

Comprehensive RF Dataset Collection and Release: A Deep Learning-Based Device Fingerprinting Use Case

Abdurrahman Elmaghub and Bechir Hamdaoui

School of Electrical Engineering and Computer Science, Oregon State University

{elmaghba, hamdaoui}@oregonstate.edu

Abstract—Deep learning-based RF fingerprinting has recently been recognized as a potential solution for enabling newly emerging wireless network applications, such as spectrum access policy enforcement, automated network device authentication, and unauthorized network access monitoring and control. Real, comprehensive RF datasets are now needed more than ever to enable the study, assessment, and validation of newly developed RF fingerprinting approaches. In this paper, we present and release a large-scale RF fingerprinting dataset, collected from 25 different LoRa-enabled IoT transmitting devices using USRP B210 receivers. Our dataset consists of a large number of SigMF-compliant binary files representing the I/Q time-domain samples and their corresponding FFT-based files of LoRa transmissions. This dataset provides a comprehensive set of essential experimental scenarios, considering both indoor and outdoor environments and various network deployments and configurations, such as the distance between the transmitters and the receiver, the configuration of the considered LoRa modulation, the physical location of the conducted experiment, and the receiver hardware used for training and testing the neural network models.

Index Terms—IoT Testbed, RF Dataset Collection and Release, RF Fingerprinting, Deep Learning, LoRa Protocol.

I. INTRODUCTION

This paper presents and releases a comprehensive dataset consisting of massive RF signal data captured from 25 LoRa-enabled transmitters using Ettus USRP B210 receivers. The RF dataset provides the research community with a set of first-of-its-kind real-world setups, all helpful in enabling the study of the effectiveness and robustness of deep learning-based wireless techniques, such as RF/device fingerprinting. These experimental setups cover and provide a comprehensive set of practical scenarios for indoor and outdoor environments while considering various realistic network deployment variability across time, location, hardware, and modulation configuration. These are obtained by varying network deployment and configuration parameters, such as the distance between the transmitters and the receiver, the configuration of the LoRa protocol, the physical location of the experiment, and the receiver hardware used for collecting the data samples and for training/testing the neural network models.

As research communities move away from model-based to data-driven solutions, many recently proposed frameworks on device/RF fingerprinting have also shifted from model-based classification approaches to deep learning-based approaches. Although recently proposed deep learning-based approaches have shown promising results, some still rely on synthesized

data for evaluation and validation, and most of those that rely on real datasets are evaluated on general indoor and outdoor setups, leaving behind many questions that need to be answered before claiming the feasibility and superiority of their proposed techniques in real-world settings.

Other essential experimental scenarios, which are very useful for assessing the capability and robustness of deep learning-based fingerprinting techniques, have not been investigated. These essential scenarios allow for the study of the impact of various deployment parameters on the performances achievable by these techniques, such as the impact of the distance between the transmitters and the receiver, the configuration of the used protocol, the physical location of the conducted measurements, and the hardware impairments of the receiver used in the training and inference stages.

Another important scenario that is also missing is the study of the performance of deep learning models when the RF signals are captured during concurrent transmissions where some portions of the incoming packets overlap.

A major challenge the wireless research community has been facing is the lack of comprehensive, publicly available datasets that could serve as benchmarks for the device/RF classification and fingerprinting techniques [1]. Having public and easily accessible dataset benchmarks has indeed been one of the main drivers for innovation and idea maturity in closely related fields, like image recognition and natural language processing, in addition to creating collaboration opportunities among researchers. Therefore, such efforts must be mimicked in the wireless community to be able to foster innovation in this domain as well.

A. What Distinguishes Our Dataset From Existing Ones?

Researchers in [2] presented three LPWAN (Sigfox and LoRaWAN) datasets collected in an outdoor environment aimed at evaluating location fingerprinting algorithms. These datasets have been collected over three months in both rural and urban areas. Sigfox-rural and Sigfox-urban datasets consist respectively of 25k+ and 14k+ Sigfox messages, whereas LoRaWAN dataset consists of 123k+ LoRaWAN messages. These messages include a couple of protocol information, time of reception, and GPS location information but do not include device labels, and therefore, can not be used for supervised deep learning-based device classification. The closest existing work to our work is the recently released work at Northeastern

Setups	Number of Transmitters	Number of Receivers	Protocol	Number of Days	Transmissions per Device	Duration per Transmission	Distances	Environment	Representation
1) Diff Days Indoor	25	1	LoRa	5	10	20s	5m	Indoor	IQ/FFT
2) Diff Days Outdoor	25	1	LoRa	5	10	20s	5m	Outdoor	IQ/FFT
3) Diff Days Wired	25	1	LoRa	5	10	20s	5m	Wired	IQ/FFT
4) Diff Distances	25	1	LoRa	1	4	20s	5,10,15,20m	Outdoor	IQ/FFT
5) Diff Configurations	25	1	LoRa	1	4	20s	5m	Indoor	IQ/FFT
6) Diff Locations	25	1	LoRa	1	3	20s	5m	2 Indoor, 1 Outdoor	IQ/FFT
7) Diff Receivers	25	2	LoRa	1	2	20s	5m	Indoor	IQ/FFT

TABLE I: Summary of Experimental Setups/Scenarios.

University [1], which collected and released a massive dataset of IEEE 802.11 a/g (WiFi) standard data obtained from 20 wireless devices with identical RF circuitry over several days in (a) an anechoic chamber, (b) in-the-wild testbed, and (c) with cable connections. The focus of the Northeastern dataset is to explore the impact of the wireless channel on the performance of deep learning-based RF fingerprinting models. The dataset is limited in terms of the covered scenarios, and it is for WiFi signals only. Unfortunately, to the best of our knowledge, there are still no public datasets for LoRa device fingerprinting nor datasets that include the diverse experimental scenarios we mentioned. Our dataset presented in this paper fulfills the need for a large-scale LoRa dataset covering a wide range of diverse scenarios for a more comprehensive evaluation and validation.

B. Our LoRa Dataset in Brief

Our RF dataset provides both time-domain I/Q samples and corresponding FFT samples collected using an IoT testbed consisting of 25 identical Pycom IoT devices and a USRP B210 receiver, operating at a center frequency of 915MHz, used for recording the received signals sampled at 1MS/s. Recorded data in the form of both the time-domain I/Q samples and FFT samples are stored into binary files in compliance with SigMF [3] by creating, for each binary file, a metafile written in plain-text JSON to include recording information such as sampling rate, time and day of recording, and carrier frequency, among other parameters.

This dataset covers multiple experimental setup scenarios, which are summarized in Table I and can be downloaded at <http://research.engr.oregonstate.edu/hamdaoui/datasets>.

The experimental scenarios are briefly described next, with more details provided in [4].

- **Setup 1—Different Days Indoor Scenario:** Indoor setup with data collected over 5 consecutive days. For each day, 10 transmissions were captured from each of the 25 transmitters, with each transmission having a duration of 20s. All transmissions took place 1 minute apart from one another.
- **Setup 2—Different Days Outdoor Scenario:** Outdoor setup with data collected over 5 consecutive days. For each day, 10 transmissions were captured from each of the 25 transmitters, with each transmission having a duration of 20s. All transmissions took place 1 minute apart from one another.
- **Setup 3—Different Days Wired Scenario:** Wired setup with data collected over 5 consecutive days. For each

day, 10 transmissions were captured from each of the 25 transmitters, with each transmission having a duration of 20s. All transmissions are 1 minute apart from one another.

- **Setup 4—Different Distances Scenario:** Outdoor setup with data collected from 4 different distances: 5m, 10m, 15m, and 20m away from the receiver. For each distance, one transmission was captured from each of the 25 transmitters, with each transmission having a duration of 20s. All transmissions are 1 minute apart from one another.
- **Setup 5—Different Configurations Scenario:** Indoor setup with data collected from 4 different LoRa configurations. For each configuration, one transmission was captured from each of the 25 transmitters, with each transmission having a duration of 20s. All transmissions are 1 minute apart from one another.
- **Setup 6—Different Locations Scenario:** This data has been collected in 3 different locations: room, outdoor, and office environments. At each location, one transmission was captured from each of the 25 transmitters, with each transmission having a duration of 20s. All transmissions are 1 minute apart from one another.
- **Setup 7—Different Receivers Scenario:** Indoor setup with data collected using 2 different receivers. For each receiver, one transmission was captured from each of the 25 transmitters, with each having a duration of 20s. All transmissions are 1 minute apart from one another.

The rest of the paper is organized as follows. Section II describes the testbed components. Sections III and IV describe the different experimental setups and the dataset, respectively. Section V presents a use case for the dataset. The challenges, limitations, and new opportunities are discussed in Section VI and the paper is concluded in Section VII.

II. TESTBED

In this section, we describe the hardware, software, and protocol components used in building our testbed.

A. Hardware Description

Our IoT testbed, shown in Fig. 1, consists of 25 Pycom devices with Semtech SX1276 LoRa transceivers, 25 Pycom sensor shields, and an Ettus USRP (Universal Software Radio Peripheral) B210 for data sampling, which was configured with a center frequency of 915MHz and a sample rate of 1MS/s. Our collection of Pycom devices is made up of 23



Fig. 1: IoT Testbed: 25 Pycom devices and USRP B210.

Lopy4 boards and 2 Fipy boards, which are MicroPython-enabled development boards with multiple network capabilities: LoRa, Sigfox, WiFi, Bluetooth, and NB-IoT. The sensor shield collection comprises 22 PySense boards equipped with sensing capability, 2 PyScan boards equipped with RFID scanning capability, and 1 PyTrack board equipped with GPS/tracking capability. We used lipo batteries to power the devices. Each Pycom device was connected to a dedicated LoRa antenna and configured to transmit LoRa transmissions at the 915MHz US band adapting the following configuration: Raw-LoRa mode, 125KHz bandwidth, a spreading factor (SF) of 7, a preamble of 8, a TX power of 20dBm, and a coding rate of 4/5.

B. Software Description

1) *Transmission Subsystem*: We programmed and configured our Pycom boards using MicroPython [5], which is an efficient implementation of Python3 that is composed of a subset of standard Python libraries and optimized to run on microcontrollers and constrained environments. Also, we used Pymkr plugin as a REPL console that connects to Pycom boards to run codes or upload files.

2) *Reception Subsystem*: We used the GNURadio software [6], a real-time signal processing graphical tool, to set up and configure the USRP receiver to capture LoRa transmissions, plot their time and spectrum domains, implement some pre-processing techniques and store the samples into their files. Fig. 2 shows the general flowgraph used for data acquisition.

C. LoRa Protocol Description

We transmitted/captured LoRa modulation signals, a proprietary physical layer implementation that employs Chirp Spread Spectrum (CSS) in the sub-GHz ISM band and trades data rate for coverage range, power consumption, or link robustness. LoRa does so by providing a tunable parameter, called a spreading factor (SF), which varies from 7 to 12 and determines the sequence length of an encoded symbol within a fixed bandwidth. A higher spreading factor means longer ranges with lower data rates. Unlike other spread spectrum techniques, the chirp-based modulation allows LoRa to maintain the same coding gain and immunity to noise

and interference while meeting the low-cost, low-power consumption requirements. A LoRa modulator generates both raw chirp signals with fixed amplitude and continuously varying frequency with constant rate and a set of modulated chirps that are cyclically time-shifted raw-chirps where the initial frequency determines the content of the chirp symbol.

III. EXPERIMENTAL SETUPS

We use our testbed described in Section II to create and collect large-scale, comprehensive LoRa RF fingerprinting dataset for multiple experimental scenarios that are specifically designed to allow intensive and comprehensive performance evaluation of various deep learning-based wireless networks techniques, with a special focus on RF/device fingerprinting. We chose a bandwidth of 125KHz for all LoRa transmissions. However, in [7], we demonstrated that considering out of band spectrum enhances the performance of the model, and hence we provide a total bandwidth of 1MHz that covers the in-band as well as an adjacent out of band spectrum of LoRa transmissions for all setups. In the remainder of this section, we present each of the seven considered experimental setups. Throughout this section, we will be referring to Fig. 3 for experimental setups 1, 2, and 3, and for Fig. 4 for experimental setups 4, 5, 6 and 7. Table I summarizes these 7 setups.

A. Setup 1: Different Days Indoor Scenario

In order to enable performance evaluation while masking the impact of the outside environment, we created an indoor setup, ran experiments, and collected datasets for this setup. These indoor experiments were carried out in a typical occupied room environment over 5 consecutive days. All devices transmitted the same message from the same location, 5m away from the receiver so that all devices experience similar channel conditions. As shown in Fig. 3, for each day, each transmitter generated 10 transmissions, each of 20s duration, all spaced apart by 1 minute. Hence, we collected about 200M complex-valued samples from each device per day. We used GNURadio packages to store the sampled raw-I/Q values and their corresponding FFT-Based representation into binary files as depicted in Fig. 3.

B. Setup 2: Different Days Outdoor Scenario

In order to allow for performance evaluation while considering the impact of outdoor wireless channel impairments, we carried out the experiments in an outdoor environment at nighttime. Here again, all devices transmitted the same message from the exact location, situated 5m away from the receiver, so that all devices experience similar channel conditions. Like in the indoor setup case and as shown in Fig. 3, for five consecutive days, each transmitter generated 10 transmissions per day, each of 20s duration, all spaced 1 minute apart from one another. This resulted in about 200M complex-valued samples per device per day. We ran this experiment over 5 consecutive days and provided 5-day worth of data to study the robustness of deep learning models when trained on data collected on one day but tested on data captured

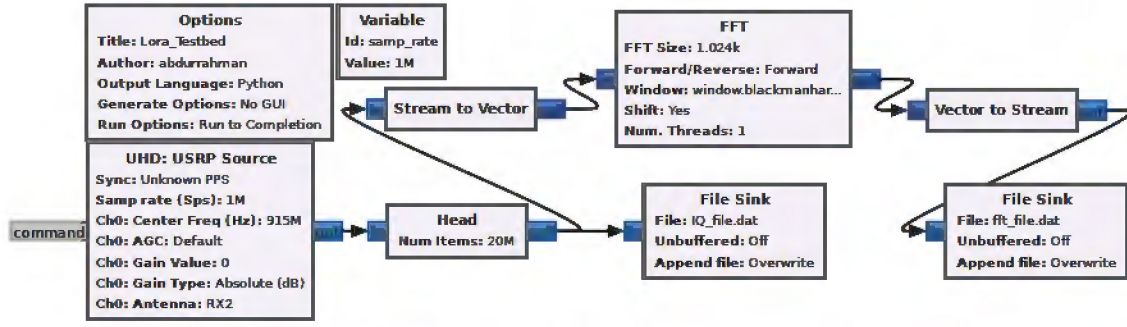


Fig. 2: The flowgraph of our data collection.

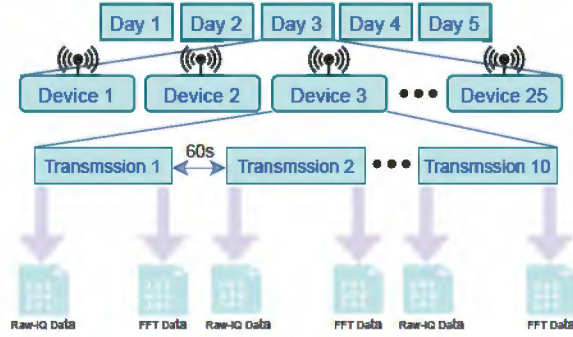


Fig. 3: Setup diagram: setups 1, 2 and 3.

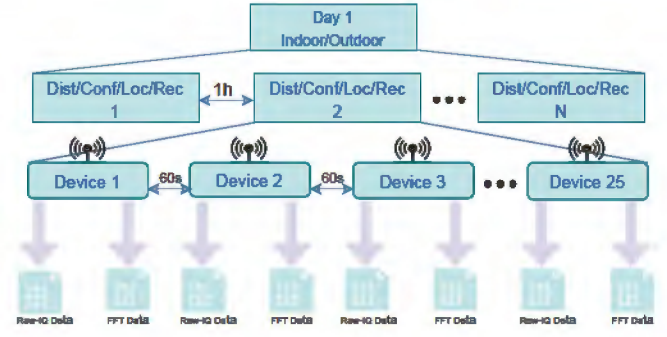


Fig. 4: Setup diagram: setups 4, 5, 6, and 7.

on a different day. We used GNURadio packages to store the sampled raw-I/Q values and their corresponding FFT-based representation into binary files as depicted in Fig. 3.

C. Setup 3: Different Days Wired Scenario

The wireless channel has a notable impact on the performance of deep learning models and presents its unique challenges [1]. Hence, to assess how well these models perform in the absence of the wireless channel's impact, we created a wired setup where the Pycom boards are directly connected to the USRP via an SMA cable and 30dB attenuator. Similar to the Indoor and Outdoor experiments, we ran this experiment over 5 consecutive days. For every day, each device transmitted 10 bursts each of 20s duration. Therefore, the total amount of collected data is 200M complex-valued samples per device per day. We again used GNURadio packages to store the sampled raw-I/Q values and their corresponding FFT-based representation into binary files as depicted in Fig. 3.

D. Setup 4: Different Distances Scenario

Some end-devices constantly change their positions, so it is critical to explore the impact of distance on the performance of classifiers and see whether or not a classifier would still recognize a device when it moves to a position that is different from the one used for training. This experiment was carried out in a typical outdoor environment in a sunny day. We considered four different distances, 5m, 10m, 15m, and 20m, and for each distance, we collected 1 transmission of 20s

for each of the 25 devices. We kept the receiver at the same location for all the transmissions while locating the transmitters at 4 different distances away from the receiver base (5m, 10m, 15m, and 20m). The transmissions were captured consecutively in time with only 60s apart from one another. Each transmitter generated 4 transmissions each of 20s duration, resulting in 80M complex-valued samples from each device. We again used GNURadio packages to store the sampled raw-I/Q values and corresponding FFT-based samples into binary files as depicted in Fig. 4.

E. Setup 5: Different Configurations Scenario

LoRaWAN uses the Adaptive Data Rate (ADR) mechanism to optimize the data rates, air-time, and energy consumption in the network to accommodate the varying RF conditions. This mechanism allows the network server to inform end devices to adjust their power consumption and data rate as needed. This is achievable by controlling the following parameters at end devices: spreading factor, bandwidth, and power consumption. Changing the spreading factor, for example, in LoRa modulation results in a change in the data rate, receiver sensitivity, time in the air, and power consumption. Fig. 5 shows the frequency spectrum of a snapshot of the four LoRa configurations that we included in our dataset. Ideally, a classification model (e.g., a deep learning fingerprinting model) should identify a device even if it changes its configuration; i.e., models that are trained using one configuration but tested on a different configuration should still perform well. Therefore, in order

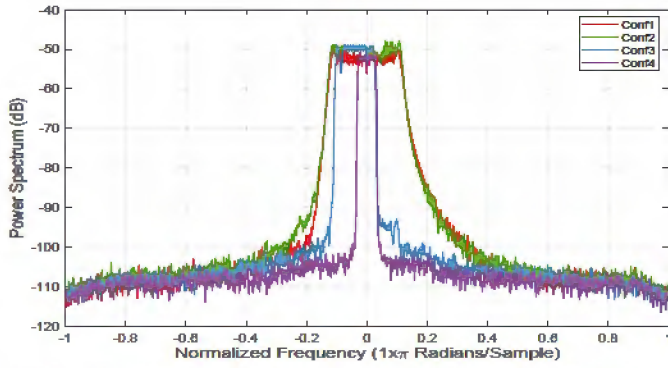


Fig. 5: Spectrum of the 4 LoRa configurations from Setup 5

Configurations	Spread Factor	Bandwidth	Bit Rate	Tx Power	Coding Rate
Configuration 1	7	125KHz	5470 bps	20dBm	4/5
Configuration 2	8	125KHz	3125 bps	20dBm	4/5
Configuration 3	11	125KHz	537 bps	20dBm	4/5
Configuration 4	12	125KHz	293 bps	20dBm	4/5

TABLE II: LoRa Configurations

to enable the assessment of how agnostic these models are to protocol configuration, we captured transmissions using 4 different configurations, as presented in Table II. For this, we collected a single LoRa transmission of 20s from each device for each configuration in an indoor setup with 5m as the distance between the receiver and transmitters. Like other setups, for this setup, we used GNURadio packages to store the sampled raw-I/Q values and their corresponding FFT-based representation into binary files as depicted in Fig. 4.

F. Setup 6: Different Locations Scenario

Another practical scenario we consider here aims at allowing deep learning models to be trained on data captured in one location but tested on data collected in another location. For this, we captured LoRa transmissions in three different location deployments, room environment, office environment, and outdoor environment, all taken on the same day. Here, we kept the distance between the receiver and transmitters (i.e., 5m) and the LoRa configuration the same. We captured a single transmission of 20s from each device at each location with a 60s period between the devices, resulting in 60M complex samples from each device. We again used GNURadio packages to store the sampled raw-I/Q values and their corresponding FFT-based representation into binary files as depicted in Fig. 4.

G. Setup 7: Different Receivers Scenario

Like transmitters, receivers also suffer from hardware impairments due to hardware imperfection. Therefore, deep learning models trained using data collected by one receiver but tested using data collected by a different receiver may not perform well due to the possible additional impairments caused by the receiver's reception. To allow researchers to study the impact of such a change in the receiving device, we provided a dataset for the 25 Pycom transmitting devices,

collected using two different USRP B210 receivers. In this experiment, we considered an indoor setup where the transmitters (the 25 Pycom devices) were located 5m away from the receiver. For each receiver, we captured a single transmission of 20s from each of the 25 Pycom transmitters, yielding a total of 40M samples for each device. Like other setups, for this setup, we used GNURadio packages to store the sampled raw-I/Q values and their corresponding FFT-based representation into binary files as depicted in Fig. 4.

IV. DATASET DESCRIPTION

Our dataset contains LoRa transmissions of 25 devices and a total of 43 transmissions on average for each device. An Ettus USRP B210 is used to record each transmission, operating at a center frequency of 915MHz with a sampling rate of 1MS/s. Each recording consists, on average, of 20M I/Q samples. For each recording, we stored the time-domain I/Q samples and FFT-based samples into binary files. The binary files are encoded with Float32, and the complex-valued sampled are interleaved where the I-values are in the odd indices, and the Q-values are in the even ones. In order to be SigMF (Signal Metadata Format) [3] compliant, we created a metadata file written in plain-text JSON adapting SigMF for each binary file to describe the essential information about the collected samples, the system that generated them, and the features of the signal itself. In our case, we stored in the metadata files information regarding (i) the sampling rate, (ii) time and day of recording, and (iii) the carrier frequency, among others.

More details and use cases of these LoRa datasets can be found in [4]. The datasets can be downloaded from NetSTAR Lab at <http://research.engr.oregonstate.edu/hamdaoui/datasets>. Users of the datasets may refer to Fig. 6 for help with the file organization and notation. Specifically, these are:

- **Diff_Days_Indoor_Setup**, **Diff_Days_Outdoor_Setup**, and **Diff_Days_Wired_Setup** directories (correspond to Setups 1, 2 and 3), each having 5 subdirectories, one for each day. Each day subdirectory has 25 subdirectories, one for each device. Each device subdirectory has 40 files (for 10 transmissions): 10 I/Q data files plus their corresponding 10 metadata files and 10 fft data files plus their corresponding 10 metadata files.
- **Diff_Distances_Setup** directory (corresponds to Setup 4), having 4 subdirectories representing the four distances. Each subdirectory includes 100 files: 25 I/Q data files plus their corresponding 25 metadata files and 25 fft data files plus their corresponding 25 metadata files.
- **Diff_Configurations_Setup** directory (corresponds to Setup 5), having 4 subdirectories representing the four configurations. Each of these 4 subdirectories includes 100 files: 25 I/Q data files plus their corresponding 25 metadata files and 25 fft data files plus their corresponding 25 metadata files.
- **Diff_Locations_Setup** directory (corresponds to Setup 6), having 3 subdirectories representing the three locations. Each subdirectory includes 100 files: 25 I/Q data

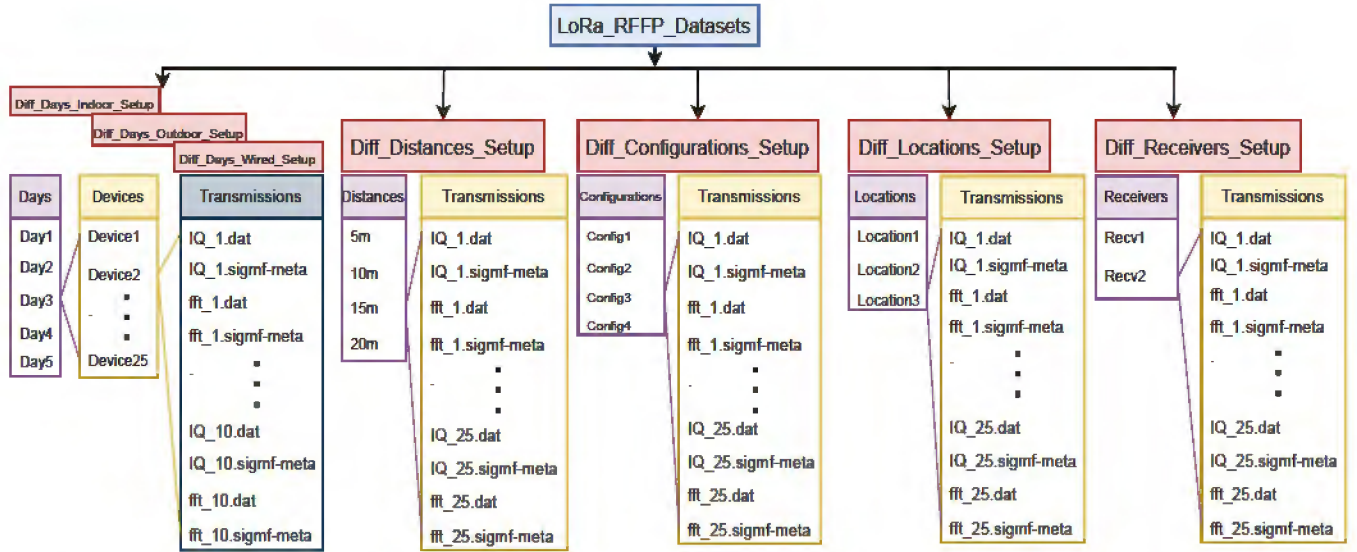


Fig. 6: Online link/file organization of the datasets. Note that `Diff_Days_Indoor_Setup`, `Diff_Days_Outdoor_Setup`, and `Diff_Days_Wired_Setup` directories have the same file system structure.

files plus their corresponding 25 metadata files and 25 fft data files plus their corresponding 25 metadata files.

- **Diff_Receivers_Setup** directory (corresponds to Setup 7), having 2 subdirectories representing the two receivers. Each subdirectory includes 100 files: 25 I/Q data files plus their corresponding 25 metadata files and 25 fft data files plus their corresponding 25 metadata files.

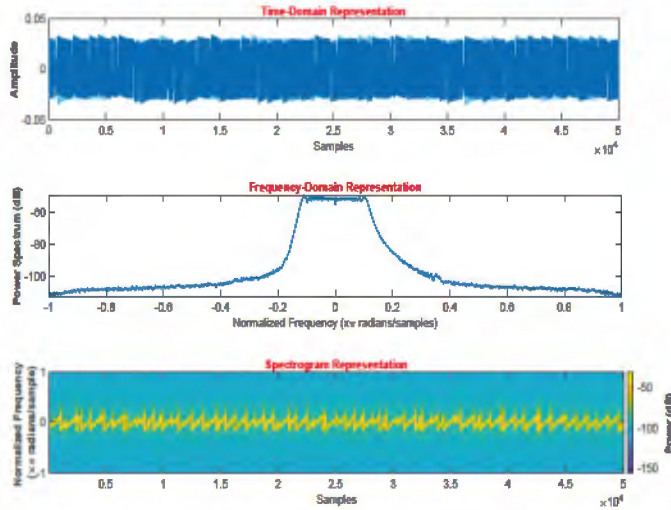


Fig. 7: Top: I/Q time-domain representation, Middle: Frequency spectrum representation, Bottom: Spectrogram representation from Device 1, Indoor Setup.

To visualize the captured LoRa signals, we plotted, in Fig. 7, the time-domain, frequency spectrum, and spectrogram of a LoRa transmission captured in an Indoor environment. The top figure in Fig. 7 represents the in-phase (I) and quadrature (Q) components of the time-domain signal from device 1 in an indoor setup, while the figure in the middle shows its

frequency spectrum and the figure in the bottom shows the upward chirps in the spectrogram of the same LoRa signal.

V. USE CASE: LORA DEVICE FINGERPRINTING

LoRaWAN [8], a low-power wide-area network technology, has emerged as the de facto standard for connecting thousands of million devices serving many IoT ecosystems and applications. Its ability to transmit over long distances with an optimized low-power consumption, leading to a long battery life of up to 10x that of cellular M2M technologies, makes it a good fit for next-generation IoT connections. LoRaWAN defines two layers of security: network-level security that deals with LoRa device authenticity and message integrity, and application-level security that provides end-to-end data encryption to ensure private device-to-server communication [9]. The functionality of these layers relies on generating and storing session keys (AppKey and NwkKey) in a secured manner. Some IDs and keys, such as DevEUI and AppKey, can be hard-coded on devices' tags or software, and thus simple but common human errors, such as failing in removing the tags or replacing the source code before deployment, can expose the network to security risks [10].

It is essential to ensure that conventional high-layer security mechanisms are complemented with unclonable, physical-layer security solutions so as to increase the security robustness of such systems [11]. One technique that has been considered for providing physical-layer security is to exploit wave-level distortions in the received RF signals that are generated by hardware impairments to provide unique signatures (aka fingerprints) of the devices, which can serve as device identifiers (e.g., [12], [13]). The uniqueness of RF fingerprints comes from the collective random deviations of a tremendous number of RF analog components from their ideal values during the manufacturing process. Hence, we can confidently

claim that there are no two analog circuits with identical collective deviations profiles, and therefore, there are no two circuits with the same RF fingerprint. IQ imbalance, DC offset, phase noise, and power amplifier nonlinearity are among other transceiver hardware impairments that manifest in uniquely distinguishing features that increase devices' separability in RF fingerprinting techniques [14].

Until recently, the feature selection process for RF fingerprinting techniques has been done manually, which requires expert knowledge and many trials and error iterations to find the features that guarantee the best performance. Nevertheless, in most cases, they end up with protocol-specific or vendor-specific solutions. Following the unprecedented achievements of deep learning models in computer vision and natural language processing in recent years, researchers have shown that the function approximation power of deep learning models can be leveraged to better improve the classification performance of RF fingerprinting techniques and other RF-domain challenges. These results led to a rush in research activities in the area of machine learning for RF systems from both industry and academia, creating an urgent need for comprehensive RF datasets that can be used for validating their proposed models.

VI. CHALLENGES, LIMITATIONS AND OPPORTUNITIES

We next describe some limitations and challenges faced when creating RF fingerprint datasets.

- **Same channel condition.** The time between a given device's first and last transmissions may not be short enough for all transmissions to be assumed to be taken under the same condition. This makes it challenging to study the performance of a learning model with training and testing being done under the same channel condition. This is because all 25 devices must first be sampled for their first transmissions before moving to the subsequent transmission. Even if one tries to collect all 10 transmissions sequentially for each device to minimize this timing effect between transmissions, we run into the problem of increasing the experiment time between devices.
- **Same power levels.** Maintaining the same power level for all devices is important so that to mask the power impact. In our experiments, to mitigate this issue, we start with a full-charged battery every time, though this still cannot guarantee that all have the same power level.
- **Concurrent transmissions.** One interesting scenario but difficult to realize is the ability to collect data from multiple devices while all transmitting concurrently, as it often occurs in random access procedures for wireless base stations. Excelling on this scenario would open the door for incorporating deep learning-based fingerprinting into next-generation cellular networks.
- **Devices at scale.** Increasing the number of transmitters in our testbed is another to-do-item that would add more credibility to the evaluation and allow us to assess the scalability performance of the proposed models.
- **Beyond RF fingerprinting.** Leveraging the multi-bearer capability of our testbed, one can use the testbed to

collect datasets for modulation recognition using other technologies like LoRaWAN, SigFox, Bluetooth, WiFi, and NB-IoT technologies. Another use is for creating datasets for studying indoor device positioning problems.

VII. CONCLUSION

This paper presents a comprehensive LoRa RF fingerprint datasets for multiple experimental scenarios specifically designed to allow thorough performance assessment of deep learning-based wireless networks techniques, such as RF/device fingerprinting. This dataset is made available to the research community to serve as a benchmark for testing RF classification and fingerprinting techniques.

VIII. ACKNOWLEDGMENT

This research is supported in part by Intel/NSF MLWiNS Award No. 2003273. We would like to thank Intel researchers, Dr. Kathiravetpillai Sivanesan, Dr. Lily Yang, and Dr. Richard Dorrance, for their constructive feedback.

REFERENCES

- [1] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia, "Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 646–655.
- [2] M. Aermouts, R. Berkmans, K. Van Vlaenderen, and M. Weyn, "Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, p. 13, 2018.
- [3] B. Hilburn, N. West, T. O'Shea, and T. Roy, "Sigmf: the signal metadata format," in *Proceedings of the GNU Radio Conf.*, vol. 3, no. 1, 2018.
- [4] A. Elmaghub and B. Hamdaoui, "LoRa device fingerprinting in the wild: Disclosing RF data-driven fingerprint sensitivity to deployment variability," *IEEE Access*, 2021.
- [5] N. H. Tollervey, *Programming with MicroPython: embedded programming with microcontrollers and Python*. "O'Reilly Media, Inc.", 2017.
- [6] D. Valerio, "Open source software-defined radio: A survey on gnuradio and its applications," *Forschungszentrum Telekommunikation Wien, Vienna, Technical Report FTW-TR-2008-002*, 2008.
- [7] A. Elmaghub, B. Hamdaoui, and A. Natarajan, "Widescan: Exploiting out-of-band distortion for device classification using deep learning," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.
- [8] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on lora networking: Research problems, current solutions, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 371–388, 2019.
- [9] D. Basu, T. Gu, and P. Mohapatra, "Security issues of low power wide area networks in the context of lora networks," *arXiv preprint arXiv:2006.16554*, 2020.
- [10] M. S. Cesar Cerrudo, Esteban Martinez Fayó, "LoRaWAN Networks Susceptible to Hacking: Common Cyber Security Problems, How to Detect and Prevent Them," <https://ioac.tv/30J8ZuE>, 2020, [Online; accessed 28-April-2021].
- [11] Y.-S. Shiu, S. Y. Chang, H.-C. Wu, S. C.-H. Huang, and H.-H. Chen, "Physical layer security in wireless networks: a tutorial," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 66–74, 2011.
- [12] 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.
- [13] B. Hamdaoui, A. Elmaghub, and S. Mejri, "Deep neural network feature designs for RF data-driven wireless device classification," *IEEE Network*, vol. 35, no. 3, pp. 191–197, 2020.
- [14] A. Elmaghub and B. Hamdaoui, "Leveraging hardware-impaired out-of-band information through deep neural networks for robust wireless device classification," *arXiv preprint arXiv:2004.11126*, 2020.