Towards Channel-Robust Radio Frequency Fingerprint Identification Using Contrastive Learning

Jie Ma*, Junqing Zhang*, Guanxiong Shen†, Linning Peng†, and Alan Marshall*
* Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom
† School of Cyber Science and Engineering, Southeast University, Nanjing, China

***Abstract*—Radio frequency fingerprint identification (RFFI) is an emerging device authentication technique that is based on intrinsic hardware impairments. Internet of things (IoT) devices can be identified and classified based on their wireless signals using RFFI. Developing a robust RFFI system that can maintain high classification accuracy across diverse communication scenarios is a critical challenge. In this paper, we proposed a contrastive learning-based RFFI approach to establish a channel-robust system using the spectrogram. Specifically, we leverage contrastive learning in the training stage, which has been implemented with data augmentation techniques to mitigate the influence of channels on RFFI. We carried out extensive experimental evaluations involving a public dataset and a self-collected dataset, both with ten LoRa devices. Utilizing these datasets, the performance of the system has been tested in various channel environments, including stationary, mobile, line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios. The results demonstrated that our approach is effective and robust to channel variation, achieving 93% and 82% on static and dynamic channels, respectively.**

***Index Terms*—Contrastive Learning, Radio Frequency Fingerprint Identification, Deep Learning**

I. Introduction

THE increasing use of wireless devices has raised security concerns regarding device authentication in Internet of Things (IoT) networks. Conventional authentication methods rely on cryptographic algorithms combined with software addresses, e.g., media access control (MAC) addresses. However, cryptographic authentication approaches heavily rely on substantial computational resources, which is not affordable to low-cost IoT devices [1]. Moreover, software addresses can be easily tampered with. Therefore, there is a strong need for a low-complexity and reliable IoT authentication scheme.

Radio frequency fingerprint identification (RFFI) is an effective and secure method for physical layer identity authentication. It identifies wireless devices by analyzing the unique distortion of the received waveform caused by the imperfections of hardware components. The imperfection comes from the transmitter chain of the wireless device [2], such as mixers, oscillators, power amplifiers, etc. These components deviate slightly from their nominal specifications due to the inevitable manufacturing process variations. It manifests as hardware imperfections such as IQ imbalance, carrier frequency offset (CFO), and power amplifier non-linearity [3]–[5]. These hardware impairments slightly impact the wireless signal waveform, but do not affect communication transmission [2]. The RFFI system extracts unique features from wireless signals emitted by IoT devices to determine their identities.

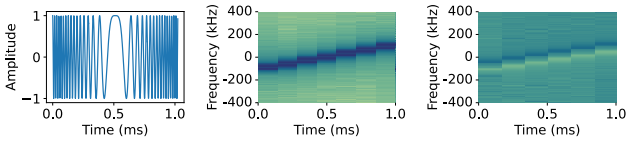
Corresponding author: J. Zhang. (Email: junqing.zhang@liverpool.ac.uk)

Deep learning (DL) has the excellent capability of feature extraction and classification, which has been widely applied in RFFI [6]–[8]. It contributes to automatically extracting the discriminative features from interrelated hardware impairments and benefits to build a well-performance RFFI system. However, there are several challenges with the RFFI, with one of the most crucial issues being to develop a system that is robust to channel variations.

Addressing the effects of channel variations in RFFI is essential, as wireless devices frequently change locations or move, causing signals to be influenced by different channels. The work in [9] extracts the channel-independent spectrogram by dividing adjacent columns in the spectrogram. This process eliminates channel interference, enabling the extraction of device characteristics. However, the algorithm assumes that the channel does not undergo significant changes over short periods of time, which may become invalid in certain mobile scenarios. The work in [10] utilized spectral quotient sequences and moment-based statistical feature extractor to build a channel-agnostic RFFI system. However, only simulation results are provided and its performance in real environments remains unknown. Despite several techniques have been proposed to reduce the impact of channel effects [9]–[12], certain limitations still persist. These challenges motivate further exploration to address the channel issues in RFFI.

Contrastive learning is a DL method that helps neural networks learn meaningful representations by comparing the similarities and differences between samples. Prevalent methods in contrastive learning include SimCLR [13], triplets [14], InfoNCE [15], etc. The goal of the above methods is to bring similar samples closer together in the latent space while pushing dissimilar samples further apart. There are some initial explorations in applying contrastive learning in RFFI [16], [17]. A scheme that alleviates performance degradation in the cross-receiver scenario has been proposed in [16]. [17] proposed a noise-tolerant RFFI system by data augmentation and contrastive learning. It considers the impact of different signal-to-noise ratios (SNR) on the system but overlooks different communication scenarios, such as dynamic channel scenarios.

We proposed to employ contrastive learning to mitigate



(a) (b) (c)

Fig. 1. (a) Time domain signal of the preamble (I branch). (b) Spectrogram of the preamble. (c) Channel-independent spectrogram of the preamble.

the effects of channel variations. To assess its performance, we conducted evaluations on two datasets including a public LoRa dataset [9] and a self-collected LoRa dataset. Both datasets were gathered from diverse real-world environments, encompassing both static and mobile channels. The technical contributions of this paper are summarized as follows.

- We designed a contrastive learning-based RFFI system, which leverages spectrogram, data augmentation, and contrastive loss to extract device-specific features, making the system robust to channel variations.
- We created a dataset involving 10 LoRa devices and a USRP N210 SDR platform. Data collection was performed in both line-of-sight (LOS) and non-line-of-sight (NLOS) mobile scenarios.
- We evaluated the performance of the approach in both static and dynamic channels using a public LoRa dataset and a self-collected dataset. The classification accuracy remains above 93% across all positions in static channels. Our method is also robust in dynamic channels, and its accuracy reaches over 82% in mobile scenarios.

The rest of the paper is organized as follows. Section II presents the signal model of LoRa in the RFFI system. The design details of the RFFI system are introduced in Section III. Section IV illustrates the implementation of contrastive learning-based RFFI system. Section V demonstrates the experimental results and Section VI concludes the paper.

II. Problem Statement

A. LoRa Primer

LoRa is a widely used wireless modulation technique for long-range IoT applications. It employs chirp spread spectrum (CSS) technology, and there are several up-chirps at the beginning of each LoRa packet, named preambles. The contents of the preambles are identical across all packets and are unaffected by the payload, making it ideal for RFFI.

The LoRa signal's instantaneous frequency varies continuously over time, and a baseband LoRa preamble is given as

$$s(t) = Ae^{j\left(-\pi Bt+\pi\frac{B}{T}t^2\right)} \quad (0 \le t \le T), \tag{1}$$

where A, B, T are the amplitude, bandwidth and symbol duration, respectively. Fig. 1(a) shows the time-domain baseband signal of the preamble (I branch).

B. Channel Effect in IQ Samples

The received signal is expressed as

$$x(t) = h(t) * \mathcal{F}_m(s(t)) + n(t), \tag{2}$$

where $h(t)$ is the wireless channel impulse response, $*$ represents the convolutional operation, $\mathcal{F}_m(\cdot)$ denotes the effect of the m^{th} DUT's hardware impairments on the waveform, and $n(t)$ denotes additive white Gaussian noise (AWGN).

In a DL-based RFFI, the channel effect $h(t)$ will change the data distribution of $x(t)$ both in the training and test stage. Hence, mitigating the channel effect is essential. However, it is quite challenging to eliminate the channel effect in the time domain due to the convolutional process.

C. Spectrogram

IQ signals are typically distorted by wireless channels, while the frequency-varying nature of LoRa signals makes time-frequency analysis techniques, such as the short-time Fourier transform (STFT), a common choice for representation [18]–[20]. After STFT, the time-domain IQ signal, $x(t)$, can be transformed to $X(t,f)$, given as

$$X(t,f) = \mathcal{S}\big(x(t)\big) = \mathcal{S}\big(h(t)\big)\mathcal{S}\big(\mathcal{F}_m(s(t))\big) + \mathcal{S}\big(n(t)\big), \tag{3}$$

where $\mathcal{S}(\cdot)$ represents STFT operation. We then convert the amplitude of the spectrogram from a linear scale to a logarithmic scale, which can be expressed as

$$\begin{aligned}\log|X(t,f)| &= \log\left|\mathcal{S}\big(h(t)\big)\mathcal{S}\big(\mathcal{F}_m(s(t))\big) + \mathcal{S}\big(n(t)\big)\right| \\ &\approx \log|\mathcal{S}(h(t))| + \log|\mathcal{S}(\mathcal{F}_m(s(t))|.\end{aligned} \tag{4}$$

A spectrogram of the preamble part is exemplified in Fig. 1(b).

As can be observed from (4), the channel effect still exists. However, it becomes a summation term in the spectrogram, which is easier to separate. This paper employs contrastive learning to eliminate the channel effect and improve the robustness of RFFI against channel variations.

III. System Overview

As shown in Fig. 2, our RFFI system consists of M DUT and a receiver. It involves two stages, namely training and inference.

A. Training

In the training stage, each DUT emits hundreds of packets, which are captured by a receiver. The receiver will then create a training dataset, $\boldsymbol{\mathcal{D}_{train}}$, and train a neural network.

1) Preprocessing: Once the signal is captured by the receiver, it undergoes signal preprocessing, including synchronization, preamble extraction, CFO compensation, and normalization.

- **Synchronization** identifies the packet's start to capture the signal accurately.
- **Preamble extraction** is designed to capture the preamble part of the signal. It is beneficial for the system to acquire the unique inherent characteristics of the device.

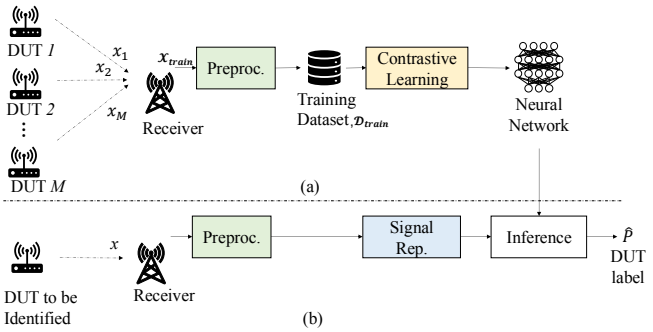


Fig. 2. A contrastive learning-based RFFI system. (a) Training stage. (b) Inference stage.

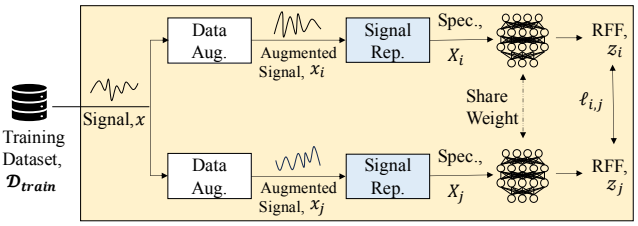


Fig. 3. Contrastive learning in RFFI with spectrogram.

- **CFO compensation** is employed to enhance system stability, which mitigates the influence of temperature fluctuations and oscillator frequency drift.
- **Normalization** prevents the neural network from being affected by amplitude variations in the received signal. The preamble part is normalized by dividing its root mean square (RMS).

Please refer to the work in [18] for a detailed description of the above algorithms.

2) Contrastive Learning: We incorporate contrastive learning during the training stage to strengthen the system's ability to handle channel variation. As will be elaborated in Section IV, the key components of contrastive learning are data augmentation and contrastive loss. Data augmentation is used to imitate variations from channels, and contrastive loss is employed to alleviate the interferences.

The loss function $\mathcal{L}$ used in this stage consists of contrastive loss $\mathcal{L}_{cl}$ and cross-entropy $\mathcal{L}_{ce}$, which can be given as

$$\mathcal{L} = \mathcal{L}_{cl} + \mathcal{L}_{ce}. \tag{5}$$

A channel-robust neural network, i.e., $f(\cdot;\theta)$, will be obtained when contrastive learning is completed.

B. Inference

The receiver acquires a packet from the device to be identified. After preprocessing and signal representation, the signal will be sent into the neural network $f(\cdot;\theta)$. Finally, the prediction of the signal will be given as

$$\hat{P} = f(x;\theta), \tag{6}$$

where $\hat{P}$ is a label predicted by the model f for the most likely transmitter of the input signal.

IV. Contrastive Learning in RFFI

A. Overview

As shown in Fig. 3, contrastive learning compares two augmented signals and brings their feature representation closer together in the latent space, because the two signals are from the same class. Specifically, a signal, x, is selected from the training dataset, and two augmented signals, x_i and x_j, are generated by data augmentation. They are converted to spectrograms, X_i and X_j, via signal representation and then fed into the neural networks for training. It is worth noting the two neural networks share the same architecture and weights. The contrastive loss is then calculated based on the outputs of the two neural networks.

B. Data Augmentation

Data augmentation is pivotal in contrastive learning, as it enriches training datasets and facilitates the construction of comparison pairs. In RFFI, the two branches will independently augment the same signal, creating two augmented versions of the same example. The stochastic data augmentation module randomly transforms any given data example, creating different views of the example. After data augmentation, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ will be obtained, which we consider as a positive pair.

In the proposed system, both channel and AWGN augmentation are used. For the channel model, the tapped delay line (TDL) channel has been chosen, which is provided in the PyPhysim library[1]. The multipath effect is modelled using the exponential power delay profile (PDP), while the Doppler effect is characterized by the Jakes model. Additionally, the synthetic AWGN noise generated using Python is incorporated in data augmentation. Unless otherwise specified, the ranges for RMS delay spread, Doppler frequency, and SNR are $[5, 300]$ ns, $[0, 5]$ Hz, and $[10, 40]$ dB, respectively.

The simulation parameters mentioned above are randomized for each specific LoRa signal. Online data augmentation [6] is utilized, which applies data augmentation in real-time, typically during each epoch in training. It contributes to increasing the training dataset and improving the model's generalization.

C. Signal Representation

Signal representation determines the network's input, which could be time-domain, frequency-domain, or other forms [18]. In our work, we chose spectrogram as the representation, which was introduced in Section II-C. After signal representation, the positive pair denoted as X_i and X_j.

D. Neural Network

The neural network extracts the representation features from the augmented signals. Various architectures such as CNN, LSTM, transformer, etc, can be used for RFFI. The final layer typically employs an MLP to project high-dimensional vectors

[1]https://pypi.org/project/pyphysim/

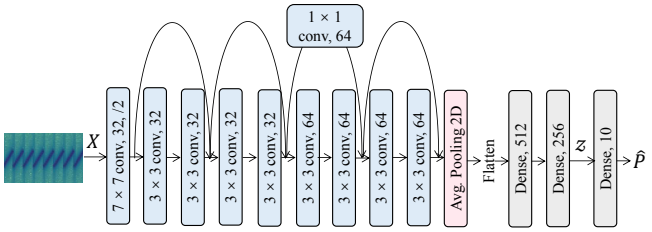


Fig. 4. Architecture of the neural network.

into low-dimensional vectors. Reducing the dimension will be more beneficial for calculating contrastive loss.

Considering that the spectrograms have been chosen to feed into the network, they can be regarded as 2D images when sent to the DL models. Therefore, we refer to ResNet, which is widely applied in computer vision, to design the neural network in our system. The architecture of the neural network is shown in Fig. 4, which consists of nine convolutional layers, one average pooling layer and three dense layers. The first convolutional layer uses 32 7×7 kernels. The second to the fifth layers use 32 3×3 kernels, while the sixth to the ninth layers employ 64 3×3 kernels. Three dense layers containing 512, 256 and 10 neurons, respectively. Specifically, the contrastive loss has been calculated by the outputs of the second dense layer.

E. NT-Xent Loss Function $\ell_{i,j}$

The contrastive loss encourages positive pairs to be close in the feature space and minimizes the agreement of all other negative pairs. In this paper, we chose SimCLR [13], which utilizes the normalized temperature-scaled cross-entropy (NT-Xent) loss. The neural network learns representations by maximizing the similarity between differently augmented views of the signal from the same device via NT-Xent loss.

The loss function is defined to obtain the representative features. Given a set $\{\boldsymbol{x}_k\}$ including a positive pair of examples $\boldsymbol{x}_i$ (X_i) and $\boldsymbol{x}_j$ (X_j), the contrastive loss helps to identify $\boldsymbol{x}_j$ (X_j) in $\{\boldsymbol{x}_k\}_{k\neq i}$ for a given $\boldsymbol{x}_i$ (X_i). Let $\text{sim}(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{u}^\top\boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$ denote the dot product between l_2 normalized $\boldsymbol{u}$ and $\boldsymbol{v}$ (i.e. cosine similarity). Consequently, the loss function for each positive pair (i,j) is defined as

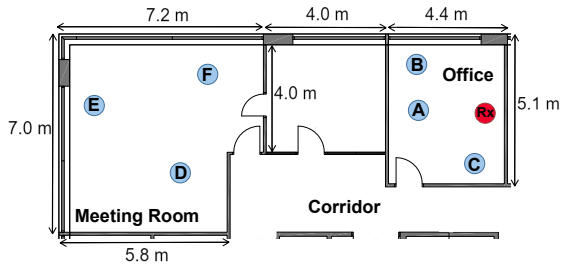
$$\ell_{i,j} = -\log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}_i,\boldsymbol{z}_j\right)/\tau\right)}{\sum_{k=1}^{2|\mathcal{B}|}\mathbb{1}_{[k\neq i]}\exp\left(\text{sim}\left(\boldsymbol{z}_i,\boldsymbol{z}_k\right)/\tau\right)}, \tag{7}$$

where $\mathbb{1}_{[k\neq i]} \in \{0,1\}$ is an indicator function that evaluates to 1 if $k \neq i$, τ denotes a temperature parameter (set as 0.05 in our work), and $|\mathcal{B}|$ is batch size. The similarity between RFF $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ will be maximized by (7), and the total loss $\mathcal{L}_{cl}$ is computed across all positive pairs in a batch $\mathcal{B}$.

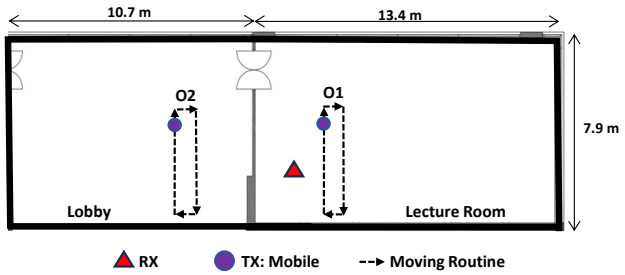
V. Experimental Evaluation

A. Setup

Two datasets were utilized for evaluation, including one public dataset [9], and another self-collected one.



(a)



(b)

Fig. 5. Floor plan of the experimental environment. (a) Public dataset [9]. (b) Self-collected dataset.

1) Public Dataset: The work in [9] released a public LoRa RFFI dataset and a subset of the dataset was used in this paper. Specifically, the DUTs were ten LoPy4 devices and the receiver was a USRP N210 SDR platform. The floorplan is shown in Fig. 5(a).

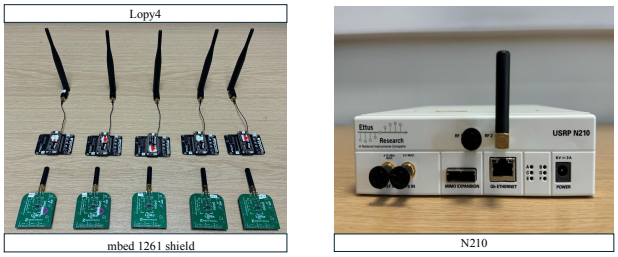
Training dataset: There were 400 LoRa packets for each device, which were collected in a residential room with LOS available. The DUT and the receiver was kept close (about half meter), in order to obtain high SNR data.

Test dataset: There were 200 packets per device. There were two scenarios, namely static and dynamic channels.

- Static channel: Testing datasets D1-D6 were collected at six different locations, labelled as A-F, respectively. The DUTs and the USRP N210 remained stationary, with no moving objects nearby.
- Dynamic channel: There were two cases. (i) Objective moving: D7 and D8 were collected at Location B and F, respectively, with the DUTs and USRP kept stationary while a person walked randomly around the room at a speed of 2 m/s. (ii) Mobile device: D9 and D10 were respectively collected in the office and meeting room, with the USRP remaining stationary as a person walked around carrying the DUTs at a speed of 2 m/s.

Datasets D1, D2, D3, D7, and D9 were collected in an LOS environment, while D4, D5, D6, D8, and D10 were collected in an NLOS environment.

2) Self-collected Dataset: We also collected our own dataset. As shown in Fig. 6(a), five LoPy4 and five mbed 1261 shields were used as DUTs, which are sequentially designated as DUT1 to DUT10. The receiver end was also a USRP N210 SDR platform, as shown in Fig. 6(b).



(a) (b)

Fig. 6. Experimental devices. (a) DUT: ten LoRa devices. (b) Receiver: USRP N210 SDR.

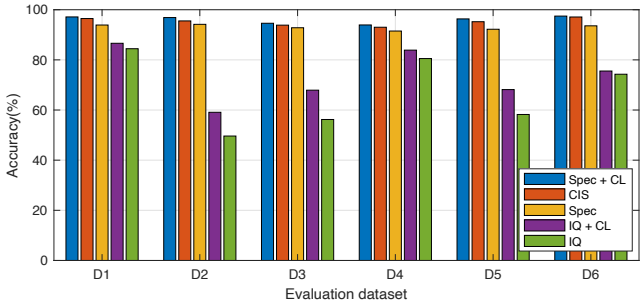
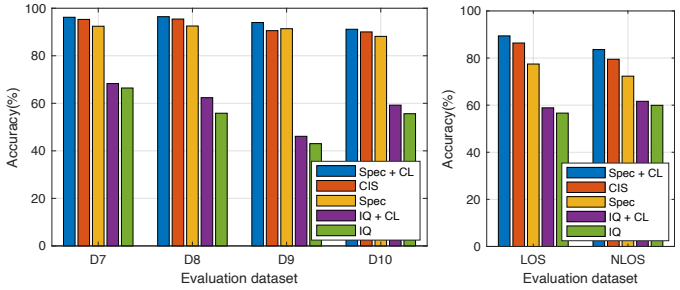


Fig. 7. Classification results in the static channels. D1-D3 were collected in LOS scenarios while D4-D6 were collected in NLOS scenarios.



(a) (b)

Fig. 8. Classification results in the dynamic channels. (a) Public dataset. (b) Self-collected dataset.

Training dataset: There were 800 LoRa packets for each device. During the collection of the training dataset, the transmitter and the receiver were placed stationary, one meter apart, with a LOS between them.

Test dataset: Each testing dataset includes 200 packets per device. We considered a mobile scenario, which is the most challenging. The receiver was kept stationary while a person took the DUTs walking along routes O1 and O2 at a speed of 2 m/s for LOS and NLOS mobile scenarios, respectively. Fig. 5(b) displays the floor plan of the environment where the testing datasets were collected.

3) Device Configuration: Both datasets used the same device configurations. The DUTs were configured with bandwidth $B = 125$ kHz and carrier frequency $f_c = 868.1$ MHz. The sampling rate of the receiver end USRP N210 is $f_s = 1$ MHz. The Communications Toolbox Support Package for USRP Radio of MATLAB R2021b is used for accessing data from USRP[2]. Eight preambles at the start of each LoRa packet were captured for RFFI.

4) Training Configuration: The deep learning training was carried out on a PC equipped with a GPU of NVIDIA GeForce RTX 4090. PyTorch was used. The neural network parameters were optimized by the adaptive moment estimation (Adam) optimizer with an initial learning rate of 0.0003 and a batch size of 32. The validation loss was monitored during training, and the learning rate was reduced by a factor of 0.5 when the validation loss did not drop within 10 epochs. Training stopped when the validation loss did not change within 30 epochs.

B. Benchmark and Metrics

We employed four benchmark algorithms for comparison.

- CIS [9]: An effective algorithm to mitigate channel effects while preserving the device-specific characteristics. The representation is shown in Fig. 1(c).
- Spec: spectrogram as input and without contrastive learning.
- IQ + CL: IQ samples as input and the neural network trained with contrastive learning.
- IQ: IQ samples as input and without contrastive learning.

[2]https://uk.mathworks.com/help/comm/usrpradio.html

For IQ signals, the network architecture is also adapted from ResNet, as Fig. 4 shows, but the kernel size is adjusted according to the dimensions of the IQ.

Accuracy is the primary evaluation criterion for the RFFI system, expressed as

$$\text{Accuracy} = \frac{\text{Number of correctly classified packets}}{\text{Total number of packets}}.$$

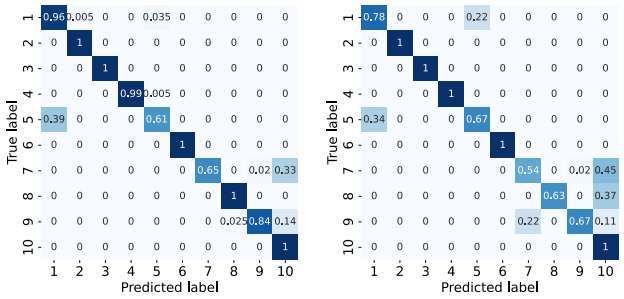
The experiments of all the evaluations were repeated 4 times and the average accuracy was provided.

C. Results with Static Channels

Fig. 7 shows the classification results in various static channels. Spectrogram with contrastive learning demonstrates the best performance across all datasets, regardless of whether in a LOS or NLOS environment. The accuracy of the spectrogram with contrastive learning remains above 94% in LOS scenarios and 93% in NLOS scenarios. The performance of IQ is lower than the spectrogram, even with contrastive learning. It is related to the fact that channel interference, which presents as a convolution in the time domain, is more difficult to eliminate compared to channel effects in the frequency domain.

D. Results with Dynamic Channels

The classification results in dynamic channels using the public dataset are presented in Fig. 8(a). It can be observed that the spectrogram with contrastive learning performs best on all datasets, which indicates the proposed RFFI system is robust to varying channels. The accuracy on the Dataset



(a) (b)

Fig. 9. Confusion matrix of the spectrogram with contrastive learning. (a) LOS. Overall accuracy: 90.55% (b) NLOS. Overall accuracy: 82.8%.

D9 and D10 is lower than D7 and D8. It suggests that in a dynamic channel, a moving device will experience a greater impact on RFF compared to a stationary device. The lower overall classification accuracy in Fig. 8(a) compared to Fig. 7 is attributed to the increased multipath and Doppler effects present in dynamic channels, which make it more difficult to eliminate the channel effects.

Fig. 8(b) displays the outcomes obtained from our self-collected dataset under a dynamic channel. The performance of spectrogram with contrastive learning still outperforms other methods in both LOS and NLOS scenarios. It outperforms channel-independent spectrograms and spectrograms by over 3% and 10%, respectively. Even in the challenging NLOS environments, the accuracy of our approach exceeds 82%. The confusion matrix of the spectrogram with contrastive learning under LOS and NLOS is shown in Fig. 9. It can be observed that the DUTs are more likely to be misclassified as devices of the same type (DUT 1-5 are LoPy4, 6-10 are mbed 1261). It could be due to the fact that devices from the same manufacturer share similar physical features, making them more difficult to distinguish.

VI. Conclusion

In this paper, we presented a contrastive learning-based RFFI system which is robust to channel variation. Specifically, the system mitigates the impact of the channel with data augmentation and contrastive learning. We tested our system on two different datasets, including a public LoRa dataset and a self-collected dataset. Both datasets include wireless signals under various channel conditions, such as LOS, NLOS, static, and mobile scenarios. The results reveal that the contrastive learning-based RFFI system with spectrogram is robust to channel variations. Compared with IQ signal, spectrogram, and channel-independent spectrogram, spectrogram with contrastive learning shows higher accuracy in both static and dynamic channels. The performance in NLOS is slightly lower than in LOS conditions, but the accuracy of our method remains above 82% and is higher than the other methods.

Acknowledgement

The work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant ID EP/Y037197/1 and in part by the UK Royal Society Research Grants RGS\R1\231435. The work of J. Ma was also supported by the China Scholarship Council CSC under Grant ID 202006470007.

References

[1] 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.

[2] 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.

[3] Y. Li, Y. Ding, J. Zhang, G. Goussetis, and S. K. Podilchak, "Radio frequency fingerprinting exploiting non-linear memory effect," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 8, no. 4, pp. 1618–1631, 2022.

[4] 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.

[5] 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.

[6] G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. Cavallaro, "Towards length-versatile and noise-robust radio frequency fingerprint identification," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2355–2367, 2023.

[7] 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.

[8] W. Wang and L. Gan, "Radio frequency fingerprinting improved by statistical noise reduction," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 8, no. 3, pp. 1444–1452, 2022.

[9] G. Shen, J. Zhang, A. Marshall, and J. R. Cavallaro, "Towards scalable and channel-robust radio frequency fingerprint identification for LoRa," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 774–787, 2022.

[10] J. He, S. Huang, Z. Yang, K. Yu, H. Huan, and Z. Feng, "Channel-agnostic radio frequency fingerprint identification using spectral quotient constellation errors," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 158–170, 2023.

[11] Y. Xing, A. Hu, J. Zhang, L. Peng, and X. Wang, "Design of a channel robust radio frequency fingerprint identification scheme," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6946–6959, 2022.

[12] J. He, S. Huang, Y. Chen, S. Chang, Y. Zhang, and Z. Feng, "Radio frequency fingerprint identification for ofdm system considering unknown multipath fading channel," *IEEE Trans. on Cogn. Commun. Netw.*, 2024.

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2020, pp. 1597–1607.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, June 2015.

[15] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[16] X. Zha, T. Li, Z. Qiu, and F. Li, "Cross-receiver radio frequency fingerprint identification based on contrastive learning and subdomain adaptation," *IEEE Signal Process. Lett.*, vol. 30, pp. 70–74, 2023.

[17] Z. Ren, P. Ren, D. Xu, and T. Zhang, "Noise-tolerant radio frequency fingerprinting with data augmentation and contrastive learning," in *Proc. IEEE Wireless Commun. and Netw. Conf. (WCNC)*, Glasgow, UK, March 2023, pp. 1–6.

[18] 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.

[19] M. A. Ben Temim, G. Ferré, B. Laporte-Fauret, D. Dallet, B. Minger, and L. Fuché, "An enhanced receiver to decode superposed lora-like signals," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7419–7431, 2020.

[20] C. Li, H. Guo, S. Tong, X. Zeng, Z. Cao, M. Zhang, Q. Yan, L. Xiao, J. Wang, and Y. Liu, "Nelora: Towards ultra-low snr lora communication with neural-enhanced demodulation," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, New York, NY, USA, 2021, p. 56–68.