume that the transmitter supports a maximum impairment level up to I_4, as shown in Fig. 12. The receiver first randomizes the ordered set $[I_1, I_2, I_3, I_4]$ and finds the $2^V = 8$ permutation pairs listed in Table A of Fig. 12, where $V = \lfloor \log_2({}^4P_2) \rfloor = 3$. Out of total permutations of ${}^SP_2$, the receiver selects first 2^V permutations, where $V = \lfloor \log_2({}^SP_2) \rfloor$ to generate a table. Each row in the table is a pair of impairments $\{I_k, I_l\}$ to be used to represent a binary 0 and 1 respectively, and this selection is constant for that entire binary sequence of the '*identification key*'.

Note that each transmitter can have a different table due to different upper bound on impairment level I_S and due to randomized permutations giving the pairwise impairments. This further enhances the security of our identification approach, while guaranteeing the BER constraint for each radio.

For each binary value 0 or 1 in the *identification key*, the transmitter maps the impairments as: $\{0 \rightarrow I_k, 1 \rightarrow I_l\}$ for a fixed number of baseband symbols. For synchronization, the transmitter radio conveys a fixed length of binary preamble sequence before *identification key* with a pair of fixed impairments $\{I_1, I_2\}$, where $BER_{I_1}, BER_{I_2} < BER_{I_j}$ for $j > 2$. We use GNU Radio functions `set_iq_imbalance` and `set_dc_offset` to introduce specific levels of IQ imbalance and DC offset respectively.

After successful identification, the transmitter selects first V bits of the binary sequence output of the PN generator to select the next pair of impairments. As shown in Fig. 12, the transmitter selects $\{I_4, I_3\}$ as next pair of impairments based on the first $V = 3$ bits of the binary sequence output. The receiver also chooses the same pair of impairments for the

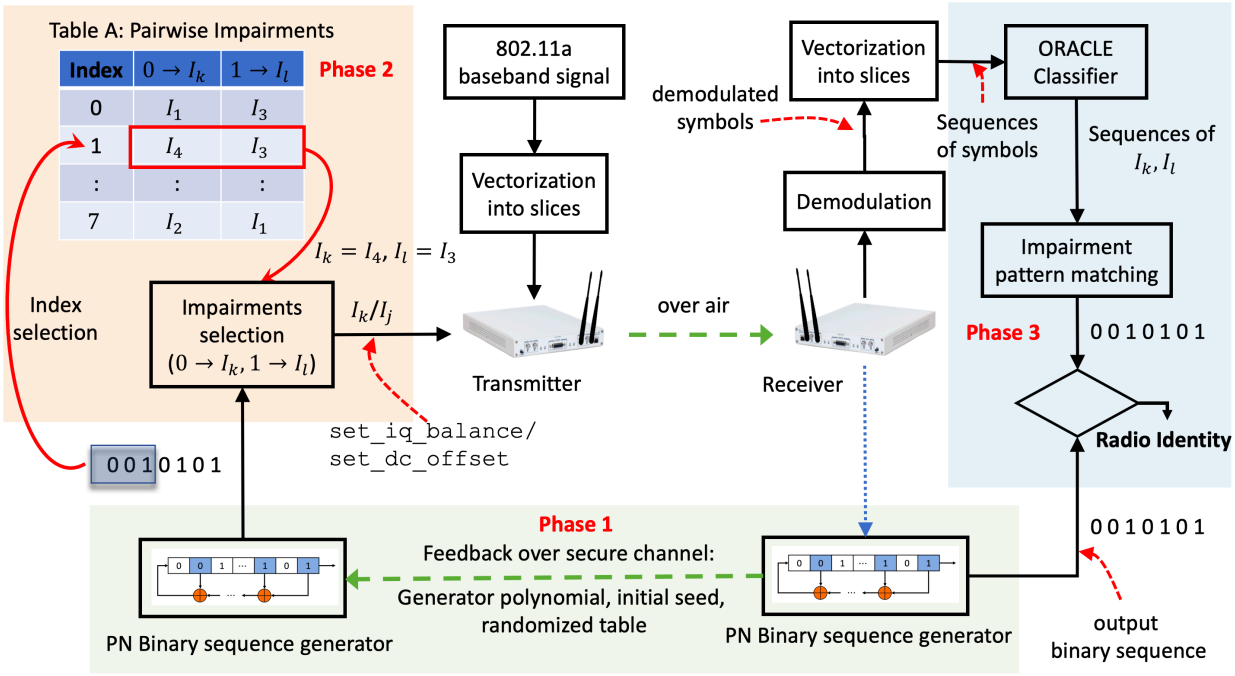


Figure 12: Illustration of radio identification using impairment hopping spread spectrum. PN generator at both transmitter and receiver produce exactly identical binary sequence '0010101', which we use as "identification key". After referring the table A, the transmitter maps the impairments $\{0 \to I_4, 1 \to I_3\}$ to convey identification key.

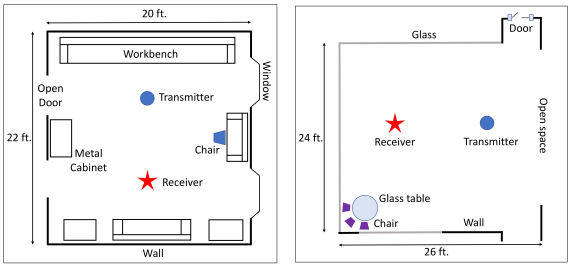
identification.

Phase 3: Transmitter identification at the receiver The receiver uses ORACLE's trained CNN classifier described in Sec. VI-C to determine the sequence of impairments used by the transmitter radio. A CNN classifier uses an input slice of demodulated symbols to get the prediction probabilities over all feasible impairments. Since the receiver knows the pair of impairments used by each transmitter radio, it uses prediction probabilities of those specific impairments $\{I_k, I_l\}$ to determine the binary output as $i = \arg\max\{p_k, p_l\}$. The receiver first synchronizes using the known preamble sequence and using fixed impairment pair $\{I_1, I_2\}$ and later uses $\{I_k, I_l\}$ to determine identification key. The receiver performs this operation for each transmitter radio separately. This binary sequence is then matched with the binary sequence output from its own PN generator specific to a particular transmitter radio to determine its identify. After successful identification (or later in time, through the mechanism of a link layer ACK), the receiver notifies the transmitter to generate a new binary sequence output for a different identification key, which also changes the pair of impairments as described in Phase 2.

The radio identification through PN sequence allows IHOP to scale to thousands of radios requiring a minimum of just two impairments whereas frequent hopping of the identification key and the pair of impairments make it hard for the adversary to find the pattern to perform spoofing attack. Thus, IHOP based radio identification is *scalable as well as secure*.

VIII. Performance Evaluation

In this section, we present the performance of ORACLE showing: (1) it increases the classification accuracy for bit-similar radios, and that accuracy is not influenced by variation in wireless channel conditions (Sec. VIII-A); (2) it achieves a near perfect identification accuracy without compromising on the security and BER performance of each radio. (Sec. VII).



(a) (b)

Figure 13: Two different experimental environments: (a) closed lab area (location 1); (b) open recreation area with much less reflections (location 2).

Experiment setup: We first identify a set S of 16 impairments which generates unique patterns as discussed in Sec. VI-B. Next, we collect demodulated data from WiFi packets that are transmitted over a cable from a single radio, after introducing these impairments through GNU Radio API. We replicate and augment demodulated data by adding a random Gaussian noise. We limit the power of noise to be under -13 dB to ensure that EMD lies below the threshold of 0.1 between patterns generated from original and altered data. Finally, we train the classifier with the augmented dataset using the same CNN architecture as described in Sec. IV.

A. Classification accuracy with different channel conditions

We test the performance of the trained CNN classifier with 16 X310 radios. To do so, we first collect samples from

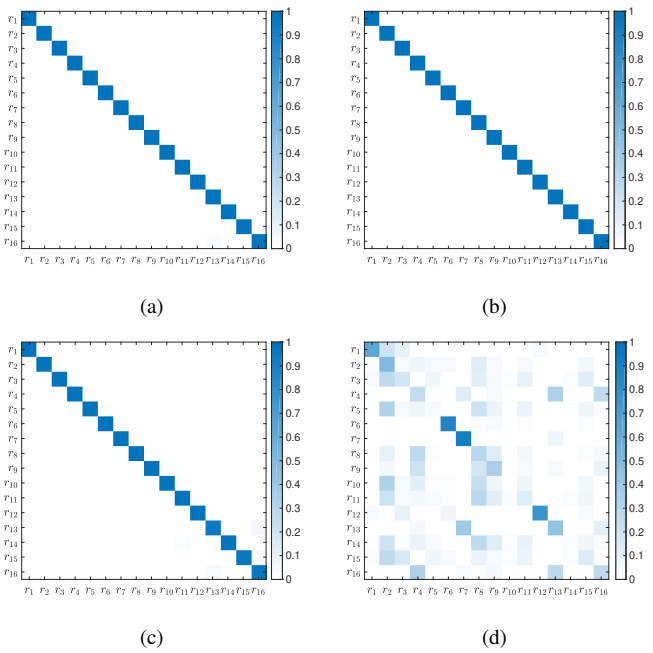


Figure 14: Classification accuracy (a) via cable; (b) over air in location 1 (Fig.13a); (c) over air in location 2 (Fig.13b). (d) shows the accuracy without ORACLE (data collected in location 2).

these radios through cable. All radios are uniquely configured with one of 16 impairments selected from set S. As shown in Fig. 14a, ORACLE easily distinguishes bit-similar radios that are intentionally introduced with unique impairments by achieving a classification accuracy of 99.76%. This indicates that our pre-trained classifier is able to identify bit-similar radios accurately when mapped to one of the hardware impairments.

Next, we evaluate the performance of ORACLE with data collected over the wireless channel. To show robustness to variation in channel conditions, we conduct the experiments in two different locations: (1) our lab, which represents a typical in-indoor environment (Fig. 13a) and (2) a more open recreation area which has fewer reflections (Fig. 13b). The confusion matrix of classification accuracy is shown in Fig. 14b and Fig. 14c respectively. In general, in both environments ORACLE can achieve higher than 99.5% accuracy, which proves that the unique patterns created by the impairments can still be detected, even with random noise.

In comparison, training the same classifier with these 16 X310 devices without any kind of artificially introduced hardware impairments results in a poor classification performance. As shown in Fig. 14d, the classification accuracy is only 35.96% for these bit-similar radios, which shows the benefits of the careful impairment allocation process.

B. Radio identification using IHOP

We evaluate the performance of our proposed IHOP based radio identification technique using the data collected over the air for 16 X310 radios in location 1. For each radio, we collect raw IQ samples for all 16 impairments. In our experimental evaluation, each radio supports all 16 impairments satisfying the BER constraint of 10^{-4}. For each transmitter radio, a receiver first randomize these 16 impairments and then find different permutations of a pair of impairments to get a total permutations ${}^{16}P_2 = 240$. It selects first $2^V = 128$ permutations where $V = \lfloor (\log_2({}^{16}P_2) \rfloor = 7$ to generate a table where each entry is a pair of impairments to be used to convey binary 0 or 1. We assume the receiver shares this table along with a unique generator polynomial, initial seed and a initial impairment pair to be used over a secure feedback channel.

Table II: A list of generator polynomial selected for a specific PN sequence length.

PN sequence length l_{pn}	Generator Polynomial
7	[7 6 0]
13	[13 12 10 9 0]
19	[19 18 17 14 0]
25	[25 22 0]
31	[31 280]

We perform 10000 trials to evaluate the performance of our proposed identification technique. In each trial, we randomly select a transmitter out of 16 X310 radios that uses a shared generator polynomial (as shown in Table II) and random initial seed to produce a PN binary sequence ('*identification key*') of length l_{pn}, that is exactly identical to a sequence generated by the receiver. The transmitter refers its own pairwise impairment table to map each binary value 0 or 1 in the identification key to the impairment as: $\{0 \rightarrow I_k, 1 \rightarrow I_l\}$ for a n_{slice} number of baseband symbols. We generate a new pairwise impairments table in each trial. We choose the value of n_{slice} same as the input slice length used by ORACLE' CNN classifier. For synchronization, the transmitter uses a fixed length preamble sequence '10101011', that is conveyed before *'identification key'* using $\{I_1, I_2\}$. The receiver uses ORACLE's trained classifier described in Sec. VI-C to obtain the sequence of impairments, which are then demapped to a binary sequence. This sequence is then matched with the output of its own PN generator to determine the success or failure in identifying the transmitter. In each trial, we repeat the identification process for the same transmitter 10 times by generating new identification key each time.

Fig. 15a shows the accuracy of our proposed IHOP based radio identification as a function PN sequence length for a slice size $n_{slice} = 128$. As the length of PN sequence increases, the identification accuracy reduces. It is because even a single inaccurate identification of the impairment I_k or I_l causes radio identification to fail which requires matching of the entire sequence. To improve the identification accuracy, we propose a simple repetition technique, in which the transmitter introduces the same impairment for $n_{rep} \times n_{slice}$ number of baseband symbols, instead of n_{slice} and where n_{rep} is the number of repetitions. As shown in figure, repetition of $n_{rep} = 3$ significantly improves the identification accuracy.

Fig. 15c shows ORACLE's classification accuracy for different length of input slice size. This is to show that input of a smaller slice size can also identify radios if we introduce artificial impairments. Fig. 15b shows the identification accuracy for PN sequence length $l_{pn} = 31$ as a function of slice size n_{slice}. It is evident that IHOP supports radio identification even with shorter WiFi packets. This also allows

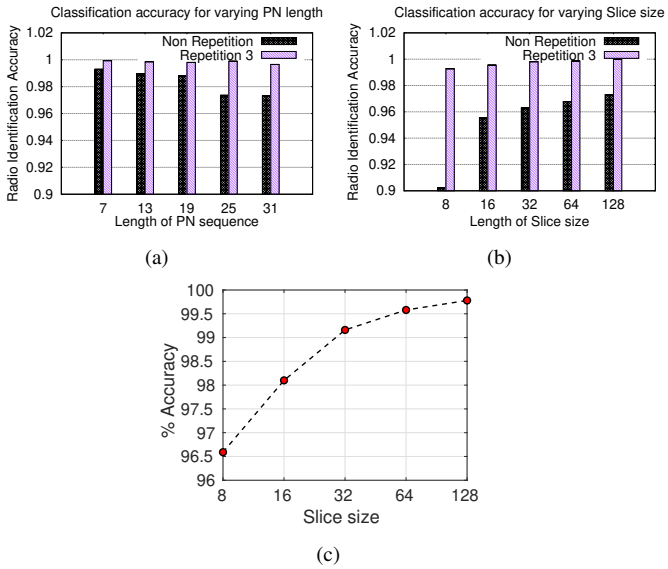


Figure 15: (a)Radio identification accuracy as a function of length of PN sequence l_{pn} for slice length of n_{slice} = 128 b) Radio identification accuracy as a function of slice length n_{slice} for a fixed PN sequence length of l_{pn} = 31 c) ORACLE's CNN classification accuracy as a function of slice length

to further improve the security by using a higher PN sequence length l_{pn} for the same size of the WiFi packet. For example, 'identification key' of l_{pn} = 31 conveyed by using n_{slice} = 8 with repetition n_{rep} = 3 achieves similar accuracy compared to that conveyed using n_{slice} = 128 with no repetition, but requires 5x times less number of symbols to represent.

IX. Conclusions

We presented ORACLE, a fingerprinting technique for identification of specific radios based on the hardware-centric features within the transmitter chain. We showed that our CNN classifier achieves an accuracy of $80 - 95\%$ using raw and feature engineering on IQ samples for $>$ 100+ COTS WiFi devices and 16 X310 USRP radios in static environment. To further improve the classification accuracy in dynamic environment, we showed how feedback-driven transmitter-side modifications can increase differentiability for bit-similar devices. Furthermore, we demonstrated a scalable and secure method to uniquely identify radios through a random binary sequence key, conveyed by hopping impairments that easily thwarts spoofing attack from an adversary. The key innovation lies in its 'train once and deploy anywhere' feature. We demonstrate experimental $> 99\%$ accuracy with bit-similar X310 radios, regardless of different channel conditions and wireless transmission environments.

Acknowledgment

This work is supported by the Defense Advanced Research Projects Agengy (DARPA) under RFMLS program contract N00164-18-R-WQ80. We are grateful to Paul Tilghman, program manager at DARPA, and Esko Jaska for their insightful comments and suggestions.

References

[1] Cisco Systems, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021 White Paper," http://tinyurl.com/zzo6766, 2017.

[2] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, Privacy and Trust in Internet of Things: The Road Ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[3] B. Danev, D. Zanetti, and S. Capkun, "On physical-layer identification of wireless devices," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, p. 6, 2012.

[4] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.

[5] T. J. O'Shea and J. Corgan, "Convolutional radio modulation recognition networks," 2016. [Online]. Available: http://arxiv.org/abs/1602.04105

[6] A. Selim, F. Paisana, J. A. Arokkiam, Y. Zhang, L. Doyle, and L. A. DaSilva, "Spectrum monitoring for radar bands using deep convolutional neural networks," in *IEEE GLOBECOM 2017*.

[7] K. B. Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," in *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*. IEEE, 2007, pp. 331–340.

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

[9] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, "Device fingerprinting to enhance wireless security using nonparametric bayesian method," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1404–1412.

[10] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Fingerprinting wi-fi devices using software defined radios," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2016, pp. 3–14.

[11] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid RF fingerprint extraction and device classification scheme," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[12] F. Xie, H. Wen, Y. Li, S. Chen, L. Hu, Y. Chen, and H. Song, "Optimized coherent integration-based radio frequency fingerprinting in internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3967–3977, Oct 2018.

[13] Y. Xing, A. Hu, J. Zhang, L. Peng, and G. Li, "On radio frequency fingerprint identification for dsss systems in low snr scenarios," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2326–2329, Nov 2018.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[15] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[16] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Feb 2018.

[17] F. Restuccia and T. Melodia, "Big data goes small: Real-time spectrum driven embedded wireless networking through deep learning in the rf loop," in *To appear in Proceedings of IEEE INFOCOM 2019. Available at https://tinyurl.com/RFLearnINFOCOM*, 2019.

[18] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Communications*, vol. 14, no. 11, pp. 92–111, 2017.

[19] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *arXiv preprint arXiv:1803.04311*, 2018.

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

[21] X. W. Wang, Xuyu and S. Mao, "Rf sensing in the internet of things: A general deep learning framework," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 62–67, 2018.

[22] X. W. et al., "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Vehic. Tech.*, vol. 66, no. 1, p. 763–76, 2017.

[23] C. Y. X. Wang and S. Mao, "Phasebeat: Exploiting csi phase data for vital sign monitoring with commodity wifi devices," in *IEEE ICDCS '17*, 2017, p. 1230–39.

[24] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive internet of things systems," *IEEE Transactions on Communications*, 2018.

[25] T. J. O'Shea and J. Hoydis, "An introduction to machine learning communications systems," 2017. [Online]. Available: http://arxiv.org/abs/1702.00832

[26] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, Sept 2018.

[27] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks," in *Proceedings of IEEE INFOCOM 2019*, 2019.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS 2012*.

[29] E. Sourour, H. El-Ghoroury, and D. McNeill, "Frequency offset estimation and correction in the ieee 802.11 a wlan," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 7. IEEE, 2004, pp. 4923–4927.

[30] Defense Advanced Research Projects Agency (DARPA), "The Radio Frequency Spectrum + Machine Learning = A New Wave in Radio Technology," https://www.darpa.mil/news-events/2017-08-11a, 2017.