Towards Scalable and Channel-Robust Radio Frequency Fingerprint Identification for LoRa

Guanxiong Shen, Junqing Zhang, *Member, IEEE*, Alan Marshall, *Senior Member, IEEE*, and Joseph Cavallaro, *Fellow, IEEE*

***Abstract*—Radio frequency fingerprint identification (RFFI) is a promising device authentication technique based on transmitter hardware impairments. The device-specific hardware features can be extracted at the receiver by analyzing the received signal and used for authentication. In this paper, we propose a scalable and channel-robust RFFI framework achieved by deep learning powered radio frequency fingerprint (RFF) extractor and channel independent features. Specifically, we leverage deep metric learning to train an RFF extractor, which has excellent generalization ability and can extract RFFs from previously unseen devices. Any devices can be enrolled via the pre-trained RFF extractor and the RFF database can be maintained efficiently for allowing devices to join and leave. Wireless channel impacts the RFF extraction and is tackled by exploiting channel independent features and data augmentation. We carried out extensive experimental evaluation involving 60 commercial off-the-shelf LoRa devices and a USRP N210 software defined radio platform. The results have successfully demonstrated that our framework can achieve excellent generalization abilities for rogue device detection and device classification as well as effective channel mitigation.**

***Index Terms*—Internet of things, device authentication, radio frequency fingerprint identification, deep learning, LoRa**

I. Introduction

With the rapid growth in the population of the Internet of things (IoT) devices, their security is becoming increasingly important. Device authentication is required to prevent the intrusion of rogue devices that may execute harmful processes or access to private data as well as infer the device identity [1]. It is usually achieved by cryptographic schemes such as the common challenge-response-based protocols. These schemes require a common key pre-shared which can be done by public-key cryptography (PKC) such as Diffie-Hellman key exchange. However, PKC may not be affordable for many IoT devices because they are usually low-cost and power constrained [2]. They also rely on software addresses such as MAC addresses, which can be tampered with easily.

Lightweight and reliable device authentication is thus urgently required for ensuring IoT security. Radio frequency fingerprint identification (RFFI) is a promising non-cryptographic technique that relies on the device intrinsic hardware impairments of radio frequency (RF) components generated during the manufacturing process. The hardware characteristics of these components slightly deviate from their nominal values but the deviation is so small that the normal communication functionality is not affected. These hardware features are unique and difficult to be tampered with, thus they can be treated as the fingerprint of IoT devices, in a similar manner as the biometrics of human beings. The transmitter hardware impairments will distort the transmission waveform, from which the receiver can infer the transmitter identity by extracting the device-specific radio frequency fingerprints (RFFs). RFFI can thus identify transmitters on a per-packet basis. All the RFFI operations are completed at the receiver and there is no modification of the transmitter. Hence, it is particularly suitable for power-constrained and low-cost IoT devices.

The existing RFFI work can be categorized into traditional RFF extractor-based and deep learning-based approaches. The former relies on a manually designed RFF extractor to obtain hardware features such as IQ imbalance [3], [4], carrier frequency offset (CFO) [4]–[9], power spectral density [10], [11], power amplifier non-linearity [12], beam pattern [13], Hilbert-Huang spectrum [14], [15], etc. However, such schemes are highly dependent on the quality of the designed feature extraction algorithms and require a deep understanding of the adopted communication protocol. Apart from this, some hardware characteristics are interrelated with each other, making it challenging to extract each feature individually. In contrast, deep learning-based approaches usually rely on a classification neural network to process raw signals and directly infer device identity without the need for feature engineering, which has attracted wide attention in recent years [16]–[35].

As shown in Fig. 1, there are some basic requirements for IoT RFFI. First, it needs to support fast device joining and leaving, and be able to detect the presence of rogue devices. Second, it should be robust to channel variations as device moving is common in IoT networks. However, previous deep learning-based RFFI systems cannot meet these requirements. They usually leverage a classification neural network with softmax layer to directly make decisions [16], [18]–[22], [32], [36]. However, the number of neurons of the softmax layer is unchangeable after training, which restricts the RFFI to a close-set classification problem. This classification neural network-based system design lacks indispensable capabilities:

Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. The work was in part supported by UK Royal Society Research Grants under grant ID RGS/R1/191241 and National Key Research and Development Program of China under grant ID 2020YFE0200600. 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)

J. Cavallaro is with the Department of Electrical and Computer Engineering, Rice University, Houston, USA. (email: cavallar@rice.edu)

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

Digital Object Identifier xxx

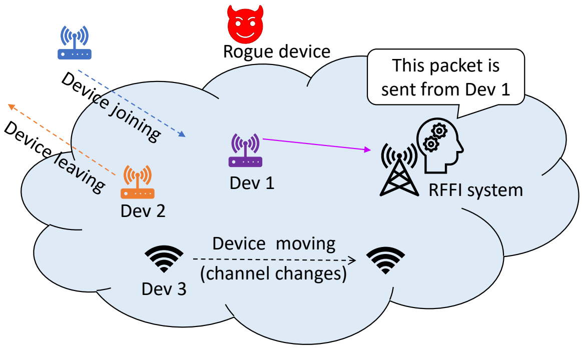


Fig. 1: IoT networks secured by RFFI. Deep learning-based RFFI systems suffer from limited scalability and channel robustness.

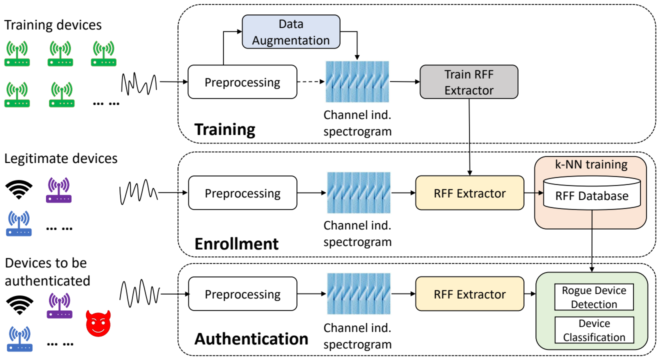


Fig. 2: System diagram of the proposed RFFI system.

- **System scalability**: As the number of softmax layer neurons is fixed, the classification neural network can only classify the devices that are present during training. It needs to be retrained whenever legitimate devices join and leave the IoT network, which is time-consuming and not practical.
- **Rogue device detection capability**: A complete RFF authentication procedure requires two steps, namely rogue device detection and classification. The former determines whether the device belongs to the legitimate group and the latter further predicts a device label. However, the softmax layer-based classification neural network does not have the first function. The rogue device will be classified as a legitimate one with the most similar characteristics in the training categories [20], [22], [28], which is not acceptable.

Moreover, all the RFFI systems are subject to channel effects as the received signal is not only distorted by hardware impairments but also the wireless channel. It is still an open problem to enable the RFFI protocols channel-robust.

This paper aims to design a scalable and channel-agnostic RFFI system with rogue device detection capability. Our approach introduces an enrollment stage and uses k-nearest neighbor (k-NN) to replace the classification neural network, making it scalable and able to detect rogue devices. Moreover, the channel independent spectrogram is proposed to mitigate channel effects. Extensive experiments were carried out with 60 commercial off-the-shelf LoRa devices of four models at various channel conditions to demonstrate the excellent performance of the proposed RFFI system. Our contributions are highlighted as follows.

- We design a scalable RFFI framework based on a deep metric learning-powered RFF extractor, which enables device joining and leaving without the need for retraining. It maintains an RFF database by enrolling a new device using the pre-trained RFF extractor or deleting the record of a leaving device.
- We propose a channel robust RFFI protocol by constructing the channel independent spectrogram and exploiting data augmentation. The channel independent spectrogram can mitigate the channel effect in the time-frequency domain while reserving the RFF of the LoRa signal. The data augmentation is carefully designed to represent real channel conditions with both multipath and Doppler shift.
- We conduct extensive experiments involving different LoRa devices, various channel conditions and antenna polarization. We experimentally demonstrate that the RFF extractor can extract features from devices that are not present during training, even they are produced by other manufacturers. The proposed channel independent spectrogram is shown to be effective in mitigating the channel effect and data augmentation can further increase the system robustness. Antenna polarization is found to affect classification performance.

The codes[1] and datasets[2] created in this research are openly available online.

This paper provides a general RFFI framework and studies LoRa as an example. LoRa devices are manufactured with low-cost components therefore have abundant hardware impairments, which are suitable for RFFI. Moreover, the commercial LoRaWAN specification defines cryptography-based device authentication schemes. The root keys are assigned to the end-nodes during fabrication and secure storage of them is a huge challenge. The proposed RFFI can provide an alternative method to authenticate LoRa devices.

The rest of the paper is organized as follows. Section II shows the system overview. Section III introduces the LoRa signal processing. Section IV presents the procedure of constructing the channel independent spectrogram. The training of RFF extractor is introduced in Section V while the enrollment and identification of RFFI systems is introduced in Section VI. Section VII provides extensive evaluation results to show the system performance. Section VIII and Section IX introduces related work and concludes the paper, respectively.

II. System Overview

The proposed system is based on the deep learning-powered RFF extractor. As shown in Fig. 2, it involves training, enrollment and authentication stages.

Training: The training stage leverages the outstanding feature extraction capability of deep learning to generate an

[1] https://github.com/gxhen/LoRa_RFFI
[2] https://ieee-dataport.org/open-access/lorarffidataset

RFF extractor rather than a classification neural network. Specifically, a large number of labelled packets are collected from numerous training devices. The data augmentation is used to increase the channel diversity. The training packets are then transformed to channel independent spectrograms to mitigate channel effects. Finally, we leverage the triplet loss in deep metric learning to train the RFF extractor. The input of the extractor is a 2-D channel independent spectrogram, and the output is a vector consisting of 512 elements, which is the unique RFF of that device.

The training only needs to be done once as the trained RFF extractor is able to extract unique RFFs from out-of-library (unseen) devices. The training devices are not necessarily the same as the ones for enrollment and authentication. The training can be done by a third party that owns massive data collected from a huge number of devices to train an RFF extractor with excellent generalization capability.

Enrollment: The enrollment will obtain the RFFs of legitimate devices using the RFF extractor. These are the actual working devices in an IoT network, which are probably different from the training devices. These legitimate devices are required to send several packets to the RFFI system. Then RFFs can be extracted from the received packets using the trained RFF extractor and stored in a database. The RFFs of the newly joined devices will be added to the database and the RFFs of devices that leave the system will be deleted, which allows the system to be updated efficiently. The enrollment should be carried out in a controlled environment to ensure the RFFs of rogue devices are not enrolled.

Authentication: A complete authentication system should consist of two parts, namely rogue device detection[3] and device classification. Rogue device detection first determines whether the transmitter belongs to the legitimate group (previously enrolled), and device classification further infers its label. Both of them are implemented by the k-NN algorithm.

III. LoRa Signal Processing

A. LoRa Signal

LoRa is based on the chirp spread spectrum (CSS) technology which uses chirps for communication. The instantaneous frequency of the LoRa signal changes continuously over time, and a basic LoRa symbol (up-chirp) can be written as

$$u(t) = Ae^{j2\pi(-\frac{B}{2}+\frac{B}{2T}t)t} \quad (0 \leq t \leq T), \tag{1}$$

where A and B denote amplitude and bandwidth, respectively. T is the LoRa symbol duration, given as

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

where SF is the spreading factor. There are eight repeating up-chirps at the beginning of a LoRa packet called the preamble, which is identical in every LoRa packet regardless of the device type.

B. Signal Acquisition

The baseband transmitted signal, $x(t)$, undergoes signal modulation and up-conversion via hardware components such as oscillator and power amplifier, etc. These components have specific impairments and their overall effect is denoted as $f(\cdot)$. The signal then travels via the wireless channel and is captured by the receiver. The baseband received signal is given as

$$y(t) = h(\tau, t) * f(x(t)) + n(t), \tag{3}$$

where $h(\tau, t)$ is the time-varying channel impulse response, $n(t)$ is the additive white Gaussian noise and $*$ denotes convolution operation. The received signal, $y(t)$, is further converted to digital samples by an analog-digital-converter (ADC) with a sampling interval of T_s, denoted as $y[nT_s]$. It is simplified to $y[n]$ for easy notation.

C. Preprocessing

The received signal needs to be pre-processed to meet the basic requirements of RFFI, including synchronization, CFO compensation and normalization. These algorithms are briefly described below and detailed descriptions can be found in our prior work [36].

Synchronization: Synchronization locates the starting point of a packet. Inaccurate synchronization introduces a segment of channel noise, which will affect the RFFI performance.

Preamble Extraction: The deep learning model can learn the identity-related information such as the MAC address if the entire packet is used for RFFI. To prevent this, only the preamble part, $s'[n]$, is employed in the proposed system.

CFO Compensation: CFO compensation is essentially required in RFFI for system stability. It has been demonstrated that crystal oscillators are particularly sensitive to temperature changes [37], and the drift of oscillator frequency can seriously degrade the system performance [18], [20].

Normalization: Normalization prevents the system from learning the received power that is not device-specific. The preamble part is normalized by dividing the root mean square (RMS) of it. The preprocessed signal is denoted as $s[n]$.

IV. Channel Independent Spectrogram

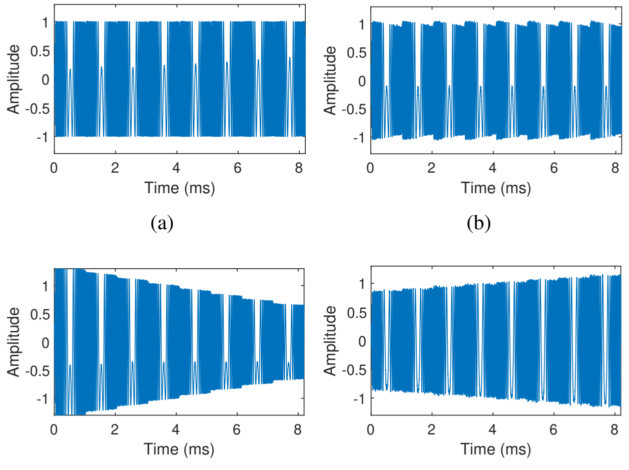
The received signal is not only distorted by the hardware impairments but also by the wireless channel. It is impossible to ensure that devices experience the same channel at different times. We propose the channel independent spectrogram to mitigate the channel effect in the time-frequency domain while preserving the device-specific characteristics.

A. Observation of Channel Effect

LoRa supports three bandwidth options, namely 125 kHz, 250 kHz and 500 kHz. LoRa transmissions are sometimes assumed to experience flat fading where the wireless channel causes the same magnitude and phase changes to all the frequency components.

However, we experimentally found that the wireless channel significantly distorts the LoRa signal whose effect is evident from the received waveform. We collected LoRa packets in

[3] Some work uses the term 'identification' for referring to detecting rogue devices.



(a) (b)

(c) (d)

Fig. 3: Normalized received LoRa waveform. (a) LOS stationary scenario. (b) NLOS stationary scenario. (c) LOS mobile scenario. (d) NLOS mobile scenario.

line-of-sight (LOS) stationary and non-line-of-sight (NLOS) stationary scenarios as well as LOS mobile and NLOS mobile scenarios. Detailed experimental setting can be found in Section VII. The preamble part (I-branch) is shown in Fig. 3. It can be observed that the waveforms collected under various scenarios are different. We will later in Section V-A demonstrate that the sawtooth shapes are caused by the time dispersion (multipath effect) and the amplitude variation is due to channel changes (Doppler effect).

B. Short-time Fourier Transform

The LoRa signal is usually analyzed in the time-frequency domain because of its non-stationary property. Short-time Fourier transform (STFT) is an efficient time-frequency analysis algorithm which can reveal the time-frequency features of the signal. The discrete STFT is mathematically written as

$$S_{k,m} = \sum_{n=0}^{N-1} s[n]w[n-mR]e^{-j2\pi\frac{k}{N}n} \qquad (4)$$
$$\text{for } k = 1,2,...,N \text{ and } m = 1,2,...,M,$$

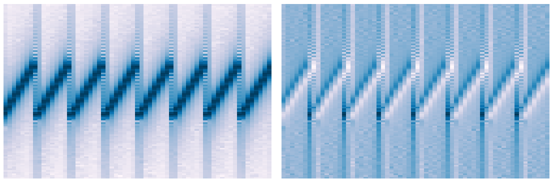
where $S_{k,m}$ is the element of the STFT complex matrix $\mathbf{S}$. M is number of columns of $\mathbf{S}$, given as

$$M = \frac{8 \cdot \frac{2^{SF}}{B} \cdot \frac{1}{T_s} - N}{R} + 1. \qquad (5)$$

N is the number of rows of $\mathbf{S}$, which is also the length of window function $w[n]$. R is the hop size. In our experimental implementation, SF = 7, B = 125 kHz, sampling rate T_s = 1 μs, N is 256, R is 128, and M is calculated as 63. The spectrogram in dB scale, $\widetilde{\mathbf{S}}$, is given as

$$\widetilde{\mathbf{S}} = 10\log_{10}(|\mathbf{S}|^2), \qquad (6)$$

where $|\cdot|$ returns the amplitude. The spectrogram of the LoRa preamble part is shown in Fig. 4a.



(a) (b)

Fig. 4: (a) Spectrogram of LoRa preambles. (b) Channel independent spectrogram of LoRa preambles.

C. Constructing Channel Independent Spectrogram

The spectrogram is affected by the wireless channel, which can be mitigated by dividing the adjacent columns. STFT can be considered as splitting $s[n]$ into M segments of N samples, then performing FFT on each segment. According to our STFT implementation, M = 63 and N = 256. The overall effect of hardware distortion to the transmitted signal in the frequency domain is denoted as $F(\cdot)$. Therefore, the STFT complex matrix $\mathbf{S}$ can be arranged as

$$\begin{aligned}\mathbf{S} &= \begin{bmatrix} H_{1,1}F(X_{1,1}) & H_{1,2}F(X_{1,2}) & \cdots & H_{1,M}F(X_{1,M}) \\ H_{2,1}F(X_{2,1}) & H_{2,2}F(X_{2,2}) & \cdots & H_{2,M}\ F(X_{2,M}) \\ \vdots & \vdots & \cdots & \vdots \\ H_{N,1}F(X_{N,1}) & H_{N,2}F(X_{N,2}) & \cdots & H_{N,M}F(X_{N,M}) \end{bmatrix} \\ &= [\mathbf{H}_1 \odot F(\mathbf{X}_1) \quad \mathbf{H}_2 \odot F(\mathbf{X}_2) \ \cdots \ \mathbf{H}_M \odot F(\mathbf{X}_M)],\end{aligned} \qquad (7)$$

where $\mathbf{X}_m = [X_{1,m}, X_{2,m} \cdots X_{N,m}]^T$ denotes the ideal frequency spectrum of the m-th signal segment, the column vector $\mathbf{H}_m = [H_{1,m}, H_{2,m} \cdots H_{N,m}]^T$ represents the channel frequency response experienced by the m-th signal segment and $\odot$ denotes element-wise product.

The phase information of $\mathbf{S}$ is noisy and can be affected by several issues such as phase noise, CFO, and sampling time offset. Therefore, only the amplitude of $\mathbf{S}$ is used, formulated as

$$|\mathbf{S}| = [|\mathbf{H}_1| \odot |F(\mathbf{X}_1)| \quad |\mathbf{H}_2| \odot |F(\mathbf{X}_2)| \ \cdots \ |\mathbf{H}_M| \odot |F(\mathbf{X}_M)|]. \qquad (8)$$

In our experimental settings, the time gap between two adjacent signal segments is only 128 μs. It is reasonable to assume $|\mathbf{H}_m| \approx |\mathbf{H}_{m+1}|$ since the wireless channel will not change dramatically in such a short period. Based on this assumption, we can divide the m-th column by the (m+1)-th one so that the channel-related information can be eliminated. The matrix after division is given as

$$\mathbf{Q} = \left[\frac{|F(\mathbf{X}_2)|}{|F(\mathbf{X}_1)|} \quad \frac{|F(\mathbf{X}_3)|}{|F(\mathbf{X}_2)|} \quad \cdots \quad \frac{|F(\mathbf{X}_M)|}{|F(\mathbf{X}_{M-1})|}\right]. \qquad (9)$$

Compared to (8), the channel information is eliminated but the device-specific hardware distortions $F(\cdot)$ are preserved. Similar to (6), the amplitude of $\mathbf{Q}$ is expressed in dB scale

$$\widetilde{\mathbf{Q}} = 10\log_{10}(|\mathbf{Q}|^2). \qquad (10)$$

$\widetilde{\mathbf{Q}}$ is the proposed channel independent spectrogram which is used as the input of RFF extractor. It is also worth noting that the top and bottom of $\widetilde{\mathbf{Q}}$ are cropped by 30% since these parts are far from the central frequency and therefore contain nearly no useful information but noise. The size of the cropped channel independent spectrogram is 102×62 according to our experimental settings. The channel independent spectrogram of the LoRa preamble part is shown in Fig. 4b.

V. RFF Extractor Training

The RFF extractor is the core module in the proposed RFFI system. It should extract channel independent and discriminative RFFs from the received signal and generalize well on previously unseen devices. In our system, the training data can be collected in a controlled environment, and correctly labelling each LoRa packet is not difficult. This enables us to use supervised learning approaches rather than unsupervised ones such as autoencoders to better learn identity-related features.

A. Data Augmentation

Data augmentation is an efficient approach in deep learning to improve performance and has been recently applied in the area of RFFI to increase its robustness to the wireless channel [18], [38], [39]. Data augmentation can generate more training data to increase the performance of the RFF extractor and reduce the overhead for data collection. It can also mitigate the channel effect by injecting various channel distortions into the training data so that the RFF extractor automatically learns how to deal with them.

1) Channel Effect: In this paper, the channel effect for data augmentation includes both multipath and Doppler shift. The multipath is described by the power delay profile (PDP). The exponential PDP is selected and the discrete model is given as

$$P(p) = \frac{1}{\tau_d} e^{-pT_s/\tau_d}, \quad p = 0, 1, \cdots, p_{max}, \tag{11}$$

where τ_d is the RMS delay spread and p_{max} is the index of the last path. The PDP is normalized.

This paper also considers Doppler shift, which is overlooked in previous work as the channel is assumed to be constant in a packet time [18], [38], [39]. However, this assumption might not hold for some LoRa transmissions. Each LoRa preamble typically lasts about 1 ms and the channel effect may not remain constant in mobile scenarios. Actually, the Doppler effect is observed from the received waveform such as Fig. 3c and Fig. 3d. The Doppler effect can be characterized by the Doppler spectrum. This paper adopts the popular Jakes model whose spectrum is defined as

$$S(f) = \frac{1}{\pi f_d \sqrt{1 - (f/f_d)^2}}, \tag{12}$$

where f_d is the maximum Doppler shift.

2) Procedure: The data augmentation is carried out as follows.

1) The training data should be collected in a short-distance LOS stationary scenario so that it can be assumed as experiencing frequency-flat slow fading.

TABLE I: Parameter of the Channel Simulator

Paramter	Range
RMS delay spread τ_d (ns)	[5,300]
Maximum Doppler frequency f_d (Hz)	[0,10]
Rician K-factor	[0,10]
SNR (dB)	[20,80]

2) The number of training samples is increased by replication. In our implementation, the training set is doubled. Note that more replications may improve system performance but will significantly increase the requirements on disk and memory storage.
3) The channel effect with multipath and Doppler effect is generated with the parameters randomly selected within specific ranges given in Table I.
4) The packet generated in step 2) passes through the channel from step 3). The artificial white Gaussian noise is added to the signal to emulate different SNR levels.

The augmentation is completed by repeating the above process for all the training packets.

We show the waveform of three cases, namely strong multipath[4] no Doppler effect, strong Doppler effect with weak multipath as well as both strong multipath and Doppler effects in Fig. 5b, Fig. 5c and Fig. 5d, respectively. It can be observed that the augmented waveforms match well with the real collected ones shown in Fig. 3. We can also conclude that the wireless channel distorts the received LoRa signal significantly. The sawtooth shapes are caused by multipath and amplitude variation is caused by the Doppler effect.

B. Model Architecture

The channel independent spectrogram can be regarded as a 2-D image so convolutional neural network (CNN) is used to extract RFFs from it. The CNN in this paper is designed with reference to the well-known ResNet [40]. It is further optimized to be more lightweight and suitable for the size of channel independent spectrograms.

The architecture of the RFF extractor is shown in Fig. 6, where /2 denotes strides two. It consists of nine convolutional layers, one average pooling layer and one dense layer of 512 neurons, residual structure is also adopted. The first convolution layer uses 32 7×7 filters with stride 2, the second to the fifth layers use 32 3×3 filters, and the sixth to the ninth layers employ 64 3×3 filters. All the convolutional layers are activated by rectified linear unit (ReLU) and padding is used. We leverage an L2 normalization layer to make the RFF extractor learns better features [41]. The output of the RFF extractor is a vector consists of 512 elements which can be considered as the RFF extracted from the received packet. Note that the model hyper-parameters such as the dimension

[4] The level of multipath relies on both the symbol period and the RMS delay spread. As LoRa has a long symbol period, it does not suffer from multipath much. ‘Strong multipath’ in this paper is just a contrast to the ‘weak multipath’ scenario of 5 ns RMS delay. Similarly, ‘strong Doppler effect’ is a contrast to the ‘no Doppler’ scenario.

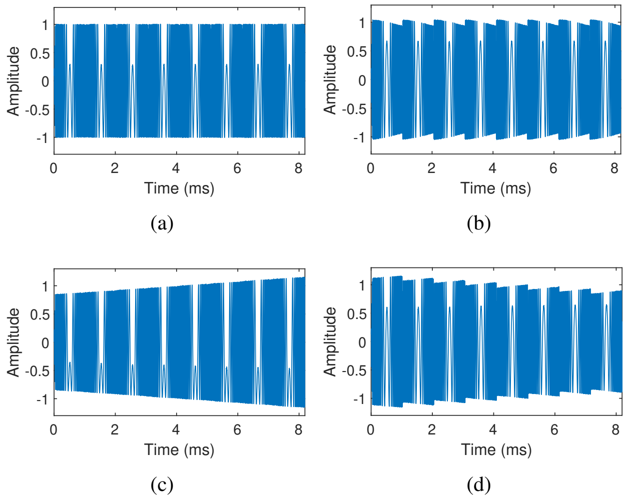


Fig. 5: Augmented LoRa preambles. (a) Original waveform. (b) Strong multipath no Doppler effect (τ_d = 300 ns, f_d = 0 Hz). (c) Strong Doppler effect with weak multipath (τ_d = 5 ns, f_d = 10 Hz). (d) Both multipath and Doppler effects are strong (τ_d = 300 ns, f_d = 10 Hz).

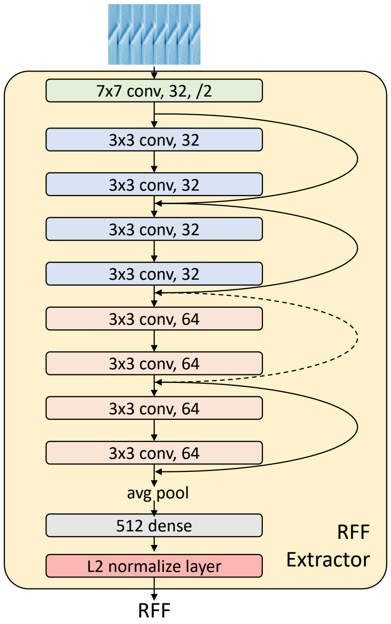


Fig. 6: Architecture of the RFF extractor, revised from ResNet.

of feature vectors are adjusted based on LoRa-RFFI, and can be optimized on a third-party dataset.

The designed feature extractor has 12,458,496 learnable parameters in total, which corresponds to a 47.5 MB file. This is affordable for many embedded systems.

C. Deep Metric Learning

Deep metric learning aims to train a neural network that maps the input to a 1D vector/embedding. There are many loss functions available in deep metric learning such as contrastive loss, L2-softmax loss, etc. Interested readers please refer to [42], [43] for detailed information. Triplet loss is a well-known ranking loss used in deep metric learning and has been successfully adopted in face recognition [44]. It projects the input data into a space where similar samples are close to each other and dissimilar ones are far away. During the training process, a triplet consisting of anchor, positive and negative samples is selected from the training set at each step. Specific to RFFI, anchor and positive samples are packets from the same device and the negative sample is from a different one. As shown in Fig. 7, the RFFs of anchor, positive and negative samples are extracted and the triplet loss can be calculated. The goal of triplet loss is to minimize the Euclidean distance between the anchor and positive samples while maximizing the distance between the anchor and negative samples, which is expressed as

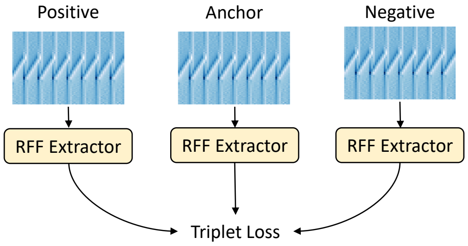


Fig. 7: Triplet loss.

$$Loss = \max(D(Anc, Pos) - D(Anc, Neg) + \alpha, 0) \quad (13)$$

where Anc, Pos, Neg denote features extracted from anchor, positive and negative samples, respectively. $D(\cdot,\cdot)$ denotes the Euclidean distance between two vectors. α is a predefined hyper-parameter which denotes the margin between positive and negative pairs. It is set to 0.1 in our implementation.

VI. Enrollment and Authentication

A. Enrollment

The legitimate devices are required to send several packets for enrollment before joining the system. These enrollment packets are first preprocessed and transformed to channel independent spectrograms. Then the RFF extractor is used to extract the RFF templates and store them in a database. The enrollment can be regarded as the training phase of a k-NN classifier, which simply memorizes all the training samples. Unlike deep learning, k-NN is not a data-hungry model. This is very desirable as it can significantly decrease the overhead of data collection. For example, we collect 100 packets from each device for enrollment, which is sufficient, as will be explained in Section VII.

B. Authentication

A complete authentication system should include two parts, namely rogue device detection and device classification. In our implementation, both of them are based on the k-NN algorithm with simple distance measures. The RFF extractor is trained with triplet loss which is defined by the Euclidean distance. The k-NN algorithm also relies on Euclidean distance measures thus their principles match well.

1) Rogue Device Detection: Rogue device detection is to determine whether the received packet is from an enrolled legitimate device, which is an anomaly detection problem. It is necessary before device classification, otherwise, the RFFI system will assign legitimate labels even to rogue devices.

The rogue device detection is implemented by the distance-based k-NN anomaly detection algorithm. The RFF of the received packet is extracted and the average distance to its K nearest neighbors in the RFF database is calculated as the detection score, which can be mathematically expressed as

$$D_{avg} = \frac{1}{K}\sum_{i=1}^{K} D_i, \tag{14}$$

where D_i is the Euclidean distance to the i-th neighbor. Then a predefined threshold λ is used to determine whether the received packet is from an enrolled device. When the detection score D_{avg} is above λ, the packet is considered to be sent from a rogue device and the authentication is failed. In contrast, the packet is considered to come from an enrolled device when D_{avg} is below the threshold λ and will be further classified. This can be formulated as

$$Decision = \begin{cases} \text{enrolled device}, when\ D_{avg} \leq \lambda \\ \text{rogue device}, when\ D_{avg} > \lambda \end{cases} \tag{15}$$

The detection threshold λ can be determined based on the application requirement. A higher λ leads to higher security, while a lower λ makes the RFFI system more user-friendly (access will not often be denied). When there is no special requirement, how to find the optimal λ will be introduced in Section VII-B.

2) Device Classification: Device classification is to infer the specific label of a transmitter from which the received packet is sent, which is a classification problem. It outputs a previously enrolled label according to the templates stored in the RFF database.

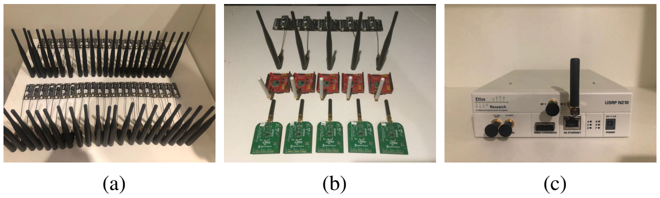
The proposed device classification system is implemented with the majority voting k-NN algorithm. The RFF of the received packet is first extracted and its K nearest neighbors are selected from the database according to the Euclidean distance. This packet is then assigned to the label that is most frequent among the K neighbors.

VII. Experimental Evaluation

A. Experimental Settings

1) DUT and Receiver: We employed 60 commercial off-the-shelf LoRa devices as devices under test (DUTs), and a USRP N210 software defined radio (SDR) platform as the receiver, as shown in Fig. 8. Detailed DUTs information can be found in Table II. The carrier frequency is 868.1 MHz and the transmission interval is set to 0.3 s. The receiver sampling rate is 1 MHz.

2) Training RFF Extractor: We first collect 500 packets from each of DUTs 1-30 in a residential room with LOS between the DUT and the receiver. The distance is about half a meter and DUT antennas are vertical to the ground.



(a) (b) (c)

Fig. 8: Experimental devices. (a) DUTs 1-45: LoPy4. (b) DUTs 46-60: mbed SX1261 shields, FiPy and Dragino SX1276 shields. (c) Receiver: USRP N210 SDR.

TABLE II: LoRa DUTs.

DUT index	Model	Chipset
1 - 45	Pycom LoPy4	SX1276
46 - 50	mbed SX1261 shield	SX1261
51 - 55	Pycom FiPy	SX1272
56 - 60	Dragino SX1276 shield	SX1276

TABLE III: Extractor Information.

	Extractor Input	Training DUTs	Data Augmentation
Extractor 1	Channel ind. spectrogram	DUT 1-30	Yes, multipath and Doppler
Extractor 2	Channel ind. spectrogram	DUT 1-20	Yes, multipath and Doppler
Extractor 3	Channel ind. spectrogram	DUT 1-10	Yes, multipath and Doppler
Extractor 4	Channel ind. spectrogram	DUT 1-30	No
Extractor 5	Spectrogram	DUT 1-30	No
Extractor 6	Channel ind. spectrogram	DUT 1-30	Yes, multipath only

We trained six RFF extractors with their detailed configuration given in Table III. Extractor 1 is trained as a baseline model involving channel independent spectrogram and data augmentation. Extractors 2-6 are trained for comparison to show the effects of different procedures during training. Extractor 1-3 are trained with the dataset augmented by both multipath and Doppler effects, while during the training of Extractor 6 only the multipath effect is emulated.

All the RFF extractors are trained with the same settings, including validation set ratio, optimizer, learning rate schedule, batch size and stop condition. 10% of the training data are randomly separated for validation. The model is optimized using RMSprop with an initial learning rate of 0.001. The learning rate drops every time the validation loss does not decrease for 10 epochs and the drop factor is 0.2. The batch size is set to 32. The training stops when validation loss does not decrease for 30 epochs. The model is implemented using Keras and trained with NVIDIA GeForce GTX 1660.

3) Enrollment and Authentication: We collect 100 packets from each DUT for enrollment. Unless otherwise specified (Section VII-H), the enrollment sets are collected in a residential room with LOS and vertical antenna placement.

We collect another 100 packets from each DUT for authentication. The experimental setup varies according to the tests. We use identification and classification sets to represent datasets for rogue device detection and device classification, respectively. The number of neighbors K is set to 15 for both rogue device detection and device classification.

The RFF extractors training only needs to be done once. Enrollment and authentication are carried out multiple times for evaluating system performance in various configurations. Table IV summarizes our experimental studies and their configurations.

B. Evaluation Metrics

1) Rogue Device Detection: Rogue device detection can be evaluated using the receiver operating characteristic (ROC) curve. As shown in (15), the result of rogue device detection is related to the value of threshold λ. This makes it unfair to compare rogue device detection performance since RFFI systems may set different threshold λ. ROC curve can be leveraged to overcome this, which reveals the trade-off between false-positive rate (FPR) and true-positive rate (TPR) at various threshold settings. More specifically, a pair of FPR and TPR are calculated at each threshold setting, and then we plot these pairs in the same figure to obtain the ROC curve. Two important metrics can be further calculated from the ROC curve, namely the area under the curve (AUC) and the equal error rate (EER). AUC refers to the area under the ROC curve, and the larger it is, the better the system performance. EER refers to the point on the ROC curve where FPR equals (1-TPR), and the smaller the better. EER can also help find the optimal threshold. The threshold λ that makes the system meet the EER point can be selected as the optimal value.

2) Classification: The metrics for device classification are the confusion matrix and overall accuracy that is defined as the correctly classified samples divided by the total number of test samples.

C. System Scalability

RFFI system should be scalable to allow device joining and leaving. Previous deep learning-based RFFI systems use a single classification neural network therefore the model output can only be device labels that are present during training. For example, if we train with the packets collected from DUTs 1-10, the model can only output a label between 1-10. Therefore, the deep learning model must be retrained when a new DUT joins the system, which is time-consuming and not practical.

In this paper, we train the deep learning model as an RFF extractor rather than for classification. The training of RFF extractor only needs to be done once. We then introduce an enrollment stage to obtain RFFs of any devices in the IoT network via the pre-trained RFF extractor. In the enrollment stage, we only need to train a k-NN classifier (store template RFFs), which can be done very quickly when new devices join. The RFF database can also be managed efficiently by only recording the RFF of devices that are active and present in the IoT network. In summary, our protocol has excellent scalability in terms of maintaining an up-to-date device list.

D. Generalization Ability for Rogue Device Detection

Rogue devices are out-of-library devices that are inaccessible during training. Whether the number of training DUTs affects the performance of rogue device detection should be evaluated. The hardest case for rogue device detection is the rogue device has similar hardware characteristics to the legitimate ones. Therefore, we specifically select DUT 31-40 as legitimate devices and DUT 41-45 as rogue ones, which are all LoPy4 devices. The identification dataset consists of 100 packets from each DUT 31-45, collected from the same residential room using the same setup.

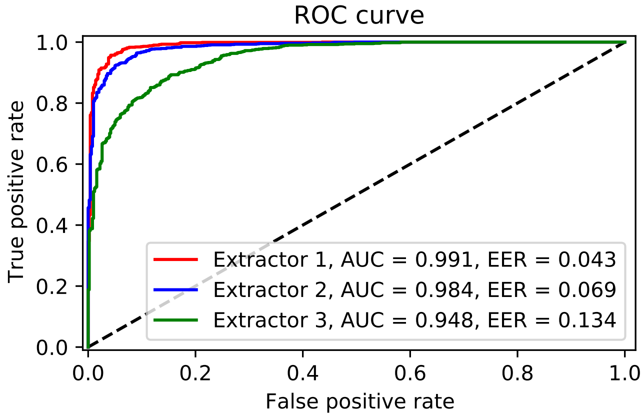


Fig. 9: ROC curve of rogue device detection.

The ROC curves are shown in Fig. 9. It can be observed that the AUC of Extractor 1 is 0.9905, indicating excellent detection performance. The AUC of Extractor 3 is the worst, which is 0.9479. This demonstrates the more DUTs involved in the training stage, the better the rogue device detection capability is.

E. Generalization Ability for Device Classification

To achieve good scalability, the RFF extractor must be able to extract RFFs from newly added devices that are out-of-library during the training stage. In other words, the RFF extractor should have an excellent generalization ability on previously unseen devices. The rule of thumb in deep metric learning is that the more training data, the better the generalization ability. We select Extractor 1, 2, 3 for comparison since they are trained with different numbers of DUTs.

We specifically select three groups of DUTs for evaluation, namely DUTs 1-10, DUTs 31-40 and DUTs 46-60.

- **Seen DUTs**: DUTs 1-10 are present during the training of Extractor 1, 2, 3 therefore they are used to evaluate the extractor performance on seen devices.
- **Unseen DUTs, same manufacturer**: DUTs 31-40 are disjoint with the training devices but with the same model (LoPy4). They can be used to validate the extractor performance on unseen devices.
- **Unseen DUTs, different manufacturers**: DUTs 46-60 are produced by other manufacturers, whose hardware characteristics are probably different from the training LoPy4 devices. This is the most challenging case as it requires a much higher generalization ability of the RFF extractor. This is also unavoidable in practice since involving devices from all the manufacturers during training is impossible.

TABLE IV: Summary of Experimental Evaluation.

Section	Extractor	Enrollment Set	Identification/Classification Set	Purpose
Section VII-D	1,2,3	Legitimate DUTs (residential room)	Legitimate and rogue DUTs (residential room)	Generalization ability of rogue device detection versus the number of training DUTs
Section VII-E	1,2,3	Seen, unseen, and different model DUTs (residential room)	Seen, unseen, and different model DUTs (residential room)	Generalization ability of device classification versus the number of training DUTs
Section VII-F	1,4,5	Data collected in stationary scenarios (residential room)	Ten datasets with different channel conditions (office building)	Evaluate data augmentation and channel independent spectrogram
Section VII-G	1,6	Data collected in stationary scenarios (residential room)	Emulated datasets with various moving speeds	Evaluate data augmentation (Doppler effect)
Section VII-H	1	Datasets with different antenna directions (office)	Datasets with different antenna directions (office)	Effect of antenna polarization

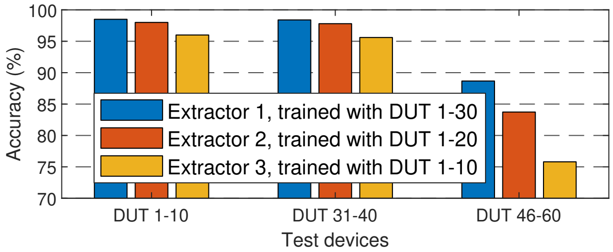


Fig. 10: Performance of RFF extractors.

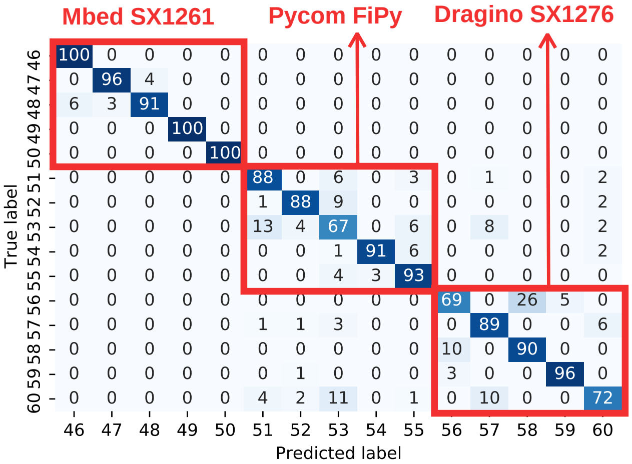
The classification datasets are collected from the same residential room with the same setup as the enrollment set.

The classification results on these three groups are shown in Fig. 10. It can be observed that Extractor 1, 2, 3 perform excellently on DUTs 1-10 and DUTs 31-40. The overall accuracies are always above 95%. The highest accuracy is reached by Extractor 1 with 98.50% on DUTs 1-10. The accuracy is 98.40% on classifying DUTs 31-40 which demonstrates the trained RFF extractor can efficiently extract RFFs from the devices that are not present during training.

We can see that there is a significant performance gap between Extractor 1, 2, 3 on DUT 46-60. Extractor 3, trained with only 10 DUTs, has the worst classification result, i.e., 75.80%. Training with 30 devices (Extractor 1) can increase it to 88.67% and the confusion matrix is shown in Fig. 11. It is found that the DUTs of the same type have similar hardware characteristics as almost all the misclassified packets fall into the three red boxes. The classification performance on DUT 46-50 is quite excellent but that on DUT 56-60 is not satisfying. This may because the hardware characteristics of DUT 56-60 (Dragino SX1276) are more different from the training DUTs (LoPy4). In summary, more devices should be included for training to achieve good generalization ability on out-of-library devices.

F. Effect of Data Augmentation and Channel Independent Spectrogram

The RFFI system should be robust to locations and channel variations. We mitigate the channel effect in the time-frequency domain and propose the channel independent spectrogram as the model input. Then we use data augmentation to further increase its robustness to the wireless channel.



	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
46	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	96	4	0	0	0	0	0	0	0	0	0	0	0	0
48	6	3	91	0	0	0	0	0	0	0	0	0	0	0	0
49	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	88	0	6	0	3	0	1	0	0	2
52	0	0	0	0	0	1	88	9	0	0	0	0	0	0	2
53	0	0	0	0	0	13	4	67	0	6	0	8	0	0	2
54	0	0	0	0	0	0	0	1	91	6	0	0	0	0	2
55	0	0	0	0	0	0	0	4	3	93	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	69	0	26	5	0
57	0	0	0	0	0	1	1	3	0	0	0	89	0	0	6
58	0	0	0	0	0	0	0	0	0	0	10	0	90	0	0
59	0	0	0	0	0	0	1	0	0	0	3	0	0	96	0
60	0	0	0	0	0	4	2	11	0	1	0	10	0	0	72

Fig. 11: Classification result on other LoRa types, overall accuracy 88.67%.

To verify that they are effective, we collect classification datasets in an office building whose environment is completely different from the enrollment residential room. The experiments are conducted in an office and a meeting room. The photos and floor plan of them can be found in Fig. 12. We collect 10 classification datasets, D1-D10, and they can be categorized into three scenarios, namely stationary, object moving and mobile scenarios. A summary of these datasets is given in Table V.

1) Stationary Scenario: Six datasets, D1-D6, are collected at Location A-F, respectively. During the data collection, DUTs and USRP N210 are stationary and there is no object moving around.

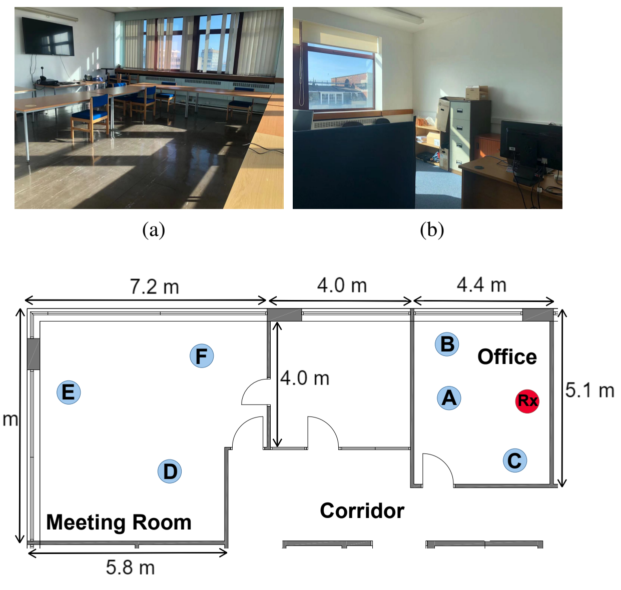
Locations A-F lead to multipath effects of varying severity. For instance, the waveform of DUT 6 at Location F (D6) is similar to that shown in Fig. 3b, showing a distinct sawtooth shape. While the waveform at Location A is almost the same as Fig. 3a, which is nearly flat. As discussed in Section III-A, the sawtooth is caused by the multipath effect therefore the channels at Locations A and F are different.

2) Object Moving Scenario: Two datasets, D7 & D8, are collected at location B and F, respectively. The DUTs and USRP are kept stationary while a person randomly walks around the room at a speed of 2 m/s.

In this scenario, the collected LoRa packets show different waveforms since the channel changes as a result of people

TABLE V: Summary of Classification Datasets Collected in an Office Room and a Meeting Room.

Evaluation Purpose	Data Augmentation and Channel Independent Spectrogram										Antenna Polarization	
Dataset	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
Location	A	B	C	D	E	F	B	F			B	F
Channel Effect	LOS	LOS	LOS	NLOS	NLOS	NLOS	LOS	NLOS	LOS	NLOS	LOS	NLOS
Movement	Stationary						Objective moving		Mobile		Stationary	
Antenna	Vertical										Parallel	


(a)

(b)

(c)

Fig. 12: Experimental environments. (a) Meeting room. (b) Office. (c) Floor plan.

walking. However, the Doppler effect is not quite serious due to the low walking speed.

3) Mobile Scenario: Two datasets, D9 & D10, are collected in the office and meeting room, respectively. The USRP is kept stationary and a person takes the DUTs walking around at a speed of 2 m/s.

Both multipath and Doppler effects are serious in this scenario since the sawtooth shapes and amplitude variations can be frequently observed. Many packets show waveforms similar to those shown in Fig. 3c and Fig. 3d.

4) Discussion: Extractor 1, 4, 5 are selected for comparison to demonstrate the effectiveness of channel independent spectrogram and data augmentation. Their training details can be found in Table III and the classification results are summarized in Fig. 13.

It can be observed that Extractor 1 performs well on all datasets, which indicates the proposed RFFI system is robust to locations and channel variations. It performs slightly worse on D9 and D10 that are collected in mobile scenarios (lower than 90%). This is possibly due to the inevitable shaking of the DUT antenna during moving, which will result in the change of antenna polarization. The effect of antenna polarization is discussed in Section VII-H.

Compared with Extractor 1, the training of Extractor 4 does not employ data augmentation. As can be seen, its performance on D5 is significantly worse with only 78.80% accuracy. D5 is collected at Location E which is the farthest from the receiver. This indicates the channel independent spectrogram is affected by the noise in lower SNR scenarios. Data augmentation must be used to further improve its robustness.

As for Extractor 5, it employs neither channel independent spectrogram nor the data augmentation. It can be observed that Extractor 5 only performs well on D1 which is a short-distance LOS stationary scenario. It is presumed that the channel condition at Location A is similar to the enrollment residential room. Once the channel is changed, the classification performance will degrade significantly. The worst results occur on D9 and D10 (mobile scenario) whose accuracies are 55.90% and 49.40%, respectively, which are 30% lower than Extractor 1.

We also try to explore whether data augmentation can solve the channel problem without the employment of channel independent feature. However, we found out the deep learning model cannot converge as the training loss does not decrease at all. In other words, the RFF extractor is not capable to extract channel independent RFFs from the spectrogram when training data is distorted by wireless channels. In contrast, when we use channel independent spectrogram instead of the spectrogram as model input (Extractor 1), the channel effects are mitigated and device-specific features can be learned so that the model is trained successfully.

G. Effect of Doppler Shift in Data Augmentation

Data augmentation must be considered to improve the classification performance in high-speed scenarios (serious Doppler effects). Due to the experimental constraints, we emulate the communication scenarios at various moving speeds by filtering the real collected D1 to a channel simulator to generate classification sets. The process is almost the same with data augmentation introduced in Section V-A. The maximum Doppler frequency, f_d, is fixed to 0, 10, 30, 50, 100 Hz, respectively. These Doppler frequencies are equivalent to moving speeds of 0, 12.46, 37.33, 62.21 and 124.4 km/h for a LoRa system operating at 868 MHz. We select Extractor 1 and Extractor 6 for comparison to demonstrate the Doppler effect must be considered during data augmentation.

The results are summarized in Fig. 14. Compared with Extractor 1, Extractor 6 does not consider the Doppler effect during data augmentation. Its maximum Doppler frequency f_d is fixed to 0 Hz. It can be seen that both these two extractors

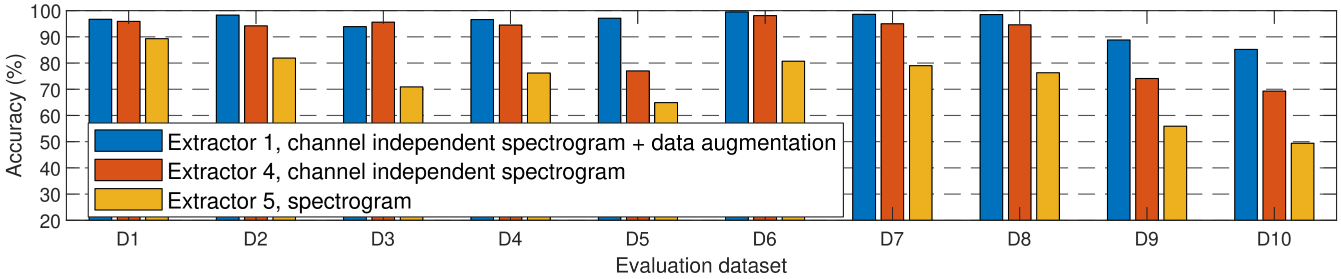


Fig. 13: Classification results in various channel conditions.

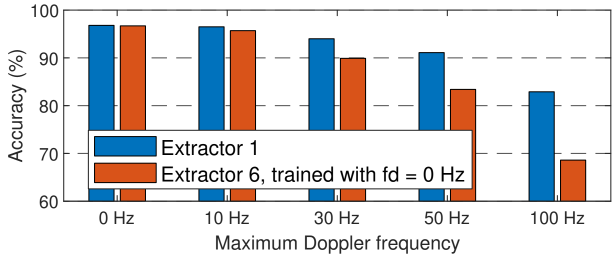


Fig. 14: Classification results under various Doppler frequencies of the identification sets.

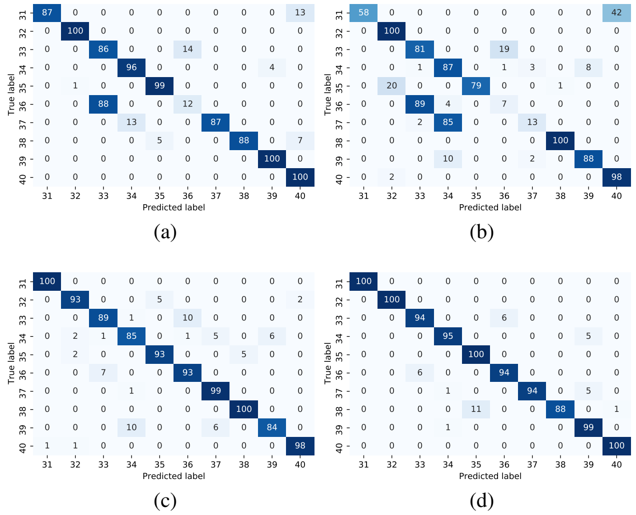
achieved a good accuracy, i.e., around 97.50%, in stationary scenarios (f_d = 0 Hz).

However, when the Doppler frequency f_d increases, the performance gap between Extractors 1 and 6 becomes larger. In the scenario of f_d = 100 Hz, the accuracy of Extractor 6 is only 68.60%, probably because the channel independent spectrogram cannot perfectly eliminate channel effects in high-speed scenarios. As given in (8), $|\mathbf{H}_m| \approx |\mathbf{H}_{m+1}|$ holds when the Doppler effect is not severe. Involving Doppler effect during data augmentation can mitigate this. As shown in Fig. 14, Extractor 1 can boost the accuracy to 80% in the f_d = 100 Hz scenario. However, there is still a 20% accuracy drop compared to low-mobility scenarios (e.g., $f_d = 10$ Hz), indicating that channel effect is not fully eliminated. Therefore, algorithm enhancement is required for high-speed scenarios.

H. Effect of Antenna Polarization

Antenna polarization refers to the direction of the electric field produced by an antenna. It was found to affect the transient-based RFFI system [45]. To the best knowledge of the authors, there is no study on antenna polarization in deep learning-based approaches that use the non-transient signal for classification. To explore this, we additionally collect two datasets, D11 & D12, at Location B and Location F, respectively. The DUT antenna is parallel to the ground and points towards the USRP. Both DUTs and USRP are kept stationary. Extractor 1 is used to extract RFFs.

Dual polarization refers to the antennas of transmitter and receiver have the orthogonal polarization directions. We use D2 and D11 for enrollment and classification sets, respectively, which are collected at Location B and the only difference is the antenna direction. The classification result is shown in Fig. 15a. It can be seen that nearly all the packets from DUT 36



	31	32	33	34	35	36	37	38	39	40
31	87	0	0	0	0	0	0	0	0	13
32	0	100	0	0	0	0	0	0	0	0
33	0	0	86	0	0	14	0	0	0	0
34	0	0	0	96	0	0	0	0	4	0
35	0	1	0	0	99	0	0	0	0	0
36	0	0	88	0	0	12	0	0	0	0
37	0	0	0	13	0	0	87	0	0	0
38	0	0	0	0	5	0	0	88	0	7
39	0	0	0	0	0	0	0	0	100	0
40	0	0	0	0	0	0	0	0	0	100

(a)

	31	32	33	34	35	36	37	38	39	40
31	58	0	0	0	0	0	0	0	0	42
32	0	100	0	0	0	0	0	0	0	0
33	0	0	81	0	0	19	0	0	0	0
34	0	0	1	87	0	1	3	0	8	0
35	0	20	0	0	79	0	0	1	0	0
36	0	0	89	4	0	7	0	0	0	0
37	0	0	2	85	0	0	13	0	0	0
38	0	0	0	0	0	0	0	100	0	0
39	0	0	0	10	0	0	2	0	88	0
40	0	2	0	0	0	0	0	0	0	98

(b)

	31	32	33	34	35	36	37	38	39	40
31	100	0	0	0	0	0	0	0	0	0
32	0	93	0	0	5	0	0	0	0	2
33	0	0	89	1	0	10	0	0	0	0
34	0	2	1	85	0	1	5	0	6	0
35	0	2	0	0	93	0	0	5	0	0
36	0	0	7	0	0	93	0	0	0	0
37	0	0	0	1	0	0	99	0	0	0
38	0	0	0	0	0	0	0	100	0	0
39	0	0	0	10	0	0	6	0	84	0
40	1	1	0	0	0	0	0	0	0	98

(c)

	31	32	33	34	35	36	37	38	39	40
31	100	0	0	0	0	0	0	0	0	0
32	0	100	0	0	0	0	0	0	0	0
33	0	0	94	0	0	6	0	0	0	0
34	0	0	0	95	0	0	0	0	5	0
35	0	0	0	0	100	0	0	0	0	0
36	0	0	6	0	0	94	0	0	0	0
37	0	0	0	1	0	0	94	0	5	0
38	0	0	0	0	11	0	0	88	0	1
39	0	0	0	1	0	0	0	0	99	0
40	0	0	0	0	0	0	0	0	0	100

(d)

Fig. 15: Classification results of antenna polarization. (a) Enrollment set D2, classification set D11, overall accuracy 85.50%. (b) Enrollment set D6, classification set D12, overall accuracy 71.10%. (c) Enrollment set D11, classification set D12, overall accuracy 93.40%. (d) Enrollment set D12, classification set D11, overall accuracy 96.40%.

are misclassified as DUT 33. A similar result is obtained when D6 is used for enrollment and D12 for classification, both of which are collected at Location F.

Linear polarization refers to the antennas of transmitter and receiver have the same polarization direction. We use D11 and D12 for enrollment and classification, respectively, which are collected at different locations but with the same antenna direction. The results are given in Fig. 15c and Fig. 15d. Their accuracies are both above 90% and there are no seriously misclassified DUTs.

The above results show the classification is not affected by the device location but the antenna polarization. We infer this may be the reason for the slightly lower accuracies of the mobile scenarios (D9 and D10) in Section VII-F. There will probably be inevitable shaking of the antenna during moving, which results in the change of antenna polarization.

VIII. Related work

Deep learning-based RFFI eliminates the need for feature engineering and usually leads to better performance. However, deep learning-based RFFI still has some shortcomings, including the lack of rogue device detection capability and system scalability, as well as sensitivity to the variations of wireless channel.

Many previous deep learning-based RFFI schemes are designed for close-set problems, making them lack system scalability and rogue device detection capability [28], [36], [46], [47]. More specifically, a close-set RFFI system neither supports efficient device joining and leaving nor can it distinguish rogue devices from legitimate ones. This is because previous methods usually rely on the softmax layer for classification, and once the training is completed, the number of neurons in this layer cannot be changed. When a new device joins or an old device leaves, the neural network should be updated by retraining, which is not practical and time-consuming. Chen *et al.* use transfer learning to speed up the retraining process, but it still requires the RFFI system to have an GPU [48]. Furthermore, the softmax layer-based systems can only output the label present in the training set, but rogue devices are never available during training. In this case, the rogue devices will be classified as one of the legitimate devices which is unacceptable.

To overcome these limitations, researchers start to consider RFFI as an open-set problem rather than a close-set one. Xie *et al.* propose a similarity-based RFFI system, which is the most relevant work to ours [49]. They trained an RFF extractor using the softmax-based loss. Then the cosine similarity and a threshold are leveraged for the back-end authentication task. The similarity-based open-set solution supports efficient device joining without high complexity. Hanna *et al.* evaluate a number of open-set classifiers, including autoencoder, OpenMax, One Vs All (OvA), etc [50]. They conclude that the OvA classifier can reach the best performance. Gritsenko *et al.* propose a novel device detection scheme that leverages the probabilistic characteristics of classification neural networks [51]. The device is labelled as a new one when the confidence level during classification is low. Soltani *et al.* further extend this scheme to the case of multiple classifiers [25]. The generative adversarial network (GAN) is also used for open-set RFFI problem [17]. Before feeding the signal to a typical classification neural network, a GAN is additionally introduced to determine whether the device is a rogue or not.

Last but not least, deep learning-based RFFI is not robust to wireless channels [16]. Sankhe *et al.* propose the ORACLE system to combat channel effects [47], [52]. However, it needs to intentionally introduce impairments to the transmitter, which is costly and not suitable for IoT applications. Morin *et al.* indicated that the training dataset should contain as many channel conditions as possible to make the neural network automatically learn how to mitigate it [34], [35]. However, this may dramatically increase the cost of collecting the training set. Data augmentation is an effective alternative to this approach, in which we can collect the training set in a static scenario and pass it into a well-designed channel simulator to generate signals under various channel conditions [18], [32], [33], [38], [39]. However, designing an accurate channel simulator that matches the real application scenarios is challenging. This paper for the first time involves Doppler shift to emulate the real channel and evaluate its performance.

IX. Conclusion

In this paper, we propose a scalable and channel-robust RFFI framework that exploits the device-intrinsic hardware impairments for device authentication. Specifically, we leverage the deep metric learning to train an RFF extractor that has excellent generalization ability. When a new device joins the system, it only needs to send several packets for enrollment and the RFF extractor does not need to be retrained. The k-NN algorithm is used for rogue device detection and device classification. To overcome the channel effect, we design the channel independent spectrogram and further use data augmentation to improve the system robustness to channel variations. We conduct extensive experiments using 60 commercial off-the-shelf LoRa devices. We demonstrate our framework has an excellent generalization performance for both device classification and rogue device detection. The channel independent spectrogram and data augmentation are shown to be effective under extensive tests with various channel conditions. We also find the antenna polarization affects the classification performance. This paper only uses the amplitude of the channel independent spectrogram, and leveraging phase information can be a promising future work.

References

[1] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, “A survey on security and privacy issues in Internet-of-Things,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, 2017.

[2] J. Zhang, G. Li, A. Marshall, A. Hu, and L. Hanzo, “A new frontier for IoT security emerging from three decades of key generation relying on wireless channels,” *IEEE Access*, vol. 8, pp. 138 406–138 446, 2020.

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

[4] Y. Shi and M. A. Jensen, “Improved radiometric identification of wireless devices using MIMO transmission,” *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1346–1354, 2011.

[5] 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.

[6] K. Joo, W. Choi, and D. H. Lee, “Hold the door! fingerprinting your car key to prevent keyless entry car theft,” in *Proc. Netw. Distrib. Syst. Security Symposium (NDSS)*, Virtual Conference, Feb. 2020.

[7] 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.

[8] A. C. Polak and D. L. Goeckel, “Wireless device identification based on RF oscillator imperfections,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2492–2501, 2015.

[9] 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, 2018.

[10] 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.

[11] B. Danev, T. S. Heydt-Benjamin, and S. Capkun, “Physical-layer identification of RFID devices.” in *Proc. USENIX Security Symposium*, 2009, pp. 199–214.

[12] 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.

[13] S. Balakrishnan, S. Gupta, A. Bhuyan, P. Wang, D. Koutsonikolas, and Z. Sun, “Physical layer identification based on spatial–temporal beam features for millimeter-wave wireless networks,” *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1831–1845, 2019.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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 Security Privacy (EuroS&P)*, Virtual Conference, 2020, pp. 322–338.

[20] 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.

[21] 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)*, 2017, pp. 58–63.

[22] 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)*, 2018, pp. 1–6.

[23] G. Reus-Muns and K. Chowdhury, “Classifying UAVs with proprietary waveforms via preamble feature extraction and federated learning,” *IEEE Trans. Veh. Technol.*, 2021.

[24] T. Jian, Y. Gong, Z. Zhan, R. Shi, N. Soltani, Z. Wang, J. G. Dy, K. R. Chowdhury, Y. Wang, and S. Ioannidis, “Radio frequency fingerprinting on the edge,” *IEEE Trans. Mobile Comput.*, 2021.

[25] N. Soltani, G. Reus-Muns, B. Salehihikouei, J. Dy, S. Ioannidis, and K. Chowdhury, “RF fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15 518–15 531, 2020.

[26] 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, 2019.

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

[28] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, “Radio frequency fingerprint identification for narrowband systems, modelling and classification,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3974 – 3987, 2021.

[29] Y. Qian, J. Qi, X. Kuai, G. Han, H. Sun, and S. Hong, “Specific emitter identification based on multi-level sparse representation in automatic identification system,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2872–2884, 2021.

[30] 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.

[31] S. Rajendran, Z. Sun, F. Lin, and K. Ren, “Injecting reliable radio frequency fingerprints using metasurface for the Internet of Things,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1896–1911, 2020.

[32] A. Al-Shawabka, P. Pietraski, S. B Pattar, F. Restuccia, and T. Melodia, “Deeplora: Fingerprinting LoRa devices at scale through deep learning and data augmentation,” in *Proc. ACM Int. Symposium Mob. Ad Hoc Netw. Comput. (MobiHoc)*, Shanghai, China, Jul. 2021.

[33] M. Piva, G. Maselli, and F. Restuccia, “The tags are alright: Robust large-scale rfid clone detection through federated data-augmented radio fingerprinting,” in *Proc. ACM Int. Symposium Mob. Ad Hoc Netw. Comput. (MobiHoc)*, Shanghai, China, Jul. 2021.

[34] C. Morin, L. S. Cardoso, J. Hoydis, J.-M. Gorce, and T. Vial, “Transmitter classification with supervised deep learning,” in *Proc. Int. Conf. Cogn. Radio Oriented Wireless Netw.* Springer, 2019, pp. 73–86.

[35] C. Morin, L. Cardoso, J. Hoydis, and J.-M. Gorce, “Deep Learning-based Transmitter identification on the physical layer,” Dec. 2020, working paper or preprint. [Online]. Available: https://hal.inria.fr/hal-03117090

[36] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, “Radio frequency fingerprint identification for LoRa using deep learning,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2604–2616, 2021.

[37] S. D. Andrews, “Extensions to radio frequency fingerprinting,” Ph.D. dissertation, Virginia Tech, 2019.

[38] N. Soltani, K. Sankhe, J. Dy, S. Ioannidis, and K. Chowdhury, “More is better: Data augmentation for channel-resilient RF fingerprinting,” *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 66–72, 2020.

[39] K. Merchant and B. Nousain, “Enhanced RF fingerprinting for IoT devices with recurrent neural networks,” in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2019, pp. 590–597.

[40] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[41] R. Ranjan, C. D. Castillo, and R. Chellappa, “L2-constrained softmax loss for discriminative face verification,” *arXiv preprint arXiv:1703.09507*, 2017.

[42] K. Musgrave, S. Belongie, and S.-N. Lim, “A metric learning reality check,” in *Proc. European Conf. Computer Vision (ECCV)*, 2020, pp. 681–699.

[43] K. Kobs, M. Steininger, A. Dulny, and A. Hotho, “Do different deep metric learning losses lead to similar learned features?” in *Proc. Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 10 644–10 654.

[44] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[45] B. Danev and S. Capkun, “Transient-based identification of wireless sensor nodes,” in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, NW Washington, DC, USA, 2009, pp. 25–36.

[46] 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.

[47] 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.

[48] S. Chen, S. Zheng, L. Yang, and X. Yang, “Deep learning for large-scale real-world ACARS and ADS-B radio signal classification,” *IEEE Access*, vol. 7, pp. 89 256–89 264, 2019.

[49] R. Xie, W. Xu, Y. Chen, J. Yu, A. Hu, D. W. K. Ng, and A. L. Swindlehurst, “A generalizable model-and-data driven approach for open-set RFF authentication,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4435–4450, 2021.

[50] S. Hanna, S. Karunaratne, and D. Cabric, “Open set wireless transmitter authorization: Deep learning approaches and dataset considerations,” *IEEE Trans. on Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 59–72, 2020.

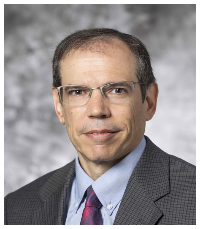
[51] A. Gritsenko, Z. Wang, T. Jian, J. Dy, K. Chowdhury, and S. Ioannidis, “Finding a ‘new’ needle in the haystack: Unseen radio detection in large populations using deep learning,” in *Proc. IEEE Int. Symposium Dynamic Spectr. Access Netw. (DySPAN)*, Newark, NJ, USA, 2019, pp. 1–10.

[52] 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, 2019, pp. 370–378.

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.

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, radio frequency fingerprint identification, and WiFi sensing. He is the receipt of the UK EPSRC New Investigator Award.

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.



Joseph R. Cavallaro (S'78, M'82, SM'05, F'15) received the B.S. from the University of Pennsylvania, Philadelphia, Pa, in 1981, the M.S. from Princeton University, Princeton, NJ, in 1982, and the Ph.D. from Cornell University, Ithaca, NY, in 1988, all in electrical engineering. He is an IEEE Fellow. From 1981 to 1983, he was with AT&T Bell Laboratories, Holmdel, NJ. In 1988, he joined the faculty of Rice University, Houston, TX, where he is currently a Professor of Electrical and Computer Engineering and Associate Chair. His research interests include computer arithmetic, and DSP, GPU, FPGA, and VLSI architectures for applications in wireless communications. During the 1996–1997 academic year, he served at the US National Science Foundation as Director of the Prototyping Tools and Methodology Program. He was a Nokia Foundation Fellow and a Visiting Professor at the University of Oulu, Finland in 2005. He is currently the Director of the Center for Multimedia Communication at Rice University. He is an advisory board member of the IEEE SPS TC on Applied Signal Processing Systems and past Chair of the IEEE CAS TC on Circuits and Systems for Communications. He was an Associate Editor of the IEEE Transactions on Signal Processing and the IEEE Signal Processing Letters, and currently serves as an AE for the Journal of Signal Processing Systems and an SE for the IEEE Journal of Emerging and Selected Topics in Circuits and Systems. He was General/Program Co-chair of the 2003, 2004, and 2011 IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP), and General/Program Co-chair for the 2012, 2014 ACM/IEEE GLSVLSI conferences. He was TPC Co-Chair in 2016 and General Co-Chair in 2020 and 2021 of the IEEE SiPS workshops. He was TPC Chair in 2017 and General Chair in 2020 of the IEEE Asilomar Conference on Signals, Systems, and Computers. He served on the IEEE CAS Society Board of Governors during 2014.