# Radio Frequency Fingerprint Identification for LoRa Using Spectrogram and CNN

Guanxiong Shen*, Junqing Zhang*§, Alan Marshall*, Linning Peng†, and Xianbin Wang‡
* Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom
Email: {Guanxiong.Shen, junqing.zhang, alan.marshall}@liverpool.ac.uk
† School of Cyber Science and Engineering, Southeast University, No. 2 Sipailou, Nanjing, China
Email: pengln@seu.edu.cn
‡ Department of Electrical and Computer Engineering, Western University, London, Ontario, N6A 5B9, Canada
Email: xianbin.wang@uwo.ca
§ Corresponding Author

***Abstract*—Radio frequency fingerprint identification (RFFI) is an emerging device authentication technique that relies on intrinsic hardware characteristics of wireless devices. We designed an RFFI scheme for Long Range (LoRa) systems based on spectrogram and convolutional neural network (CNN). Specifically, we used spectrogram to represent the fine-grained time-frequency characteristics of LoRa signals. In addition, we revealed that the instantaneous carrier frequency offset (CFO) is drifting, which will result in misclassification and significantly compromise the system stability; we demonstrated CFO compensation is an effective mitigation. Finally, we designed a hybrid classifier that can adjust CNN outputs with the estimated CFO. The mean value of CFO remains relatively stable, hence it can be used to rule out CNN predictions whose estimated CFO falls out of the range. We performed experiments in real wireless environments using 20 LoRa devices under test (DUTs) and a Universal Software Radio Peripheral (USRP) N210 receiver. By comparing with the IQ-based and FFT-based RFFI schemes, our spectrogram-based scheme can reach the best classification accuracy, i.e., 97.61% for 20 LoRa DUTs.**

***Index Terms*—Internet of Things, LoRa, device authentication, radio frequency fingerprint, convolutional neural network, carrier frequency offset**

## I. Introduction

The Internet of things (IoT) applications are blooming with numerous exciting applications such as connected healthcare, smart cities and intelligent industries [1]. Statista estimated there would be 75.44 billion IoT devices by 2025[1]. Device authentication is critical to safeguard IoT applications for allowing legitimate users to access the network while preventing malicious users [2]. This task is becoming more challenging with the rapid growth of low cost IoT devices. Conventional authentication schemes rely on software addresses such as Internet Protocol (IP) and/or Media Access Control (MAC) addresses, which are prone to be tampered or forged [3]. Once the security credentials are obtained by malicious users, they can masquerade as the legitimate users to access the private data or launch fatal attacks on the IoT networks.

Radio frequency fingerprint identification (RFFI) is a promising authentication scheme that can identify wireless devices from their emitted transmissions [2], [4], [5]. Radio frequency fingerprint (RFF) is originated from the hardware imperfections introduced during the manufacturing process, which is inherent to the analog front-end components. These imperfections deviate slightly from their nominal specifications hence do not affect the normal communication functions; but we can design advanced algorithms to extract them as a device identifier. Similar to a biometric fingerprint, RFF is unique and hard to tamper without tremendous efforts.

RFFI consists of two stages, namely training and classification. During the training stage, an authenticator will collect sufficient wireless packets from devices under test (DUTs), then extract features from the received packets to train a classifier. Various features are considered in previous work including Hilbert spectrum [6], carrier frequency offset (CFO) [7]–[11], inphase and quadrature (IQ) offset [12], spectrum [13], time-frequency statistics [14], phase error [12], and power amplifier nonlinearity [15], etc. During the classification stage, the authenticator will extract the same type of features from received packets, feed them to the trained classifier and infer the device identity.

Compared with conventional cryptography-based security schemes [3], one major advantage of RFFI is that it does not impose any additional computational burden and power consumption on the devices to be authenticated [2], [16]. This is particularly desirable for many IoT applications because most of the end nodes are low-cost with limited computational and energy resources. For instance, RFFI could be utilized in Long Range (LoRa) networks to relieve the severe battery power constraint of LoRa devices.

RFFI can be considered as a multi-class classification problem, hence the most recent development on deep learning could be leveraged [17]–[26]. Manually extracting handcrafted features requires comprehensive knowledge on the adopted communication technology and protocol. In addition, it is difficult to estimate each individual feature accurately as the hardware imperfections are interrelated [27]. Deep learning algorithms can automatically extract features from the received signals and can extract more distinguishable and high-level fingerprints [18]. Deep learning-based RFFI systems are built

[1]https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

with the latest convolutional neural network (CNN) [17]–[25] or recurrent neural network (RNN) such as long short-term memory (LSTM) [26]. Most of them often use the IQ samples as the network input [17]–[21], which may not be the best solution as the signal characteristics is not explicit in the time domain. The system performance can be improved by transforming IQ samples and obtaining more distinguishable signal representations, such as bispectrum [22], [23], differential constellation trace figure [24], Hilbert-Huang spectrum [25], error signal [17], etc. During the classification stage, softmax function is often used in neural networks to return a list of probabilities with respect to the classes, indicating the confidence of the predictions. In some cases, the classifier is not confident about its prediction, i.e, confidence score is low and the probabilities of several classes are quite close. However, there is no existing work to leverage this confidence information to calibrate the uncertain predictions.

As a device authentication scheme, RFFI should remain stable [2]. Robyns *et al.* [28] indicated that the accuracy of their system dropped over time and inferred this was caused by oscillator frequency drift. However, the authors did not provide an in-depth analysis or mitigation methods. Andrews *et al.* [29] experimentally examined the effect of temperature variation on different analog components, e.g. oscillator, power amplifier, phase locked loop, mixer, etc., and concluded that the oscillator is particularly sensitive to temperature fluctuations. While CFO has been successfully used to identify WiFi devices [10], [11], it was also observed that low-cost ZigBee devices have severe CFO variations even within 15 minutes [18], [30]. A comprehensive investigation of the CFO variation on low cost IoT devices and its effect on the RFFI is still missing.

In this paper, we take LoRa as a case study to investigate the above challenges. LoRa is a physical layer standard developed by Cycleo and patented by Semtech in 2014 [31], which has been widely used for long range IoT applications. LoRaWAN, a higher layer protocol for LoRa, relies on cryptography-based schemes for device registration, namely Over The Air Activation (OTAA) and Activation by Personalization (ABP) [32], which are prone to be tampered by malicious users. Therefore, the emerging RFFI technique is promising for LoRa device authentication. To the best knowledge of the authors, there are three papers on the LoRa RFFI [26], [28], [33]. LoRa employes chirp spread spectrum (CSS) modulation which exhibits time-frequency characteristics, which can be explicitly revealed in spectrogram. However, none of them considered the unique modulation techniques of LoRa, which may not be able to reach optimal performance. In addition, LoRa devices are low cost and usually manufactured with cheap components including oscillators. The effect of CFO variation on the LoRa RFFI has never been investigated.

This paper designs a CNN-based RFFI system to classify LoRa devices. We aim to answer three questions: (1) Can we employ a signal representation that is unique to LoRa modulation and improve the classification accuracy? (2) How does the CFO variation affect the RFFI stability and can we mitigate it? (3) Can we leverage the probabilities of the softmax output to further enhance the deep learning-based RFFI? We carried out an in-depth investigation and extensive experiments that involved 20 LoRa devices as DUTs and a Universal Software Radio Peripheral (USRP) N210 software defined radio (SDR) platform as the authenticator to answer these questions. Our contributions are listed as follows.

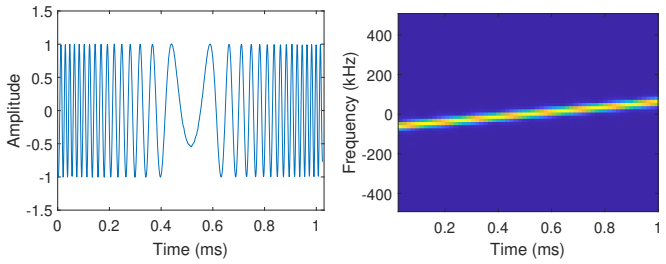
- We experimentally compare three signal representations for LoRa signals, namely IQ samples, Fast Fourier transform (FFT) results and spectrogram. It is found spectrogram can reach the highest accuracy of 96.44% while the IQ samples and FFT can reach 83.36% and 87.36%, respectively. In addition to this, the training time of spectrogram-based model (20 minutes) is much shorter than that of IQ/FFT-based model (one hour), which indicates the training cost can be significantly reduced.
- We experimentally demonstrate that CFO is unstable and degrades the system performance. We established a bespoke setup by connecting a LoRa DUT and USRP with an attenuator to eliminate channel effects. CFO is found to vary in a short time frame but stays relatively stable in long terms. CFO compensation is found to be effective in mitigating the performance degradation, which can improve the classification accuracy from 75.59% to 96.44% for spectrogram CNN-based scheme.
- We design a hybrid classifier based on the softmax output and CFO to further increase the classification accuracy. The CNN may be uncertain when some devices have very similar hardware characteristics and their softmax output probabilities will be close. As the CFO has long term stability, we calibrate the output of CNN according to the estimated CFO. The designed hybrid classifier can significantly improve the system performance, namely from 83.36% to 92.01% in the best case for the IQ-based RFFI.

The rest of the paper is organized as follows. Section II briefly introduces the background of LoRa modulation and spectrogram. Then we present the LoRa receiver operations in Section III. The design details of the RFFI system and the CNN architecture are introduced in Section IV and Section V, respectively. In Section VI, we experimentally demonstrate CFO drift and its effect on the stability of RFFI, and in Section VII the performance of the proposed RFFI systems is thoroughly evaluated in a real wireless environment. The paper is finally concluded in Section VIII.

## II. Preliminary

### A. LoRa Modulation Technique

LoRa employs CSS modulation which uses linear chirps for communications. The linear chirps are also known as linear frequency modulation (LFM) signals, of which the frequency increases or decreases linearly with time. The information in each symbol is encoded to the initial phase of the chirp. A




(a) (b)

Fig. 1. LoRa preamble. (a) Time domain representation of one preamble (I branch). (b) Spectrogram of one preamble.

standard basic chirp $c(t)$ of the RF band can be mathematically expressed as

$$c^{rf}(t) = Ae^{j(-\pi Bt+\pi\frac{B}{T}t^2+2\pi f_c t)} = u(t)e^{j2\pi f_c t}, \quad (1)$$

where $f_c$ is the carrier frequency, $B$ is the bandwidth, $T$ is the symbol duration, and $A$ is the amplitude. The baseband signal can be given as

$$u(t) = Ae^{j(-\pi Bt+\pi\frac{B}{T}t^2)} = Ae^{j\phi(t)}, \quad (2)$$

where $\phi(t)$ denotes the phase of the baseband chirp signal. The symbol duration $T$ is given as

$$T = \frac{2^{SF}}{B}, \quad (3)$$

where $SF$ is the spreading factor ranging from 7 to 12. The instantaneous frequency of $u(t)$ is defined as

$$f(t) = \frac{1}{2\pi}\frac{d\phi(t)}{dt} = -\frac{B}{2} + \frac{B}{T}t. \quad (4)$$

The LoRa standard specifies several basic chirps called preambles in a packet [34]. The preamble is the same in every packet and for any LoRa device type. Fig. 1a shows the time-domain baseband signal (I branch) of one preamble in a LoRa packet.

### B. Short-Time Fourier Transform and Spectrogram

Short-time Fourier transform (STFT) is a well-known time-frequency analysis algorithm which has been extensively utilized to analyze non-stationary signals, including LoRa signals. STFT divides a long signal into short segments and then performs Fourier transform separately on each segment. The discrete-time STFT can be mathematically given as

$$STFT(m,f) = \sum_{n=-\infty}^{\infty} s[n]w[n-mR]e^{-j2\pi f n T_s}, \quad (5)$$

where $s[n]$ is the signal to be analyzed, $w[n]$ is the window function of length $M$, $m$ is the column index of the matrix and $R$ is the hop size. The spectrogram can be given as

$$Spectrogram(m,f) = |STFT(m,f)|^2, \quad (6)$$

where $|\cdot|$ returns the amplitude. Fig. 1b is the spectrogram of one LoRa preamble. Spectrogram can efficiently represent how the instantaneous frequency changes over time, as well as some signal parameters such as bandwidth $B$ and symbol duration $T$.

## III. LoRa Receiver Operation

### A. Signal Reception

The LoRa signal is first received by the receiver antenna, i.e., $r^{rf}(t)$. Then it is down-converted to the baseband by the mixer. The received baseband signal is sampled by an analog-to-digital converter (ADC) to obtain the digital baseband signal $r[nT_s]$, which can be mathematically expressed as

$$\begin{aligned} r[nT_s] =& r^{rf}[nT_s]e^{-j2\pi f_c^{rx} nT_s} \\ =& u'[nT_s]e^{j2\pi f_c^{tx} nT_s}e^{-j2\pi f_c^{rx} nT_s} \\ =& u'[nT_s]e^{j2\pi\Delta f nT_s}, \end{aligned} \quad (7)$$

where $u'[nT_s]$ is the received baseband signal, $T_s$ is the sampling interval, $f_c^{tx}$ and $f_c^{rx}$ are the carrier frequencies of the transmitter and receiver, respectively, and $\Delta f = f_c^{tx} - f_c^{rx}$ is the CFO between them. For the simplicity of notations, $T_s$ is omitted. The digital baseband signal can be rewritten as

$$r[n] = u'[n]e^{j2\pi\Delta f n T_s}. \quad (8)$$

### B. CFO Estimation and Compensation

In this section, we will introduce the CFO estimation and compensation for LoRa signals.

*1) Coarse CFO Estimation:* The ideal instantaneous frequency of the baseband basic chirp, $f_{ideal}[n]$, increases linearly from $-\frac{B}{2}$ to $\frac{B}{2}$. However, there is an inevitable frequency offset, $\Delta f$, in the received baseband signal $r[n]$. Its instantaneous frequency $f[n]$ thus becomes

$$f[n] = -\frac{B}{2} + \Delta f + \frac{B}{T}nT_s. \quad (9)$$

Thanks to the linearity of $f[n]$, the CFO can be roughly estimated by calculating the mean value of $f[n]$ of the received preambles. The estimated CFO $\widehat{f}_{coarse}$ is given as

$$\Delta\widehat{f}_{coarse} = \frac{1}{L}\sum_{n=0}^{L-1} f[n], \quad (10)$$

where $L$ is the symbol length, defined as

$$L = \frac{T}{T_s} = \frac{2^{SF}}{B\cdot T_s}. \quad (11)$$

The received signal can be coarsely compensated by the estimated frequency offset, given as

$$r'[n] = r[n]\cdot e^{-j2\pi\Delta\widehat{f}_{coarse} nT_s}. \quad (12)$$

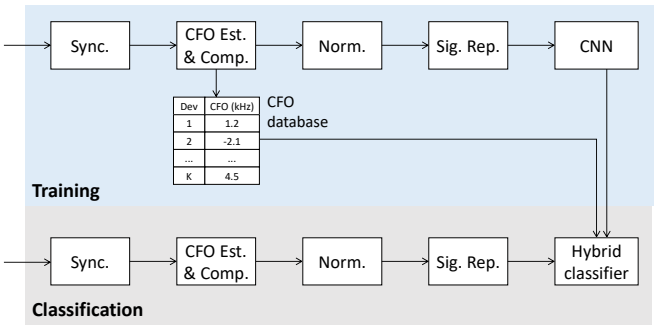



Fig. 2. A CNN-based RFFI scheme. CFO compensation is adopted.

*2) Fine CFO Estimation:* There will be residual frequency offset after the above coarse frequency compensation, hence we further employ a fine estimation algorithm. The residual offset can be estimated based on the repeating property of preambles, given as

$$\Delta\widehat{f}_{fine} = -\frac{1}{2\pi \cdot T_s L} \cdot \angle\Big(\sum_{n=0}^{L-1} r'[n] \cdot r'^{*}[n+L]\Big), \quad (13)$$

where $\angle\cdot$ returns the angle of the variable and $(\cdot)^*$ denotes conjugation. The received signal can be further finely compensated as

$$r''[n] = r'[n] \cdot e^{-j2\pi\Delta\widehat{f}_{fine} n T_s}. \quad (14)$$

As the phase can only be resolved in $[-\pi, \pi]$, the range of CFO that can be estimated by (13) is

$$|\Delta\widehat{f}_{fine}| < \frac{\pi}{2\pi \cdot T_s L} = \frac{B}{2^{SF+1}}. \quad (15)$$

When the LoRa transmission is configured with $SF$=7 and $B$=125 kHz, the estimation capability is within ±488.3 Hz. It is common that the frequency drift of the oscillator of LoRa devices is ±10 ppm [35], approximately 8.68 kHz for an 868 MHz carrier frequency, which is much higher than 488.3 Hz. Hence, the coarse estimation should be employed to limit the residual offset before the fine estimation.

After the coarse and fine CFO estimation, the overall estimated CFO, $\Delta\widehat{f}$, can be represented as

$$\Delta\widehat{f} = \Delta\widehat{f}_{coarse} + \Delta\widehat{f}_{fine}. \quad (16)$$

## IV. RFFI System

The architecture of the proposed RFFI system is shown in Fig. 2. This section will introduce each step in detail.

### A. Synchronization and CFO Compensation

Synchronization detects the signal arrival and locates the packet relying on the repeated preambles, which is a standard process in the communication system. Interested readers please refer to the work in [36] for detailed information.

CFO estimation and compensation are standard procedures in wireless communication systems as well. However, some previous studies did not perform these steps as they used the raw IQ samples directly. Some work also used CFO as one of the RFF features [7]–[11]. However, studies also revealed CFO would cause performance degradation [28]. The effect of CFO on the RFFI for low-cost IoT devices is not experimentally investigated.

We adopted the CFO estimation and compensation algorithms introduced in Section III-B. During the training, a CFO database is generated, which contains the estimated CFO of each DUT. This CFO database will be used for the hybrid classifier which will be introduced in Section IV-E.

### B. Normalization

RFFI systems are not expected to differentiate devices by power differences because signal power is susceptible to distance. Normalization has been a standard process in RFFI. The normalized signal $s[n]$ can be given as

$$s[n] = \frac{r''[n]}{x_{rms}}, \quad (17)$$

where $x_{rms}$ is the root mean square of the amplitude of $r''[n]$.

### C. Signal Representation

Signal representation employs signal processing algorithms to reveal the underlying signal characteristics, which can be better learned by the classifier. This paper only uses the preamble part to prevent the deep learning model learning protocol-specific and data-specific knowledge.

*1) IQ Samples:* IQ samples represent the time-domain signals which are captured from the receiver chain directly. Some previous work aims to design protocol-agnostic RFFI systems without considering the physical modulation schemes so they employ IQ samples as system inputs [17]–[21].

*2) FFT Results:* FFT converts the time-domain signal to the frequency domain. Features that are not obvious in the time domain may be easily observed in the frequency domain. The FFT coefficients are readily available in WiFi OFDM systems [21].

*3) Spectrogram:* The spectrogram can be a better signal representation for LoRa signals since it converts the time domain IQ samples to the time-frequency domain characteristics, which not only provides information in the frequency domain but also reveals how it changes over time. Logarithmic compression of magnitudes was found to be effective in improving the performance and has been a standard strategy in preprocessing spectrograms [37], which is also used in this paper.

### D. Convolutional Neural Network

CNN has attracted many research interests from both academia and industry thanks to its excellent performance in image recognition and computer vision. It can find patterns in the data automatically which eliminates the need for manual feature extraction. CNN is usually composed of convolutional layers, fully connected layers, as well as some pooling layers that reduce the number of parameters to prevent overfitting. The convolutional and pooling layers act as a feature extractor

that directly extracts features from the input data. The extracted high-level features are then fed into the fully connected layers for classification.

In classification problems, softmax function is usually used at the last layer of CNN to map its outputs to a list of probabilities $\mathbf{S} = (S_1, S_2, ..., S_K)$ over all the predicted classes. $S_k$ is the predicted probability of the $k$-th class, which can be mathematically expressed as

$$S_k = \sigma(\mathbf{z})_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for} \quad k = 1, 2, ..., K, \tag{18}$$

where $K$ is the number of classes, and $\mathbf{z} = (z_1, z_2, ..., z_K)$ is the output of the layer before softmax activation. The most common way to make a prediction is to select the class with the highest probability as the final predicted label.

The CNN architectures used in this paper will be elaborated in Section V.

### E. Hybrid Classifier

CNN cannot perfectly distinguish devices whose hardware characteristics are quite similar, particularly when they are from the same manufacturer. Then, the output probabilities of these classes are close to each other, e.g., $S_1$=0.51 and $S_2$=0.49. In this case, simply selecting the device with the highest probability is likely to cause misclassification.

As we will demonstrate later in Section VI and Fig. 6, the mean values of CFO among different days remained relatively stable. Hence, it inspires us to use the estimated CFO to calibrate the CNN predictions.

We propose a hybrid classifier to exclude unreliable predictions derived by the CNN classifier, which is described in Algorithm 1. We first create a reference CFO database for all the $K$ devices during the training stage, namely, $\{\Delta\widehat{f}_k\}$. Then, for each DUT during the classifications stage, we will estimate its CFO, $\{\Delta\widehat{f}_{DUT}\}$, and compare it with the CFO database. The operation can be formulated as a hypothesis test

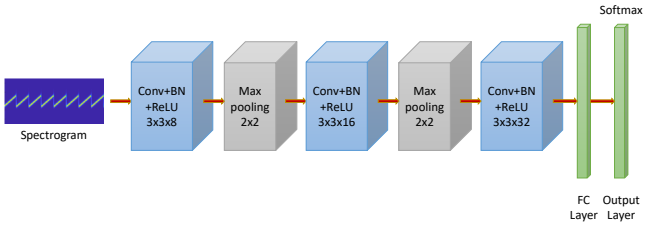
$$\left|\Delta\widehat{f}_{DUT} - \Delta\widehat{f}_k\right| \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda, \tag{19}$$

where $\lambda$ is the predefined threshold based on the range of CFO variations. Hypothesis $\mathcal{H}_1$ means that the packet is impossible to be sent from the $k$-th device due to the large difference between $\Delta\widehat{f}_{DUT}$ and the reference $\Delta\widehat{f}_k$. When this occurs, the probability of $k$-th class, $S_k$, is set to zero. In contrast, hypothesis $\mathcal{H}_0$ means the prediction of CNN is correct, thus $S_k$ maintains the original value. After such calibration, the device with the highest probability in $\mathbf{S}$ is selected as the final predicted label.

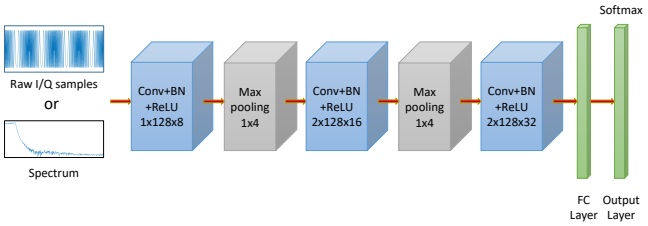
**Algorithm 1** Hybrid Classifier

**INPUT:** $\mathbf{S}$, The softmax output which denotes the probability of each device;
**INPUT:** $\Delta\widehat{f}_{DUT}$, The estimated CFO of the DUT;
**INPUT:** $\Delta\widehat{f}_k$, The reference CFO of the $k$-th device stored in the database;
**INPUT:** $\lambda$, The CFO threshold.
**OUTPUT:** $l$, The eventually predicted label.

```
for k = 1 to K do
    if |Δf̂_DUT − Δf̂_k| > λ then
        S_k = 0
    else
        S_k = S_k
    end if
end for
Select the device with the highest probability in S as the predicted label.
```




(a)




(b)

Fig. 3. CNN architectures. (a) Spectrogram-based model. (b) IQ/FFT-based model.

## V. CNN Architecture

### A. Spectrogram-based CNN

The architecture of spectrogram-based CNN model is illustrated in Fig. 3a. It consists of three convolutional layers of 8, 16, and 32 3×3 filters, respectively. Each convolutional layer is followed by a batch normalization layer, the rectified linear unit (ReLU) activation and a 2×2 max pooling layer with stride 2. The output of the last convolutional layer is then fed into the fully connected layer for classification. The softmax function is used in the last layer of the neural network to output the probabilities for different classes.

Adam is selected as the optimizer for the training process. The initial training rate is set to 0.0003 and drops every 10 epochs with a drop factor of 0.3. The mini-batch size is set to 32.

The spectrogram used in this paper is generated with the rectangular window, with a window length $M$ of 256 and hop size $R$ of 128. Our network is designed for this specific parameter setting.

### B. IQ/FFT-based CNN

We also establish another CNN model which can input the complex IQ samples or FFT results, i.e, consider the real

and imaginary part of the complex signal as two independent dimensions [19]. It is worth noting that IQ samples and corresponding FFT results are complex vectors of the same length. Hence exactly the same CNN can be used for both IQ and FFT data.

The architecture of the IQ/FFT-based CNN is shown in Fig. 3b, which also consists of three convolutional layers and one fully connected layer. The three convolutional layers are composed of 8, 16, and 32 filters, respectively, and the filter sizes are $1 \times 128$, $2 \times 128$, and $2 \times 128$, respectively. Each convolutional layer is followed by a batch normalization layer and ReLU function is selected as the activation function. There are two max pooling layers of size $1 \times 4$ following the first and second convolutional layer, respectively. The output of the third convolutional layer is fed into a fully connected layer for classification, and the softmax activation function is selected to output the probability of each class.

To make a fair comparison, the IQ/FFT-based and the spectrogram-based CNNs are deliberately designed with similar network architectures and trained under the same settings such as initial learning rate and mini-batch size. Both of them are implemented using the Matlab Deep Learning Toolbox[2].

## VI. Experimental Results of CFO Drift

The RF fingerprints must be time-invariant in the presence of environmental changes as they represent the user identities. In this section, we experimentally demonstrated that the CFO of LoRa devices drifts over time and CFO compensation is an essential procedure to mitigate performance degradation.

### A. Experimental Setup

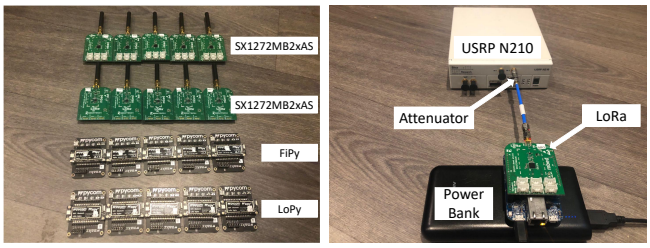
We used ten LoRa devices of two models, namely five SX1272MB2xAS mbed shields and five SX126xMB2xAS mbed shields, as listed in Table I and shown in Fig. 4a. All the LoRa devices were configured with $SF$ = 7, bandwidth $B$ = 125 kHz, and carrier frequency $f_c$ = 868.1 MHz. The receiver was a USRP N210 SDR and configured with carrier frequency $f_c$ = 868.1 MHz and 1 MS/s sampling rate. We used the Communications Toolbox Support Package for USRP Radio of Matlab[3] to control the USRP and access IQ samples from it. In order to eliminate channel effects and focus on CFO variations, we created a bespoke setup by connecting the LoRa DUT and USRP N210 receiver by a 40 dB attenuator, as shown in Fig. 4b.

The data collection for each device lasted for about one hour and was repeated for four days. The transmission interval was set to 1 second and 3,000 packets were collected in about one hour, considering the packet duration and processing time. We named the four datasets as Day 1, Day 2, Day 3 and Day 4 dataset.

[2]https://mathworks.com/products/deep-learning.html
[3]https://mathworks.com/help/supportpkg/usrpradio/
[4]https://os.mbed.com/components/SX1272MB2xAS/
[5]https://os.mbed.com/components/SX126xMB2xAS/
[6]https://pycom.io/product/fipy/
[7]https://pycom.io/product/lopy4/

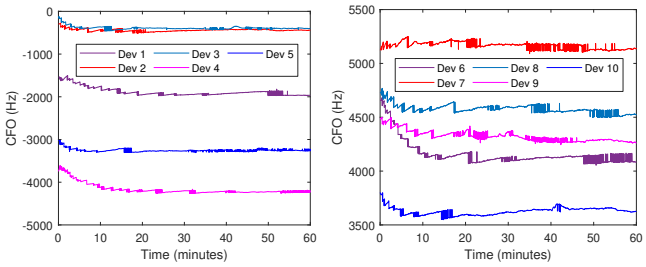
TABLE I
LoRa DUTs.

| DUT Index | Model | Chipset |
|---|---|---|
| 1 - 5 | SX1272MB2xAS mbed shield[4] | SX1272 |
| 6 - 10 | SX126xMB2xAS mbed shield[5] | SX1261 |
| 11 - 15 | Pycom FiPy[6] | SX2172 |
| 16 - 20 | Pycom LoPy[7] | SX1276 |




(a) (b)

Fig. 4. Experimental devices and setup. (a) LoRa DUTs. (b) The LoRa transmitter and USRP receiver connected by a 40 dB attenuator.




(a) (b)

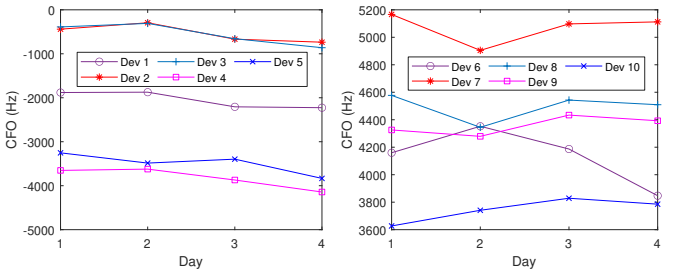
Fig. 5. CFO variations within one hour. (a) Dev 1-5. (b) Dev 6-10.

### B. CFO Drift

The CFO drift is demonstrated from two aspects, namely short-time and long-time variation. The short-time variation refers to the CFO of devices changed rapidly within a short time after they are powered on while the long-time variation shows that the average CFO drifts within four days but remains relatively stable.

*1) Short-time Variation:* We analyzed the Day 1 dataset as an example to observe how the CFO changes within one hour. The CFO of each packet was estimated using the algorithm introduced in Section III-B. As shown in Fig. 5, the CFO of each device decreased over the first 20 minutes and then remained relatively constant. This is reasonable because the temperature gradually increases after the device is powered on (self-heating) and the oscillator is sensitive to temperature variations [29].

*2) Long-time Variation:* We further investigated CFO drifts over different days. We estimated the CFO of the packets collected on the same day and calculated the average value. The results are shown in Fig. 6. There is a non-negligible and unpredictable CFO change on different days. The drift




(a) (b)

Fig. 6. Variations of average CFO within four days. (a) Dev 1-5. (b) Dev 6-10.

is probably caused by the uncontrollable changes in room temperature. The long-time variation indicates that the classification accuracy may decrease when the training and test data are not collected on the same day since they probably have different CFOs.

However, it can also be observed their average CFO remained relatively stable over the four days. While the time-varying CFO might not be suitable as a device identifier, it can be used to assist classification by ruling out devices whose estimated CFO deviates from the range too much.

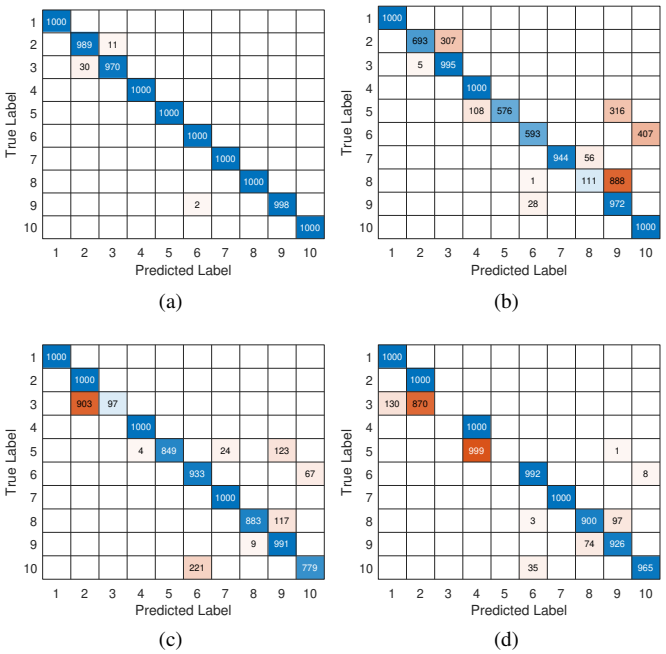
### C. The Effect of CFO Drift on RFFI

We carried out extensive experiments to evaluate CFO effects on the RFFI. We used the spectrograms of eight preambles and the CNN model shown in Fig. 3a.

The CNN was trained with the first 1,000 packets of each device (1,000×10 packets in total) from Day 1 dataset, among which 90% were randomly selected for training and the rest 10% were for validation. Then we used another 1,000 packets of each device from Day 1 dataset to test the trained CNN classifier. For Day 2-4 datasets, the first 1,000 packets of each device were used as the test data. This allowed us to evaluate the trained CNN classifier with packets collected on four different days.

Fig. 7 shows the confusion matrices obtained by CNN-only classifier when CFO compensation was not applied. Figs. 7a, 7b, 7c, and 7d represent the classification results when the test data was collected on Day 1, Day 2, Day 3, and Day 4, respectively. When the training and test sets were collected on the same day (Fig. 7a), the classification accuracy reached 99.57% which was almost no classification error. However, when the training and test data were collected on different days (Figs. 7b, 7c, and 7d), the classification results were unacceptable as several devices were completely misclassified, e.g., Dev 3 and Dev 5 in Fig. 7d.

The worst case happened on Day 4 (Fig. 7d) where the packets from Dev 3 and Dev 5 were completely misclassified. As shown in Fig. 8, it can be observed that the CFO of Dev 3 and Dev 5 drifted by hundreds of hertz from Day 1 to Day 4. The CFO of Dev 3 in the test data (purple line) was closer to Dev 2 (red line) in the training data. Similarly, the CFO of Dev 5 (orange line) in the test data was closer to Dev 4 (pink




(a) (b) (c) (d)

Fig. 7. Experimental results without CFO compensation (CNN-only classifier). (a) Day 1 Training, Day 1 Test, overall accuracy: 99.57%. (b) Day 1 Training, Day 2 Test, overall accuracy: 78.84%. (c) Day 1 Training, Day 3 Test, overall accuracy: 85.32%. (d) Day 1 Training, Day 4 Test, overall accuracy: 77.83%.

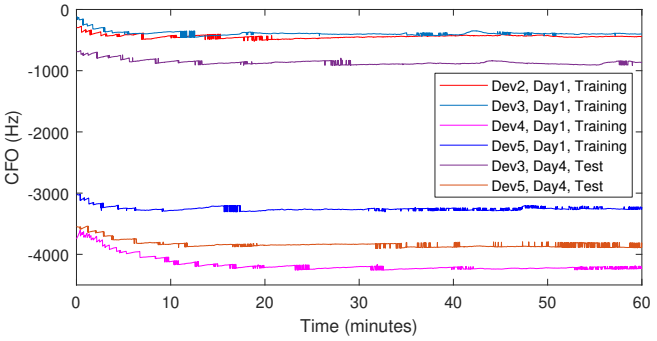



Fig. 8. The comparison of CFO between the Day 1 training data and Day 4 test data.

line) in the training data. It is inferred that CFO drift was the main reason for performance degradation and a slight drift of CFO would cause the classifier to make a wrong decision.

Fig. 9 shows the confusion matrices obtained by CNN-only classifier after CFO compensation was applied. Compared with the results in Fig. 7, there is no performance degradation after compensating the CFO, the accuracy always maintained above 96% on the four days. These results reveal that CNN can identify different devices with high accuracy after CFO compensation and performance degradation is significantly mitigated.

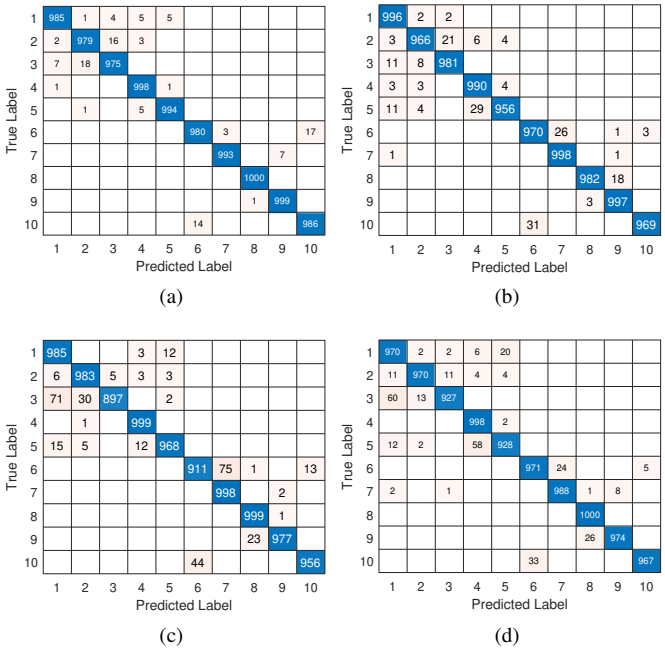



Fig. 9. Experimental results with CFO compensation (CNN-only classifier). (a) Day 1 Training, Day 1 Test, overall accuracy: 98.89%. (b) Day 1 Training, Day 2 Test, overall accuracy: 98.05%. (c) Day 1 Training, Day 3 Test, overall accuracy: 96.73%. (d) Day 1 Training, Day 4 Test, overall accuracy: 96.93%.

## VII. Experimental Evaluations in a Real Wireless Environment

In Section VI, the LoRa DUT and USRP were connected using an attenuator, which allowed us to investigate the CFO effect on RFFI without the channel effect. However, this is not a practical application scenario. Hence in this section, our proposed spectrogram-based RFFI system is evaluated in a real wireless environment. First, we compare the performance of three signal representations introduced in Section IV-C. Then we further evaluate the CFO effect in the wireless environment, and demonstrate that CFO compensation is an essential step in deep learning-based RFFI systems. Finally, the calibration function of the proposed hybrid classifier is experimentally demonstrated.

### A. Experimental Setup

We increased the number of LoRa DUTs to 20 in this section. As shown in Table I and Fig. 4a, these LoRa devices were from four different manufacturers. The same USRP N210 platform was used as the receiver. The LoRa DUTs and USRP platform were configured with the same parameters as described in Section VI-A. The difference is that we shortened the transmission interval to 0.3 seconds to speed up signal collection.

The experiments were carried out in a typical indoor environment, with chairs and tables distributed in the room. The distance between the LoRa DUT and USRP receiver was approximately three meters and there was line of sight (LOS) between them. We collected 2,000 packets continuously from each device for about 15 minutes. All the devices were placed at the same location and the environment was kept the same. Therefore, the same channel condition can be assumed for all the signal transmissions.

TABLE II
Experimental Results. Overall Classification Accuracy.

| | CNN-only Classifier | | Hybrid Classifier | |
|---|---|---|---|---|
| | w/o CFO Comp. | w/ CFO Comp. | w/o CFO Comp. | w/ CFO Comp. |
| IQ samples | 59.44% | 83.36% | 59.45% | 92.01% |
| FFT results | 51.62% | 87.36% | 51.63% | 92.31% |
| Spectrogram | 75.59% | 96.44% | 75.59% | 97.61% |

We used the first 1,000 packets of each device as training data, 90% of which were randomly selected for training and the rest 10% were for validation. The second 1,000 packets of each device were used as the test data to evaluate the RFFI system. The experimental results are presented in Table II. We analyzed the results from three aspects: the selection of signal representations, the impact of CFO in a wireless environment, and the calibration performance of our proposed hybrid classifier.

### B. Selection of Signal Representations

We compare the classification accuracy of the three signal representations. The IQ/FFT-based CNN has a similar network structure with the spectrogram-based model hence a relatively fair comparison can be carried out.

As shown in Table II, when the CNN-only classifiers were used, the spectrogram-based model reached the highest accuracy of 96.44%, while the IQ and FFT-based model only reached 83.36% and 87.36%, respectively. This shows that for LoRa signals whose frequency components are time-changing, device fingerprints can be detected more easily in the time-frequency domain.

Besides the classification results demonstrated in Table II, we found that the training time of our spectrogram-based model and the IQ/FFT-based model was about 20 minutes and one hour, respectively, when both were trained on the same PC. In addition to this, the loss of spectrogram-based model drops earlier and faster than the IQ/FFT-based model. This is another advantage of spectrogram-CNN model in terms of training costs.

### C. Impact of CFO drift

As can be observed in Table II, when there was no CFO compensation, the accuracies of IQ, FFT, and spectrogram-based RFFI systems were only 59.44%, 51.62% and 75.59%, respectively. After CFO compensation was applied, the corresponding accuracies significantly increased to 83.36%, 87.36% and 96.44%, respectively.

We take the dataset of Dev 1 as an example to explain the results without CFO compensation. Fig. 10 shows the CFO of each packet collected from Dev 1, and presents a similar pattern with Fig. 5 that the CFO decreased continuously after

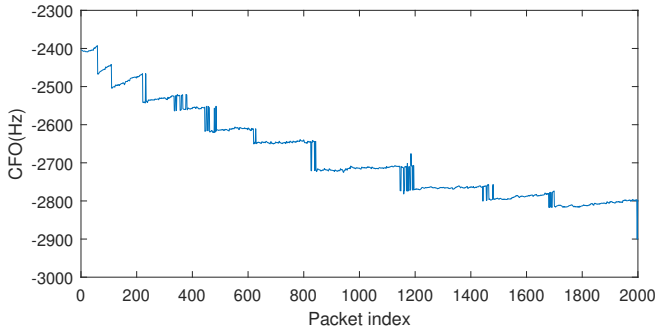
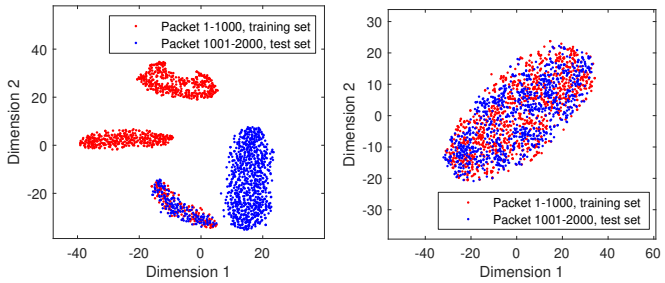



Fig. 10. CFO of each packet in the dataset of Dev 1.




(a) (b)

Fig. 11. t-SNE visualization of the training and test sets of Dev1. (a) Without CFO compensation. (b) With CFO compensation.

the device was powered on. In the wireless experiments, we used packet 1-1,000 to train the CNN and packet 1,001-2,000 to evaluate its performance. However, the packets in the test set have different CFOs with those in the training set. In other words, the test data has different distributions from the training data. In machine learning tasks, the training and test sets are often required to have the same, at least similar data distributions, otherwise, the trained model will face severe generalization problems. This might also be the reason for the low classification accuracy in [28] when CFO compensation is not involved.

Then we verify the argument that the training data and test data have different distributions using the well-known t-SNE visualization algorithm. The visualization result is shown in Fig. 11, in which each point represents a packet collected from Dev 1. There are 2,000 points in total and the red points represent packets 1-1,000 (training data) and blue points represent packets 1,001-2,000 (test data). From Fig. 11a it can be observed that there are four distinct clusters when there is no CFO compensation, which indicates that the training data and test data have different features/distributions. In contrast, as shown in Fig. 11b, the blue and red points are mixed after CFO compensation and cannot be separated intuitively. This is what we expected because the features of each device should be time-invariant, i.e., the first 1,000 packets should have the same features with the second 1,000 packets, which leads to overlapping in the visualization.

### *D. Effectiveness of The Hybrid Classifier*

The hybrid classifier introduced in Section IV-E calibrates the softmax output of CNN according to the estimated CFO. As shown in Fig. 6, the CFO varies over different days and some devices may have similar CFOs, hence it cannot be used as a fingerprint to identify numerous low-cost IoT devices. However, the average values of CFO stay relatively stable in a small range, which can be used for calibration to rule out predictions whose estimated CFO is much different from the reference one.

As shown in Table II, it can be observed that the hybrid classifier can increase the accuracy for all the three signal representations. The most significant improvement was the input type of IQ samples after applying CFO compensation, the accuracy with hybrid classifier for IQ data reached 92.01% while the accuracy using CNN-only classifier was 83.36%, which was an 8.65% improvement. For the signal representation of FFT results, there was an accuracy improvement from 87.36% to 92.31%, and for the spectrogram, the accuracy increased from 96.44% to 97.61%.

It is also observed that the hybrid classifier does not work when there was no CFO compensation. This is reasonable because the CFO has contributed to the prediction when compensation is not involved and the hybrid classifier cannot provide additional helpful information.

## VIII. Conclusion

In this paper, we proposed a spectrogram-based RFFI system and carried out extensive experimental evaluations. We used 20 LoRa devices of four models as the DUTs and a USRP N210 SDR as the receiver. Firstly, as LoRa uses chirp modulation, we employed spectrogram to represent the time-frequency characteristics of LoRa signals. We found that using spectrogram can achieve a better classification accuracy compared to the IQ samples and FFT results. Secondly, we experimentally found that CFO is not stable as it was varying over time probably due to temperature changes. Hence it will compromise the system stability. CFO compensation was experimentally found to be effective in mitigating the performance degradation. Finally, we proposed a hybrid classifier that calibrates the softmax output of CNN using the estimated CFO. Although CFO is varying over time, its average value stays relatively stable over days. CFO must be compensated to avoid performance degradation but is helpful to rule out predictions when the estimated CFO deviates greatly from the reference CFO. Our proposed RFFI system finally achieved a classification accuracy of 97.61% in distinguishing 20 LoRa devices in real wireless environments.

## Acknowledgement

The work was in part supported by the UK Royal Society Research Grants under grant ID RGS/R1/191241 and national key research and development program of China under grant ID 2020YFE0200600.

## REFERENCES


[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[2] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 94–104, 2015.

[3] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

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

[5] J. Zhang, S. Rajendran, Z. Sun, R. Woods, and L. Hanzo, "Physical layer security for the internet of things: Authentication and key generation," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 92–98, 2019.

[6] J. Zhang, F. Wang, O. A. Dobre, and Z. Zhong, "Specific emitter identification via Hilbert–Huang transform in single-hop and relaying scenarios," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1192–1205, 2016.

[7] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, "Device fingerprinting to enhance wireless security using nonparametric Bayesian method," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Shanghai, China, Apr. 2011, pp. 1404–1412.

[8] W. Hou, X. Wang, J.-Y. Chouinard, and A. Refaey, "Physical layer authentication for mobile systems with time-varying carrier frequency offsets," *IEEE Trans. Commun.*, vol. 62, no. 5, pp. 1658–1667, 2014.

[9] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Fingerprinting Wi-Fi devices using software defined radios," in *Proc. ACM Conf. Security Privacy Wireless Mobile Netw. (WiSec)*, Darmstadt, Germany, Jul. 2016, pp. 3–14.

[10] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong, "Accurate and efficient wireless device fingerprinting using channel state information," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 1700–1708.

[11] P. Liu, P. Yang, W.-Z. Song, Y. Yan, and X.-Y. Li, "Real-time identification of rogue WiFi connections using environment-independent physical features," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 190–198.

[12] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, San Francisco, CA, USA, Sep. 2008, pp. 116–127.

[13] W. Wang, Z. Sun, S. Piao, B. Zhu, and K. Ren, "Wireless physical-layer identification: Modeling and validation," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2091–2106, 2016.

[14] T. J. Bihl, K. W. Bauer, and M. A. Temple, "Feature selection for RF fingerprinting with multiple discriminant analysis and using ZigBee device emissions," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1862–1874, 2016.

[15] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, "Identifying wireless users via transmitter imperfections," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 7, pp. 1469–1479, 2011.

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

[17] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, 2018.

[18] J. Yu, A. Hu, G. Li, and L. Peng, "A robust RF fingerprinting approach using multisampling convolutional neural network," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6786–6799, 2019.

[19] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "Oracle: Optimized radio classification through convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 370–378.

[20] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, "No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 165–178, 2019.

[21] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, K. Chowdhury, S. Ioannidis, and T. Melodia, "Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 646–655.

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

[23] J. Gong, X. Xu, and Y. Lei, "Unsupervised specific emitter identification method using radio-frequency fingerprint embedded InfoGAN," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2898–2913, 2020.

[24] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based RF fingerprint identification using differential constellation trace figure," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1091 – 1095, 2020.

[25] Y. Pan, S. Yang, H. Peng, T. Li, and W. Wang, "Specific emitter identification based on deep residual networks," *IEEE Access*, vol. 7, pp. 54 425–54 434, 2019.

[26] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. Moura, "A deep learning approach to IoT authentication," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[27] Z. Zhu, H. Leung, and X. Huang, "Challenges in reconfigurable radio transceivers and application of nonlinear signal processing for rf impairment mitigation," *IEEE Circuits Syst. Mag.*, vol. 13, no. 1, pp. 44–65, 2013.

[28] P. Robyns, E. Marin, W. Lamotte, P. Quax, D. Singelée, and B. Preneel, "Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning," in *Proc. ACM Conf. Security Privacy Wireless Mobile Netw. (WiSec)*, Boston, MA, USA, Jul. 2017, pp. 58–63.

[29] S. D. Andrews, "Extensions to radio frequency fingerprinting," Ph.D. dissertation, Virginia Tech, 2019.

[30] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid rf fingerprint extraction and device classification scheme," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 349–360, 2019.

[31] O. B. Seller and N. Sornin, "Low power long range transmitter," U.S. Patent 9,252,834, Feb., 2016.

[32] "LoRa and LoRaWAN: A Technical Overview," Semtech, Tech. Rep., Feb. 2020, accessed on 15 Aug., 2020. [Online]. Available: https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/

[33] Y. Jiang, L. Peng, A. Hu, S. Wang, Y. Huang, and L. Zhang, "Physical layer identification of LoRa devices using constellation trace figure," *EURASIP J. Wireless Communications and Networking*, vol. 2019, no. 1, p. 223, 2019.

[34] "LoRaWAN® Regional Parameters," Semtech, Tech. Rep., Feb. 2020, accessed on 14 Aug., 2020. [Online]. Available: https://lora-alliance.org/sites/default/files/2020-06/rp_2-1.0.1.pdf

[35] "LoRa Modulation Crystal Oscillator Guidance," Semtech, Tech. Rep. AN1200.14, Jul. 2019, accessed on 4 Jun., 2020. [Online]. Available: https://lora-developers.semtech.com/library/product-documents/

[36] P. Robyns, P. Quax, W. Lamotte, and W. Thenaers, "A multi-channel software decoder for the LoRa modulation scheme," in *Proc. Int. Conf. Internet Things, Big Data Secur. (IoTBDS)*, Mar. 2018, pp. 41–51.

[37] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "A comparison of audio signal preprocessing methods for deep neural networks on music tagging," in *Proc. European Signal Processing Conference (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 1870–1874.