Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting

Amani Al-Shawabka*, Francesco Restuccia*, Salvatore D'Oro, Tong Jian, Bruno Costa Rendon, Nasim Soltani, Jennifer Dy, Kaushik Chowdhury, Stratis Ioannidis and Tommaso Melodia
Institute for the Wireless Internet of Things
Northeastern University, Boston, MA 02115 USA
*Corresponding Authors: {a.al-shawabka, frestuc}@northeastern.edu

***Abstract*—Radio fingerprinting uniquely identifies wireless devices by leveraging tiny hardware-level imperfections inevitably present in off-the-shelf radio circuitry. This way, devices can be directly identified at the physical layer by analyzing the unprocessed received waveform – thus avoiding energy-expensive upper-layer cryptography that resource-challenged embedded devices may not be able to afford. Recent advances have proven that convolutional neural networks (CNNs) – thanks to their multidimensional mappings – can achieve fingerprinting accuracy levels impossible to achieve by traditional low-dimensional algorithms. The same research, however, has also suggested that the wireless channel may negatively impact the accuracy of CNN-based radio fingerprinting algorithms by making device-unique hardware imperfections much harder to recognize.**

In spite of the growing interest in radio fingerprinting research by academia and DARPA, the wireless research community still lacks (i) a large-scale open dataset for radio fingerprinting collected in diverse environments and rich, diverse, channel conditions; and (ii) a full-fledged, systematic, quantitative investigation of the impact of the wireless channel on the accuracy of CNN-based radio fingerprinting algorithms. The key contribution of this paper is to bridge this gap by (i) collecting and sharing with the community more than 7TB of wireless data obtained from 20 wireless devices with identical RF circuitry (and thus, worst-case scenario for fingerprinting) over the course of several days in (a) an anechoic chamber, (b) in-the-wild testbed, and (c) with cable connections; and (ii) providing a first-of-its-kind evaluation of the impact of the wireless channel on CNN-based fingerprinting algorithms through (a) the 7TB experimental dataset and (b) a 400GB dataset provided by DARPA containing hundreds of thousands of transmissions from thousands of WiFi and ADS-B devices with different SNR conditions. Experimental results conclude that (i) the wireless channel impacts the classification accuracy significantly, *i.e.*, from 85% to 9% and from 30% to 17% in the experimental and DARPA dataset, respectively; and that (ii) equalizing I/Q data can increase the accuracy to a significant extent (*i.e.*, by up to 23%) when the number of devices increases significantly.

I. Introduction and Motivation

The Internet of Things (IoT) will soon realize our long-standing vision where tiny embedded wireless devices are deployed just about everywhere – around us [1] and also inside us [2]. The never-before-seen pervasiveness of IoT devices – coupled with their peculiarly resource-constrained nature – ultimately makes the design of low-power identification techniques a compelling necessity [3]. Regrettably, legacy radio identification techniques mostly rely on energy-hungry cryptographic techniques such as private- or public-key cryptography, which operate at the MAC layer and above and thus require extra computational resources – making them hardly suitable for devices possessing CPUs with a handful of megahertz and memory in the order of hundreds of kilobytes.

Thanks to its proven efficacy, *radio fingerprinting* has recently received significant interest from both academia and government [4–11]. In short, radio fingerprinting uses waveform-level imperfections imposed by the radio-frequency (RF) circuitry to obtain a "fingerprint" of the wireless device, which cannot be imitated by adversarial devices. These imperfections include I/Q imbalance, phase noise, frequency offset, sampling offset, and phase noise, among others [12]. Moreover, by operating at the physical layer, radio fingerprinting has a very low impact on the device's resources. To further attest to its crucial importance, the Defense Advanced Research Projects Agency (DARPA) has sponsored the radio-frequency machine learning systems (RFMLS) program [11], where the main task is to learn to recognize a specific transmitter based on the RF hardware imperfections.

Recently, convolutional neural networks (CNNs) have been proposed to fingerprint radios though deep learning of the hardware impairments [13, 14]. CNNs are becoming more and more popular in the wireless community [3, 15–18], thanks to their ability to avoid manual – and thus, significantly cumbersome and necessarily sub-optimal – feature extraction techniques, such as Zigbee's O-QPSK modulation [8] or the WiFi training symbols [5, 7]. Another downside is that the derived algorithm is highly protocol-dependent and therefore not entirely applicable to general waveforms. Conversely, by taking as input "raw" (*i.e.*, unprocessed) I/Q samples, CNNs can fingerprint wireless devices using *any* wireless technology of choice. This key aspect makes deep learning-based radio fingerprinting particularly desirable for the IoT, where different wireless technologies co-exist [19].

Cutting-edge advances, however, have hinted that the non-stationary, dynamic and unpredictable effect of the wireless channel may cause the accuracy of the CNN to plummet (*i.e.*, from 98% to 71%) when tested with a waveform collected after a number of days the CNN was trained [16]. This is because, at their core, CNNs make the somewhat strong assumption that the input data is (i) time-stationary, *i.e.*, the same input does not change over time; and (ii) inputs are drawn from independent and identically distributed (i.i.d.) random variables. However, this may not always be the case for the wireless environment, where (i) hardware impairments may change over time due to unpredictable phenomena such as temperature and voltage oscillations; and (ii) the wireless channel is convolved with the transmitted waveform with time-

varying parameters dependent on the current level of fading and noise – which can only be estimated in real-time through packet-level pilot symbols. To address the issue, the authors in [16] point out that since retraining a CNN in real-time may be prohibitive, a dynamic, channel- and device-tailored strategy must be devised. To this end, they show that a carefully-tailored finite input response (FIR) applied at the transmitter's side can be used to partially bring back the accuracy to an acceptable level.

What is Missing Today

Despite the recent rush of research activity on radio fingerprinting, there are still a number of stark and fundamental questions that need to be addressed. This is not at all surprising – the highly non-linear behavior of CNNs, joint with the non-stationary nature of impairments, fading and noise, make understanding the impact of the wireless channel a formidable challenge. The towering issue, among others, is gaining insight on whether the CNN is learning (i) only the device's hardware impairments or (ii) only the channel or (iii) a combination of impairments and channel. Moreover, we still do not know (i) if channel equalization can be truly beneficial to the learning process of the CNN, and (ii) what is the best environment to train and test the CNN algorithms. It is clear that without a full-scale experimental investigation, any advance in the field will necessarily be plagued by the following question: "is the action of the wireless channel impairing the learning, or is my model too small and thus underfitting?"

Moreover, it is a matter of fact that groundbreaking innovation in this critical field has been so far severely stymied by the lack of large-scale waveform datasets providing a common benchmark for researchers working in the field. Without a common waveform dataset collected under rich and diverse wireless conditions, it is indeed unavoidable that every paper on radio fingerprinting will claim to be "better than the previous ones." This does not happen in more mature and widely-explored learning domains such as computer vision and natural language processing, where massive-scale labeled datasets such as MNIST for image classification [20] and IMDb for sentiment analysis [21] have been completely available to the research community for many years. Unfortunately, as far as we know, existing work on radio fingerprinting has so far refrained from releasing its testing data to the community, thus creating a fundamental need for data that is yet to be fulfilled.

Technical Contributions

The paper's key contribution is to report the largest data collection campaign and experimental evaluation ever conducted to evaluate radio fingerprinting algorithms. Specifically, we make the following technical contributions:

• We conduct a massive data collection campaign aimed at evaluating the impact of the wireless channel on CNN-based radio fingerprinting algorithms. We first consider the *worst-case* scenario for radio fingerprinting, *i.e.*, devices with the same RF circuitry. To this end, we collect data in the following scenarios, described in Section IV: **(i)** a 20-radio "in-the-wild" setup where nominally-identical USRP devices transmit the same baseband signal to a single receiver. We investigate the case where the devices **(a)** use the same antenna and distance from the receiver, and **(b)** use different antennas and distance; **(ii)** a 10-radio setup where we use a 50x50x22ft anechoic chamber to study radio fingerprinting in optimal RF conditions; and **(iii)** a 20-radio testbed where we collect data using an RF cable to collect the waveforms from the USRP devices (Section V). To the best of our knowledge, no such data has ever been collected and made available before, making our experimental results and the related dataset first-of-its-kind and thus extremely valuable;

• Through a large-scale dataset provided by DARPA[1], we analyze the fingerprinting performance on 10,000 WiFi and ADS-B [22] devices emitting more than 300,000 transmissions (Section VI). We analyze the impact of the number of devices and the signal-to-noise ratio (SNR) level on the performance. As far as we know, no previous paper has ever reported fingerprinting performance for thousands of devices;

• Experimental results on both testbed and dataset unequivocally conclude that (i) the wireless channel impacts on the classification accuracy significantly, *i.e.*, from 85% to 9% and from 30% to 17% in the experimental and DARPA dataset, respectively; and that (ii) equalizing I/Q data can increase the accuracy to a significant extent (*i.e.*, by up to 23%) when the number of devices increases significantly.

II. Related Work

The main focus of early work on radio fingerprinting has been devising hand-tailored feature extraction techniques [5–10, 23]. Nguyen *et al.* [6] propose a non-parametric Bayesian method to detect the number of devices through device-dependent channel-invariant radio-metrics, however, the effectiveness of the methodology is tested on 4 ZigBee nodes only. Brik *et al.* [5] consider a large (i.e., 130 devices) WiFi testbed, and through carefully-tailored transients and offset-based features show that 99% accuracy can be achieved. On the other hand, Brik *et al.* only consider experiments in an anechoic chamber, and thus the effect of the channel is not considered. Conversely, [7] evaluates, on an in-the-wild WiFi testbed, several feature-based algorithms based on the WiFi scrambling seed, frequency offset and transients, achieving accuracy up to 50% on about 100 devices. Peng *et al.* [8] devise features based on the ZigBee's PSK constellation to fingerprint 54 radios with about 95% accuracy. Recently, Zheng *et al.* [4] proposed an $\mathcal{F}(\cdot)$ function to model a device's modulation and timing errors, frequency offsets and power amplifier noise, and show that high-accuracy can be achieved on a series of 33 devices. However, the multipath evaluation in [4] is somewhat limited, and thus it is not definitely clear whether the algorithm is indeed learning the impairments or learning the channel. Moreover, Zheng *et al.* state that ".. if we want to identify WiFi signals whose symbol rate is 20M/s in multipath environments, channel estimation and deconvolution is an indispensable step."

Recent work has demonstrated that deep learning can be successfully used to fingerprint wireless devices with high

[1]Unfortunately this dataset cannot be released to the community due to contract obligations. We hope this will change in the future.

accuracy [13, 14, 16, 24, 25]. Merchant *et al.* [24] and Das *et al.* [25] leverage CNN and RNNs to achieve respectively 92% accuracy on a testbed of 7 ZigBee devices and 90% on 30 LoRa devices. However, the effect of the channel on the performance is not studied. The works in [13, 14] are the first to explicitly evaluate the impact of impairments on the performance of CNN-based fingerpriting algorithms, and propose the introduction of artificial impairments to improve the accuracy. However, the approaches in [13, 14] do not show how the receiver can accurately compensate the introduced impairments, and it is not clear how to connect the increase in accuracy and the introduction of the hardware impairments. Gopalakrishnan *et al.* [26] explore the use of complex-valued CNNs for radio fingerprinting.

The closest work to ours is [16], where Restuccia *et al.* investigate the wireless channel issue in radio fingerprinting, and propose to compensate the wireless channel by developing a FIR filter that applied at the transmitter's side can compensate the distortion in the received I/Q samples. However, the authors do not investigate what the CNN is actually learning, or how the channel conditions affect the classification results. In this work, for the first time, we focus on evaluating the channel (wireless and wired) impact in real-world experimental environments (in-the-wild and anechoic chambers) and through a large-scale dataset.

III. Learning and Data Collection methodology

We first provide some background on CNNs in Section III-A, then describe our CNN architectures in Section III-B, and finally the performance metrics in Section III-C.

A. Background on Convolutional Neural Networks (CNNs)

CNNs have found tremendous success in the computer vision and language processing domains [27–29]. In the wireless domain, we have seen groundbreaking advances in CNN-based modulation classification [15, 30, 31] and radio fingerprinting [13, 14, 16]. Differently from image-based domains, and due to its immediate applicability to I/Q data, we consider CNNs with one-dimensional convolutions (1D) instead of two-dimensional (2D) convolutions. Conv1D networks, indeed, perform extremely well in locating patterns (in our case, hardware imperfections) regardless of where the imperfection actually starts. This property is called *shift-invariance* of neural networks.

The core component of a CNN is the convolution layer, which in a nutshell convolves a number of equally-sized kernels with the input. The *stride* parameter controls the step size of the kernel across the input. The output of the convolution layer is a vector of feature maps with decreased dimension compared to the original input. More formally, a 1D convolution layer is composed of F filters $\mathbf{K}_n, 1 \leq n \leq F$. By defining D and W as the depth and the width of the kernel, each filter generates a mapping $\mathbf{Y}^n \in \mathbb{R}^{F-W+1}$ from an input $\mathbf{X} \in \mathbb{R}^{D \times L}$ as follows:

$$\mathbf{Y}_j^n = \sum_{x=0}^{L-W} K_{j-x}^n \cdot \mathbf{X}_{n,x} \tag{1}$$

where L is the length of the input.

By choosing the depth of the kernel as $D = 2$, we force each kernel to jointly learn a pattern across the I and Q components of the received waveform. To introduce non-linearity in the CNN, we use the popular rectified linear units (ReLU) activation functions. Since the ReLU sets negative values to zero, the output becomes sparser than other activation functions, which provides robustness to noise. We also use a maximum pooling (MaxPool) layer to reduce the number of parameters. What the MaxPool layer does is to aggregate all values in a pool by using the max operator. Finally, we use dense layers to generate high-level features after the convolution layers, and one classifier layer with as many outputs as the number of classes and using the softmax activation function. Given a weight matrix $\mathbf{W} \in \mathbb{R}^{M \times N}$ and a bias vector $\mathbf{b} \in \mathbb{R}^{1 \times N}$, each dense layer generates an output $\mathbf{Y}^n \in \mathbb{R}^{1 \times N}$ from an input $\mathbf{X} \in \mathbb{R}^{1 \times M}$ through the following:

$$\mathbf{Y} = \mathbf{W} \cdot \mathbf{X} + \mathbf{b} \tag{2}$$

As in the case of the convolution layer, non-linearities can be introduced by the ReLU activation. By using the categorical cross-entropy as loss function, the output of the network is the probability of each class.

B. CNN Architectures and Inputs

Figure 1 shows the three different CNN architectures we use to understand to what extent model architecture can mitigate the channel impact. The smallest model is called *Homegrown* and is shown in Figure 1(a). It is a very simple architecture composed by two one-dimensional (1D) convolutional layers (ConvLayers), each operating with 50 filters – of size 1x7 and 2x7 for the first and second ConvLayer, respectively – and rectified linear units (ReLU) as activation function. The output of the ConvLayers is then fed to two dense layers of 256, and 80 neurons, respectively. Finally, a softmax layer is used to get the classification probability. A dropout of rate 0.5 is also used to reduce overfitting.

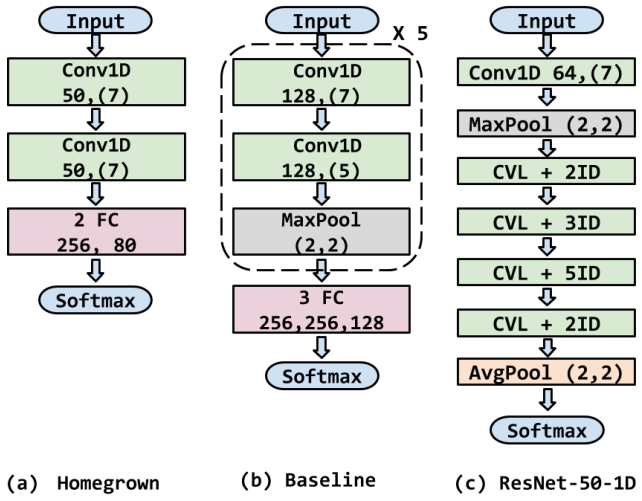

Figure 1. The CNN architectures used in this Paper.

While we show that increasing the CNN depth leads to a significant increase in fingerprinting accuracy, it also inevitably

slows the training time. For this reason, we utilized deeper CNN architectures to study whether such architectures can tame the channel impact and increase fingerprinting accuracy in the larger DARPA dataset.

The *Baseline* model, shown in Figure 1(b), is a modified version of the well-known AlexNet CNN [**?**] and consists of two one-dimensional (1D) convolution layers followed by ReLU activations and a 2x2 MaxPooling layer to decrease overfitting. This sequence is repeated five times and the output is then fed to three fully connected layers of 256, 256, and 128 neurons, respectively.

Another architecture we consider in our investigation is the *ResNet-50-1D* model shown in Figure 1(c), which is inspired from the well-known residual network (ResNet) architecture [28]. Similarly to it 2D version, *ResNet-50-1D* consists of 50 layers which have been converted into 1D to reduce the overall complexity of the architecture. The convolution block (CVL) and identity block (ID) in Figure 1(c) both consist of three Conv1D layers with 64, 64, and 256 filters, respectively. Each of these layers has a filter size of (1x1), (1x3), and (1x1), respectively. All three models were implemented in Keras with TensorFlow backend.

A major challenge in applying CNNs to the wireless domain is that they are restricted to operate on inputs with fixed length. However, wireless transmissions are obviously variable in size. For this reason, we use a *sliding window* approach to cut each transmission (in both training and test set) as shown in Figure. 2. Specifically, given an I/Q sequence k of length M_k, we generate $M_k - j + 1$ *slices* of length j by sliding a window with stride 1. Each slice is then labeled individually, receiving the ID of the device that generated the transmission as a label. We do not use all the generated slices, but only $\frac{M_k}{j}\kappa$ slices uniformly at random from each transmission, where $1 \leq \kappa \leq j$. Since an I/Q sample appears in approximately $\kappa \in [0, j]$ slices, we refer to κ as the *replication factor*.

Slicing and randomization have several advantages. First, they ensure we can always train our models on fixed input sizes. Second, they enhance the shift invariance of the features learned by the network – indeed, RF imperfections can manifest anywhere in a slice, and randomizing the slice origin enforces this invariance in the data. Third, the slices belonging to an entire transmission can be used to *boost* the classification accuracy, for example, by taking the majority label. Finally, slicing allows us to avoid varying length input issues like vanishing gradients.

C. Performance Metrics

As per usual machine learning common practice, we use a training and a testing dataset to evaluate the model's learning. To assess the performance of our models we use the following performance metrics:

- ***"Per-slice" Accuracy (PSA)***, the number of correctly predicted slices over the total number of tested slices from the testing dataset;
- ***"Train-and-Test-One-Day" Accuracy (TTOD)***, the PSA over a testing dataset consisting of slices collected during the same day of that used in the training process;

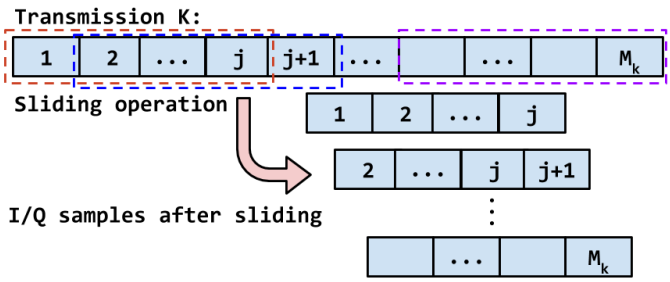

Figure 2. Given a time-series of length M_k, we create $M_k - j + 1$ subsequences of length j by sliding a window of length j over the larger sequence (or stream) of I/Q samples (with stride 1). This leads to inputs of a fixed length, but also enhances shift invariance.

- ***"Train-One-Day-Test-Another" Accuracy (TDTA)***, the PSA over a testing dataset consisting of slices collected during a different day of that used in the training process;
- ***"Per-Transmission" Accuracy*** *(PTA)*, which reports the CNN accuracy in predicting to which device a transmission belongs to. Specifically, each data transmission is divided into a set of consecutive slices with predefined slice length. This set of slices is fed to a pre-trained model that individually classifies every slice in the set and map it to the predicted device, which is the device that got the highest prediction probability over all tested slices. For example, let us consider the case the total number of labeled devices is equal to K, and for each device we collected $[1, 2, ..., n, ...N]$ transmissions. Each transmission is divided into S slices, each with length L. For each slice j belonging to transmission n, the per-slice prediction probability is equal to $\sum_{j=1}^{S} P_{x,j}$, where $P_{x,j}$ is the predicted probability that slice j belongs to device x. The predicted device $\hat{x}$ is thus computed as follows:

$$\hat{x} = \arg\max_{x} \sum_{j=1}^{S_n} P_{x,j} \tag{3}$$

D. Data Collection and Experimental Dataset: An Overview

As in most of previous radio fingerprinting work [5, 7, 13, 14, 16], we chose to collect I/Q samples transmitted using the IEEE 802.11 a/g (WiFi) standard [32], which is arguably the most common wireless communication used nowadays. The WiFi standard uses orthogonal frequency division multiplexing (OFDM) and thus multiple subcarriers to transmit each digital symbol, which can be modulated using BPSK, QPSK, 16QAM or 64QAM, with different levels of convolutional coding (1/2 or 3/4). A short training sequence (STS) is used for frame detection, while a long training sequence (LTS) is used to perform time synchronization. Then, a fast Fourier transform (FFT) is used to bring the I/Q samples back to the frequency domain from the time domain and thus recover the transmitted digital symbols. Finally, the LTS is again used to perform channel estimation and offset correction. The equalized I/Q samples are then decoded and sent to higher layers.

Figure 3 summarizes the I/Q data collection procedure. To analyze the effect of the channel at different stages of the WiFi demodulation process, we collect the following I/Q samples: (i) *Raw I/Q before FFT*; (ii) *Raw I/Q after*

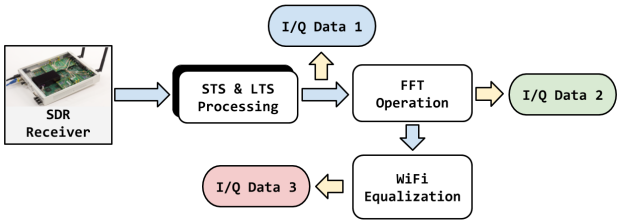

Figure 3. Architecture of collected I/Q data.

FFT; and (iii) *Equalized I/Q*. We use the Gnuradio WiFi implementation by Bloessl *et al.* [33] to both transmit and collect I/Q samples. This allowed us to have perfect control of each WiFi demodulation stage and thus ease significantly the data collection process. Samples are streamed at 2.432 GHz (5-th WiFi Channel), at sampling rate of 20 MS/s and with BPSK 1/2 as modulation and coding scheme. Each transmitter sends exactly the same WiFi frame over and over again. The samples are also received at 20 MS/s.

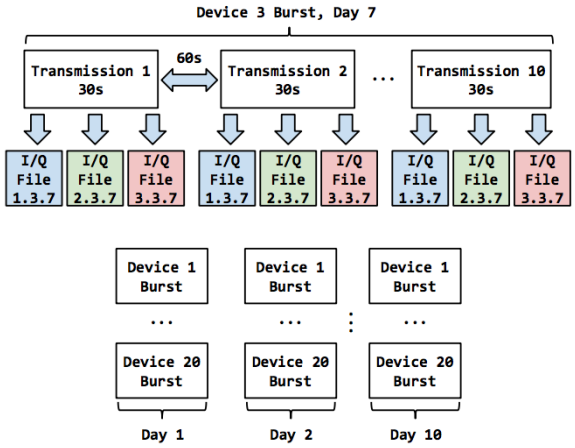

Figure 4. Organization of Testbed Dataset.

Figure 4 shows the structure of our data collection process and the generated files. We define as *device burst* a set of 10 different transmissions of about 30 seconds each, with each transmission spaced in time approximately 1 minute from each other. During each recording day, we collect a burst from each device, for 10 days. Each device burst generates 3 files, as shown in Figure 3. During each transmission, we append the I/Q samples to the appropriate file. Each device burst generates approximately 26GB of data.

We label each of the 3 files using the *Signal Metadata Format* (SigMF) [34], which describes sets of recorded digital signal samples with metadata written in plain-text JSON. We used SigMF since it is widely used yet simple format that can be parsed by any JSON parser in any programming language. By using SigMF, one recording consists of (i) the binary file containing the I/Q samples, and (ii) a metadata file containing general information about the recorded I/Q samples. In our case, we store in the metadata file information regarding (i) the sampling rate, (ii) time and day of recording, and (iii) testbed type (described in Section IV), among others. The dataset is available on the Institute for the Wireless Internet of Things (WIOT) website, *https://northeastern.edu/wiot*.

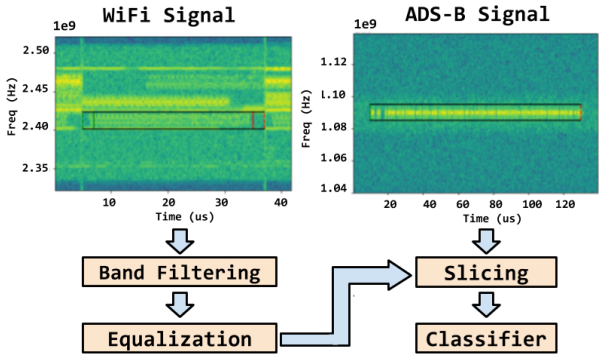

Figure 5. Pre-processing and Classification Pipeline. For WiFi transmissions (left), we perform band filtering (i.e., extracting the signal shown by the bounding box). The sliding window creates multiple slices that are fed to the classifier for training and testing.

E. DARPA Dataset: An Overview

The dataset contains both WiFi and ADS-B transmissions. The WiFi portion of the dataset contains 5117 devices and 166 transmissions on average for each device. A spectrum analyzer is used to record each transmission, operating at center frequency of either 2.4 GHz or 5.8 GHz with sampling rate 200 MS/s. Each recording consists on average of 18686 I/Q samples. The ADS-B dataset contains 5000 devices and 76 transmissions on average for each device. Each transmission is recorded at the center frequency of 1.09 GHz with sampling rate 100 MS/s. Each recording on average consists of 9156 IQ samples. Due to the high sampling rate, WiFi signals in the DARPA dataset include multiple devices transmitting simultaneously on multiple bands. Thus, as shown in Figure. 5, we move the signal to baseband and apply a low pass filter to remove any interference or noise generated out of band.

IV. Experimental Testbeds and Setups

The "in-the-wild" testbed is shown in Figure 6 and consists of an 8x8 VERT2450 antenna grid testbed covering an indoor area of 6000 square ft. The 64 antennas are connected to 24 USRPs controlled by 12 host servers. The USRPs are synchronized both in phase and frequency through four National Instruments OctoClock clock distributors and are connected to the antennas mounted on ceiling rails through 100ft-long low-attenuation coaxial cables. The radio rack is connected to the server rack through 10 Gigabit/s Ethernet cables. The server rack includes 12 Dell PowerEdge R340 running Ubuntu 16.04 LTS, where each server controls a subset of the USRPs only.

This testbed is an open space area where the antennas are located in the laboratory ceiling as shown in Figure

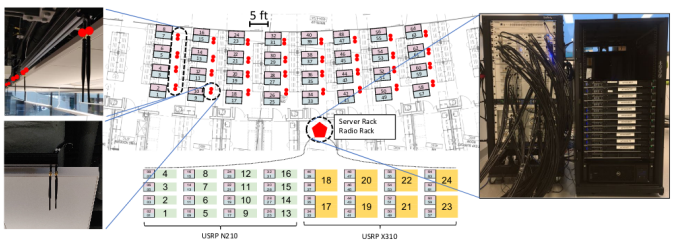

Figure 6. "In-the-wild" Experimental Testbed.

6. The experiments in this testbed are conducted using 20 SDR (software define radios), composed of 13 N210 and 7 X310 USRPs, each equipped with a 1200-6000 MHz CBX daughterboard with 40 MHz instantaneous bandwidth. The receiver is an N210 also equipped with a CBX. The testbed presents heterogeneous obstacles, resulting in a challenging scenario with rich multipath, external interference, human mobility, uncontrolled electrical hum, among others.

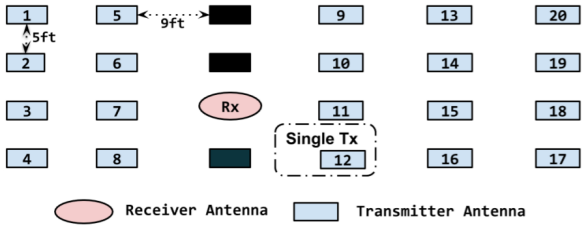

Figure 7. Non-anechoic chamber antenna setup

Figure 8 shows the anechoic chamber used for our data collection. This chamber measures 50 feet by 50 feet with 22 feet of headroom, designed to absorb unwanted radio frequency waves by lining the chamber with hundreds of blue foam protruding arrowheads. Using this isolated environment can enhance our understanding of the channel impact since (i) external RF activity does not affect the ongoing transmission inside the chamber, thanks to the external Faraday cage; and (ii) the cones deployed in the chamber absorb signals generated internally, preventing multipath.

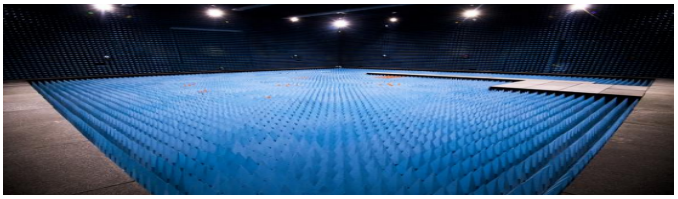

Figure 8. Anechoic Chamber Testbed.

We use four different experimental setups:

- **Setup A - In-the-Wild, Different Antennas:** devices are connected to dedicated antennas (one per SDR) deployed as shown in Figure 7. In this setup, not only do the SDRs differ in their hardware impairments, but also in their antenna, distance from the receiver, and experienced multi-path. The data collection process under this setup is repeated for ten days;
- **Setup B - In-the-Wild, Single Antenna:** all devices are connected to the same antenna, as indicated in Figure 7. This way, all devices are equally distant from the receiver and experience similar channel and multi-path conditions. Data collection is repeated for two days; I/Q samples collected during the first day are used to train our models. Instead, data collected on the second day are used for testing purposes;
- **Setup C - Wired Connection**: each transmitter is connected to the receiver with a coaxial RF SMA cable and a 5db attenuator. We use the same cable and attenuator to connect all SDRs to the receiver one at a time. By using wired connections, in this setup all transmitters are not affected by multi-path and experience exactly the same channel conditions. similarly to Setup B, data collection is repeated for two days, one for training and one for testing purposes;
- **Setup D - Anechoic Chamber, Single Antenna:** device are located in the anechoic chamber and are connected to the same transmitting antenna. Data collection has been performed for one day only.

V. Experiment Results

A. Setup A: In-the-Wild, Different Antennas

To evaluate the train-and-test-one-day (TTOD) and the train-one-day-test-another (TOTA) metrics defined in Section III-C, Figure 9 shows the 10x10 confusion matrices (CMs) representing the average per-slice accuracy (PSA) of 20 devices over 10 training days using 10 testing days. The Y-axis represents the training day while the X-axis represents the testing day. To train our CNNs, we generated 100k slices from each device, as defined in Figure 4. For our testbed experiments, we used a slice size of $L = 288$ I/Q samples, which corresponds to 6 OFDM symbols each containing 48 payload I/Q samples (6x48= 288 I/Q samples). The set of 100k slices is partitioned as follows: 70% for training, 20% for validation, and 10% for testing our models.

Figure 9 shows unequivocally that the CNN trained with I/Q samples collected one day is not able to generalize to I/Q samples collected in a different day. That is, the CNN learns to distinguish the devices based on the channel only. Indeed, different noise, fading, multipath and interference conditions establish a unique channel condition for each day. In all the cases, the TDTA accuracy drops to 5% (*i.e.*, 1/20, corresponding to random guess), while the TTOD accuracy remains close to 100%. Overall, the average PSA is pretty poor: 14.5%, 5% and 14.8%, respectively. This suggest that the CNN is "overfitting to the channel", meaning that the CNN is using the unique channel conditions of each radio, and not its hardware impairments, as the feature to discriminate them.

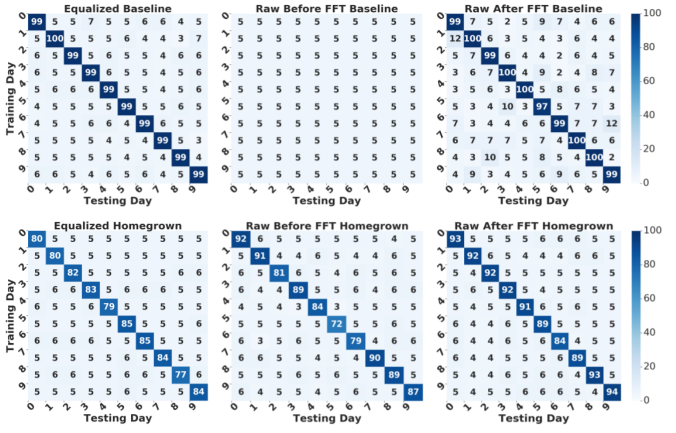

Figure 9. **"TDTA"** analysis in Setup A with (top) the *Baseline* and (bottom) *Homegrown* CNNs. If X-axis (Testing day) is the same as Y-axis (Training Day) then it becomes **"TTOD"**. From left to right: (a) Equalized (b) Raw I/Q before FFT, (c) Raw I/Q after FFT.

Interestingly enough, Figure 9 also suggests that there is no difference between using raw-after-FFT samples and equalized

I/Q samples, since they both show poor TDTA accuracy. Conversely, the DARPA dataset results shown in Section VI indicate that when the CNN is trained on one day and tested on another, the equalized I/Q samples show better performance. This difference can be explained as follows – in our testbed experiments, the devices are identical in RF circuitry (*i.e.*, same RF daughterboard). This arguably corresponds to the worst case for radio fingerprinting. On the contrary, in the DARPA datasets the devices do not have identical RF chips. In this latter case, therefore, the impairments between the different devices – which are brought to light by the channel equalization process – are more evident. Indeed, impairments of RF circuitry produced by the same manufacturer are similar. Therefore, they are not different enough to serve as a good feature to distinguish among different radios.

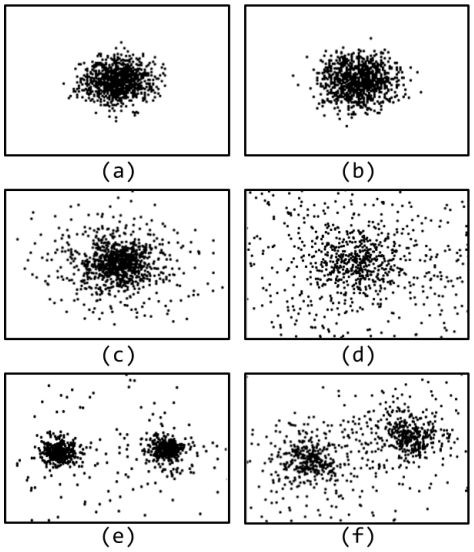

Figure 10. I/Q constellation in Setup A for devices D1 and D2; (a) D1 before FFT, (b) D2 before FFT, (c) D1 after FFT, (d) D2 after FFT, (e) D1 Equalized, (f) D2 Equalized.

Another fundamental result brought to light by Figure 9 is that the raw-before-FFT I/Q samples are not sufficient to obtain acceptable levels of accuracy. Recall that these symbols are in the time domain and therefore, I/Q samples assigned to multiple subcarriers are summed with each other. Therefore, here the I/Q samples do not contain any useful data. Differently, the raw-after-FFT I/Q samples represent the payload symbols we are sending (in our case, BPSK) – thus, the resulting I/Q samples are much more distinguishable from each other and the effect of the channel is more evident to the CNN to learn. Figure 10 shows the I/Q constellations for two devices in the testbed, and confirms the intuition above. Indeed, it shows that (i) the raw-before-FFT data "collapses" around zero in both cases and is thus indistinguishable between the two devices; (ii) the raw-after-FFT I/Q data still does not show a clear constellation but the channel impact is more evident in both cases; (iii) the equalized I/Q data shows clearly the effect of the channel and the impairments.

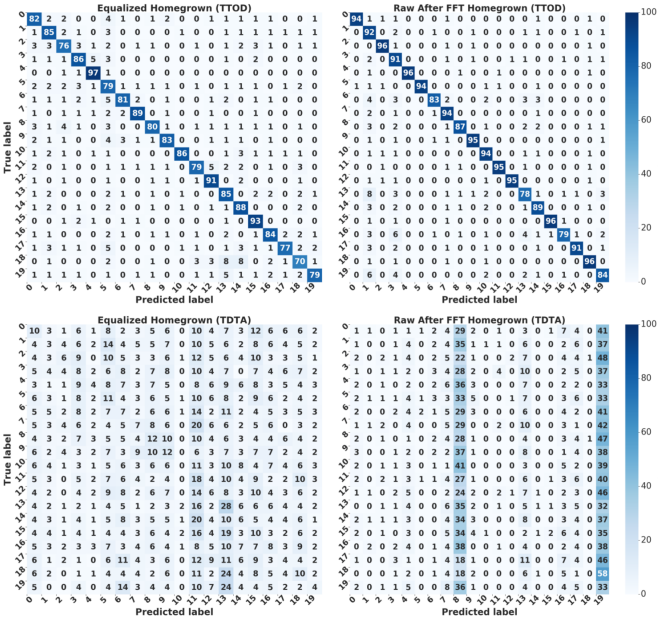

Figure 11. **"Setup B – In-the-Wild, Single Antenna"**. Top, from left to right: (a) Equalized I/Q **TTOD**, (b) Raw I/Q after FFT **TTOD**. Bottom, from left to right: (c) Equalized I/Q **TDTA**, (d) Raw I/Q after FFT **TDTA**.

B. Setup B: Same Antenna and Setup C: Wired

The top and bottom sides of Figure 11 show respectively the TTOD and the TDTA results obtained on Setup B (Same Antenna) and with the *Homegrown* CNN. We plot the same quantities in Figure 12 for Setup C (Wired). We only show the results obtained for equalized I/Q and raw-after-FFT I/Q samples since the raw-before-FFT I/Q results are similar to those obtained in the raw-after-FFT case. As we can see, the results confirm that the TDTA performance is not satisfying in both cases. Overall, in Setup B the TTOD and TDTA accuracy for equalized I/Q is 83.45% and 8.72% respectively. The discrepancy between TTOD and TDTA demonstrates that the CNN is still learning channel conditions.

Particularly, in Setup C we show how exposing all devices to the same channel conditions (i.e., coaxial RF SMA cable with 5db attenuator) impacts the fingerprinting results. Here, there are two interesting aspects that are worth mentioning. First, Figure 12 clearly shows that the TTOD for raw-after-FFT data is worse in the wired case than in the wireless one (i.e., Setup A (Different Antennas) and Setup B). This means that, as the channel becomes less evident (i.e., Setup C), the raw-after-FFT features become less effective. Moreover, we notice that the TDTA for equalized I/Q data is better in Setup C than in Setup B. Indeed, the TTOD and TDTA here are 87.41% and 29.68%, respectively. This suggests that the model learns better to discern devices from their impairments in the wired scenario, where channel action becomes a stationary process for all devices and, therefore, does not represents a relevant feature to be learned.

C. Setup D: Anechoic Chamber, Single Antenna

The setup consists of 10 USRP transmitters (6 X310 and 4 N210) with one Receiver (N210). The distance between the transmitter and receiver antennas is fixed and equal to 12 feet in all transmissions, and all devices uses the same transmitter

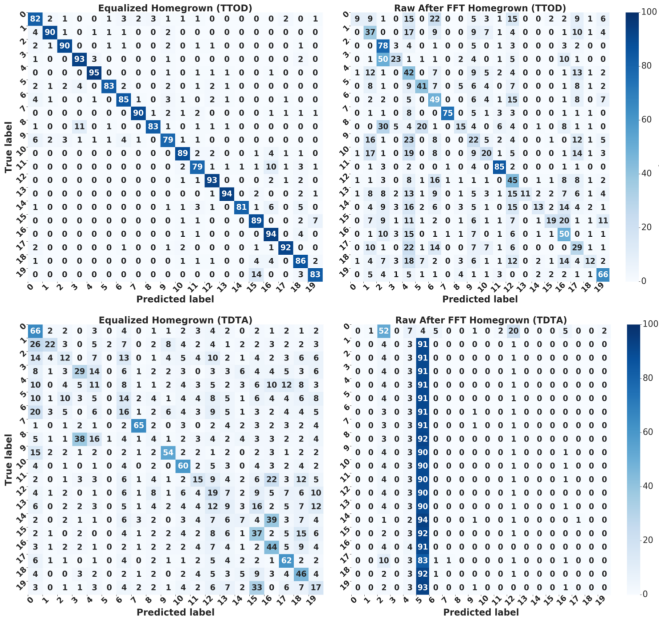

Figure 12. **"Setup C – Wired Connection"**. Top, from left to right: (a) Equalized I/Q **TTOD**, (b) Raw I/Q after FFT **TTOD**. Bottom, from left to right: (c) Equalized I/Q **TDTA**, (d) Raw I/Q after FFT **TDTA**.

antenna one at a time. The aim of using anechoic chamber is to analyze the performance of radio fingerprinting in a wireless environment without interference, and to confirm whether we would obtain results similar to Setup C (Wired). As shown in Figure 13, the TTOD and TDTA results for Setup D are similar to those reported for Setup C. Indeed, the equalized I/Q data results in better TDTA accuracy if compared to Setup B (Single Antenna).

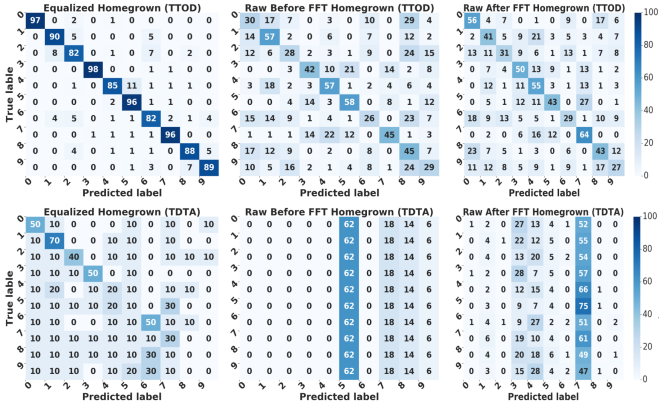

Figure 13. **"Setup D – Anechoic Chamber"**. Top, from left to right: (a) Equalized I/Q **TTOD**, (b) Raw I/Q before FFT **TTOD**, (c) Raw I/Q after FFT **TTOD**, Bottom, from left to right: (d) Equalized I/Q **TDTA**, (e) Raw I/Q before FFT **TDTA**, (f) Raw I/Q after FFT **TDTA**.

D. Learning Performance Comparison

In this section we focus on evaluating the learning performance by comparing and analyzing the validation loss comparison, and the final prediction/classification results. Figure 14 compares the validation loss during the training process over different setups for the equalized I/Q model. We also show the per-transmission accuracy (PTA) comparison between Setup B

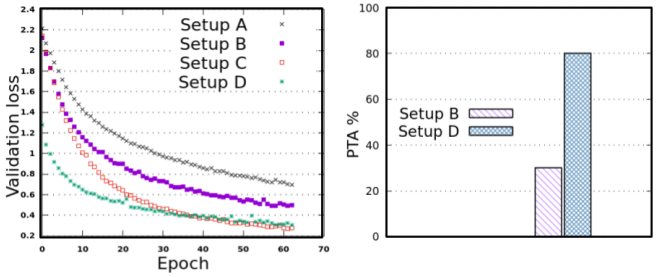

Figure 14. Validation loss and PTA% per Experimental Setup.

(Single Antenna) and Setup D (Anechoic Chamber). We notice that Setup D performs the best, followed by Setup C (Wired) and Setup B. This supports our results and confirms that the lowest loss occurs with the wireless channel condition of the anechoic testbed. In this case, the lack of interference, multipath and same channel conditions helped CNN models to learn the hardware impairments rather than the actual channel conditions. This is also confirmed by the PTA results, which show that the PTA is much higher in Setup D.

VI. Large-scale Dataset Results

In this section, we report the results on the large-scale dataset provided by DARPA. To analyze the effect of different parameters on the learning performance, we have split the entire WiFi/ADS-B dataset into a set of different learning tasks, summarized in Table 1. Specifically, Task A evaluates the performance as a function of the number of devices (from 100 to 10000, equally split between WiFi and ADS-B), while Task B assesses the effect of environment conditions on classifier accuracy, using a dataset of 310k WiFi transmissions generated by 350 devices. Each subtask aims to simulate a different environmental condition. Task C includes a dataset of 120k ADS-B transmissions generated by 100 devices, encountering high (15.3 to 5.1dB), medium (5.0 to 2.0dB), and low (1.9 to -13.3dB) SNR levels.

We evaluate the performance of the Baseline and ResNet-50-1D CNNs on both WiFi and ADS-B transmission. For WiFi we consider both raw and equalized data. Conversely, thanks to its simplicity (on-off keying) and lower amount of interference, we train the ADS-B models on raw I/Q data. Table 2 reports the per-slice accuracy (PSA) and per-transmission accuracy (PTA) for each CNN, task and level of equalization. We report in bold the best performance in each subtask.

As far as Task A (Scalability) is concerned, we notice that both Baseline and ResNet-50-1D scale well with the number of devices. Moreover, ResNet-50-1D performs well over raw-before-FFT data for the WiFi dataset, while Baseline performs better over equalized data. For the ADS-B dataset, ResNet-50-1D obtains 77% and 90% accuracy over 5000 and 500 devices, respectively, while Baseline outperforms ResNet-50-1D over fewer classes, attaining 88% and 92% accuracy over 250 and 50 devices, respectively.

The results obtained in Task B (Training Data) confirm the results obtained by using our experimental testbed, and indeed show that the environmental conditions affect the learning process significantly. The most interesting result regards Task

Task	Description	# of Devices
A1	Very High Population	10,000
A2	High Population	1000
A3	Medium Population	500
A4	Low Population	100
B1	Train One Day Test Another	50
B2	Train on a Mix of Days Test on a Mix	100
B3	Train and Test on a Single Day	100
C1	SNR: Train High Test Medium	100
C2	SNR: Train High Test Low	100
C3	SNR: Train Medium Test High	100
C4	SNR: Train Medium Test Low	100
C5	SNR: Train Low Test High	100
C6	SNR: Train Low Test Medium	100

Table 1. Summary of large-scale learning subtasks. Task A measures the scalability of the model, Task B and C measure the effect of environmental and channel conditions.

Task	Testing Accuracy Per-Slice / Per-Transmission Accuracy (PSA/PTA)			
	WiFi			
	Raw I/Q before FFT		Equalized	
	Baseline	ResNet-50-1D	Baseline	ResNet-50-1D
A1	0.082 / 0.130	**0.164 / 0.262**	0.062 / 0.101	0.014 / 0.030
A2	(0.299 / 0.378	**0.393 / 0.612**	0.327 / 0.434	0.392 / 0.555
A3	0.354 / 0.398	**0.467 / 0.629**	0.454 / 0.478	0.430 / 0.549
A4	0.335 / 0.575	0.490 / 0.631	**0.762 / 0.639**	0.699 / 0.637
B1	0.017 / 0.016	0.013 / 0.012	**0.232 / 0.335**	0.175 / 0.258
B2	0.444 / 0.695	**0.520 / 0.811**	0.678 / 0.674	0.751 / 0.735
B3	0.310 / 0.598	**0.441 / 0.746**	0.210 / 0.432	0.308 / 0.542

Table 2. Large-scale DARPA WiFi Dataset Results.

B1 (Train One Day Test Another), which confirms what experienced in the testbed. Indeed, we experience a drop in accuracy from 44% PSA in B3 (Train and Test on a Single Day) to almost 0% in B1 for the *ResNet-50-1D* CNN on raw-before-FFT data. For equalized data, we notice that the drop in accuracy is smaller – from 30% to 17%. This confirms that equalizing I/Q transmissions indeed improves the performance, as prediction over raw-before-FFT data is close to random guessing when raw data is used and testing is done on days different than the training one. Indeed, notice that in B1 the PSA increases from almost 0% to 23% and 17% when equalized I/Q samples are considered. Overall, these experiments unequivocally conclude that the wireless channel does impact the accuracy when a large population of device is considered.

Table 3 shows the results obtained on the subtasks involving ADS-B. The first important thing to notice here is that the performance is much better than WiFi. This is mainly thanks to the simplicity of ADS-B and the presence of less interference in the ADS-B channel. Interestingly, we also notice that training the model on high-SNR data and testing on low-SNR samples (subtask C2) leads to lower accuracy. Instead, when the opposite is true (subtask C4), we experience high accuracy. The phenomenon is even more pronounced on Subtask C3, where we train on medium SNR and test on high SNR, and C6 , where we train with low SNR and test on medium SNR, where we observe very high accuracy (92% and 93%, respectively). This means that adding noise to the training data makes the CNN less affected by noise in the test set, which could be further used to increase accuracy in future work.

Task	Testing Accuracy Per-Slice / Per-Transmission Accuracy (PSA/PTA)	
	ADS-B - Raw I/Q	
	Baseline	ResNet-50-1D
A1	0.374 / 0.529	**0.574 / 0.770**
A2	0.665 / 0.826	**0.803 / 0.896**
A3	**0.732 / 0.877**	0.646 / 0.813
A4	**0.810 / 0.919**	0.717 / 0.862
C1	**0.375 / 0.526**	0.376 / 0.494
C2	**0.115 / 0.156**	0.104 / 0.132
C3	**0.790 / 0.916**	0.720 / 0.828
C4	**0.378 / 0.600**	0.333 / 0.509
C5	**0.695 / 0.886**	0.567 / 0.693
C6	**0.649 / 0.925**	0.594 / 0.810

Table 3. Large-scale DARPA ADS-B Dataset Results.

VII. Conclusions

This paper has presented (i) a large-scale open dataset for radio fingerprinting collected in diverse environments and rich, diverse, channel conditions; and (ii) a full-fledged, systematic investigation of the impact of the wireless channel on the accuracy of CNN-based radio fingerprinting algorithms. Specifically, we have collected more than 7TB of wireless data obtained from 20 wireless devices over the course of 10 days in an anechoic chamber, in-the-wild, and with cable connections. We have also provided an exhaustive evaluation of the impact of the wireless channel on CNN-based fingerprinting algorithms through the testbed data and a 400GB dataset provided by DARPA. Experimental results conclude that (i) the wireless channel impacts the classification accuracy significantly, *i.e.*, from 85% to 9% and from 30% to 17% in the experimental and DARPA dataset, respectively; and that (ii) equalizing I/Q data can increase the accuracy to a significant extent (*i.e.*, by 23%) when the number of devices increases significantly.

Acknowledgments

This work is supported by the Defense Advanced Research Projects Agengy (DARPA) under RFMLS program contract N00164-18-R-WQ80. We thank Esko Jaska and Paul Tilghman for their precious comments, which have helped us significantly improve our manuscript. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

References

[1] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How Can Heterogeneous Internet of Things Build our future: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2011–2027, 2018.

[2] A. S. Yeole and D. Kalbande, "Use of Internet of Things (IoT) in Healthcare: A Survey," in *Proceedings of the ACM Symposium on Women in Research 2016*. ACM, 2016, pp. 71–76.

[3] F. Restuccia, S. D'Oro, and T. Melodia, "Securing the Internet of Things in the Age of Machine Learning and Software-Defined Networking," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4829–4842, Dec 2018.

[4] T. Zheng, Z. Sun, and K. Ren, "FID: Function Modeling-based Data-Independent and Channel-Robust Physical-Layer Identification," in *Proc. of the IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2019, pp. 199–207.

[5] 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.

[6] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, "Device Fingerprinting to Enhance Wireless Security Using Nonparametric Bayesian Method," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2011, pp. 1404–1412.

[7] 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.

[8] 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*, vol. 6, no. 1, pp. 349–360, Feb 2019.

[9] 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.

[10] 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.

[11] 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.

[12] E. Johnson, "Physical Limitations on Frequency and Power Parameters of Transistors," in *1958 IRE International Convention Record*, vol. 13. IEEE, 1966, pp. 27–34.

[13] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 370–378.

[14] 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.

[15] 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.

[16] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms," in *Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)*. ACM, 2019, pp. 51–60.

[17] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine Learning for Wireless Communications in the Internet of Things: A Comprehensive Survey," *Ad Hoc Networks*, vol. 93, p. 101913, 2019.

[18] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, "No Radio Left Behind: Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments," *IEEE Transactions on Cognitive Communications and Networking (TCCN)*, 2019.

[19] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's Intranet of Things to a future Internet of Things: a Wireless-and Mobility-related View," *IEEE Wireless communications*, vol. 17, no. 6, 2010.

[20] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[21] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to Generate Reviews and Discovering Sentiment," *arXiv preprint arXiv:1704.01444*, 2017.

[22] Y. Zhang and J. Qiao, "ADS-B Radar System," Aug. 19 2008, uS Patent 7,414,567.

[23] 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.

[24] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep Learning for RF Device Fingerprinting in Cognitive Communication Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 160–167, 2018.

[25] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. Moura, "A Deep Learning Approach to IoT Authentication," in *Proc. of the IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks."

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[30] T. J. 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.

[31] K. Karra, S. Kuzdeba, and J. Petersen, "Modulation Recognition Using Hierarchical Deep Neural Networks," in *Proc. of IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Baltimore, MD, USA, March 2017, pp. 1–3.

[32] IEEE, "IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Redline," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007) - Redline*, pp. 1–5229, March 2012.

[33] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio," in *Proceedings of the second workshop on Software radio implementation forum*. ACM, 2013, pp. 9–16.

[34] B. Hilburn, N. West, T. O'Shea, and T. Roy, "SigMF: The Signal Metadata Format," in *Proceedings of the GNU Radio Conference*, vol. 3, no. 1, 2018.