DeepLoRa: Fingerprinting LoRa Devices at Scale Through Deep Learning and Data Augmentation

Amani Al-Shawabka[†], Philip Pietraski[‡], Sudhir B Pattar[‡],
Francesco Restuccia[†] and Tommaso Melodia[†]
[†]Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA
[‡]InterDigital Communications, United States

ABSTRACT

The Long Range (LoRa) protocol for low-power wide-area networks (LPWANs) is a strong candidate to enable the massive roll-out of the Internet of Things (IoT) because of its low cost, impressive sensitivity (-137dBm), and massive scalability potential. As tens of thousands of tiny LoRa devices are deployed over large geographic areas, a key component to the success of LoRa will be the development of reliable and robust authentication mechanisms. To this end, Radio Frequency Fingerprinting (RFFP) through deep learning (DL) has been heralded as an effective zero-power supplement or alternative to energy-hungry cryptography. Existing work on LoRa RFFP has mostly focused on small-scale testbeds and low-dimensional learning techniques; however, many challenges remain. Key among them are authentication techniques robust to a wide variety of channel variations over time and supporting a vast population of devices.

In this work, we advance the state of the art by presenting (i) the first massive experimental evaluation of DL RFFP and (ii) new data augmentation techniques for LoRa designed to counter the degradation introduced by the wireless channel. Specifically, we collected and publicly shared more than 1TB of waveform data from 100 bit-similar devices (with identical manufacturing processes) over different deployment scenarios (outdoor vs. indoor) and spanning several days. We train and test diverse DL models (convolutional and recurrent neural networks) using either preamble or payload data slices. We compare three different representations of the received signal: (i) IQ, (ii) amplitude-phase, and (iii) spectrogram. Finally, we propose a novel data augmentation technique called *DeepLoRa* to enhance the LoRa RFFP performance. Results show that (i) training the CNN models with IQ representation is not always the best combo in fingerprinting LoRa radios; training CNNs and RNN-LSTMs with amplitude-phase and spectrogram representations may increase the fingerprinting performance in small and medium-scale testbeds; (ii) using only payload data in the fingerprinting process outperforms preamble only data, and (iii) *DeepLoRa* data augmentation technique improves the classification accuracy from 19% to 36% in the **RFFP challenging case of training on data collected on a different day than the testing data**. Moreover, *DeepLoRa* raises the accuracy from 82% to 91% when training and testing 100 devices with data collected on the same day.

MobiHoc '21, July 26–29, 2021, Shanghai, China

ACM ISBN 978-1-4503-8558-9/21/07…$15.00
https://doi.org/10.1145/3466772.3467054

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**; • **Security and privacy → Mobile and wireless security**; • **Networks → Network experimentation**.

KEYWORDS

Radio Fingerprinting, Deep Learning, LoRa, Datasets, Testbed.

ACM Reference Format:
Amani Al-Shawabka[†], Philip Pietraski[‡], Sudhir B Pattar[‡],, Francesco Restuccia[†] and Tommaso Melodia[†]. 2021. *DeepLoRa*: Fingerprinting LoRa Devices at Scale Through Deep Learning and Data Augmentation. In *The Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '21), July 26–29, 2021, Shanghai, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3466772.3467054

1 INTRODUCTION

The massive scale of the IoT – expected to reach 30B devices by 2025 – implies that tens of thousands of tiny, low-power nodes will be deployed under large geographical areas for sensing and communication purposes [7]. Thanks to the promise of low-power communications (up to 10 years of battery lifetime) with very long-range connectivity (up to tens of kilometers), LPWAN protocols such as LoRa [6], Sigfox [32], and NB-IoT [11] are gaining significant momentum in both the industrial and academic communities [2]. For useful surveys of LPWANs, we refer the reader to [18, 26].

Due to the extremely low computation capabilities, the success of LPWANs relies on performing on-board operations – for both communication and computation – in the most efficient and effective way possible [13]. Indeed, LoRa relies on a very simple and low-power communication scheme called Chirp Spread Spectrum (CSS), allowing low-rate long-distance communications with throughput in the order of tens of kilobytes per second. The simplicity of LoRa can lead to security issues. Most of the existing work has focused on improving its proprietary communication protocol to improve throughput [27] or energy consumption [29]. However, a critical aspect that has so far been mostly neglected is designing scalable and reliable authentication techniques for LoRa devices without relying on energy-hungry cryptography [20]. Classic authentication techniques are off-limits in low power LoRa application since they would severely diminish the battery lifetime, and in most cases, exceed the computational capability that a tiny LoRa device can offer. To address this vital issue, authentication techniques based on RFFP

have received a surge of interest from the research community [17]. RFFP leverages tiny imperfections in the analog circuitry of radio devices, *e.g.*, IQ imbalance, frequency/sampling offsets, to derive a unique "signature" of the device, which can hardly be imitated by an adversarial device [19]. Most importantly, RFFP operates on existing device transmissions without impacting the device operations, and thus, it has no impact on the device's required computational or energy resources. The Radio-Frequency Machine Learning Systems (RFMLS) program by the Defense Advanced Research Projects Agency (DARPA) [8], focused on RFFP, is just one example to attest to the interest that this topic is generating in the community.

Recently, CNNs have been proposed to fingerprint radios through deep learning of the hardware impairments [24]. CNN's are becoming more and more popular in the wireless community [15, 19, 20], thanks to their ability to avoid manual – and thus, cumbersome and necessarily sub-optimal – feature extraction techniques, such as Zigbee's O-QPSK modulation in [17] or with the WiFi training symbols in [5]. Another downside of waveform specific methods is that the derived algorithm is highly protocol-dependent and not entirely applicable to general waveforms. Conversely, using "raw" (*i.e.*, unprocessed) IQ 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 [31].

Challenges. One of the most crucial challenges in radio fingerprinting is the wireless channel's time-varying variations introduced to the IQ samples that are typically fed to the decision maker. The channel "action" significantly decreases radio fingerprinting accuracy. The authors in [1] carried out an extensive experimental campaign on 20 WiFi devices. They illustrated that the wireless channel significantly degrades the classification accuracy, i.e., the fingerprinting accuracy of equalized IQ dataset dropped from 85% (train-and-test-one-day) to 9% (train-one-day-test-another). Another crucial challenge in the RFFP domain is the testbed scalability; existing work in LoRa RFFP has mostly focused on small or medium-scale testbeds and low-dimensional learning techniques. However, these problems are clearly best addressed through a large-scale data collection campaign. Today, large-scale datasets of LoRa transmissions in rich and diverse indoor and outdoor environments are missing.

This paper attempts to address the above challenges and to create a benchmark dataset for RFFP. We also propose newly designed data augmentation techniques that improve the LoRa RFFP by mitigating the wireless channel's impact. We propose *DeepLoRa* as a distinct data augmentation technique. This technique applies a wide variety of multipath channels generated and used as complex-valued finite input response (FIR) filter taps. Applying these synthetic FIR filters to the collected LoRa samples during the DL training phase increases the diversity of the propagation channels. According to statistical learning theory, data augmentation regularizes the deep learning models and develops generalizations [9]. Through *DeepLoRa*, we expect our models to become more robust to channel variations by "forcing" them to learn to recognize specific transmitter impairments rather than the channel conditions under which these were collected. To the best of our knowledge, this paper, for the first time, improves the RFFP worst-case scenario (train-one-day-test-another) using data augmentation.

Contributions. Our paper makes the following contributions:

• **Massive Data Collection Campaign.** We conduct a massive data collection campaign to assess the RFFP performance in authenticating 100 bit-similar LoRa devices, thus generating a dataset that will continue to grow. We train different NN models (two CNNs and three RNN-LSTM models) over different deployments (outdoor vs. indoor vs. mix of environments) and spanning several days. We label our dataset using SigMF-metafiles. Moreover, we extend this format to incorporate a new extension designed to capture environmental and LoRa specific information. Each SigMF record consists of (i) the binary file containing the IQ samples and (ii) the extended SigMF metadata file[10]. As a significant additional contribution both SigMF records used in this paper are made available on *https://www.interdigital.com/data_sets/lora-radio-data.*

• **Study of the impact of various packet components on the RFFP accuracy.** We customize our DL models' input by extracting the preamble, and the payload data from LoRa received packets. Then we investigate the impact of training and testing a DL model using preamble data only or payload data only. This process can help identify which part of the packet holds specific device hardware impairments that are most valuable in increasing the RFFP accuracy and expedite the training process to avoid training models with the full packet data.

• **Comparison of different representations.** We employ different representations of the received signal, (i) raw IQ, (ii) amplitude-phase, and (iii) spectrogram. Each representation highlights distinct characteristics in the time and frequency domains. By analyzing these divers representations against varying testbed scales and NN models, we shed light on which representation is a good fit for each specific setup. For example we show that the spectrogram representation results in a better RFFP performance in small-scale testbeds using RNN-LSTM NNs, while amplitude-phase and IQ representations provide better robustness in medium and large-scale testbeds.

• **DeepLoRa.** We propose *DeepLoRa*, a novel data augmentation technique based on ITU-R channel models, to enhance the reliability and robustness of the RFFP algorithms. This augmentation technique is shown to improve performance regardless of which day the test data was collected. The key innovation in *DeepLoRa* is the ability to apply a wide variety of ITU standard multipath channels generated and employed as FIR filter taps; this action extends the propagated channels' diversity of the received signal. To evaluate the scalability of *DeepLoRa*, we conducted extensive experiments using various modeling scenarios over different fading distributions (Rayleigh and Rician) and among several signal-to-noise powers.

2 RELATED WORK

Deep learning in LPWANs. Neural Networks (NNs) succeeded in improving the fingerprinting accuracy of the wireless devices by analyzing and tracking the unique hardware impairments emitted from the radios during the wireless transmissions process [19, 24]. NNs are employed to identify the radio's unique hardware impairments and patterns by exploring a massive amount of raw data emitted from them. In most cases, IoT networks have a dense deployment, and applying DL techniques to fingerprint these IoT devices benefit the devices' authentication process, security, and accessibility to the services. The authors in [21] proposed a supervised

per-symbol classification methodology, including MLPs and CNNs, to fingerprint 22 LoRa devices with three different chipset vendors. The fingerprinting accuracy for these three chipset vendors was 99% to 100%. Additionally, they used a zero-shot learning model; however, they could not distinguish other chipset vendors when training the model with devices from one vendor. Their testing accuracy degraded to 22% and 26% using different channel conditions and sampling rates. In [14] the authors compared the CNN, DNN, and LSTM networks in fingerprinting 6 ZigBee devices over several SNR levels to guarantee the proposed models' resilience to various wireless channel conditions. Experimental results demonstrate high accuracy using the DNN method. However, a newer paper[12] showed promising results for fingerprinting radio modulation based on LSTM. All previous works in LPWANs did not consider (i) evaluating the RFFP using a large-scale indoor and outdoor data collection campaign, (ii) testing with samples collected on a day different than the training day using bit-similar devices and without altering the collected samples, and (iii) excluding any useless information of the received packet that is not helping in expediting the DL process.

Data Augmentation in LPWANs. The Computer Vision and Natural Language Processing (NLP) domains use data augmentation techniques to improve the deep learning models' robustness and reduce the models' overfitting. To the best of our knowledge, no prior work has investigated data augmentation on LPWAN. However, there are few works related to data augmentation in radio fingerprinting—a recent work [12] applied data augmentation on deep learning-based radio modulation classification. The authors considered three methods to apply data augmentation by rotating, flipping, and adding Gaussian noise to the IQ samples. They improved the classification accuracy using the three methods. This work is different from our work since they classify and augment data collected using various modulation schemes, each scheme defined by district constellations characteristics. In contrast, in our study, we classify and augment data collected from bit-similar devices and transmitted using one modulation scheme. The authors in [25] employed two approaches for WiFi RFFP data augmentation: (i) the first approach augments the transmitter data "pure signals without channel distortion." The authors then simulated the device-specific RF impairments and indoor wireless channel instances before classifying the synthetically created devices, and (ii) their second approach augments "over-the-air" transmitted signals using samples collected in the same days to train and test the model. In our work, we did not embed any RF hardware impairments to the transmitted signal as in their first approach. While in their second approach, the authors did not consider the worst-case scenario, which includes testing a large scale of bit-similar devices using data collected in a day different from the training day "Train one day, test another." As per our knowledge, we are the first to report an improvement of this worst-case scenario. Our proposal expedites the DL process by specifying and extracting the part of the packet that holds the most remarkable impairments and uses it as an input to the DL model. We exclude any device identification in the received signal to ensure that we do not learn the device-specific ID. We use different representations of the received signal to train our models IQ, amplitude-phase, and spectrogram. Our paper presents a new data augmentation technique *DeepLoRa* that successfully improved the RFFP performance when testing with samples collected on a day different than the training day.

2.1 Background on LoRa Technology

LoRa operates under the Industrial, Scientific, and Medical (ISM) radio frequency bands such as 868 MHz in Europe, 915 MHz in North America, and 923 MHz in Asia. LoRa refers to the physical layer, and it does not involve any encryption method. Typical LoRa deployment uses a star network topology and uses a CSS modulation technique and defines a "chirp" as one symbol [23].The LoRa signal is very robust to noise, interference [26]. The LoRa signal is scalable to different Bandwidths (BWs), Spreading Factors (SFs), and code rates (CR). Optimizing the CR, SF, and BW values trade off the data rate for range. Increasing the SF or lowering the CR allows more extended coverage at the expense of reducing the data rate. LoRa transmission can be customized using different parameters as follows:

• **LoRa BW:** it is scalable to three BW settings 125kHz, 250kHz, and 500kHz. Higher BW provides a higher data rate but degrades the sensitivity. A LoRa symbol is one such chirp spanning the BW. The chirp is cyclically shifted (modulated) by one of 2^{SF} values to carry SF bits of information. Each of the possible shifts is referred to as a chip. The chip rate is equal to the radio BW.

• **LoRa SF:** LoRa employs six different SFs ranging from 7 to 12; the SF represents the ratio between the symbol and chip rate. The SF also trades data rate with the transmission range. Increasing the SF improves the transmission range; however, the packet duration will also increase, thus- lowering the data rate. For example, if SF equals 7 (each symbol carries seven bits but symbol lasts 2^7 chips), then the length of the chirp is $2^{SF} = 128$ chips or 7/128 bits per chip =8.8k bps.

• **Coding Rate (CR):** Forward error correction (FEC) techniques applied to further improve the receiver sensitivity. LoRa supports four CR values between 1/2 and 1.0. The CR equals $4/(4+n)$, with $n \in \{1, 2, 3, 4\}$. FEC technique protects LoRa transmission from interference. Increasing the CR enhances the protection but lowers the information bit rate.

Customizing these parameters determines the radio energy consumption, the transmission range, and the resilience to noise [4]. LoRa Frame structure illustrated in Figure 1 starts with a preamble that is composed of (i) multiple (typically 8) un-modulated up-chirps employed to detect the start of packet, (ii) two modulated up-chirps used for frame synchronization, and finally, (iii) 2.25 un-modulated down-chirps for frequency synchronization. An optional physical Header and a Header CRC follow the preamble. The final part consists of the physical payload and the payload CRC. This paper focuses on extracting the preamble and frame payload data. The reader can refer to [30] for more details about LoRa frame.

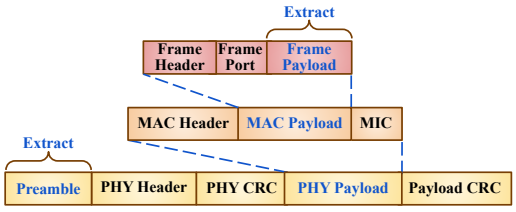


Fig. 1: LoRa Frame structure: Extracting Preamble and Payload.

2.2 Preamble and Payload Extraction Methodology

We use the Gnuradio gr-lora implementation by Robyns [22] to receive LoRa radio signals using a Software Defined Radio (SDR). The gr-lora receiver chain, as illustrated in Figure. 2 starts with frame synchronization, including (i) sampling and filtering, (ii) preamble detection, and (iii) synchronization to ensure detection of the preamble and estimating the Sampling Time Offset (STO) and Carrier Frequency Offset (CFO). Frame synchronization is followed by chirp demodulation, Gray de-mapping, de-interleaving, Hamming decoding, de-whitening, and CRC. The gr-lora implementation details are discussed briefly in [23]. We modified this gr-lora code to extract the preamble and payload IQ data as shown in Figure 1. All fields that carry address information are removed. Preamble and payload data are employed separately to determine which holds more prominent features that can improve the learning process. The preamble and payload high-level detection process is shown in Figure 2.

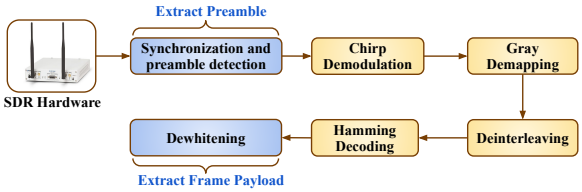


Fig. 2: The gr-lora receiver chain.

3 DEEP LEARNING AND DATA AUGMENTATION METHODOLOGY

We considered two popular deep learning approaches with distinct architectures; a CNN and an RNN. The intention behind using these different architectures is to compare their performance in large-scale and in diverse environments. We illustrate the CNN and RNN-LSTM architectures in Section 3.1. We describe the IQ representations in Section 3.2. Then, Section 3.3 presents *DeepLoRa*, a customized data augmentation technique based on ITU-R channel models. Finally, we describe the performance metrics used in this paper in Section 3.4.

3.1 NN Models Architecture

CNN. We consider two CNN models that differ in the dimension of the convolutional layers (ConvLayers).

• **1D CNN:** Figure 3 shows the architecture of the 1D CNN model used in this study, inspired by a well-known AlexNet CNN. The model consists of two one-dimension (1D) ConvLayers with rectified linear units (ReLU) activation function. The first ConvLayer operates with 128 filters of size 1x7, while the second ConvLayer also operates with 128 filters but each with 1x5 kernel size. The I and Q data are presented as separate channels to the 1D convolutions, i.e., each channel has separate convolutions, but the convolutions' results are summed at the output. The two ConvLayers are followed by a 1x2 MaxPooling layer. This setup was repeated five times and then fed to three fully connected (FC) layers with ReLU activations and 256, 256, and 128 neurons, respectively. Finally, to generate the radio classification probabilities, an FC layer with a softmax activation function is used.

• **2D CNN:** The structure is similar to 1D CNN model, except a two-dimensional ConvLayer is used. Based on this, the kernel size and the MaxPolling layer dimension were modified to match the change. For completeness, we note that the 1D CNN was implemented with Keras/TensorFlow, while the 2D model was implemented with PyTorch.

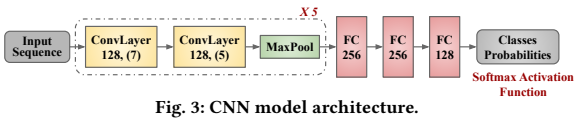


Fig. 3: CNN model architecture.

RNN-LSTM. An RNN is a generalization of a feed-forward neural network designed to retain and use information over a sequence of inputs. The RNN output depends on its current input and the computation from the previous input using feedback connections. However, training basic RNN using backpropagation through time suffers from exploding or vanishing gradients. Long Short Term Memory (LSTM) is a particular class of the RNN that helps solve the exploding/vanishing gradient problems [28]. Each LSTM cell consists of current input data, short-term memory from the previous cell, and long-term memory. The cell uses filters as gates to decide what memory to keep or discard at each cell before moving to the next one. Figure 4 shows the LSTM model architecture used in this work. We analyze the performance of three LSTMs applied to fingerprint LoRa radios. The goal is to analyze the trade-offs between the LSTM network depth and classification accuracy. Our LSTM models include one, two, or three LSTM layers, each with memory cells that equals the input slice length. The output is then fed to an FC layer, and finally, the output layer, which is another FC layer with a softmax activation function, produces the radios classification probabilities. A dropout with a probability of 0.5 is used after each LSTM layer to reduce overfitting. *LSTM models were implemented using Keras with TensorFlow backend.* We slice our dataset using the sliding window technique described in [1], and then we feed it to the LSTM network as a consecutive modulated in-phase (I) and quadrature (Q) sequences as shown in Figure 4. Each slice is mapped to a its device label during the training phase.

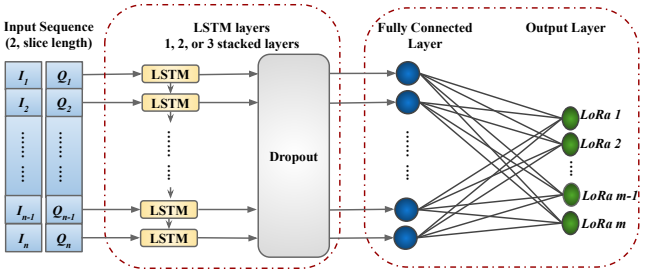


Fig. 4: LSTM model architecture.

3.2 IQ Dataset Representation

We reproduce our dataset in three different representations to recognize each representation characteristics' impact in the training and testing phases. We employ the following forms:

• Raw IQ: Figure 5 shows raw IQ representation in the time-domain for one device on different days and environments. The

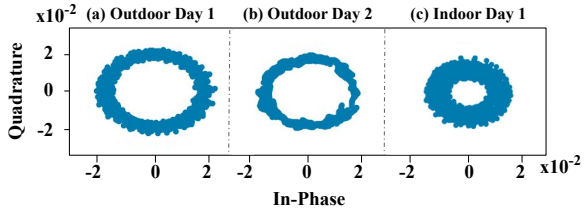


Fig. 5: Scatter representation of the received IQ samples of one device in different environments and spanning different days.

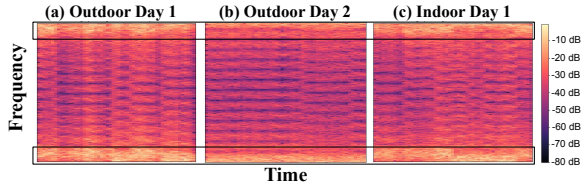


Fig. 6: Logarithmic-Frequency power spectrogram representation. One LoRa device in different environments and spanning different days.

scatter representation varies from day to day, even under the same environment.

• Spectrogram representation: a Short-Time Fourier Transform (STFT) is employed to convert the IQ raw dataset to a time-frequency dataset. The generated dataset describes both the time and frequency characteristics of the received signal. Equation 1 transforms an IQ sequence X to 2D time-frequency S representation. Where W is a window function, with a center that is located at the time index τ

$$S = \int_{-\infty}^{\infty} XW(t-\tau)e^{-j2\pi ft} \tag{1}$$

We generate this representation using scipy.signal python package. Figure 6 illustrates how the logarithmic spectrogram representation varies from day to day and in different environments.

• Amplitude-phase representation: (2), and (3) are used to convert the received IQ samples to their amplitude and phase representations, respectively. Where A and ϕ represent the amplitude and phase of the received signal, respectively.

$$A = \sqrt{I^2 + Q^2} \tag{2}$$

$$\phi = \arctan Q/I \tag{3}$$

We analyzed the impact of these different representations in the NN performance in subsection 5.1.

3.3 Data Augmentation Technique

Data augmentation techniques in the ML domain are used to synthetically expand the training or testing datasets by applying transformation on the existing samples. The newly generated dataset is (i) identically distributed and (ii) not independent from the original dataset. According to the statistical learning theory, data augmentation regularizes the deep learning models and develops generalizations [9]. There are few related works on data augmentation for radio frequency fingerprinting in the literature. To the best of our knowledge, we are the first to analyze the implementation of data augmentation on LPWANs LoRa radios. We propose *DeepLoRa* channel modeling as a novel data augmentation technique. A wide variety of multipath channels are generated and used as complex-valued FIR filter taps, then applied to the collected samples. Increasing the diversity of the propagated channels during the training phase, we expect the model to be more robust to channel variations by forcing it to learn transmitter impairments rather than channel conditions. DeeploRa creates channel taps to transform the original dataset using FIR filters. These taps are drawn from distributions based on the ITU outdoor-to-indoor pedestrian, and vehicular–high antenna channels. Indoor channel models are also available, but the delay spread is too small to have much effect on the data. Each channel realization is statistically independent, i.e., time correlation is removed to prevent correlated input sequences during training. The data augmentation also includes the addition of AWGN to produce augmented data with a variety of SNR; this process enhances the model robustness to the SNR. Similarly, the model is forced to learn transmitter impairments rather than SNR.

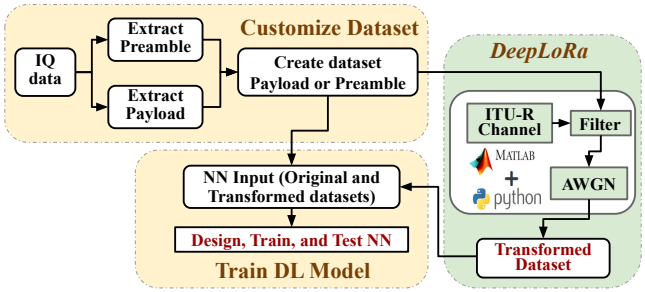


Fig. 7: *DeepLoRa* Data augmentation with channel modeling technique.

Channel modeling is a technique to predict what will happen to the transmitted signal as it propagates through the radio channel. However, our *DeepLoRa* intention is not to estimate the receiver performance in realistic environments but rather to impart further propagation effects into signals collected under 'good' channel conditions and high SNR to enhance the learning generalization.

Channel models specified in ITU-R recommendation M.1225 are commonly used as a set of empirical channel models. The recommendation defines three different test environments (i) indoor office, (ii) outdoor-to-indoor pedestrian, and (iii) vehicular–high antenna. In this work, we use (ii) and (iii). The recommendation also specifies two different delay spreads for each test environment (i) channel A-low delay spread and (ii) channel B-average delay spread [16]. We use PA, PB (pedestrian channel A and channel B), and VA, VB (vehicular–high antenna channel A and channel B).

These ITU channels are simple and reasonably realistic for Single-Input and Single-Output (SISO) conditions. The channels consist of a complex-valued FIR filter with up to 6 nonzero taps. Tables 1 and 2 describe the tap characteristics in time units with the first tap having delay equals 0 and a normalized power such that the largest tap average power is 0 dB. Each active tap is drawn from an independent zero-mean complex Gaussian distribution (i.e., Rayleigh fading). Once the channel selects parameters, the FIR filter is static for the duration of a packet or multiple packets if there is more than one packet in the given RF IQ sample stream.

We implement our *DeepLoRa* as a data augmentation technique based on IUU-R channel models using Python-Pytorch joined with MATLAB. We create a Python package to import MATLAB fading

Table 1: ITU Channel Model for Outdoor to Indoor Pedestrian Environment.

TAP	Channel A		Channel B	
	Relative Delay (ns)	Average Power (dB)	Relative Delay (ns)	Average Power (dB)
1	0	0	0	0
2	110	-9.7	200	-0.9
3	190	-19.2	800	-4.9
4	410	-22.8	1 200	-8.0
5	-	-	2 300	-7.8
6	-	-	3 700	-23.9

Table 2: ITU Channel Model for Vehicular Environment.

TAP	Channel A		Channel B	
	Relative Delay (ns)	Average Power (dB)	Relative Delay (ns)	Average Power (dB)
1	0	0	0	-2.5
2	310	-1	300	0
3	710	-9	8 900	-12.8
4	1 090	-10	12 900	-10.0
5	1 730	-15	17 100	-25.2
6	2 510	-20	20 000	-16.0

channels during the deep learning process. This package will create an FIR filter using one or more of the channel models illustrated in Tables 1, and 2 to transform the original dataset.

The FIR is described by a finite sequence ϕ of M *filter taps*, i.e., ϕ=(ϕ_1,ϕ_2,...,ϕ_M). Given input $\mathcal{X} \subseteq \mathbb{C}^N$, where N is a set of consecutive I/Q samples that constitute an input to the classifier, then for any input $\mathbf{x} \in \mathcal{X}$, the filtered n-th element $\hat{x}[n] \in \hat{\mathbf{x}}$ can be written as

$$\hat{x}[n] = \sum_{j=0}^{M-1} \phi_j x[n-j] \tag{4}$$

ϕ_j is the set of tap weights given by:

$$\phi_j = \sum_{k=1}^{K} a_k sinc([\tau_k/T_s - j]) \tag{5}$$

Where, T_s is the period of input sample to the channel, τ_k is the set of path relative delays $1 \leqslant k \leqslant K$ where, K is the total number of multipath fading channel, and a_k is the set of complex path gains of the multipath fading channel and calculated using Sum-of-sinusoids technique.

Figure 7 shows a high-level overview of the implementation process, it illustrates how we apply data augmentation using the channel modeling technique. First, we process the collected data using the sliding window technique described in [1] to generate the training, validation, and testing datasets. The generated datasets are then fed to the deep learning model. We fed the training dataset to the customized *DeepLoRa* transform to filter the dataset using a predefined channel model to agument the original dataset. We can train our model using the original dataset, the new transformed dataset, or both. Further, we can implement more than one transformation at a time. We have flexibility in changing the designed *DeepLoRa* channel characteristics by identifying various parameters such as the fading type (e.g., Rayleigh or Rician), adding AWGN with different SNR values, composing more than one channel, and other parameters.

3.4 Performance Metrics

To assess the performance of our CNN and RNN models, we use the following performance metrics:

- "Per-slice Training" Accuracy (PSTrain), defined as the number of correctly predicted slices over the total number of trained slices from the training dataset;
- "Per-slice Testing" Accuracy (PSTest), defined as the number of the correctly predicted slices over the total number of tested slices from the testing dataset;
- "Train-and-Test-Same-Day" Accuracy (TTSD), represents the PSTestA over a testing dataset consists of slices collected during the same day used to training model;
- "Train-and-Test-Other-Day" Accuracy (TTOD), represents the PSTestA over a testing dataset consist of slices collected during a day different than the day used to train the model;

4 EXPERIMENTAL CAMPAIGN

Section 4.1 summarizes the data collection methodology and the datasets organization structure, while Section 4.2 briefly describes the experimental setup and testbed layout.

4.1 Dataset Collection Methodology

As a first step, we performed an extensive data collection campaign, where we employed 100 bit-similar Pysense sensors connected using 100 bit-similar FiPy radios; the top part of Figure 8 shows the end device connection. The radios in our setup operate on a carrier of 902.3 MHz in the 915 MHz ISM band. Each transmits ten consecutive bursts of packets in each location. Each burst is separated by 1 second. The burst consists of 100 consecutive packets separated by 10 ms. Each packet contains the payload information carrying the temperature, the humidity, and the device voltage readings. A USRP N-210 equipped with a CBX 1200-6000 MHz daughterboard with 40 MHz instantaneous bandwidth is synchronized to receive the transmitted packets in the data collection testbed. The data is stored in two files (i) a dataset file, a binary file of the recorded digital samples, and (ii) the SigMF metafile, which contains information that describes the dataset in plain-text JSON format. Our binary and meta format is an extension of, and compatible with the SigMF specifications[10]. Moreover, we extend the SigMF meta format to incorporate LoRa specific details.

4.2 Experimental Testbeds

Indoor testbed: This testbed, shown in the bottom part of Figure 8, is an open-access wireless testbed based on a grid of 8x8 VERT2450 antennas mounted on the ceiling of 2240 square ft indoor office-space environment. Each of the 64 antennas is cabled to a programmable SDR through low-attenuation coaxial cable enabling sub-6 GHz 5G-and-beyond spectrum research. This testbed contains 24 SDRs controlled by 12 Dell Power Edge R340 running Ubuntu 16.04 LTS computational servers[3]. In this initial dataset, data is collected from only one of these antennas, as indicated in Figure 8.

Outdoor testbed: To collect outdoor data, we repeated the indoor experiment but move all LoRa devices outside the building and replicate the indoor transmission scenario. The receiver and the

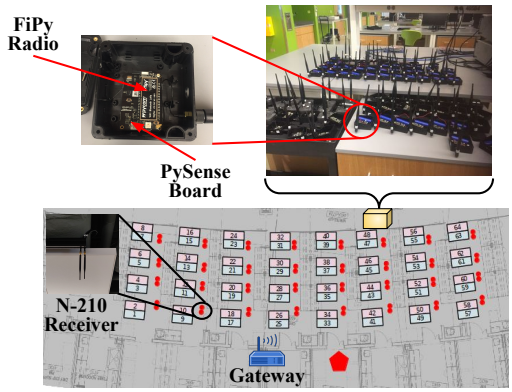


Fig. 8: LoRa Experiment: Indoor setup.

gateway were kept inside the building. The outdoor experiment was conducted in a residential area.

5 EXPERIMENT RESULTS

5.1 Performance Comparison

CNN vs. LSTM. Before training our models, we extracted the preamble and the payload dataset. We sliced and partitioned our dataset into three independent components (i) training, (ii) validation, and (iii) testing datasets. This section evaluates both CNN and LSTM models' performance over this dataset; this part was completed using the extracted IQ payload dataset collected in an outdoor environment.

Table. 3 compares the training dataset performance of the CNN and RNN-LSTM models described in section 3 for 10, 20, 50, and 100 devices. We elected the candidate devices for each model by keeping a constant distance between these devices. *e.g.*, for the 10 devices model, we utilized the dataset collected from devices labeled as [10, 20, ..,100]. Table. 3 illustrates that increasing the number of devices degrades the training accuracy for all but the 2D CNN, especially for the RNN-LSTM models. The worst decrease was from 92% to 45% when training with LSTM_1 model. To evaluate

Table 3: NN models PSTrain% performance while classifying the payload IQ dataset of 10, 20, 50, and 100 LoRa devices.

Model	10	20	50	100
LSTM_1	92	75	58	45
LSTM_2	97	91	84	64
LSTM_3	89	84	70	77
1D CNN	99	97	89	82
2D CNN	99	99	99	99

the model, we used two methodologies (i) testing with dataset collected on the same day of the training dataset (*TTSD*), and (ii) testing with a dataset collected on a different day than the training dataset (*TTOD*). *TTOD* can better judge the robustness of the model. However, *TTSD* performs better than *TTOD*. The non-stationary action of the wireless channel degrades the *TTOD* performance. This section explains the NN architecture's role in mitigating part of the channel impact to enhance the model robustness using LoRa IQ dataset representation in various setups. Results indicated in Table 4 illustrate that the CNN models perform better than RNN-LSTM models in both *TTSD* and *TTOD* matrices; Moreover, the 1D CNN model outperforms the 2D CNN. Again raising the number of devices decreases both metrics. We investigate the RFFP of the worst-case scenario using bit-similar devices in a non-stationary wireless channel condition and a dataset collected on different days. However, increasing the number of the classified devices in such a setup develops the chance of altering some radios impairments and confuses the NN. Therefore, further investigations are subject to future work.

Table 4: CNN and RNN-LSTM *TTSD* and *TTOD* performance while classifying 10, 20, 50, and 100 LoRa devices using IQ payload datasets

Model	TTSD%				TTOD%			
	10	20	50	100	10	20	50	100
LSTM_1	88	62	55	51	14	9	4	2
LSTM_2	91	72	66	55	15	11	5	2
LSTM_3	86	68	63	59	16	12	4	3
1D_CNN	99	99	99	99	28	19	8	5
2D_CNN	99	98	97	97	19	13	7	6

Preamble vs. Payload. Table 5 shows our CNN and RNN-LSTM models TTSD and TTOD performance using the preamble dataset of 10, 20, 50, and 100 LoRa devices located in an outdoor environment, analyzing Tables. 4 and 5 indicate that TTSD and TTOD performance using payload dataset outperforms in almost all models. TTOD results prove that channel conditions degrade the models' performance significantly when testing the models with either preamble or payload data collected on a day different from the data used to train the model. Generally, CNN outperforms the LSTM; however, the TTOD performance of the LSTM models show more robustness using the preamble dataset of 10 and 20 devices although other-day performance remains substantially worse than same-day.

The preamble TTOD results are not good in general, and in some cases it is worse than random guessing. This can be justified by examining the characteristics of the preamble data in the LoRa packet described in section 2.1; the preamble is composed of fixed-length chirps with a fixed frame format structure for all bursts. The preamble chirps are unmodulated up-chirps; were chirp frequency varies linearly with time, and thus- this feature makes the chirp signals resilient to noise, fading, and interference [26]. This feature might cause a drift in some radios' impairments to the point that they begin to look more like another radio than themselves, especially with large-scale testbeds, and decrease the preamble performance-badly.. Medium-scale testbeds (10 and 20 devices) trained with IQ preamble datasets using LSTM models show better robustness than CNN models. This can be explained by (i) the preamble structure fact discussed earlier in this subsection and (ii) the LSTM capability in tracking the hardware impairments rather than the channel impairments using their memory cells.

Indoor vs. Outdoor. Table 6 compares the TTSD performance using preamble and payload datasets with IQ representations collected in an indoor and outdoor environments. LSTM_2 model performance outperforms all other LSTM models. Generally, LSTM models perform better in an indoor environment using the payload dataset, followed by the payload dataset collected in the outdoor environment, the preamble indoor dataset, and the preamble outdoor.

Table 5: CNN and RNN-LSTM *TTSD* and *TTOD* performance while classifying 10, 20, 50, and 100 LoRa devices using IQ preamble datasets

Model	TTSD%				TTOD%			
	10	20	50	100	10	20	50	100
LSTM_1	77	44	47	29	15	9	0.4	0.2
LSTM_2	75	60	41	34	16	5	4	0.1
LSTM_3	63	53	26	24	13	1	1	0.2
1D_CNN	98	97	86	60	4	2	2	1
2D_CNN	98	98	97	82	15	4	4	2

In most cases, the preamble dataset collected in an indoor environment provides better performance than preamble data collected in an outdoor environment. In contrast, the payload and the preamble data collected in the outdoor environment outperform the indoor ones using the CNN models. Generally CNN models work better than LSTM in all scenarios. LoRa preamble dataset is more resilient to noise, and since we are analyzing TTSD performance, then the channel impairments will serve the payload dataset more than the preamble dataset during the classification phase.

Table 6: CNN and RNN-LSTM *TTSD%* performance while classifying 20 LoRa devices using IQ preamble and IQ payload datasets

Model	Preamble		Payload	
	Outdoor	Indoor	Outdoor	Indoor
LSTM_1	44	56	62	82
LSTM_2	60	62	72	88
LSTM_3	53	44	68	82
1D_CNN	97	88	99	99
2D_CNN	98	98	98	89

IQ vs. Amplitude-Phase vs. Spectrogram. We convert the IQ samples to their amplitude-phase and spectrogram representations. Each representation mirrors individual characteristics associated with the received waveform. We aim to understand how each form can motivate the neural network performance.

Figures 9 a, b, c, and d compare the three representations with respect to the testbed size and the employed NN. By dividing our experiment based on the testbed size into three categories: (i) small-scale (less than 10 devices), (ii) medium-scale (from 10- 49 devices), and (iii) large-scale (50 and above). Then we can notice that in the small-scale testbed, amplitude-phase and spectrogram with LSTM models outperform all other combinations. In medium-scale testbeds, the amplitude-phase representation exceeds in all cases except one case where CNN 1 slightly outperforms using IQ representation. Finally, all models and representations had poor performance with the large-scale testbeds, but IQ representations provided marginally better results.

Train Outdoor Test Indoor. This part analyzes our NN models' performance using different signal representations of data collected in different environments. We train a small-scale model with IQ, amplitude-phase, and spectrogram representations of the dataset collected in an outdoor environment. To test the models, we use a dataset collected on other day in an indoor environments.

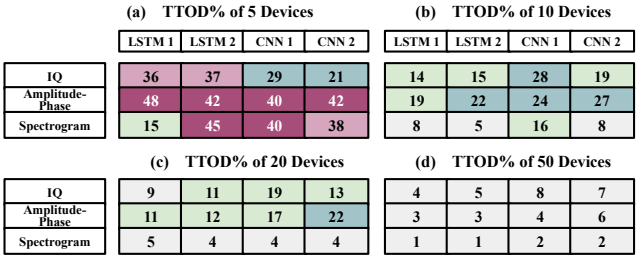


Fig. 9: IQ, amplitude-phase, and spectrogram TTOD% performance per NN model and using (a) small-scale, (b),(c) medium-scale, and (d) large-scale testbeds. The models trained with an outdoor samples while each model is tested using samples collected in an outdoor environment in a day different than the training day.

The result in Table 7 illustrates that amplitude-phase and spectrogram representations outperform the IQ representation in almost all cases. Testing with a dataset collected in a different environment impacted the NN performance badly, especially using IQ representation with the best performance of 8% using LSTM 3 model. While the performance increased using the other representations, and the most remarkable accuracy was 36% using spectrogram with LSTM 3 model.

Table 7: *TTOD%* of classifying 5 devices using IQ, amplitude-phase, and spectrogram dataset. The models trained with outdoor samples, while tested with indoor samples.

Model	IQ	Amplitude Phase	Spectrogram
LSTM_1	5	**24**	**22**
LSTM_2	4	**20**	**36**
LSTM_3	8	16	**26**
1D_CNN	0	0	**23**
2D_CNN	0	7	**20**

Train on two days, Test with the third. We investigate training NN using LoRa dataset collected on two different days; while testing these models using data collected on the third day. The data collected from 10 devices in an outdoor environment in this experiment. To train and test the model, we used the whole received frame without extracting preamble or payload data. We employed IQ, amplitude-phase, and spectrogram representations to train 1D CNN, 2D CNN, and LSTM_2 models. All results show a similar TTOD classification accuracy of 10%, which is the random guess probability of 10—mixing the training dataset using data collected on different days did not improve the training accuracy while testing with data collected on the third day. We repeated the experiment by training the model with samples collected in one day only while keeping the testing samples as-is from the third day; however, we got the same results, 10% for TTOD accuracy. It is important to mention that training NN model with payload dataset enhances the NN robustness, especially when testing with data collected in a day different than the training day as shown in Table 4.

5.2 *DeepLoRa* data Augmentation Performance

DeepLoRa data augmentation technique has a flexible structure, allowing us to design several channel conditions to augment our datasets. We can define the statistics of the augmented data by identifying the (i) channel model environment: pedestrian (ITU-P) or vehicular (ITU-V), (ii) delay spreads: Channel A with low delay

spread or Channel B with average delay spread, (iii) multipath fading type: Rayleigh or Rician, and (iv) AWGN SNR level.

We extensively assessed *DeepLoRa* by conducting several experiments to understand these parameters' impact on DL performance under several circumstances. In this section, we report some of them. The first experiment was carried out to analyze *DeepLoRa* performance with different ITU channel models. We utilized PA, PB, VA, VB, and a composition of all of them. For each ITU channel model, we used (i) Rayleigh fading, (ii) several values of AWGN SNR 10, 20, and 30, (iii) 2D CNN PyTorch model trained with both the training and the transformed dataset using IQ representations.

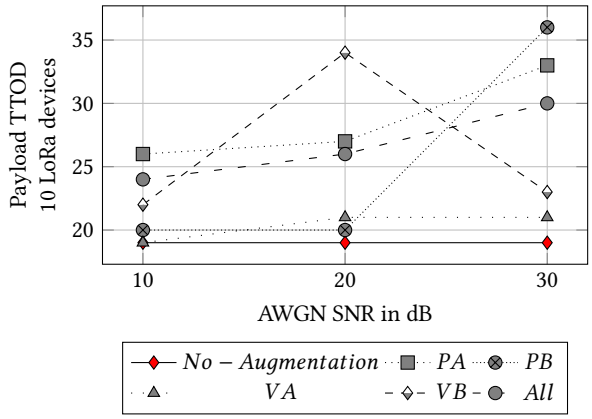


Fig. 10: *TTOD*% (i) without data augmentation, and (ii) with *DeepLoRa* using PA, PB (outdoor-to-indoor pedestrian environments with channels A and B) and VA, VB (vehicular–high antenna environments with channels A and B) and a combination of all (PA, PB, VA, and VB) . Rayleigh fading with three AWGN SNR values 10, 20, and 30dB.

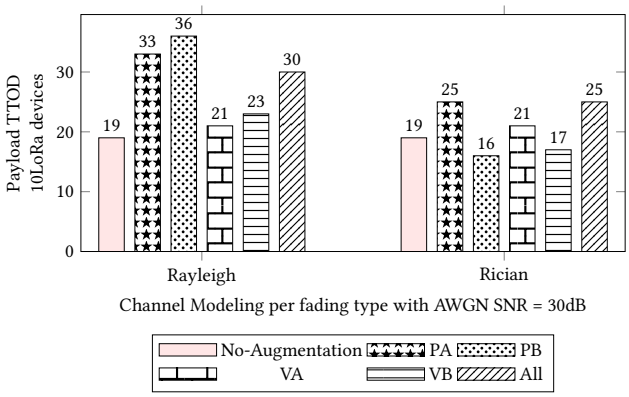
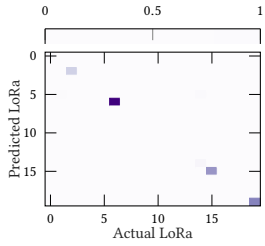


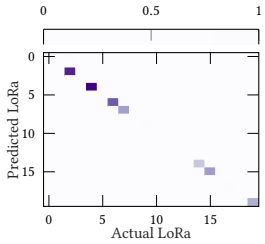
Fig. 11: DeepLoRa *TTOD*%. Rayleigh vs. Rician fading with outdoor-to-indoor pedestrian environments with channels A and B) and VA, VB (vehicular–high antenna environments with channels A and B), and a combination of all (PA, PB, VA, and VB). AWGN SNR = 30dB..

To validate and test the models, we use the validation and testing datasets without applying any augmentation. Figure 10 shows the *DeepLoRa TTOD* performance using 10 LoRa bit-similar devices. *TTOD* test our models with data collected in a day different from the data used during the training and validation phases. *DeepLoRa* succeeded in improving the model's generalization performance, despite employing different channel characteristics in each ITU model. The best results occurred using PB channel and AWGN with 30 dB, where the *TTOD* accuracy increased from 19% to 36%, with an 86% percent increase. Figure. 10 illustrates that the AWGN SNR level impacts the channel generalization performance; however, according to the results, all three AWGN SNR values enhanced the *TTOD* performance.

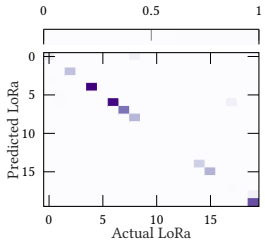
In the second experiment, we assessed *DeepLoRa* performance by implementing different fading types. We evaluated *DeepLoRa* using both Rician and Rayleigh fading channels. This scenario's result shown in Figure 13, indicating that *DeepLoRa* improved the *TTOD* accuracy using Rayleigh and Rician fading. However, the Rayleigh fading provides more benefit than Rician fading in all tested cases. Rician fading introduces less variation in the channel due to the line-of-sight component. Figure 12 shows an experiment conducted to examine the impact of applying PA, PB, VA, and VB Rayleigh channel models to augment a 2D CNN PyTorch model using the payload dataset of 20bit-similar LoRa devices. The results indicate that applying data augmentation using these diverse channels succeeded in enhancing the TTOD accuracy from 13% without augmentation to 22%, 20%, 18%, and 21% with PA, PB, VA, and VB channels augmentation, respectively. Finally, our data augmenta-



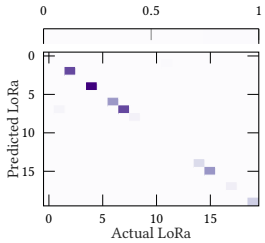
(a) *TTOD* = 13% without augmentation, 3 devices classified correctly



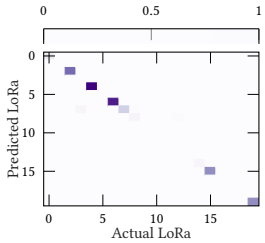
(b) *TTOD* = 22% with PA augmentation, 7 devices classified correctly



(c) *TTOD* = 20% with PB augmentation, 5 devices classified correctly



(d) *TTOD* = 18% with VA augmentation, 5 devices classified correctly



(e) *TTOD* = 21% with VB augmentation, 5 devices classified correctly

Fig. 12: *TTOD*% of IQ payload data collected in outdoor environment. 12a without data augmentation; 12b, 12c, 12d, and 12e with DeepLoRa using outdoor-to-indoor pedestrian environments with channels A and B) and VA, VB (vehicular–high antenna environments with channels A and B). Rayleigh fading, and AWGN SNR 30 dB.

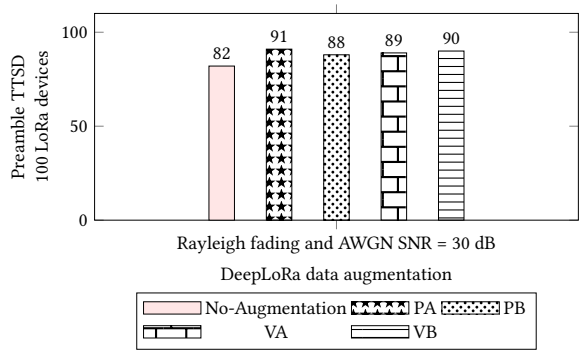


Fig. 13: *TTSD* performance using the preamble dataset of 100 LoRa devices. Without data augmentation and with DeepLoRa data augmentation using PA, PB (outdoor-to-indoor pedestrian channel A and channel B test environments) and VA, VB (vehicular–high antenna channel A and channel B test environments), with Rayleigh fading, and AWGN SNR 30 dB.

tion model succeeded in enhancing the *TTSD* performance of a 2D CNN model trained, validated, and tested using a preamble dataset collected from 100 devices in an outdoor environment. Figure 13 illustrates the increase of the *TTSD* performance from 82% to 91%, 88%, 89%, and 90% after augmenting the training dataset using the PA, PB, VA, and VB channel models, respectively.

6 CONCLUSIONS

This paper has presented a large and growing dataset for IoT fingerprinting using 100bit-similar LoRa radios collected in indoor and outdoor environments over several days. We study DL RFFP performance using the preamble only and the payload only dataset. We employed different representations for the received signal (i) IQ, (ii) amplitude-phase, and (iii) spectrogram; each representation shows different behavior depending on the testbed size and the adopted NN. We experimented with various CNN and RNN-LSTM deep learning architectures. We propose a novel data augmentation technique DeepLoRa that can mitigate part of the classification accuracy drop when training and testing on different days. To the best of our knowledge, we are the first to improve the TTOD performance in RFFP using data augmentation. The best-case scenario showed an increase from 19% to 36%. The augmentation technique also improves TTSD performance from 82%to 91% using 100bit-similar devices. Data augmentation technique proposed in this paper improved the TTOD accuracy for the worst-case scenario significantly. However, further enhancements to this case are crucial for the practical IoT Fingerprinting applications that rely on deep learning, which will be subject to future work.

ACKNOWLEDGMENTS

This work is supported by InterDigital Communications, United States. The authors would like to thank Dr. David Malone for shepherding the paper.

REFERENCES

[1] Al-Shawabka, A., Restuccia, F., D'Oro, S., Jian, T., Rendon, B. C., Soltani, N., Dy, J., Chowdhury, K., Ioannidis, S., and Melodia, T. Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting. *Proc. of INFOCOM* (2020).

[2] Bembe, M., Abu-Mahfouz, A., Masonta, M., and Ngqondi, T. A survey on low-power wide area networks for iot applications. *Telecommunication Systems* (2019).

[3] Bertizzolo, L., Bonati, L., Demirors, E., and Melodia, T. Arena: A 64-antenna sdr-based ceiling grid testbed for sub-6 ghz radio spectrum research. In *Proc. of the 13th Intl Workshop on Wireless Network Testbeds* (2019).

[4] Bor, M. C., Roedig, U., Voigt, T., and Alonso, J. M. Do LoRa Low-power Wide-area Networks Scale? In *Proc. of the 19th ACM Intl. Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (2016).

[5] Brik, V., Banerjee, S., Gruteser, M., and Oh, S. Wireless Device Identification with Radiometric Signatures. In *Proc. of MobiCom* (2008).

[6] Chiani, M., and Elzanaty, A. On the LoRa Modulation for IoT: Waveform Properties and Spectral Analysis. *IEEE Internet of Things Journal 6*, 5 (2019).

[7] Cisco Systems. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2018-2023 White Paper. https://tinyurl.com/shmk6f2, 2018.

[8] 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.

[9] Hernández-García, A., and König, P. Further advantages of data augmentation on convolutional neural networks. In *Intl. Conference on Artificial Neural Networks* (2018), Springer.

[10] Hilburn, B., West, N., O'Shea, T., and Roy, T. SigMF: The Signal Metadata Format. In *Proceedings of the GNU Radio Conference* (2018), vol. 3.

[11] Hoglund, A., Lin, X., Liberg, O., Behravan, A., Yavuz, E. A., Van Der Zee, M., Sui, Y., Tirronen, T., Ratilainen, A., and Eriksson, D. Overview of 3GPP release 14 enhanced NB-IoT. *IEEE network 31*, 6 (2017), 16–22.

[12] Huang, L., Pan, W., Zhang, Y., Qian, L., Gao, N., and Wu, Y. Data augmentation for deep learning-based radio modulation classification. *IEEE Access 8* (2019).

[13] Ismail, D., Rahman, M., and Saifullah, A. Low-power Wide-area Networks: Opportunities, Challenges, and Directions. In *Proc. of the Workshop Program of the 19th Intl. Conference on Distributed Computing and Networking* (2018).

[14] Jafari, H., Omotere, O., Adesina, D., Wu, H.-H., and Qian, L. Iot devices fingerprinting using deep learning. In *MILCOM* (2018), IEEE.

[15] Jagannath, J., Polosky, N., Jagannath, A., Restuccia, F., and Melodia, T. Machine learning for wireless communications in the internet of things: A comprehensive survey. *Ad Hoc Networks 93* (2019), 101913.

[16] Jain, R. Channel models: A tutorial. In *WiMAX forum AATG* (2007), vol. 10, Washington Univ. St. Louis, Dept. CSE.

[17] Peng, L., Hu, A., Zhang, J., Jiang, Y., Yu, J., and Yan, Y. Design of a Hybrid RF Fingerprint Extraction and Device Classification Scheme. *IEEE Internet of Things Journal 6*, 1 (Feb 2019), 349–360.

[18] Raza, U., Kulkarni, P., and Sooriyabandara, M. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys & Tutorials 19*, 2 (2017).

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

[20] Restuccia, F., D'Oro, S., and Melodia, T. Securing the Internet of Things in the Age of Machine Learning and Software-Defined Networking. *IEEE Internet of Things Journal 5*, 6 (Dec 2018), 4829–4842.

[21] Robyns, P., Marin, E., Lamotte, W., Quax, P., Singelée, D., and Preneel, B. Physical-Layer Fingerprinting of LoRa Devices Using Supervised and Zero-Shot Learning. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (2017), ACM, pp. 58–63.

[22] Robyns, P., Quax, P., Lamotte, W., and Thenaers, W. gr-lora: An efficient lora decoder for gnu radio, 2017.

[23] Robyns, P., Quax, P., Lamotte, W., and Thenaers, W. A multi-channel software decoder for the lora modulation scheme. In *IoTBDS* (2018), pp. 41–51.

[24] Sankhe, K., Belgiovine, M., Zhou, F., Riyaz, S., Ioannidis, S., and Chowdhury, K. ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks. In *IEEE INFOCOM 2019* (2019), IEEE.

[25] Soltani, N., Sankhe, K., Dy, J., Ioannidis, S., and Chowdhury, K. More is better: Data augmentation for channel-resilient rf fingerprinting.

[26] Sundaram, J. P. S., Du, W., and Zhao, Z. A survey on lora networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys & Tutorials 22*, 1 (2019), 371–388.

[27] Tong, S., Xu, Z., and Wang, J. CoLoRa: Enabling Multi-Packet Reception in LoRa.

[28] Wang, Y. A new concept using lstm neural networks for dynamic system identification. In *2017 American Control Conference (ACC)* (2017), IEEE, pp. 5324–5329.

[29] Xia, X., Zheng, Y., and Gu, T. LiteNap: Downclocking LoRa Reception.

[30] Yasmin, R. Integration of lora wide area network with the 5g test network.

[31] Zorzi, M., Gluhak, A., Lange, S., and Bassi, A. From today's Intranet of Things to a future Internet of Things: a Wireless-and Mobility-related View. *IEEE Wireless communications 17*, 6 (2010).

[32] Zuniga, J. C., and Ponsard, B. Sigfox System Description. *LPWAN@ IETF97, Nov. 14th 25* (2016).