# Radio Frequency Fingerprint Identification for LoRa Using Deep Learning

Guanxiong Shen, Junqing Zhang, Alan Marshall, *Senior Member, IEEE*, Linning Peng, and Xianbin Wang, *Fellow, IEEE*

***Abstract*—Radio frequency fingerprint identification (RFFI) is an emerging device authentication technique that relies on the intrinsic hardware characteristics of wireless devices. This paper designs a deep learning-based RFFI scheme for Long Range (LoRa) systems. Firstly, the instantaneous carrier frequency offset (CFO) is found to drift, which could result in misclassification and significantly compromise the stability of the deep learning-based RFFI system. CFO compensation is demonstrated to be effective mitigation. Secondly, three signal representations for deep learning-based RFFI are investigated in time, frequency, and time-frequency domains, namely in-phase and quadrature (IQ) samples, fast Fourier transform (FFT) results and spectrograms, respectively. For these signal representations, three deep learning models are implemented, i.e., multilayer perceptron (MLP), long short-term memory (LSTM) network and convolutional neural network (CNN), in order to explore an optimal framework. Finally, a hybrid classifier that can adjust the prediction of deep learning models with the estimated CFO is designed to further increase the classification accuracy. The CFO will not change dramatically over several continuous days, hence it can be used to correct predictions when the estimated CFO is much different from the reference one. Experimental evaluation is performed in real wireless environments involving 25 LoRa devices and a Universal Software Radio Peripheral (USRP) N210 platform. The spectrogram-CNN model is found to be optimal for classifying LoRa devices which can reach an accuracy of 96.40% with the least complexity and training time.**

***Index Terms*—Internet of Things, LoRa, device authentication, radio frequency fingerprint, deep learning, carrier frequency offset**

## I. Introduction

DEVICE authentication is critical to safeguard Internet of Things (IoT) systems for allowing legitimate users to access the network while preventing malicious users [2]. This task is becoming challenging with the rapid growth of low-cost IoT devices. Conventional authentication schemes rely on software addresses such as Media Access Control (MAC) address, which are prone to spoofing [3]. Once the security credentials are obtained by malicious users, they can masquerade as legitimate users to access the private data or launch fatal attacks on the IoT networks.

Radio frequency fingerprint identification (RFFI) is a promising device authentication scheme that can identify wireless devices based on their hardware fingerprints [2], [4], [5]. Radio frequency fingerprint (RFF) originates from hardware impairments introduced during the manufacturing process, which are inherent to the analog front-end components. These device-specific hardware features deviate from their nominal values. The deviations are too small to affect the normal communication functions but will slightly distort wireless waveforms, from which the hardware impairments can be extracted as device identifiers. Similar to a biometric fingerprint, RFF is unique and hard to tamper without tremendous efforts. RFFI is particularly suitable for low-cost IoT devices. They are made of inexpensive components thus are rich in hardware imperfections which are the source of RFFI. In addition, RFFI does not impose any additional power consumption on the devices to be authenticated [2], [6], which is particularly desirable because most IoT end nodes are with limited computational and energy resources.

RFFI can be regarded as a classification problem, hence the state-of-the-art deep learning technique is leveraged [7]–[17]. This paper will study three important aspects of deep learning-based RFFI, namely system stability, selection of signal representations and deep learning models, as well as the calibration of the unreliable predictions. Firstly, stability is the most prominent requirement for RFFI [2]. Oscillator is especially sensitive in low-cost IoT devices [8], [18], whose frequency drift is inferred to degrade system performance over time [19], [20]. However, there is currently no comprehensive study on the impact of carrier frequency offset (CFO) drift on RFFI systems. Secondly, a reasonable selection of signal representation and deep learning model can improve the performance of the RFFI system. Some RFFI systems directly use in-phase and quadrature (IQ) samples [7]–[11], while others transform IQ samples to other forms to make the deep learning model easier to learn [19]. A more thorough comparative investigation is still missing. Finally, the deep learning model inference might not be correct when the characteristics of several devices are similar. There is currently no work attempting to calibrate the unreliable predictions.

Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. 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. This work was presented in part at the IEEE INFOCOM 2021 [1]. The review of this paper was coordinated by xxx. *(Corresponding author: Junqing Zhang.)*

G. Shen, J. Zhang and A. Marshall are with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom. (email: Guanxiong.Shen@liverpool.ac.uk; junqing.zhang@liverpool.ac.uk; alan.marshall@liverpool.ac.uk)

L. Peng is with the School of Cyber Science and Engineering, Southeast University, No. 2 Sipailou, Nanjing, China and Purple Mountain Laboratories, Nanjing, China. (email: pengln@seu.edu.cn)

X. Wang is with the Department of Electrical and Computer Engineering, Western University, London, Ontario, N6A 5B9, Canada. (email: xianbin.wang@uwo.ca)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier xxx

This paper is motivated to address the above research challenges by designing a deep learning-based RFFI system to classify LoRa devices. Specifically, we will answer the following three questions: (1) How does the CFO variation affect the RFFI system and can we mitigate it? (2) Can we find a signal representation that is unique to LoRa modulation and build a suitable deep learning model for it? (3) Can we leverage the probabilities of the softmax output to further enhance the classification accuracy? We carry out an in-depth investigation and extensive experiments involving 25 LoRa devices as devices under test (DUTs) and a Universal Software Radio Peripheral (USRP) N210 software-defined radio (SDR) platform as the authenticator to answer these questions. The technical contributions of this paper are summarized as follows.

- We carry out extensive experiments over seven months to measure the CFO variation and demonstrate that CFO drifts over time and degrades the RFFI performance. A bespoke setup is established by connecting each LoRa DUT and the USRP platform with an attenuator to eliminate channel effects. The instantaneous CFO is found to vary in a short time frame (over one hour). CFO compensation is demonstrated to be an effective mitigation method, which can improve the classification accuracy from 83.53% to 95.35% for the spectrogram-CNN[1] model.
- We investigate three signal representations of LoRa signals to support deep learning-based RFFI, namely IQ samples, fast Fourier transform (FFT) results and spectrograms, corresponding to time domain, frequency domain and time-frequency domain analysis, respectively. For each signal representation, we build multilayer perceptron (MLP), convolutional neural network (CNN) and long short-term memory (LSTM) networks to explore the optimal model. The spectrogram-CNN can reach the highest accuracy (95.35%) with the least system complexity (1,545,193 parameters) and the shortest training time (20 minutes).
- We design a hybrid classifier based on the softmax output and CFO to further improve the performance of deep learning-based RFFI. The range of CFO variation remains relatively stable over days, as the environment temperature only changes slowly. Hence, the output of the deep learning model is calibrated according to the estimated CFO. The designed hybrid classifier can significantly improve the classification accuracy, namely from 58.26% to 82.81% in the best case for the FFT-LSTM model.

In our previous work [1], we have shown that CFO must be compensated for system stability, established a CNN model to compare the performance of the three signal representations, and designed a hybrid classifier to improve the system performance. This paper further extends the experimental measurements of CFO to reveal the CFO variation in seven months. In addition, we investigate two more deep learning models, namely MLP and LSTM, and discuss which signal representation better suits them. We also examine their system complexity and training cost.

The rest of the paper is organized as follows. Section II briefly introduces the related work and and Section III presents the background of LoRa modulation and spectrogram. The LoRa receiver operation is explained in Section IV. The design details of the RFFI system and the architecture of deep learning models are introduced in Section V and Section VI, respectively. Section VII experimentally demonstrates CFO drift and its effect on the RFFI system and Section VIII evaluates the performance of the proposed RFFI systems in a real wireless environment. Section IX concludes the paper.

## II. Related Work

RFFI systems can be roughly categorized into two groups, namely, handcrafted feature-based systems and deep learning-based systems. Handcrafted features include statistics of the Hilbert spectrum [21], [22], CFO [23]–[27], IQ offset [28], phase error [28], and power amplifier/digital-to-analog converter nonlinearity [29], etc. There are several major limitations to such handcrafted feature-based RFFI techniques. First, the system performance is constrained by the quality of handcrafted features and their extraction requires comprehensive knowledge of the adopted communication technology. Second, it is difficult to estimate each individual feature accurately as the hardware imperfections are interrelated [30]. Recently, deep learning-based RFFI is proposed to solve the above challenges. Sankhe *et al.* fed the IQ samples into a CNN for classification [9], [10]. Roy *et al.* investigated the performance of three different deep learning models, namely MLP, CNN and LSTM [15], but they only examined IQ data. As IQ data is a complex vector, complex-valued neural networks are adopted [31], [32].

As a device authentication scheme, RFFI should remain stable [2]. Robyns *et al.* [19] indicated that the accuracy of their system dropped over time and inferred this may be caused by the oscillator frequency drift. However, they did not further provide in-depth analysis or a mitigation method. Andrews *et al.* [20] 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 [26], [27], it was also observed that low-cost ZigBee devices have severe CFO variations even within 15 minutes [8], [18]. A comprehensive investigation of CFO variation in low-cost IoT devices and its effect on the RFFI is still missing.

Designing more discriminative signal representations is a popular topic in deep learning-based RFFI. Merchant *et al.* [7] calculated the error signal by subtracting the ideal signal from the received one. He *et al.* [33] leveraged signal processing techniques to decompose the received signals. Gong *et al.* [17] extracted the gray histogram of bispectrum to enhance individual discriminability. To the best knowledge of the authors, there are only three papers on LoRa RFFI [16], [19], [34].

[1]Spectrogram-CNN denotes using the spectrogram as signal representation and CNN as the deep learning model. Similar descriptions are used throughout the paper.

Robyns *et al.* [19] performed FFT on the received signal to make CFO easier to learn. Das *et al.* [16] directly used IQ samples as the system input. Jiang *et al.* [34] adopted the so-called differential constellation trace figures. However, none of them considered the unique modulation techniques of LoRa.

The confidence information returned by deep learning models can be further leveraged. The softmax function is used to return a list of probabilities over all the classes, and the label with the highest probability is selected as the final prediction. However, the classifier is not confident about its prediction when the confidence score is low and the probabilities of several classes are quite close. Gritsenko *et al.* [35] used the confidence information to detect unregistered devices. When the prediction confidence is low, the classifier considers the signal is from a new device that is not present in the training set. However, there has been no work reported to calibrate the predictions of deep learning.

## III. Preliminary

### A. LoRa Modulation Technique

LoRa is a physical layer standard patented by Semtech in 2014 [36], which has been widely used for long range IoT applications. LoRa employs chirp spread spectrum (CSS) modulation whose frequency increases or decreases linearly with time. It is a type of non-stationary signal that can be analyzed by time-frequency analysis.

The standard linear chirp in the RF band, $c^{rf}(t)$, can be mathematically expressed as

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

where $A$, $B$, $T$, $f_c$ are amplitude, bandwidth, time duration, and carrier frequency of $c^{rf}(t)$, respectively. In LoRa modulation, $T$ is adjustable by modifying the spreading factor $SF$ and bandwidth $B$

$$T = \frac{2^{SF}}{B}. \tag{2}$$

The baseband chirp signal, $c(t)$, can be derived from (1)

$$c(t) = Ae^{j(-\pi Bt+\pi\frac{B}{T}t^2)} = Ae^{j\phi(t)}, \tag{3}$$

where $\phi(t)$ denotes the phase of $c(t)$. The instantaneous frequency can then be calculated as

$$f(t) = \frac{1}{2\pi}\frac{d\phi(t)}{dt} = -\frac{B}{2} + \frac{B}{T}t \quad (0 \le t \le T). \tag{4}$$

The LoRa standard specifies several repeated baseband chirps at the beginning of a packet as preambles [37]. The preamble part is the same for all the packets and any LoRa device type. Fig. 1a shows the time-domain baseband signal of the preamble part and Fig. 1b is the zoom of the first preamble.

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

Short-time Fourier transform (STFT) is a well-known time-frequency analysis algorithm that has been extensively used to analyze non-stationary signals. The linear chirp employed by LoRa modulation is a typical non-stationary signal so the spectrogram obtained by STFT is usually used to visualize LoRa packets.

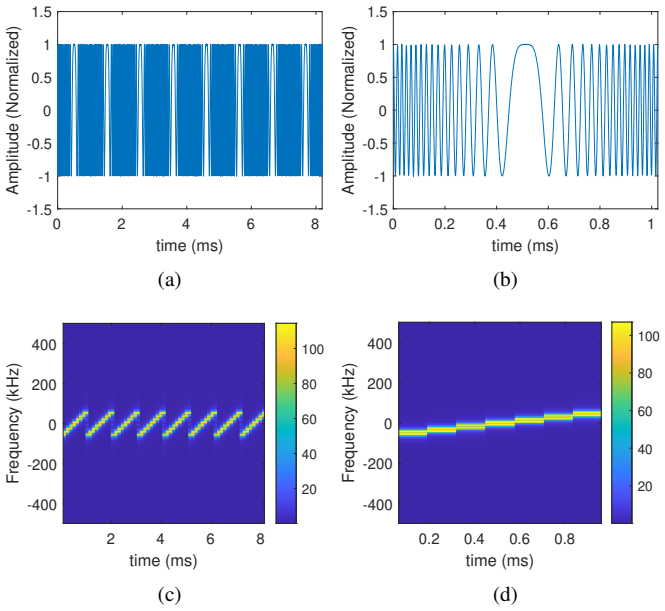



Fig. 1. Preamble part of a LoRa packet. (a) Time domain signal of all the preambles (I branch). (b) Time domain signal of the first preamble (I branch). (c) Spectrogram of all the preambles. (d) Spectrogram of the first preamble.

The discrete-time STFT is mathematically given as

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

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

$$Spectrogram(m,f) = |STFT(m,f)|^2, \tag{6}$$

where $|\cdot|$ returns the amplitude. Fig. 1c is the spectrogram of the entire preamble part and Fig. 1d shows the zoom of the first one. Spectrogram can represent how the instantaneous frequency changes over time.

## IV. LoRa Receiver Operation

### A. Signal Reception

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

$$\begin{aligned} r[nT_s] =& r^{rf}[nT_s]e^{-j2\pi f_c^{rx} nT_s} \\ =& u[nT_s]e^{j2\pi\Delta f n T_s}, \end{aligned} \tag{7}$$

where $u[nT_s]$ is the transmitted 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

$$r[n] = u[n]e^{j2\pi\Delta f n T_s}. \tag{8}$$

### B. Synchronization

Accurate synchronization is essential to detect signal arrival. The well-known Schmidl-Cox algorithm is exploited, which is based on the repeating property of the preambles [38]. The mathematical expression is shown as

$$M[n] = \frac{\left|\sum_{k=0}^{L-1} r[n+k]r^*[n+k+L]\right|}{\sum_{k=0}^{L-1} r[n+k+L]r^*[n+k+L]}, \tag{9}$$

where $r^*[n+k]$ denotes the conjugate of $r[n+k]$ and $L$ is the symbol length (in samples) of one LoRa symbol, given as

$$L = \frac{T}{T_s}, \tag{10}$$

where symbol duration $T$ must be the integer multiple of sampling time $T_s$ to make $L$ an integer. A threshold is predefined to detect the arrival of the packet: when $M[n] > threshold$, the packet arrives. However, this approach can only detect the arrival of a packet but not the exact starting point [39]. There is usually a short section of channel noise in front of the received packet.

A more precise synchronization algorithm is essential to locate the exact starting point of the packet. Robyns *et al.* [39] proposed an effective fine synchronization algorithm according to the unique characteristics of LoRa signals. First, an ideal baseband basic chirp is generated at the receiver and its instantaneous frequency $f_{ideal}[n]$ is calculated as (4). Then the receiver calculates the instantaneous frequency of the coarse synchronized baseband signal, $\widehat{f}_r[n]$. Finally, a sliding window cross-correlation is performed between $\widehat{f}_r[n]$ and $f_{ideal}[n]$. The index of the maximum value is chosen as the index of the accurate starting point, $ind$, which is given as

$$ind = \underset{i\epsilon\{0,1,\ldots,L-1\}}{\arg\max} \Big( \sum_{n=0}^{L-1} f_{ideal}[n] \cdot \widehat{f}_r[n+i] \Big). \tag{11}$$

### C. CFO Estimation and Compensation

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 $r[n]$. The instantaneous frequency $f[n]$ thus becomes

$$f[n] = -\frac{B}{2} + \Delta f + \frac{B}{T} n T_s. \tag{12}$$

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

$$\Delta\widehat{f}_{coarse} = \frac{1}{L} \sum_{n=0}^{L-1} f[n]. \tag{13}$$

The received signal can be compensated by the estimated $\widehat{f}_{coarse}$, given as

$$r'[n] = r[n] \cdot e^{-j2\pi\Delta\widehat{f}_{coarse} n T_s}. \tag{14}$$

However, the coarse compensation is not accurate enough. There will be a residual CFO after the above coarse frequency compensation, hence we further employ a fine CFO estimation algorithm. The residual offset, $\widehat{f}_{fine}$, can be estimated based on the repeating property of preambles, given as

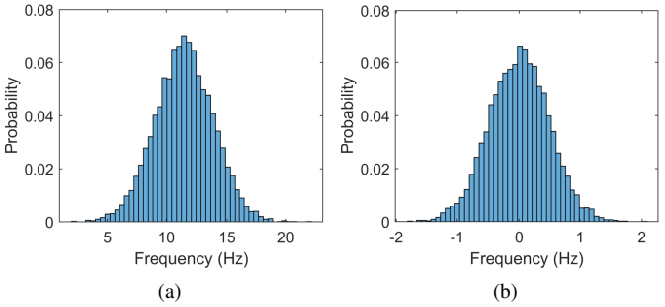



Fig. 2. Histograms of residual CFO. (a) Residual CFO after coarse compensation. (b) Residual CFO after fine compensation.

$$\Delta\widehat{f}_{fine} = -\frac{1}{2\pi T_s L} \angle \Big( \sum_{n=0}^{L-1} r'[n] r'^*[n+L] \Big), \tag{15}$$

where $\angle\cdot$ returns the angle of the variable. The received signal can be further finely compensated as

$$r''[n] = r'[n] \cdot e^{-j2\pi\Delta\widehat{f}_{fine} n T_s}. \tag{16}$$

As the phase can only be resolved in $[-\pi, \pi]$, the range of CFO that can be estimated by (15) is

$$|\Delta\widehat{f}_{fine}| < \frac{\pi}{2\pi T_s L} = \frac{B}{2^{SF+1}}. \tag{17}$$

When the LoRa transmission is configured with $SF$=7 and $B$=125 kHz, the estimation capability is within $\pm$488.3 Hz. Commonly, the oscillator drift of LoRa devices is $\pm$10 ppm [40], approximately 8.68 kHz for an 868 MHz carrier frequency, which is much higher than 488.3 Hz. Hence, the coarse CFO estimation should be employed first to limit the residual offset before the fine CFO estimation.

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

$$\Delta\widehat{f} = \Delta\widehat{f}_{coarse} + \Delta\widehat{f}_{fine}. \tag{18}$$

Simulation is carried out to evaluate the performance of CFO compensation using MATLAB. The baseband LoRa preamble, $u[n]$, and a CFO, $\Delta f_{sim}$, are generated and combined as in (8). Then the coarse estimated CFO $\Delta\widehat{f}_{coarse}$ is derived by (13). The residual CFO after coarse compensation can be calculated by $\Delta f_{sim} - \Delta\hat{f}_{coarse}$. Similarly, the fine estimated CFO, $\Delta\widehat{f}_{fine}$, can be obtained by (15) and the residual CFO after fine compensation can be calculated by $\Delta f_{sim} - \Delta\widehat{f}_{coarse} - \Delta\widehat{f}_{fine}$. We ran the simulation 10,000 times with $\Delta f_{sim}$ uniformly distributed between -10,000 Hz and +10,000 Hz. The signal-to-noise ratio (SNR) was 20 dB. The residual CFOs after coarse and fine compensation are shown by histograms in Fig. 2. The residual CFO after coarse and fine compensation is between 5 Hz to 20 Hz and between -1 Hz to +1 Hz, which is quite accurate.

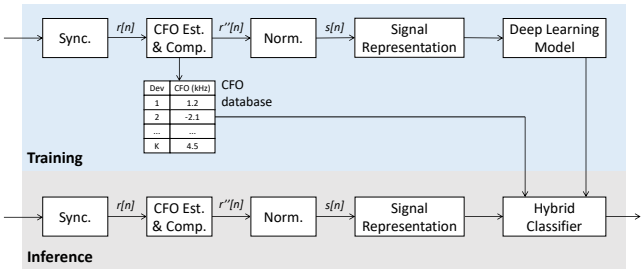



Fig. 3. A deep learning-based RFFI system. CFO compensation is adopted.

## V. RFFI System

The architecture of the proposed RFFI system is shown in Fig. 3. The proposed system only uses the preamble part to prevent the model from learning protocol-specific knowledge. In the training stage, numerous packets are collected from legitimate devices with each packet correctly labelled. The preamble parts of these packets are then pre-processed, including synchronization, normalization and conversion to the selected signal representation (IQ samples/FFT results/spectrogram). After the pre-processing, the training data is used to train the deep learning model. Once the training is completed, the newly received packet preamble can be fed into the trained deep learning model and a predicted device label will be returned, which is named the inference phase.

### 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. The algorithm used in this paper has been described in Section IV-B.

CFO estimation and compensation are also normal operations in communication systems. Some previous studies retained the CFO as a discriminative feature, which may compromise the system stability [19]. We adopted the CFO estimation and compensation algorithms introduced in Section IV-C. During the training stage, a CFO database is generated by recording the average CFO of each DUT. This CFO database will be used to calibrate the softmax output of deep learning models, which will be introduced in Section V-E.

### B. Normalization

Normalization is a standard process in RFFI since classifiers should not distinguish devices based on the power of the received signal. We normalize the received signal by dividing its root mean square value. The normalized signal $s[n]$ can be given as

$$s[n] = \frac{r''[n]}{x_{rms}}, \tag{19}$$

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

### C. Signal Representation

The received IQ samples can be directly fed into the deep learning model. They can also be converted to other representations using signal processing algorithms. This helps reveal underlying signal characteristics thus makes the classifier easier to learn.

**IQ samples**: IQ samples, consisting of samples from I and Q branches, represent the time-domain signals which are captured from the receiver chain directly. Some previous studies aim to design protocol-agnostic RFFI systems so they employ IQ samples as system inputs [7]–[11]. However, this can make training deep learning models difficult and time-consuming as some signal characteristics are not obvious in the time domain.

**FFT results**: FFT transforms the signal into the frequency domain, making some features easier to observe. For example, the CFO causes phase differences in the time domain, but is reflected as a shift of frequency spectrum in the frequency domain. The latter is easier to observe [19].

**Spectrogram**: The spectrogram reveals signal characteristics in the time-frequency domain, which not only provides information about frequency components but also demonstrates how they change over time. It is useful for analyzing non-stationary wireless signals, including the chirps employed by LoRa. Logarithmic compression of magnitudes has been a standard method in preprocessing spectrograms [41], which is also used in this work.

### D. Deep Learning Model

Deep learning models can automatically extract features from signal representations without the need for feature engineering. Considering the characteristics of IQ samples, FFT results and spectrogram, three well-known deep learning models are investigated, namely MLP, CNN and LSTM. A rule of thumb in deep learning is that CNN is efficient in processing data with spatial correlation such as images (spectrogram), while LSTM is efficient in time series with temporal correlation (IQ samples).

**Multilayer Perceptron**: MLP is the forerunner of CNN and LSTM. It consists of fully connected layers and employs non-linear activation functions to process the data that is not linearly separable. It includes numerous parameters since all the neurons are fully connected, which results in redundancy. MLP has no preferred input data type.

**Convolutional Neural Network**: CNN was popular in recent years thanks to its excellent performance in image recognition and computer vision [42]. CNN is usually composed of convolutional layers, fully connected layers and some pooling layers. CNN is designed to capture the spatial correlation in the input data thus particularly suitable for processing images. Among the three signal representations introduced in Section V-C, spectrogram can be considered as a 2D image thus may be suitable for CNN.

**Long Short-Term Memory Network**: LSTM is a kind of recurrent neural network that is designed for temporal sequences such as speech and acoustic signals [43]. It is efficient in capturing the temporal correlation in the input

sequences. Among the three signal representations, both IQ samples and spectrogram exist temporal correlation thus may be suitable for LSTM.

**Softmax Output**: In classification problems, softmax function is usually employed in the last layer of a deep learning model to map the outputs to a list of probabilities $\mathbf{S} = (S_1, S_2, ..., S_K)$ over all the classes, which can be mathematically expressed as

$$S_k = \sigma(\mathbf{z})_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for} \quad k = 1, 2, ..., K, \tag{20}$$

where $K$ is the total number of classes, $S_k$ is the probability of the $k$-th class, and $\mathbf{z} = (z_1, z_2, ..., z_K)$ is the output of the layer before softmax activation. The straightforward method to make predictions is to select the class with the highest probability as the final predicted label, which is used by most of the deep learning-based RFFI systems.

### *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 possibly results in misclassification.

As we will demonstrate later in Section VII and Fig. 6, the CFO remained relatively stable over several continuous days. This inspires us to use the estimated CFO to calibrate the prediction of deep learning models.

We propose a hybrid classifier to exclude unreliable predictions returned by the deep learning model, which is described in Algorithm 1. We first create a CFO database for all the $K$ devices during the training stage. For the $k^{th}$ DUT, the CFO database includes $\Delta\widehat{f}_k^{\min}$, $\Delta\widehat{f}_k^{\max}$, $\Delta\widehat{f}_k = (\Delta\widehat{f}_k^{\max} + \Delta\widehat{f}_k^{\min})/2$ and $\lambda_k = (\Delta\widehat{f}_k^{\max} - \Delta\widehat{f}_k^{\min})/2$. Then, for each DUT during the inference stage, we will estimate its CFO, $\{\Delta\widehat{f}_{DUT}\}$, and compare it with the reference CFO stored in the database. The process can be formulated as

$$\left|\Delta\widehat{f}_{DUT} - \Delta\widehat{f}_k\right| \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda_k. \tag{21}$$

The 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 happens, the probability of $k$-th class, $S_k$, is set to zero. In contrast, hypothesis $\mathcal{H}_0$ means the prediction of the deep learning model is correct, thus $S_k$ maintains the original value. After the 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:** $\lambda_k$, The CFO threshold of the $k$-th device
**OUTPUT:** $l$, The eventually predicted label.

```
for k = 1 to K do
    if |Δf̂_DUT − Δf̂_k| > λ_k 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.
```

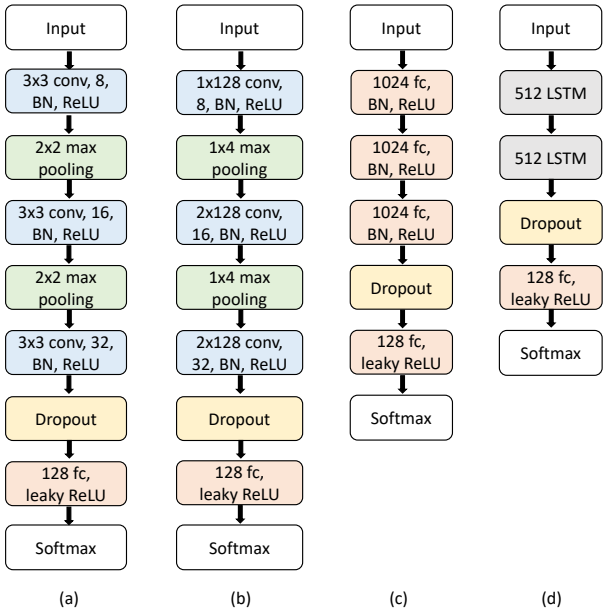



Fig. 4. Deep learning model architecture. (a) CNN for spectrograms. (b) CNN for IQ/FFT data. (c) MLP for IQ/FFT data and spectrograms. (d) LSTM for IQ/FFT data and spectrograms.

## VI. Design of Deep Learning Models

This section will introduce the architectures of all the deep learning models as well as the training details. The preamble part of LoRa packets used in our experiments contains 8,192 IQ samples. The FFT result also consists of 8,192 complex numbers. As the deep learning model cannot process complex numbers, we split I and Q branches of IQ data, and the amplitude and phase of FFT result as two independent dimensions. Therefore, the input dimension of deep learning models designed for IQ/FFT data is 2*8192. Unlike IQ and FFT data, spectrogram can be considered as a 2-D image thus leads to a different input dimension. We generate the spectrogram with a rectangular window of length 256 and hop size 128, which leads to a 256*63 spectrogram. We further crop it to a smaller size of 102*63 as the top and bottom parts contain nearly no useful information.

The proposed deep learning models are developed from the well-known ones. For instance, the CNN for spectrogram is developed from LeNet but its structure is further experimentally optimized to adapt to our applications.

### A. CNN Architectures

The architecture of CNN for spectrogram is illustrated in Fig. 4(a). It consists of three 2D convolutional layers with 8, 16, and 32 3*3 filters, respectively. Each convolutional layer is followed by a batch normalization (BN) layer and activated by the rectified linear unit (ReLU) function. There are two maxpooling layers of size 2*2 following the first and second convolutional layer. The last convolutional layer is connected to a fully connected layer activated by leaky ReLU. Dropout is adopted before fully connected layers and padding is used in each convolutional layer.

The architecture of CNN for IQ/FFT is illustrated in Fig. 4(b), which consists of three 2D convolutional layers and one fully connected layer. The three convolutional layers are composed of 8, 16, and 32 filters with filter sizes of 1*128, 2*128, and 2*128, respectively. There are two 1*4 max pooling layers following the first and second convolutional layer. Other settings are the same as the CNN designed for spectrogram.

### B. MLP Architectures

For a fair comparison, the MLPs for spectrogram and IQ/FFT have the same network architecture except for the input dimension. The architecture is shown in Fig. 4(c). It is implemented with 4 fully connected layers. The first three layers have 1024 neurons with ReLU activation and the last one has 128 neurons with leaky ReLU activation. Dropout is adopted after the third fully connected layer.

### C. LSTM Architectures

Similar to the MLP, the LSTMs designed for spectrogram and IQ/FFT have the same network architecture except for the input dimension. We use two LSTM layers with 512 units, tanh is adopted as the activation function. Then we add one fully connected layer with leaky ReLU activation after the second LSTM layer. Dropout is used after the second LSTM layer.

### D. Training Parameters

The deep learning models are trained with the same parameters. We select Adam as the optimizer and the initial learning rate is set to 0.0003. The learning rate drops every 10 epochs with a drop factor of 0.3. The mini-batch size is set to 32 and the L2 regularization factor is 0.0001. The training stops when the maximum epochs are reached (60 epochs). The training and validation loss plateaus when they reached the maximum epochs so that all the models can be considered fully trained. All the networks are implemented with the MATLAB Deep Learning Toolbox[2] and trained on the same PC with a GPU of NVIDIA GeForce GTX 1660.

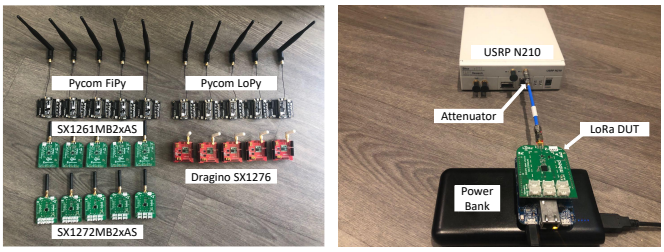
## VII. Experimental Results of CFO Drift

The RF fingerprints must be time-invariant in the presence of environmental and time changes as they represent the user identities. Hence, stability is one of the most essential attributes of an RFFI system. In this section, we experimentally demonstrated that CFO drifts over time and CFO compensation is an indispensable procedure in RFFI systems to prevent system performance from deteriorating.

[2]https://mathworks.com/products/deep-learning.html

TABLE I
LoRa DUTs.

| DUT Index | Model | Chipset |
|---|---|---|
| 1 - 5 | SX1272MB2xAS mbed shield[4] | SX1272 |
| 6 - 10 | SX1261MB2xAS mbed shield[5] | SX1261 |
| 11 - 15 | Pycom FiPy[6] | SX1272 |
| 16 - 20 | Pycom LoPy[7] | SX1276 |
| 21 - 25 | Dragino SX1276 shield[8] | SX1276 |




(a) (b)

Fig. 5. Experimental devices and setup. (a) LoRa DUTs. (b) The LoRa transmitter and USRP receiver connected by a 40 dB attenuator.

### 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. 5(a). 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 to configure the USRP N210 SDR and access data from it[3]. 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. 5(b). The packet transmission interval was set to 1 second.

### B. CFO Drift

To evaluate the CFO drift in a long term, we carried extensive experiments spanning seven months, namely April and September, October, and November 2020. The five SX1272MB2xAS LoRa devices were tested as a case study. We collected 3,000 packets per day from each device, and each collection lasted about one hour.

[3]https://mathworks.com/help/supportpkg/usrpradio/
[5]https://os.mbed.com/components/SX1272MB2xAS/
[6]https://os.mbed.com/components/SX126xMB2xAS/
[7]https://pycom.io/product/fipy/
[8]https://pycom.io/product/lopy4/

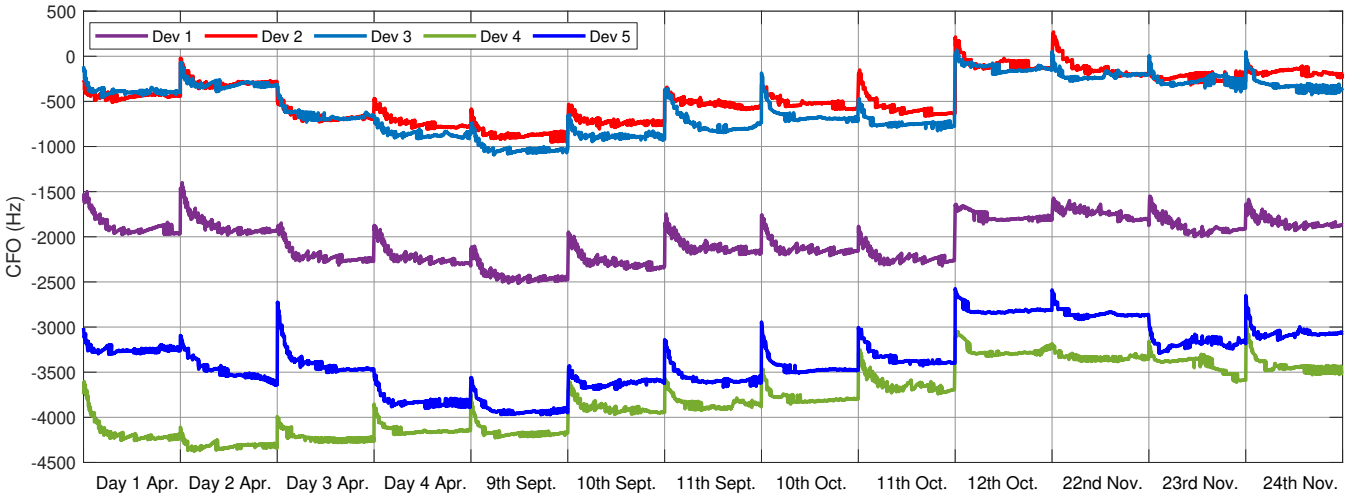



Fig. 6. CFO drift over seven months. The experiments were carried out in April, September, October, and November 2020. The data collection in April for each device were not completed on a single day thus cannot be labeled with a specific date.

The CFO of each packet was estimated using the algorithm introduced in Section IV-C. As can be observed in Fig. 6, the CFO 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 [20]. The short-time variation indicates that the CFO of devices changes rapidly within a short time after they are powered on.

It is also observed in Fig. 6 that there is a non-negligible and unpredictable CFO change over 7 months. The drift is probably caused by uncontrollable environmental conditions such as room temperature. However, it can also be found that the CFO does not change dramatically over several continuous days, which is possibly due to the relatively stable room temperature.

In machine learning tasks, training and test sets are often required to have the same, at least similar data distributions. Otherwise, the trained model will perform poorly on the test set. However, the CFO variation indicates that the training data and test data may have different CFOs when they are collected on different days, which will result in different data distributions. In such cases, CFO compensation is necessary to solve the drift problem caused by CFO variations.

The received packet contains thousands of IQ samples (dimensions). The t-distributed stochastic neighbor embedding (t-SNE) is a well-known non-linear dimensionality reduction technique to visualize high-dimensional data, which has been used for data visualization in a wide range of applications, including RFFI [19]. Hence, we can use t-SNE to visualize the IQ data collected on different days and understand the effectiveness of CFO compensation.

The results are given in Fig. 7. We used the data of device 1 as an example. Each point represents one packet, and the colors of points indicate the day on which the packet was collected. From Fig. 7(a), the IQ data collected on different days gather as several clusters when CFO was not compensated. After CFO compensation, the IQ data collected on different days were mixed and cannot be separated intuitively, as shown in Fig. 7(b). This is desired for RFFI because the IQ data collected from one device should be time-invariant. Therefore, CFO compensation is an indispensable process in RFFI systems.

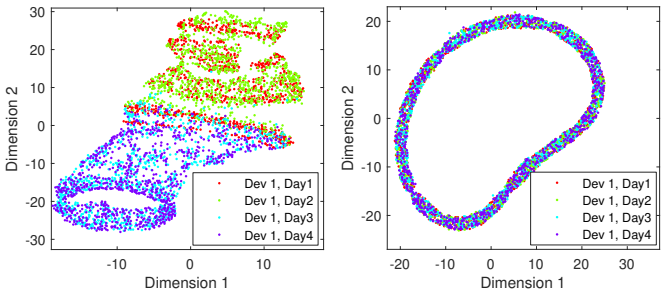



Fig. 7. t-SNE visualization of IQ data. (a) Without CFO compensation. (b) With CFO compensation.

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

Besides the data collection for five SX1272MB2xAS LoRa devices on four days in April, we also carried out the same experiments for the five SX1261MB2xAS LoRa devices. We used the data from these ten devices on four days to evaluate the effect of CFO drift on RFFI. The spectrogram was selected as the signal representation and the CNN model in Fig. 4(a) was used. The CNN was trained with the first 1,000 packets of each device (1,000*10 packets in total) from the 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.

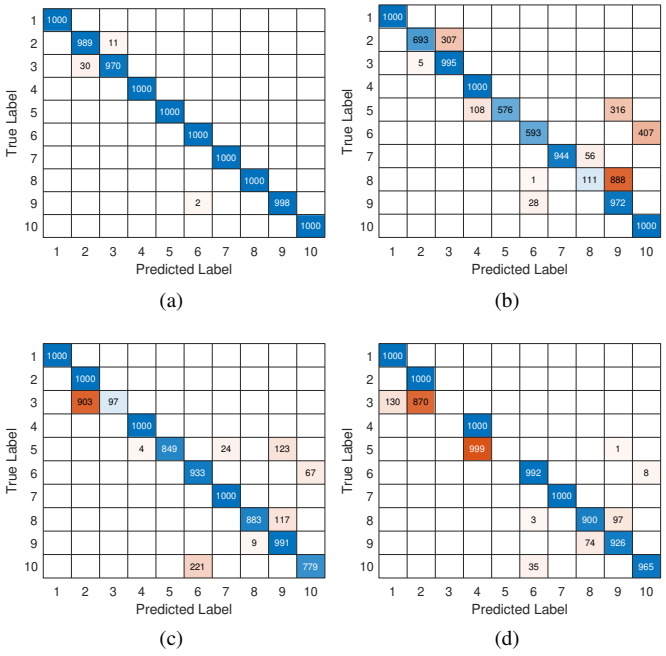



Fig. 8. Experimental results without CFO compensation (spectrogram-CNN model). (a) Day 1 Training, Day 1 Test, overall accuracy: 99.57%. (b) Day 1 Training, Day 2 Test, overall accuracy: 78.84%. (c) Day 1 Training, Day 3 Test, overall accuracy: 85.32%. (d) Day 1 Training, Day 4 Test, overall accuracy: 77.83%.

Fig. 8 shows the confusion matrices obtained by the spectrogram-CNN model when CFO compensation was not applied. Figs. 8(a), 8(b), 8(c), and 8(d) 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. 8(a)), 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. 8(b), 8(c), and 8(d)), the classification results were unacceptable as several devices were completely misclassified, e.g., Dev 3 and Dev 5 in Fig. 8(d). As shown in Fig. 9(a), it can be observed that the CFO of Dev 3 drifted by hundreds of hertz from Day 1 to Day 4. The CFO of Dev 3 test data (light blue dashed line) was closer to Dev 2 training data (red solid line) rather than Dev 3 training data (light blue solid line). Similarly, as shown in Fig. 9(b), the CFO of Dev 5 test data (blue dashed line) was closer to Dev 4 training data (green solid line) rather than Dev 5 training data (blue solid line). 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. 10 shows the confusion matrices obtained by the spectrogram-CNN model when CFO compensation was applied. The accuracy always maintained above 96% on the four days. These results reveal that CNN can classify devices with high accuracy after CFO compensation and performance degradation is significantly mitigated.

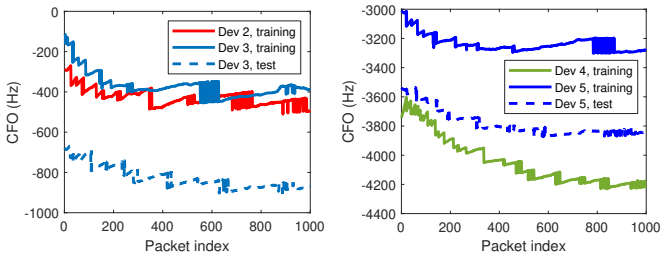



Fig. 9. The comparison of CFO between the Day 1 training data and Day 4 test data. (a) Comparsion between Dev 2 and Dev 3. (b) Comparsion between Dev 4 and Dev 5.

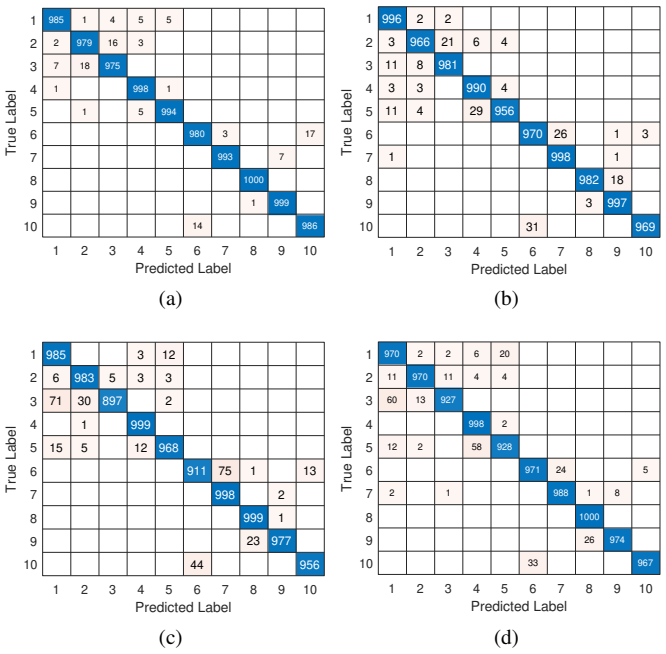



Fig. 10. Experimental results with CFO compensation (spectrogram-CNN model). (a) Day 1 Training, Day 1 Test, overall accuracy: 98.89%. (b) Day 1 Training, Day 2 Test, overall accuracy: 98.05%. (c) Day 1 Training, Day 3 Test, overall accuracy: 96.73%. (d) Day 1 Training, Day 4 Test, overall accuracy: 96.93%.

## VIII. Experimental Evaluations in a Real Wireless Environment

In Section VII, the LoRa DUT and USRP were connected using an attenuator, which allowed us to investigate the CFO effect on RFFI without the impact of the wireless channel. However, this is not a practical application scenario. Our proposed RFFI system will be further evaluated in a real wireless environment in this section.

### A. Experimental Setup

We increased the number of LoRa DUTs to 25 in this section. As shown in Table I and Fig. 5a, these LoRa devices were from five manufacturers. The same USRP N210 SDR was used as the receiver. The LoRa DUTs and USRP were configured with the same parameters as described in Section VII-A. However, we reduced the transmission interval to 0.3 seconds to accelerate the signal collection.

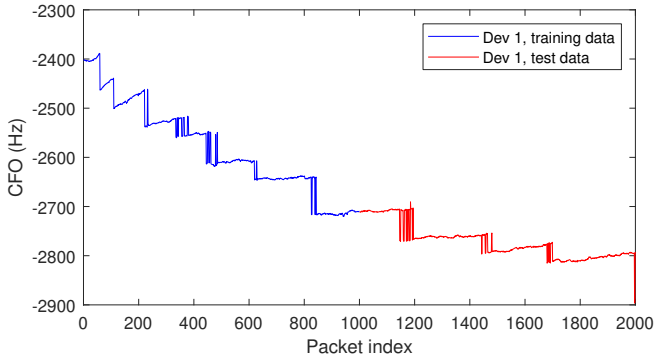



Fig. 11. CFO of each packet in the dataset of Dev 1.

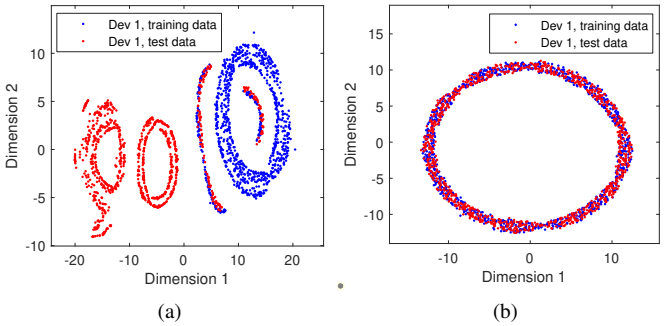



Fig. 12. t-SNE visualization of the training and test sets of Dev1. (a) Without CFO compensation. (b) With CFO compensation.

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 was about three meters and there was a line of sight (LOS) between them. We collected 2,000 packets continuously from each device, which lasted 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.

We evaluated different signal representations and deep learning models. We used the first 1,000 packets of each device as the 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 impact of CFO in a wireless environment, the selection of signal representation and deep learning models, and the effectiveness of our proposed hybrid classifier.

### *B. Impact of CFO Drift*

We take Dev 1 as an example to illustrate the CFO variation of the collected data. Fig. 11 shows the CFO of each packet collected from Dev 1, and presents a similar pattern with Fig. 6 that the CFO decreased after the device was powered on. In the wireless experiments, we used packets 1-1,000 to train the deep learning model and packets 1,001-2,000 to evaluate their performance. The packets in the test set (red curve) have different CFOs from those in the training set (blue curve).

Similar to Section VII-B, we used t-SNE to visualize the training and test data and the results are shown in Fig. 12. There are 2,000 points and the blue points represent packets 1-1,000 (training data) and red points represent packets 1,001-2,000 (test data). From Fig. 12(a) it can be observed that there are distinct clusters when there is no CFO compensation, which indicates that the training and test data have different features/distributions. In contrast, as shown in Fig. 12(b), the blue and red points are mixed after CFO compensation and cannot be separated intuitively. This is expected because the features of each device should be time-invariant after CFO compensation.

The overall classification accuracies are shown in Table II. When there was no CFO compensation, the accuracies are not satisfying for all the signal representations and deep learning models. Take spectrogram-CNN as an example, the overall accuracy was only 83.53%. After CFO compensation was applied, the accuracy significantly increased to 95.35%. The confusion matrices were given in Figs. 13(a) and 13(b) that provide more detailed information.

### *C. Selection of Signal Representation and Deep Learning Model*

The most crucial step to establishing deep learning-based RFFI systems is to select an appropriate signal representation and build a suitable deep learning model for it. The system performance can be roughly evaluated by three metrics: classification accuracy, system complexity and training time.

**Classification Accuracy**: Classification accuracy is the primary metric in evaluating the RFFI system as it measures the ability to correctly identify devices. As shown in Table. II, the spectrogram-CNN model achieves the highest classification accuracy, namely 95.35%. The FFT-MLP model has an accuracy of 94.48% while the IQ-CNN model reaches 92.26%.

In addition to this, we can observe two underperforming combinations: IQ-MLP and FFT-LSTM, whose accuracies are only 55.73% and 58.26%, respectively. For IQ-MLP, the designed MLP may lack the ability to extract distinctive features from LoRa IQ samples. For FFT-LSTM, there is no time-correlation in the FFT data since it contains only frequency-domain information, which is not suitable as the input of LSTM networks.

**System Complexity and Training Time**: System complexity and training time are additional required considerations. The RFFI systems are desired to have low complexity and as few parameters as possible so that the demand on the authenticator hardware (memory, computing power) will be reduced and the authentication will be faster. Table. II reveals that the spectrogram-based CNN has the least amount of parameters, namely 1,545,193, and the trained CNN only takes up 5,679 kb of storage space.

The training time is another essential metric. High-performance GPUs are often required in deep learning to accelerate the training process, while it is impractical to have the authenticators carry expensive GPUs which will

TABLE II
CLASSIFICATION ACCURACY OF DIFFERENT MODELS, NUMBER OF PARAMETERS AND REQUIRED TRAINING TIME.

| Signal Representation | Deep Learning Model | Accuracy | | | # of Parameters | Training Time |
|---|---|---|---|---|---|---|
| | | w/o CFO Comp. | w/ CFO Comp. | Hybrid | | |
| IQ samples | MLP | 54.08% | 55.73% | 78.26% | 19,018,009 | 25 min |
| | CNN | 64.10% | 92.26% | **98.11%** | 4,361,545 | 75 min |
| | LSTM | 61.16% | 89.54% | 95.14% | 4,267,289 | 70 min |
| FFT results | MLP | 55.44% | 94.48% | 96.17% | 19,018,009 | 25 min |
| | CNN | 61.14% | 82.10% | 85.58% | 4,361,545 | 75 min |
| | LSTM | 49.20% | 58.26% | 82.81% | 4,267,289 | 69 min |
| Spectrogram | MLP | 88.60% | 91.82% | 95.95% | 8,821,017 | 22 min |
| | CNN | 83.53% | **95.35%** | 96.40% | **1,545,193** | 20 min |
| | LSTM | 68.16% | 89.50% | 98.04% | 3,427,609 | 80 min |

significantly increase the deployment cost. As demonstrated in Table. II, the spectrogram-based CNN requires 20 mins for training, which is only a quarter of the spectrogram-based LSTM. As the number of DUT increases and the deep learning model deepens, the data sets will grow and the training time may even increase to a few days. Then the advantages of shorter training sessions will become more important.

**Comprehensive Comparison**: The combination of spectrogram-CNN leads to the highest classification accuracy with the least complexity and training time. Although FFT-MLP can reach an accuracy of 94.48% which is only 0.87% lower, the designed MLP has 19,018,009 learnable parameters, almost 12 times the number of spectrogram-based CNN's parameters. For LoRa signals, the spectrogram-CNN systems are highly recommended. The LoRa hardware imperfections will affect both time and frequency domains. The spectrogram can help reveal this information, and CNN is particularly efficient at extracting the features hidden in spectrograms (2D images).

### *D. Comparison to Previous Work*

Roy *et al.* compared the performance of MLP, CNN and LSTM on classifying eight USRP B210 devices with QPSK modulation [17]. However, they only considered IQ samples as the signal representation. The most relevant works on LoRa RFFI are [19] and [16] that employ deep learning techniques as well. Robyns *et al.* [19] selected the FFT as the signal representation and investigated three different learning models, namely support vector machine (SVM), MLP and CNN. Das *et al.* [16] directly used the IQ samples as the signal representation and compared the performance of SVM, MLP and LSTM. Their models correspond to the combination of FFT-CNN, FFT-MLP, IQ-LSTM and IQ-MLP, which have been included in our work and the comparison results can be found in Table II. Both papers concluded that the performance of SVM is much worse than deep learning models therefore SVM is not studied in this work.

### *E. Performance of the Hybrid Classifier*

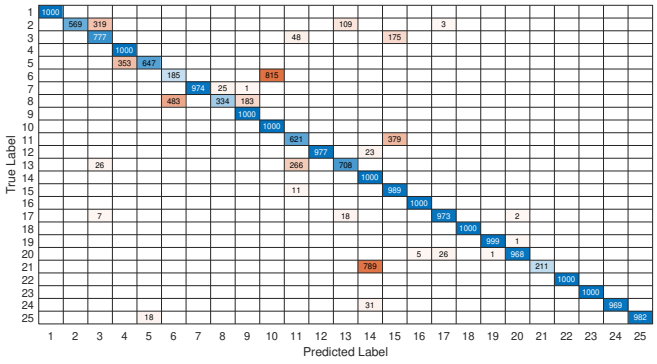
The hybrid classifier introduced in Section V-E calibrates the softmax output of the deep learning model according to the estimated CFO. As shown in Fig. 6, the CFO varies over different days and some devices may have similar CFOs, hence it cannot be used solely as an RF fingerprint to identify numerous low-cost IoT devices. However, CFO will not change dramatically thus can be used for calibration to rule out predictions whose estimated CFO is much different from the reference CFO.

As shown in Table II, it can be observed that the hybrid classifier can increase the accuracy for all the signal representations and deep learning models. The most significant improvement was the FFT-LSTM; the accuracy with the hybrid classifier increased from 58.26% to 82.81% , which was a 24.55% improvement.

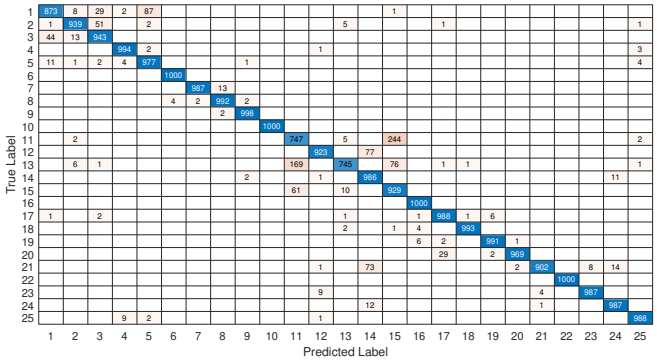
The beneficial improvement of hybrid classification is achieved at a low cost. The calibration process only requires $K$ additional comparison steps, which is easy to implement and requires only a small amount of computing resources.

## IX. CONCLUSION

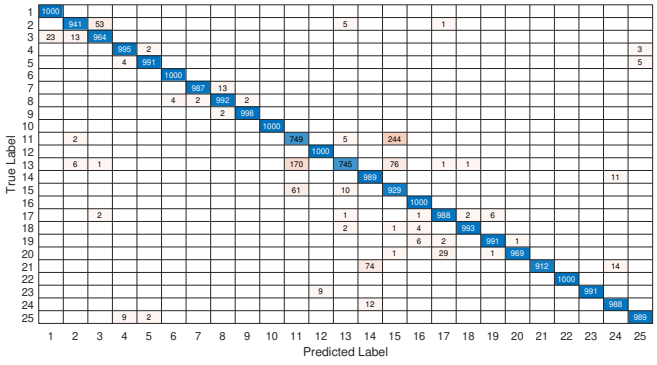
In this paper, we proposed a deep learning-based RFFI scheme to classify LoRa devices and carried out extensive experimental evaluations involving 25 LoRa devices from five manufacturers as the DUTs and a USRP N210 SDR as the receiver. Firstly, CFO was experimentally found to vary over time and compromise the system stability. CFO compensation was demonstrated to be effective in mitigating the performance degradation. Secondly, we investigated three signal representations, namely IQ samples, FFT results and spectrograms, and built three deep learning models, i.e., MLP, LSTM and CNN. The spectrogram-CNN model was shown to achieve the highest classification accuracy with the least complexity and the shortest training time. Finally, a hybrid classifier was proposed to calibrate the softmax output of deep learning models using the estimated CFO. The range of CFO variations was found to stay relatively stable over several continuous days so it is helpful to rule out predictions when the estimated CFO deviates greatly from the reference CFO. Our proposed RFFI system achieved a classification accuracy of 96.40% in distinguishing 25 LoRa devices in real wireless environments, when the hybrid spectrogram-CNN was used.




(a)




(b)




(c)

Fig. 13. Classification results of the spectrogram-based CNN. (a) Without CFO compensation, overall accuracy: 83.53%. (b) With CFO compensation, overall accuracy: 95.35%. (c) With CFO compensation, hybrid classifier, overall accuracy: 96.40%.

## REFERENCES


[1] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRa using spectrogram and CNN," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Virtual Conference, May 2021. [Online]. Available: https://arxiv.org/abs/2101.01668

[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] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, "RFAL: Adversarial learning for RF transmitter identification and classification," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 783–801, 2019.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] U. Satija, N. Trivedi, G. Biswal, and B. Ramkumar, "Specific emitter identification based on variational mode decomposition and spectral features in single hop and relaying scenarios," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 581–591, 2018.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] M. Cekic, S. Gopalakrishnan, and U. Madhow, "Robust wireless fingerprinting: Generalizing across space and time," *arXiv preprint arXiv:2002.10791*, 2020.

[32] I. Agadakos, N. Agadakos, J. Polakis, and M. R. Amer, "Chameleons' oblivion: Complex-valued deep neural networks for protocol-agnostic RF device fingerprinting," in *Proc. IEEE European Symposium on Security and Privacy (EuroS&P)*, Genoa, Italy, Sep. 2020, pp. 322–338.

[33] B. He and F. Wang, "Cooperative specific emitter identification via multiple distorted receivers," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3791–3806, 2020.

[34] 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.

[35] A. Gritsenko, Z. Wang, T. Jian, J. Dy, K. Chowdhury, and S. Ioannidis, "Finding a 'new' needle in the haystack: Unseen radio detection in large populations using deep learning," in *Proc. IEEE Int. Symp. Dynamic Spectrum Access Networks (DySPAN)*, Newark, NJ, USA, 2019, pp. 1–10.

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

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

[38] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, 1997.

[39] 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.

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

[41] 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.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference Computer Vision Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.


**Junqing Zhang** received the B.Eng and M.Eng degrees in Electrical Engineering from Tianjin University, China in 2009 and 2012, respectively, and the Ph.D degree in Electronics and Electrical Engineering from Queen's University Belfast, UK in 2016. From Feb. 2016 to Jan. 2018, he was a Postdoctoral Research Fellow with Queen's University Belfast. From Feb. 2018 to May 2020, he was a Tenure Track Fellow (Assistant Professor) with University of Liverpool, UK. Since June 2020, he is a Lecturer (Assistant Professor) with University of Liverpool. His research interests include Internet of Things, wireless security, physical layer security, key generation, and radio frequency fingerprinting identification.

**Alan Marshall** (M'88–SM'00) holds the Chair in communications networks with the University of Liverpool, where he is the Director of the Advanced Networks Group and the Head of the Department. He is a fellow of the Institution of Engineering and Technology and senior fellow of the Higher Education Academy. He has spent over 24 years working in the telecommunications and defense industries. He has published over 250 scientific papers and holds a number of joint patents in the areas of communications and network security. He formed a successful spin-out company Traffic Observation and Management Ltd. His research interests include mobile and wireless network architectures and protocols, network security and multisensory communications including haptics and olfaction. He is currently a Section Editor of the Computer Journal of the British Computer Society and an Editorial Board Member of the Journal of Networks.

**Guanxiong Shen** received the B.Eng degree from Xidian University, Xi'an, China, in 2019. He is currently pursuing the Ph.D degree at the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, U.K. His current research interests include the Internet of Things, wireless security and radio frequency fingerprint identification.

**Linning Peng** received his PhD degree from IETR (Electronics and Telecommunications Institute of Rennes) laboratory at INSA (National Institute of Applied Sciences) of Rennes, France, in 2014. From 2014, he has been an associate professor with School of Information Science and Engineering, Cyber Science and Engineering, Southeast University, China. He is author or coauthor of more than 30 technical papers in international scientific journals and conferences, and holds more than 10 Chinese patents. His current research interests are in the areas of physical layer security in wired and wireless communications such as device fingerprint based identification, secret keys generation from wireless channel, and IoT system security.

**Xianbin Wang** (S'98-M'99-SM'06-F'17) is a Professor and Tier-1 Canada Research Chair at Western University, Canada. He received his Ph.D. degree in electrical and computer engineering from the National University of Singapore in 2001.

Prior to joining Western, he was with Communications Research Centre Canada (CRC) as a Research Scientist/Senior Research Scientist between July 2002 and Dec. 2007. From Jan. 2001 to July 2002, he was a system designer at STMicroelectronics. His current research interests include 5G/6G technologies, Internet-of-Things, communications security, machine learning and intelligent communications. Dr. Wang has over 450 highly cited journal and conference papers, in addition to 30 granted and pending patents and several standard contributions.

Dr. Wang is a Fellow of Canadian Academy of Engineering, a Fellow of Engineering Institute of Canada, a Fellow of IEEE and an IEEE Distinguished Lecturer. He has received many awards and recognitions, including Canada Research Chair, CRC President's Excellence Award, Canadian Federal Government Public Service Award, Ontario Early Researcher Award and six IEEE Best Paper Awards. He currently serves/has served as an Editor-in-Chief, Associate Editor-in-Chief, Editor/Associate Editor for over 10 journals. He was involved in many IEEE conferences including GLOBECOM, ICC, VTC, PIMRC, WCNC and CWIT, in different roles such as symposium chair, tutorial instructor, track chair, session chair, TPC co-chair and keynote speaker. He has been nominated as an IEEE Distinguished Lecturer several times during the last ten years. Dr. Wang is currently serving as the Chair of IEEE London Section and the Chair of ComSoc Signal Processing and Computing for Communications (SPCC) Technical Committee.