Security and Safety, Vol. 3, 2023019 (2024)

Security and Safety in Physical Layer Systems

Research Article

Open Access

Information Network

Exploration of transferable deep learning-aided radio frequency fingerprint identification systems

Guanxiong Shen and Junqing Zhang*

Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, UK

Received: 30 April 2023 / Revised: 12 June 2023 / Accepted: 28 June 2023 / Published online: 28 September 2023

Abstract Radio frequency fingerprint identification (RFFI) shows great potential as a means for authenticating wireless devices. As RFFI can be addressed as a classification problem, deep learning techniques are widely utilized in modern RFFI systems for their outstanding performance. RFFI is suitable for securing the legacy existing Internet of Things (IoT) networks since it does not require any modifications to the existing end-node hardware and communication protocols. However, most deep learning-based RFFI systems require the collection of a great number of labelled signals for training, which is time-consuming and not ideal, especially for the IoT end nodes that are already deployed and configured with long transmission intervals. Moreover, the long time required to train a neural network from scratch also limits rapid deployment on legacy IoT networks. To address the above issues, two transferable RFFI protocols are proposed in this paper leveraging the concept of transfer learning. More specifically, they rely on fine-tuning and distance metric learning, respectively, and only require only a small amount of signals from the legacy IoT network. As the dataset used for transfer is small, we propose to apply augmentation in the transfer process to generate more training signals to improve performance. A LoRa-RFFI testbed consisting of 40 commercial-off-the-shelf (COTS) LoRa IoT devices and a software-defined radio (SDR) receiver is built to experimentally evaluate the proposed approaches. The experimental results demonstrate that both the fine-tuning and distance metric learning-based RFFI approaches can be rapidly transferred to another IoT network with less than ten signals from each LoRa device. The classification accuracy is over 90%, and the augmentation technique can improve the accuracy by up to 20%.

Keywords Device authentication, internet of things, LoRa, radio frequency fingerprint identification, deep learning, wireless security

Citation Shen G and Zhang J Exploration of transferable deep learning-aided radio frequency fingerprint identification systems. Security and Safety 2024; **3**: 2023019. https://doi.org/10.1051/sands/2023019

1 Introduction

In recent years, there has been a significant increase in the quantity of connected Internet of Things (IoT) devices, which is predicted to reach 29.42 billion by 2030 [1]. As IoT technology is becoming increasingly integrated into both industrial manufacturing and our daily activities, its security has also attracted significant attention. Authentication is a critical aspect of ensuring secure wireless connectivity of IoT devices, which prevents adversaries from gaining access to the network. Current IoT authentication solutions heavily rely on cryptographic algorithms, which require keys to be securely distributed among IoT devices. Although these cryptographic solutions theoretically offer robust and strong security, they

can be entirely compromised once the keys are leaked. As ensuring absolute security of key distribution and storage remains challenging, alternative methods for IoT authentication that do not rely on cryptographic schemes are essentially required.

Radio frequency fingerprint identification (RFFI) is an emerging technique for authenticating wireless devices. In a nutshell, an RFFI system is equipped at the receiver and identifies IoT transmitters by analyzing the physical characteristics of their emitted radio signals. These radio signals are generated by the analog front end (AFE) of IoT devices. The AFE consists of numerous inexpensive analog components, which unavoidably deviate from nominal values during the manufacturing process. The slight deviation is within the specification which does not impact the communication functions such as demodulation but its distortion to the emitted radio signal can be extracted using carefully designed algorithms. Such distortion is unique to the AFE of wireless devices, therefore it can serve as an identifier for authentication.

In recent years, the RFFI technique is gradually moving from the traditional approaches to the deep learning (DL) era. Traditional RFFI approaches focus on feature engineering. More specifically, wireless communication experts design various algorithms to manually extract features from the received signals, and then use a classic machine learning model to process them. Common handcrafted features for RFFI include frequency offset [2–7], IQ imbalance [5, 8], power amplifier characteristics [9, 10], beam patterns [11, 12], and statistics features such as kurtosis [3]. Although the handcrafted features are discriminative and effective in identification, one needs to re-design the feature extraction algorithm once the communication protocol has changed. The algorithm designed for a specific wireless technology may not suit others. Moreover, the performance of traditional RFFI systems heavily depends on the quality of the designed handcrafted features. Some information useful for identification may be eliminated by the feature extraction algorithm, thus restricting the identification accuracy. With the fast development of DL, it has been widely utilized in RFFI for its remarkable performance. In contrast to the traditional approaches that rely on handcrafted features, DL-RFFI systems automatically learn feature extraction algorithms from a large number of data. The learned feature extraction algorithms often outperform those designed manually, leading to enhanced performance. In a DL-RFFI system, a neural network (NN) is deployed at the receiver, with input being the captured radio signals and output being the predicted device identity. Various types of NNs have been leveraged to build RFFI systems, including convolutional neural networks (CNNs) [13–34], long short-term memory (LSTM) networks [16, 20, 27, 35, 36], gated recurrent unit (GRU) networks [20, 37], generative adversarial networks (GANs) [20, 38–40], and transformers [37, 41, 42]. They are demonstrated to be particularly effective in improving the identification performance of RFFI systems.

The RFFI technique is intuitively suitable for protecting legacy IoT networks that are already in operation but the training cost of NNs can become a bottleneck. The RFFI system is entirely deployed at the receiver side and no modifications to the transmitter hardware or communication protocols are required. This means that training and implementing the NN only on the receiver can effectively secure the legacy IoT network. Although the above analysis holds true, the training cost of NNs remains a major obstacle. When deploying the RFFI system to a legacy IoT network, we need to first collect a significant amount of labelled signals from the operating IoT end nodes and use them to train a classification NN. This procedure can be time-consuming for two reasons. Firstly, the end nodes in the IoT network can be configured with a very long transmission interval to save energy. As a result, a considerable amount of time is required to collect sufficient training signals from all the end nodes to train a classification NN with satisfactory performance. Secondly, even if the training signals are successfully collected, the time for training the classification NN is excessively long and cannot be quickly deployed, especially when the number of end nodes is large in the IoT network. For the aforementioned reasons, current DL-RFFI protocols lack the capability to quickly adapt to legacy existing IoT networks, and a transferable RFFI protocol is needed.

In this paper, two transferable RFFI protocols are proposed, namely fine-tuning-based and distance metric learning-based approaches. After an in-lab pre-training procedure, both protocols can be quickly deployed on a legacy existing IoT network without the need for collecting a large number of training signals. The proposed transferable RFFI protocols are evaluated with 40 commercial-off-the-shelf (COTS) LoRa IoT devices and a USRP N210 software-defined radio (SDR) receiver. Our detailed contributions are listed as follows:

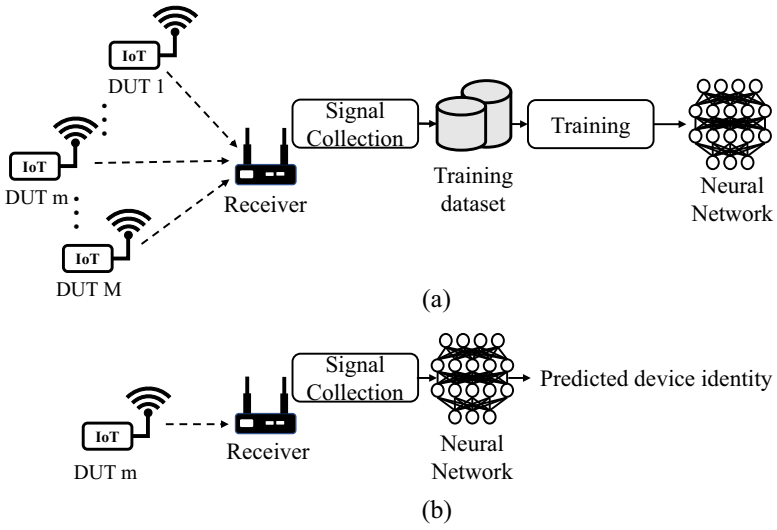

Figure 1. Overview of a conventional DL-based RFFI system. (a) Training stage. (b) Inference stage.

- We designed two transferable RFFI approaches based on fine-tuning and distance metric learning techniques, respectively. Both approaches can be rapidly deployed on legacy existing IoT networks without the need to collect large amounts of labelled signals and time-consuming training. Firstly, an NN-based feature extractor is pre-trained, and then it can be used for transfer learning with only a few labelled signals. For the fine-tuning-based approach, we train a new classifier to identify the end nodes operating in the legacy IoT network; for the metric learning-based approach, we use the k-nearest neighbour (kNN) algorithm to implement transfer learning. The experimental results show that the transferred RFFI systems can achieve over 90% classification accuracy at six different locations, and only less than five signals are needed to reach a satisfactory transfer performance.
- We propose to perform augmentation to expand the dataset collected from the legacy IoT network. Due to the fact that end nodes in existing legacy IoT networks may be configured with long transmission intervals, the dataset collected for transfer is often constrained in size and thus reducing the performance of the transferred RFFI systems. To mitigate this, artificial noise can be added to expand the collected dataset. The experimental results demonstrate that the augmentation technique can effectively improve classification accuracy by up to 20%.

The rest of the paper is organized as follows. Section 2 introduces the background of RFFI and states the problem targeted in this paper. Section 3 details the proposed two RFFI protocols, namely fine-tuning and distance metric learning-based approaches. Section 4 presents the experimental results with a LoRa-RFFI case study. Section 5 finally concludes the paper.

2 Background and problem statement

This section first introduces the technical background of DL-RFFI systems. After that, the problem that this paper aims to solve is formulated and presented.

2.1 RFFI primer

The overview of a DL-aided RFFI system is shown in Figure 1. There are M legitimate devices under test (DUTs) operating in an IoT network and a receiver capturing wireless signals from which infers the index of the transmitter. Rogue and unauthorized devices are not considered in this paper.

As illustrated in the figure, a DL-aided RFFI system consists of two successive stages: training and inference. To train a classification NN, a significant amount of labelled wireless signals need to be collected from all DUTs, *i.e.*, training category set $C^{\text{train}} = \{\text{DUT } 1, \ldots, \text{DUT } m, \ldots, \text{DUT } M\}$, and form

a training dataset $\mathcal{X}^{\text{train}}$, given as

$$\mathcal{X}^{\text{train}} = \{(\mathbf{r}_i, \mathbf{y}_i)\}_{i=1}^{I_{\text{train}}}, \mathbf{y}_i \in C^{\text{train}}, \tag{1}$$

where $\mathbf{r}_i$ denotes the ith captured wireless signal and $\mathbf{y}_i$ is the corresponding one-hot encoded label. I_{train} is the total number of training samples in $\mathcal{X}^{\text{train}}$. With the dataset consisting of a large number of labelled signals, a classification NN $c(\cdot, \theta)$ can be trained. The training process can be expressed as an optimization problem that seeks to find the parameter set θ that minimizes the loss function $\mathcal{L}(\cdot, \cdot)$, given as

$$\theta = \underset{\theta}{\text{argmin}} \sum_{(\mathbf{r},\mathbf{y})\in\mathcal{X}^{\text{train}}} \mathcal{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right), \tag{2}$$

where $\hat{\mathbf{y}} = \{\hat{y}_1, \ldots, \hat{y}_m, \ldots, \hat{y}_M\}$ is the prediction output by the classification NN $c(\cdot; \theta)$, and $\hat{y}_m$ can be interpreted as the probability that the signal is sent from DUT m. To reach the training goal, iterative optimization algorithms such as stochastic gradient descent (SGD) and adaptive moment estimation (ADAM) are often used. The loss function $\mathcal{L}(\cdot, \cdot)$ is often defined as cross-entropy, given as

$$\mathcal{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right) = -\sum_{m=1}^{M} y_m \log\left(\hat{y}_m\right). \tag{3}$$

In the inference stage shown in Figure 1b, the classification NN can predict the device identity by analyzing the captured signal. The inference stage is mathematically given as

$$\hat{\mathbf{y}} = c\left(\mathbf{r}; \theta\right). \tag{4}$$

2.2 Problem statement

While the above two-stage method is popular in the literature, collecting sufficient packets for training is time-consuming. In the existing works, data collection was carried out offline in a lab setting, where the device's configuration (*e.g.*, transmission interval) and environment conditions (*e.g.*, multipath levels) can be well controlled. In practice, when the DL-based RFFI is to be applied to a legacy IoT network where the IoT devices have already been deployed in real environments, it is extremely challenging to collect sufficient packets with good quality, *e.g.*, high signal-to-noise ratio (SNR).

Our goal is to design an RFFI protocol that enables the NN trained with the data collected from a number of training DUTs, *i.e.*, training category set C^{train}, can be rapidly transferred to a legacy IoT network, *i.e.*, legacy category set C^{legacy}, without the requirement for collecting numerous labelled signals and time-consuming retraining. Note that the category set C^{legacy} for a legacy existing IoT network is assumed to be different from the training category set, given as

$$C^{\text{legacy}} = \{\text{DUT } 1, \ldots, \text{DUT } n, \ldots, \text{DUT } N\} \neq C^{\text{train}}, \tag{5}$$

where DUTs $1 - N$ are the end nodes operating in the existing legacy IoT network.

3 Transferable RFFI protocols

In this section, two transferable RFFI protocols are proposed, namely the fine-tuning and distance metric learning-aided protocols. Both protocols require pre-training a classification NN, which is thus first described. After that, the fine-tuning and distance metric learning-aided approaches are introduced separately.

3.1 Pre-training

Both of the proposed transferable RFFI protocols require pre-training of a classification NN, whose procedure is shown in Figure 2 and detailed in this subsection.

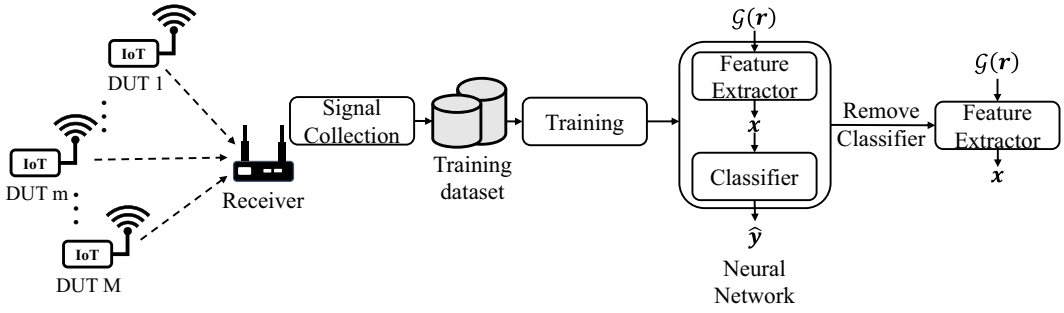

Figure 2. Pre-train a classification NN and remove the classifier.

3.1.1 Signal collection

The receiver performs signal collection algorithms to capture wireless signals over the air. To meet the requirements of RFFI systems, signal collection often includes synchronization, frequency offset compensation, power normalization, and preamble extraction. The algorithm design depends on the communication protocol used by the IoT network. Interested readers can refer to [16] for a signal collection implementation example for LoRa-RFFI systems. The captured baseband signals $\mathbf{r}$ are in the format of IQ samples, *i.e.*, a vector consisting of complex numbers, and are stored in the training dataset $\mathcal{X}^{\text{train}}$.

3.1.2 Signal representation

The IQ samples are often converted to other appropriate signal representations before being fed into the NN. This is because using the IQ samples $\mathbf{r}$ as input to the NN often cannot lead to a satisfactory identification performance due to factors like channel effects. The design of signal representation demands expertise in wireless communication knowledge, such as the channel-independent spectrogram [19] and differential constellation trace figure (DCTF) [7], etc. The symbol $\mathcal{G}(\cdot)$ represents the signal representation conversion.

3.1.3 Neural network architecture

As shown in Figure 2, the classification NN used for pre-training consists of two main components: a feature extractor and a classifier. The feature extractor can be constructed with various types of layers, such as convolutional and recurrent layers. Its output, $\mathbf{x}$, can be considered as the extracted feature as it contains high-level representations of the input data $\mathcal{G}(\mathbf{r})$. The feature $\mathbf{x}$ is then fed into a classifier composed of fully connected layers, with the output $\hat{\mathbf{y}} = \{\hat{y}_1, ..., \hat{y}_m, ..., \hat{y}_M\}$ being the predicted probabilities for each DUT.

3.1.4 Train with data augmentation

As discussed in Section 2, the training of NN is mathematically an optimization problem, and the mini-batch training is often leveraged considering the training dataset $\mathcal{X}^{\text{train}}$ is large. During the training process, a batch of training samples $\mathcal{X}^{\text{batch}}$ is first selected from $\mathcal{X}^{\text{train}}$, given as

$$\mathcal{X}^{\text{batch}} = \{(\mathbf{r}_b, \mathbf{y}_b)\}_{b=1}^{B} \subset \mathcal{X}^{\text{train}}, \tag{6}$$

where B is the batch size. The artificial noise is then added to each sample in the batch, which is called augmentation and can effectively increase the RFFI performance in low SNR scenarios [42]. The batch after augmentation is mathematically given as

$$\mathcal{X}^{\text{batch}} = \{(\mathbf{r}_b + \mathbf{n}_b, \mathbf{y}_b)\}_{b=1}^{B}, \tag{7}$$

where $\mathbf{n}_b$ is the generated artificial Gaussian noise for the bth training sample. Note that the power of $\mathbf{n}_b$ is not constant and differs for each specific training sample.

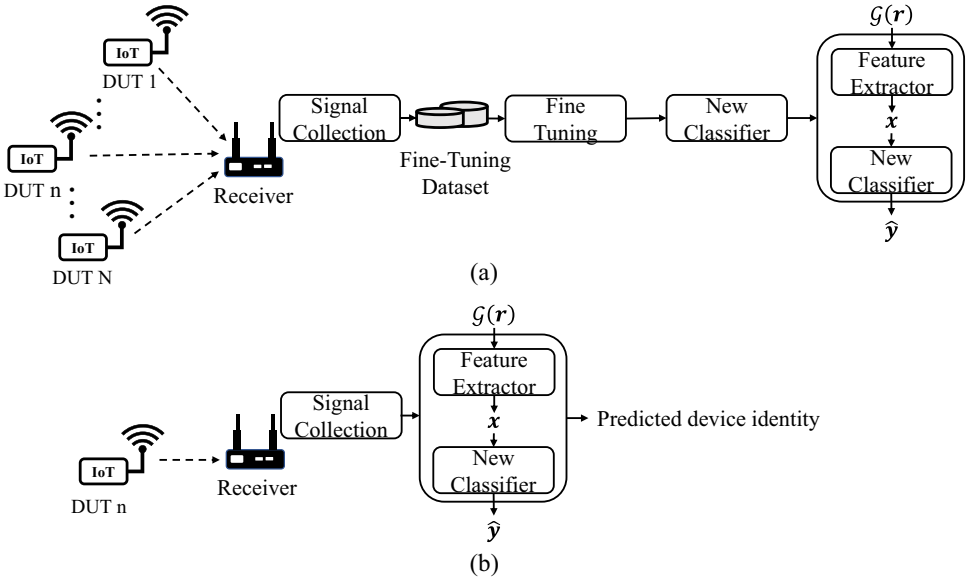

Figure 3. Fine-tuning aided transferable RFFI protocol. (a) Fine-tuning. (b) Inference.

After augmentation, the training samples in $\mathcal{X}^{\text{batch}}$ are converted to designed signal representations and used to calculate gradients, given as

$$\mathbf{g} = \frac{1}{B}\sum_{b=1}^{B} \bigtriangledown_{\theta}\mathcal{L}(c(\mathcal{G}(\mathbf{r}_b + \mathbf{n}_b); \theta), \mathbf{y}_b), \tag{8}$$

where $\mathbf{g}$ is the gradient for batch $\mathcal{X}^{\text{batch}}$ computed to update the NN parameters θ. The training process is repeated until the stop conditions are satisfied.

3.1.5 Remove classifier

Once the training is finished, the classifier is removed from the NN since it is designed to classify the DUTs in C^{train} and becomes useless when applied to a legacy existing IoT network consisting of DUTs $1-N$, *i.e.*, C^{legacy}. Note that the remaining feature extractor is still effective in extracting high-level representations from the input data $\mathcal{G}(\mathbf{r})$ that is collected from unknown categories in C^{legacy}, which will be further experimentally supported in the following sections.

3.2 Fine-tuning aided transferable RFFI protocol

This subsection introduces a fine-tuning-aided transferable RFFI protocol, which consists of fine-tuning and inference stages. The fine-tuning procedure is shown in Figure 3a. Firstly, a fine-tuning dataset X^{tune} is collected from the legacy DUTs $1-N$ that are operating in the IoT network, *i.e.*, C^{legacy}, given as

$$\mathcal{X}^{\text{tune}} = \{(\mathbf{r}_i, \mathbf{y}_i)\}_{i=1}^{I_{\text{tune}}}, \mathbf{y}_i \in C^{\text{legacy}}, \tag{9}$$

where I_{tune} is the number of packets in the fine-tuning dataset $\mathcal{X}^{\text{tune}}$. After that, we fix the parameters of the feature extractor and train a new classifier. Finally, the trained new classifier is connected after the pre-trained feature extractor, which acts as a classification NN that can classify the legacy DUTs in C^{legacy}. Note that the data augmentation introduced in Section 3.1.4 is applied as well, which is an effective mitigation of the limited size of the fine-tuning dataset $\mathcal{X}^{\text{tune}}$. The detailed procedure for data augmentation is not introduced here to reduce redundancy.

In the inference stage shown in Figure 3b, the signal sent from the DUT n in the legacy IoT network is captured, processed, and fed to the assembled classification NN. The output $\hat{\mathbf{y}}$ is the probability vector over DUTs $1-N$.

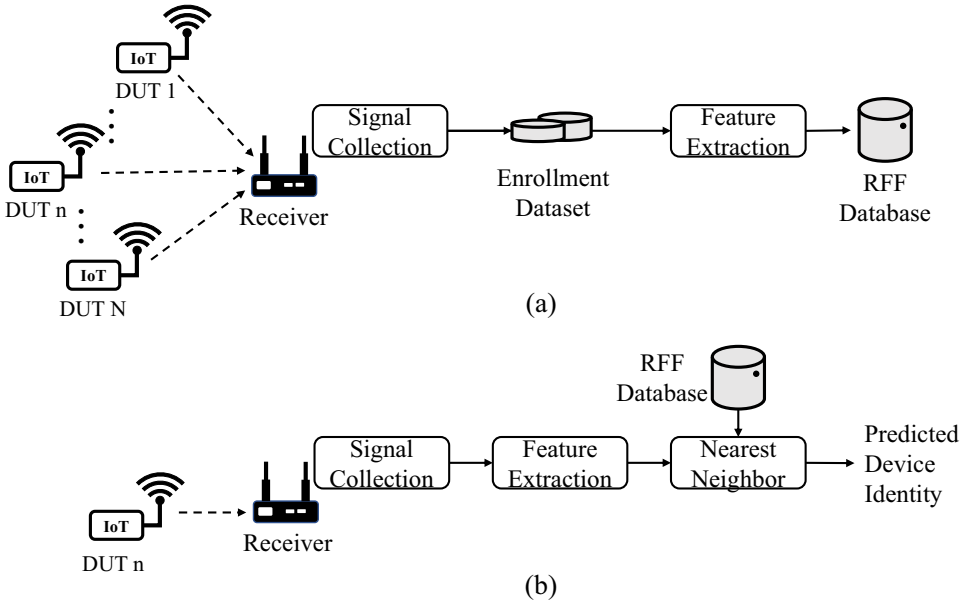

Figure 4. Distance metric learning aided transferable RFFI protocol. (a) Enrollment. (b) Inference.

3.3 Distance metric learning aided transferable RFFI protocol

This subsection introduces a distance metric learning-aided transferable RFFI protocol, which consists of enrollment and inference stages. Firstly, an enrollment dataset $\mathcal{X}^{\text{enrol}}$ is collected from the DUTs in the category set C^{legacy}, given as

$$\mathcal{X}^{\text{enrol}} = \{(\mathbf{r}_i, \mathbf{y}_i)\}_{i=1}^{I_{\text{enrol}}}, \mathbf{y}_i \in C^{\text{legacy}}, \tag{10}$$

where I_{enrol} is the number of signals in the enrollment dataset $\mathcal{X}^{\text{enrol}}$. We then augment $\mathcal{X}^{\text{enrol}}$ by replicating the dataset and adding artificial noise. The enrollment dataset after augmentation is given as

$$\mathcal{X}^{\text{enrol}} = \{(\mathbf{r}_i + \mathbf{n}_i, \mathbf{y}_i)\}_{i=1}^{A \times I_{\text{enrol}}}, \tag{11}$$

where A denotes the number of replications. $\mathbf{n}_i$ is the generated Gaussian noise for the ith enrollment sample. After that, we use the pre-trained feature extractor to process the augmented enrollment dataset $\mathcal{X}^{\text{enrol}}$ and convert it to an RFF database $\mathcal{X}^{\text{rff}}$, given as

$$\mathcal{X}^{\text{rff}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{A \times I_{\text{enrol}}}, \tag{12}$$

where $\mathbf{x}_i$ is the RFF extracted from the ith signal.

The inference stage is illustrated in Figure 4. The collected signal is converted into the designed signal representation, and the pre-trained feature extractor extracts its RFF $\mathbf{x}'$. After that, the kNN algorithm is used to determine which DUT the signal is sent from. More specifically, the distances between $\mathbf{x}'$ and all the RFFs in $\mathcal{X}^{\text{rff}}$ are calculated, and $\mathbf{x}'$ is assigned with the label that is most frequent among its k nearest data samples in $\mathcal{X}^{\text{rff}}$. The cosine distance is leveraged as the distance metric, which is given as

$$D\left(\mathbf{x}_1, \mathbf{x}_2\right) = 1 - \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\| \, \|\mathbf{x}_2\|}, \tag{13}$$

where $D(\mathbf{x}_1, \mathbf{x}_2)$ denotes the cosine distance between $\mathbf{x}_1$ and $\mathbf{x}_2$, $(\cdot)$ denotes the dot product operation, and $\|\cdot\|$ returns the vector magnitude.

4 Experimental evaluation

This section evaluates the proposed transferable RFFI protocols. We take LoRa-RFFI as a case study and use the real-collected LoRa signals for evaluations.

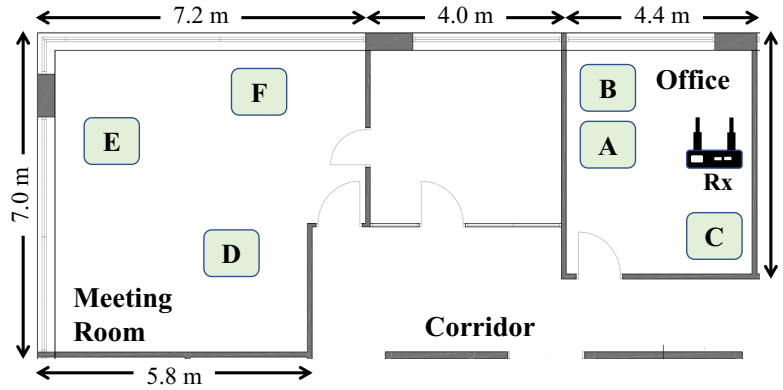

Figure 5. Floor plan.

4.1 Case study: Transferable LoRa-RFFI protocols

The LoRa-RFFI is taken as a case study to evaluate the transferable RFFI protocols proposed in Section 3. Note that the proposed RFFI protocols can be applied to any wireless technologies and are not restricted to LoRa. This subsection introduces the detailed experimental setup as well as the collected datasets.

4.1.1 LoRa modulation primer

LoRa is a wireless modulation technology designed for long-range and power-consuming communication. It is derived from the existing chirp spread spectrum (CSS) technology. More specifically, the linear chirps are used for communication and the information is encoded in the initial frequency. An unmodulated chirp $x(t)$ is mathematically expressed as

$$x(t) = Ae^{j\left(-\pi \mathrm{BW} t + \pi \frac{\mathrm{BW}}{T} t^2\right)}, \tag{14}$$

where A and BW are signal amplitude and bandwidth, respectively. T is the duration of a LoRa symbol. A LoRa packet typically starts with eight repeating $x(t)$ for packet detection and synchronization, which is named preamble. Note that this work only utilizes the preamble part for RFFI to prevent the payload information from contributing to the identification.

4.1.2 Experimental dataset

- **Transmitters (DUTs)**: 40 COTS LoPy4 development kits are used as the DUTs to be identified. The spreading factor, bandwidth, and centre frequency are set to seven, 125 kHz, and 868.1 MHz, respectively.
- **Receiver**: an USRP N210 SDR is utilized as the receiver to capture LoRa physical layer signals, whose sampling rate is set to 1 MHz. The USRP N210 is connected to a laptop that runs the MATLAB LoRa collection program.
- **Pre-training dataset**: the pre-training dataset $\mathcal{X}^{\text{train}}$ contains 30 000 signals collected from DUTs 1−30, *i.e.*, 1000 from each. The transmitter and receiver are approximately half a meter apart with a line of sight (LOS) during data collection.
- **Evaluation datasets**: the evaluation datasets are collected in an office building at six locations. The floor plan is shown in Figure 5. The dataset used for fine-tuning/enrollment is collected at Location A, and the dataset used for inference will be specified in each subsection.

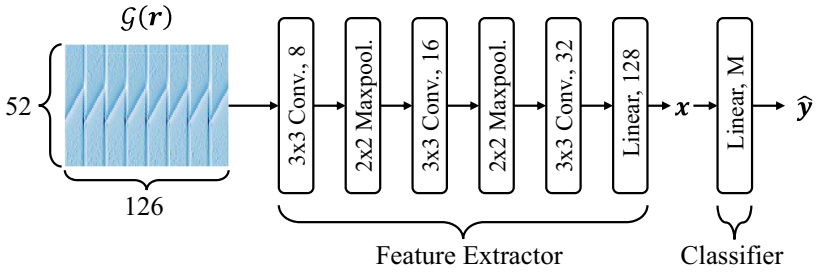

Figure 6. The architecture of the neural network during pre-training. The input is a 52×128 channel-independent spectrogram.

4.1.3 Experimental setup: Pre-training

- **Signal collection**: the LoRa receiver needs to perform signal collection algorithms to capture physical layer IQ samples. The signal collection program includes packet detection, fine synchronization, power normalization, and preamble extraction. After that, the frequency offset compensation is additionally conducted to increase system stability. The algorithms are implemented in MATLAB. More details about the LoRa signal collection algorithms can be found in [16].
- **Signal representation**: the channel-independent spectrogram proposed in [19] is leveraged as the signal representation, which is specially designed for the LoRa modulation technique and can significantly mitigate the channel effects. The conversion function $\mathcal{G}(\cdot)$ converts the collected IQ samples, *i.e.*, a complex vector, into a 2D matrix. The converted channel-independent spectrogram is shown in Figure 6 as the input to the NN. As this paper does not focus on the impact of the wireless channels, interested readers please refer to [19] for more information.
- **Neural network architecture**: the classification NN used for pre-training is shown in Figure 6, whose input is the channel-independent spectrogram. The NN consists of a feature extractor and a classifier. The feature extractor is composed of three convolutional, two 2×2 max-pooling layers, and a linear layer of 128 neurons. The number of kernels in the convolutional layers is 8, 16, and 32, respectively. The convolutional layers are activated by the ReLU function and their outputs as padded to maintain the same size as the inputs. The output of the feature extractor is a 128-element vector $\mathbf{x}$, which is considered the high-level representation extracted from the input data. The feature vector $\mathbf{x}$ is then input to a classifier, *i.e.*, a linear layer of M neurons, and $\hat{\mathbf{y}}$ is the softmax-activated NN output. The NN is implemented with the PyTorch library.
- **Train with augmentation**: after the NN is built, its parameters are updated with the collected training dataset. The augmentation is leveraged during training to increase its robustness to noise. The batch size and initial learning rate are set to 32 and 0.001, respectively. The Adam optimization algorithm is used. The SNR of the signal after augmentation is uniformly distributed in the range of $[0, 80]$ dB. 10% of the training samples are used for validation. We utilize a learning rate scheduler to control the training process, which reduces the learning rate by 0.5 when the validation loss does not change for 10 epochs. The training stops when the validation loss stays unchanged for 20 epochs. The training is conducted on a PC with NVIDIA GeForce GTX 1660.
- **Remove classifier**: as discussed in Section 3.1, the classifier, *i.e.*, fully connected layers, of the classification NN will be removed once training is complete. The rest part serves as a feature extractor. According to our CNN design shown in Figure 6, the feature extractor receives a 52×126 channel-independent spectrogram and outputs a vector of length 128.

4.1.4 Experimental setup: Fine-tuning aided transferable RFFI protocol

In the fine-tuning process, we fix the parameters of the feature extractor and train a new classifier, *i.e.*, a linear layer of N neurons. The training batch size and learning rate are set to four and 0.001, respectively. The SGD optimizer is leveraged. The learning rate scheduler and stop conditions are exactly the same as those used during pre-training. Note that the augmentation technique introduced in Section 3.1 is utilized during fine-tuning as well. The SNR range for augmentation is $[0, 80]$ dB. After fine-tuning, the captured signal can be fed into the classification NN and a predicted identity will be given.

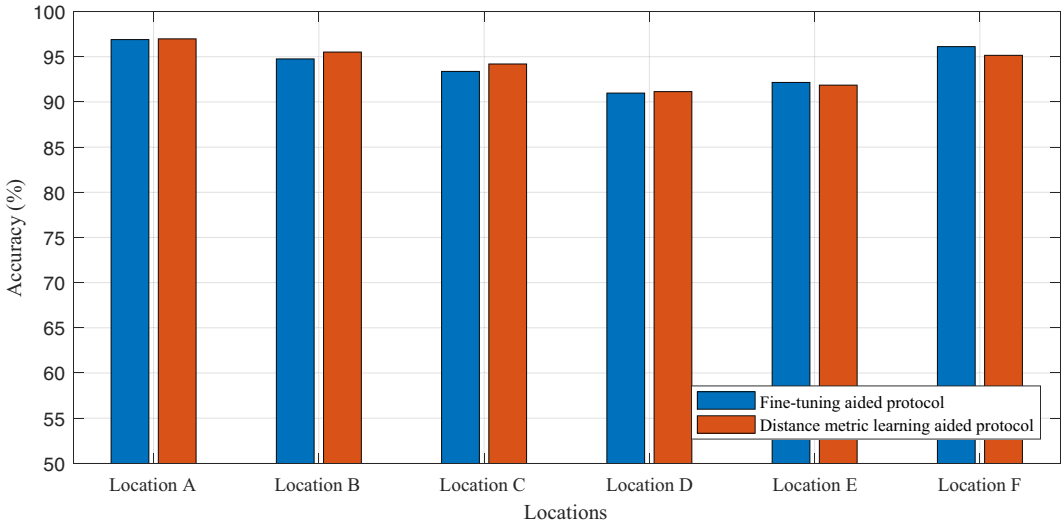

Figure 7. The experimental results of the transferred RFFI systems. The fine-tuning/enrollment dataset contains 15 signals from each DUT at Location A. The evaluation datasets are collected at six locations. The experiments are repeated five times and the average accuracy is calculated.

4.1.5 Experimental setup: Distance metric learning aided transferable RFFI protocol

In the distance metric learning-aided transferable RFFI protocol, we need to first build an RFF database with the pre-trained feature extractor, which is called the enrollment stage. Note that augmentation on the enrollment dataset is also applied by replicating it 10 times and adding artificial noise. Once the RFF database is built, the kNN algorithm can be used to classify the received LoRa signal. The k is set to 15 unless otherwise specified.

4.2 Evaluation of the transferability and robustness to location changes

This section evaluates whether the pre-trained RFFI systems can be transferred to a legacy IoT network that is already in operation without requiring a large number of labelled signals. As the locations of end nodes in an IoT network may change due to movement, we evaluate the performance of the transferred RFFI systems on the datasets collected at different locations.

The pre-training procedure is described in Section 4.1, whose training dataset contains the signals collected from DUT 1−30. Then the pre-trained RFFI system is transferred to identify DUT 31−40, emulating the end nodes operating in an existing legacy IoT network. More specifically, we use 15 signals collected from DUT 31−40 at Location A as the fine-tuning/enrollment dataset and evaluate the transferred RFFI systems on Locations A–F, respectively. The experimental results at six locations are given in Figures 7 and 8 provides the classification results at Location A as confusion matrices. It can be observed that the classification accuracy is always above 90% for both fine-tuning and distance metric learning-based protocols, which shows that the proposed RFFI protocols have excellent transferability and is robust to location changes.

4.3 Effect of fine-tuning/enrollment dataset size

This section investigates the impact of the number of signals in the fine-tuning/enrollment dataset, *i.e.*, I_{tune} and I_{enrol}, on the performance of the transferred RFFI system. It is desired that I_{tune} and I_{enrol} are restricted because the IoT end nodes may be configured with long transmission intervals and thus collecting a large number of signals is time-consuming.

The fine-tuning/enrollment datasets are collected at Location A, and the transferred RFFI systems are evaluated on another 100 signals collected at the same location. The experimental results are given in Figure 9. It can be observed that the classification accuracy increases as more signals are collected from

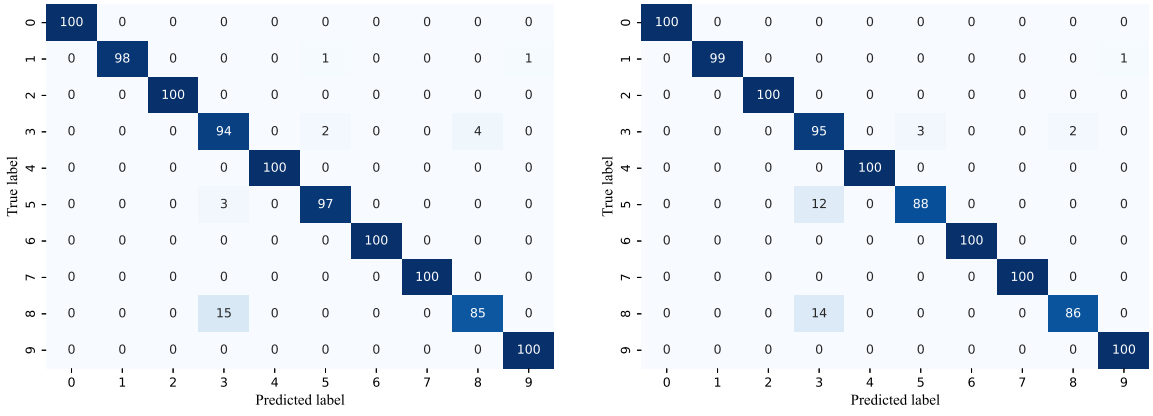

Figure 8. Confusion matrices for the dataset collected at Location A. The fine-tuning/enrollment dataset contains 15 signals from each DUT at Location A. (a) Fine-tuning-based RFFI protocol. (b) Distance metric learning-based RFFI protocol.

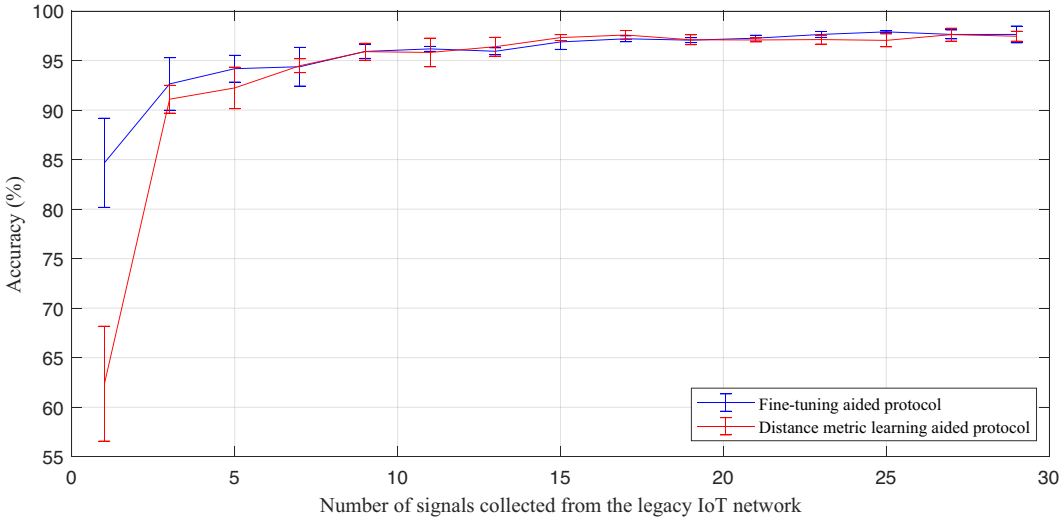

Figure 9. Effect of the number of packets in the fine-tuning/enrollment dataset. Both the fine-tuning/enrollment and evaluation datasets are collected at Location A. The evaluation dataset contains 100 signals from each DUT. The experiments were repeated five times and the error bars are provided.

the legacy IoT network. The performance improvement becomes marginal when the number of signals reaches 15 for both protocols, which implies that the end nodes in the legacy IoT network do not need to transmit numerous signals and thus the time consumption for transfer can be reduced.

4.4 Effect of augmentation on the fine-tuning/enrollment dataset

The augmentation technique can be applied to the fine-tuning/enrollment procedure to further increase the performance of the transferred RFFI systems. As described in Section 3, in the fine-tuning-based protocol, artificial noise is added to the batch, while in the metric learning-based protocol, it is added to the replicated enrollment dataset.

We collect 15 signals from DUT 31−40 at Location A for fine-tuning/enrollment and then use another 100 signals from each DUT for the test. Different levels of artificial noise are added to the test signals to simulate environments with varying SNRs. As illustrated by the experimental results in Figure 10, augmentation during transfer can significantly improve the classification performance. More specifically,

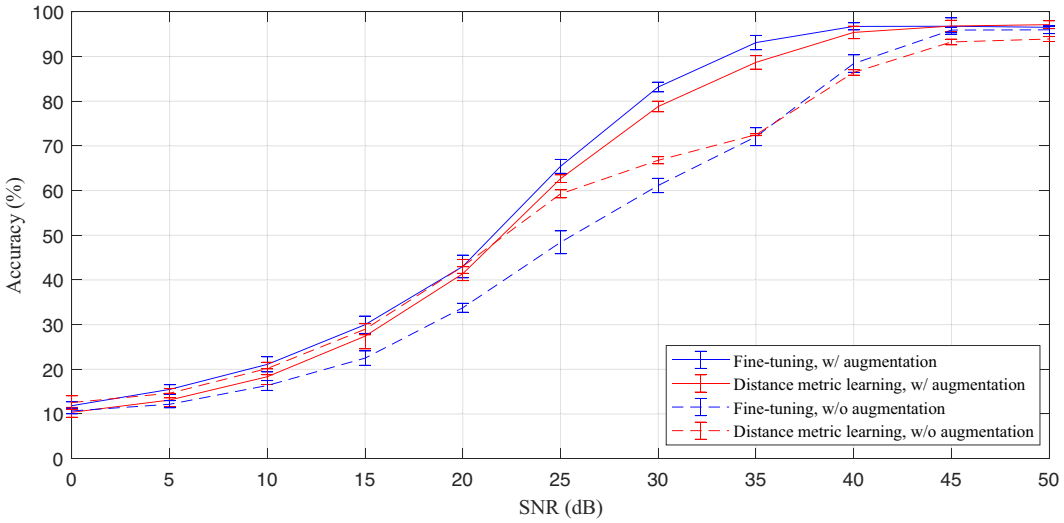

Figure 10. The classification results at different SNR conditions. The fine-tuning/enrollment dataset contains 15 signals from each DUT at Location A. The evaluation dataset contains 100 signals from each DUT at Location A. The experiments were repeated five times and the error bars are provided. Note that "w/" and 'w/o' are abbreviations for "with" and "without", respectively.

the accuracy can be increased by up to 20% when the SNR is between 20 dB and 40 dB. Therefore, augmentation should be leveraged during transfer to improve system performance.

4.5 Summary and discussion

It is experimentally demonstrated that both fine-tuning and distance metric learning-based RFFI protocols can be rapidly deployed to protect legacy existing IoT networks. The results in Figure 10 show that fine-tuning aided RFFI protocol achieves higher accuracy than the distance metric learning-based one. However, fine-tuning introduces additional training costs and thus requires the receiver/authenticator to have the ability to update the NN parameters. In contrast, the distance metric learning-based approach is training-free and is more friendly to low-cost receivers. In summary, there is a trade-off between complexity and performance. It is recommended to apply the metric learning-based approach when the receivers are computing-constrained while applying the fine-tuning-based approach when sufficient computing resources are available.

5 Conclusion

This paper aims to design RFFI protocols that can be rapidly transferred to legacy existing IoT networks without the need to collect numerous signals. Fine-tuning and distance metric learning techniques in transfer learning are utilized, which make the RFFI systems efficiently transferable. More specifically, we first pre-train a feature extractor using a large amount of data, and then only need to collect a few signals from the existing IoT networks for transfer learning. For the fine-tuning-based approach, we train a new classifier for the IoT end nodes in the legacy network, while for the metric learning-based approach, the kNN algorithm is used for classification. Since the signals used for transfer learning are limited in number, we propose to perform augmentation on the transfer dataset to further improve performance. A LoRa-RFFI system is built as a case study to evaluate the proposed transferable RFFI protocols, consisting of 40 COTS LoRa DUTs and a USRP N210 SDR receiver. The experimental results show that both the proposed RFFI protocols can achieve classification accuracy higher than 90% when transferred to a new IoT network, and require no more than five signals per DUT. It is also experimentally demonstrated that augmenting the transfer dataset can improve the identification performance by up to 20%.

Conflict of Interest

The author declares no conflict of interest.

Data Availability

Please send an email to the authors for the data.

Authors' Contributions

G. Shen designed the algorithms, carried out experiments, produced results and wrote the paper. J. Zhang supervised the work and reviewed the paper.

Acknowledgements

We greatly appreciate the comments and suggestions of all reviewers.

Funding

The work was in part supported by UK Engineering and Physical Sciences Research Council under grant ID EP/V027697/1 and in part by the National Key Research and Development Program of China under grant ID 2020YFE0200600.

References

[1] Number of IoT connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030 (in billions). https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (17 March 2023, online; accessed), 2022.

[2] Liu P, Yang P and Song W-Z et al. Real-time identification of rogue WiFi connections using environment-independent physical features. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, Paris, France, Apr. 2019, 190–8.

[3] Joo K, Choi W and Lee DH. 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.

[4] Hua J, Sun H and Shen Z et al. Accurate and efficient wireless device fingerprinting using channel state information. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE, Honolulu, HI, USA, Apr. 2018, 1700–8.

[5] Shi Y and Jensen MA. Improved radiometric identification of wireless devices using MIMO transmission. IEEE Trans Inf Forensics Secur 2011; **6**: 1346–54.

[6] Polak AC and Goeckel DL. Wireless device identification based on RF oscillator imperfections. IEEE Trans Inf Forensics Secur 2015; **10**: 2492–501.

[7] Peng L, Hu A and Zhang J et al. Design of a hybrid RF fingerprint extraction and device classification scheme. IEEE Internet Things J 2018; **6**: 349–60.

[8] Brik V, Banerjee S and Gruteser M et al. Wireless device identification with radiometric signatures. In: MobiCom '08: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, San Francisco, CA, USA, Sep. 2008, 116–27.

[9] Polak AC, Dolatshahi S and Goeckel DL. Identifying wireless users via transmitter imperfections. IEEE J Sel Areas Commun 2011; **29**: 1469–79.

[10] Li Y, Ding Y and Zhang J et al. Radio frequency fingerprinting exploiting non-linear memory effect. IEEE Trans Cogn Commun Netw 2022; **8**: 1618–31

[11] Balakrishnan S, Gupta S and Bhuyan A et al. Physical layer identification based on spatial–temporal beam features for millimeter-wave wireless networks. IEEE Trans Inf Forensics Secur 2019; **15**: 1831–45.

[12] Wang N, Li W and Jiao L et al. Orientation and channel-independent RF fingerprinting for 5G IEEE 802.11 ad devices. IEEE Internet Things J 2021; **9**: 9036–48.

[13] Robyns P, Marin E and Lamotte W et al. Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning. In: Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2017, 58–63.

[14] Peng L, Zhang J and Liu M et al. Deep learning based RF fingerprint identification using differential constellation trace figure. IEEE Trans Veh Technol 2019; **69**: 1091–95.

[15] Al-Shawabka A, Restuccia F and D'Oro S et al. Exposing the fingerprint: dissecting the impact of the wireless channel on radio fingerprinting. In: IEEE INFOCOM 2020 – IEEE Conference on Computer Communications. IEEE, Toronto, ON, Canada, Jul. 2020, 646–55.

[16] Shen G, Zhang J and Marshall A et al. Radio frequency fingerprint identification for LoRa using deep learning. IEEE J Sel Areas Commun 2021; **39**: 2604–16.

[17] Zhang J, Woods R and Sandell M et al. Radio frequency fingerprint identification for narrowband systems, modelling and classification. IEEE Trans Inf Forensics Secur 2021; **16**: 3974–87.

[18] Shen G, Zhang J and Marshall A et al. Radio frequency fingerprint identification for LoRa using spectrogram and CNN. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, Virtual Conference. IEEE, Vancouver, BC, Canada, 2021, 1–10.

[19] Shen G, Zhang J and Marshall A et al. Towards scalable and channel-robust radio frequency fingerprint identification for LoRa. IEEE Trans Inf Forensics Secur 2022; **17**: 774–87. Dataset and code are available: https://ieee-dataport.org/open-access/lorarffidataset (29 January 2023, last accessed).

[20] Roy D, Mukherjee T and Chatterjee M et al. RFAL: adversarial learning for RF transmitter identification and classification. IEEE Trans Cogn Commun Netw 2019; **6**: 783–801.

[21] Cekic M, Gopalakrishnan S and Madhow U. Wireless fingerprinting via deep learning: the impact of confounding factors. In: 2021 55th Asilomar Conference on Signals, Systems, and Computers. IEEE, Pacific Grove, CA, USA, 2021, 677–84.
[22] Yu J, Hu A and Li G et al. A robust RF fingerprinting approach using multisampling convolutional neural network. IEEE Internet Things J 2019; **6**: 6786–99.
[23] Jian T, Gong Y and Zhan Z et al. Radio frequency fingerprinting on the edge. IEEE Trans. Mobile Comput. 2021; **21**: 4078–93.
[24] Soltani N, Sankhe K and Dy J et al. More is better: data augmentation for channel-resilient RF fingerprinting. IEEE Commun Mag 2020; **58**: 66–72.
[25] Soltani N, Reus-Muns G and Salehihikouei B et al. RF fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms. IEEE Trans Veh Technol 2020; **69**: 15518–531.
[26] Qian Y, Qi J and Kuai X et al. Specific emitter identification based on multi-level sparse representation in automatic identification system. IEEE Trans Inf Forensics Secur 2021; **16**: 2872–84.
[27] Al-Shawabka A, Pietraski P and Pattar SB et al. DeepLoRa: fingerprinting LoRa devices at scale through deep learning and data augmentation. In: Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, Jul. 2021, 251–60.
[28] Piva M, Maselli G and Restuccia F. The tags are alright: robust large-scale RFID clone detection through federated data-augmented radio fingerprinting. In: Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, Jul. 2021, 41–50.
[29] Merchant K, Revay S and Stantchev G et al. Deep learning for RF device fingerprinting in cognitive communication networks. IEEE J Sel Topics Signal Process 2018; **12**: 160–7.
[30] Elmaghbub A and Hamdaoui B. LoRa device fingerprinting in the wild: disclosing RF data-driven fingerprint sensitivity to deployment variability. IEEE Access 2021; **9**: 142893–909.
[31] Hanna S, Karunaratne S and Cabric D. WiSig: a large-scale WiFi signal dataset for receiver and channel agnostic RF fingerprinting. IEEE Access 2022; **10**: 22808–18.
[32] Xie R, Xu W and Chen Y et al. A generalizable model-and-data driven approach for open-set RFF authentication. IEEE Trans Inf Forensics Secur 2021; **16**: 4435–50.
[33] Rajendran S and Sun Z. RF impairment model-based IoT physical-layer identification for enhanced domain generalization. IEEE Trans Inf Forensics Secur 2022; **17**: 1285–99.
[34] Ding L, Wang S and Wang F et al. Specific emitter identification via convolutional neural networks. IEEE Commun Lett 2018; **22**: 2591–4.
[35] Das R, Gadre A and Zhang S et al. A deep learning approach to IoT authentication. In: 2018 IEEE international conference on communications (ICC). IEEE, Kansas City, MO, USA, 2018, 1–6.
[36] He B and Wang F. Cooperative specific emitter identification via multiple distorted receivers. IEEE Trans Inf Forensics Secur 2020; **15**: 3791–3806.
[37] Shen G, Zhang J and Marshall A et al. Radio frequency fingerprint identification for security in low-cost IoT devices. In: 2021 55th Asilomar Conference on Signals, Systems, and Computers. IEEE, Pacific Grove, CA, USA, 2021, 309–13.
[38] Xu Y, Liu M and Peng L et al. Colluding RF fingerprint impersonation attack based on generative adversarial network. In: ICC 2022-IEEE International Conference on Communications. IEEE, Seoul, Republic of Korea, 2022, 3220–25.
[39] Merchant K and Nousain B. Securing IoT RF fingerprinting systems with generative adversarial networks. In: MILCOM 2019-2019 IEEE Military Communications Conference. IEEE, Norfolk, VA, USA, 2019, 584–9.
[40] Chen Z, Peng L and A. Hu et al. Generative adversarial network-based rogue device identification using differential constellation trace figure. *EURASIP J Wireless Commun Netw* 2021; **2021**: 1–27.
[41] Xu H and Xu X, A transformer based approach for open set specific emitter identification. In: 2021 7th International Conference on Computer and Communications (ICCC). IEEE, Chengdu, China, 2021, 1420–5.
[42] Shen G, Zhang J and Marshall A et al. Towards length-versatile and noise-robust radio frequency fingerprint identification. IEEE Trans Inf Forensics Secur 2023; **18**: 2355–67.

Guanxiong Shen received a B.Eng degree from Xidian University, Xi'an, China, in 2019. He is currently pursuing a 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 Queen's University Belfast. From Feb. 2018 to May 2020, he was a Tenure Track Fellow (Assistant Professor) at the University of Liverpool, UK. Since June 2020, he is a Lecturer (Assistant Professor) with University of Liverpool. His research interests include the Internet of Things, wireless security, physical layer security, key generation, radio frequency fingerprint identification, and wireless sensing.