Federated Radio Frequency Fingerprint Identification Powered by Unsupervised Contrastive Learning

Guanxiong Shen, *Member, IEEE*, Junqing Zhang, *Member, IEEE*, Xuyu Wang, *Member, IEEE*, and Shiwen Mao, *Fellow, IEEE*

***Abstract*— Radio frequency fingerprint identification (RFFI) is a promising physical layer authentication technique that utilizes the unique impairments within the analog front-end of transmitters as distinct identifiers. State-of-the-art RFFI systems are frequently powered by deep learning, which requires extensive training data to ensure satisfactory performance. However, current RFFI studies suffer from a severe lack of training data, which poses challenges in achieving high identification accuracy. In this paper, we propose a federated RFFI system that is particularly suitable for Internet of Things (IoT) networks, which holds a high potential to address the data scarcity challenge in RFFI development. Specifically, all the receivers in an IoT network can pre-train a deep learning-driven feature extractor in a federated and unsupervised manner. Subsequently, a new client can perform fine-tuning on the basis of the pre-trained feature extractor to activate its RFFI functionality. Extensive experimental evaluation was carried out, involving 60 commercial off-the-shelf (COTS) LoRa transmitters and six software-defined radio (SDR) receivers. The experimental results demonstrate that the federated RFFI protocol can effectively improve the identification accuracy from 63% to 95%, and is robust to receiver hardware and location variations.**

***Index Terms*— Device authentication, radio frequency fingerprint, Internet of Things, LoRa, deep learning, federated learning.**

I. Introduction

THE number of Internet of Things (IoT) devices has increased significantly over the past decade. Numerous techniques are emerging and contributing to the establishment of IoT networks, such as LoRa, Narrowband IoT (NB-IoT), ZigBee, Bluetooth low energy (BLE), etc. Authentication is a critical factor in ensuring the security of IoT networks. Unreliable authentication mechanisms cause the risk of exposing sensitive data to unauthorized users and can even lead to disruption of the entire IoT network. Most authentication solutions rely on software identifiers, e.g., MAC addresses, which are vulnerable to malicious modification and can result in spoofing attacks. Existing authentication mechanisms rely on cryptographic algorithms, but securely storing and managing keys remains a significant challenge [1].

Radio frequency fingerprint identification (RFFI) stands as a promising non-cryptographic device authentication technique, which utilizes the hardware impairments of a wireless transmitter as the unique identifier. More specifically, the emitted wireless signals are distorted by the RF front-end impairments such as oscillator frequency offset, power amplifier nonlinearity, in-phase/quadrature (I/Q) imbalance [2], [3]. The receiver can analyze the captured signals and extract their physical layer features as a unique identifier. The RFFI is particularly suitable for securing IoT networks because the low-cost analog components in the IoT transmitter chain may have discriminative impairments, which guarantee high identification performance. Furthermore, the RFFI system operates completely on the receiver side, freeing up the transmitter's computing resources and reducing power consumption. Hence, the RFFI technique has been studied to secure various wireless systems such as LoRa [4], [5], ZigBee [6], LTE [7], [8], WiFi [9], and satellite [10], [11].

State-of-the-art (SOTA) RFFI systems typically employ deep learning techniques to extract the subtle transmitter distortion because of their powerful feature extraction capability. In a deep learning-based RFFI system, a well-trained neural network (NN) is deployed at the receiver, taking in physical layer signals and predicting device identity. Specifically, NN models such as CNN [3], [4], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], long short-term memory (LSTM) network [14], [17], [24], [32], multiple layer perception (MLP) [4], [14], [17], and the transformer [33] have been studied, which can effectively extract discriminative features after sufficient training. However, the NN is usually data-hungry; it should undergo extensive training on large datasets to achieve an excellent feature extraction performance.

Received 3 April 2024; revised 19 August 2024; accepted 14 September 2024. Date of publication 27 September 2024; date of current version 7 October 2024. The work of Guanxiong Shen was supported in part by the National Natural Science Foundation of China under Grant 62401138. The work of Junqing Zhang was supported in part by U.K. Engineering and Physical Sciences Research Council (EPSRC) New Investigator Award under Grant EP/V027697/1 and in part by the Royal Society Research Grants under Grant RGS/R1/231435. The work of Shiwen Mao was supported in part by NSF under Grant CNS-2107190. The associate editor coordinating the review of this article and approving it for publication was Dr. Dusit Niyato. *(Corresponding author: Junqing Zhang.)*

Guanxiong Shen is with the School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: gxshen@seu.edu.cn).

Junqing Zhang is with the Department of Electrical Engineering and Electronics, University of Liverpool, L69 3GJ Liverpool, U.K. (e-mail: junqing.zhang@liverpool.ac.uk).

Xuyu Wang is with the Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL 33199 USA (e-mail: xuywang@fiu.edu).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

Digital Object Identifier 10.1109/TIFS.2024.3469820

The shortage of training data sources greatly constrains the performance of deep learning-powered RFFI systems. The RFFI community suffers from a severe lack of high-quality, large-scale datasets when compared to other fields such as face recognition. The number of categories in RFFI datasets typically ranges from tens to hundreds, whereas face recognition datasets often consist of thousands of categories. For example, the public LoRa-RFFI dataset contains 60 LoRa transmitters [34], while the labeled faces in the wild (LFW), a face recognition dataset, contains images of 5,749 people [35]. In fact, the data sources should not be a bottleneck in RFFI because an IoT network typically contains numerous distributed edges and end nodes. However, it would be impractical to request edge nodes, i.e., receivers, to upload their raw signals to a central server because of the unaffordable communication cost. It is, therefore, necessary to explore a scheme that can efficiently utilize these distributed training data sources.

Federated learning is a promising technique that can utilize these distributed data sources for RFFI. It allows every individual receiver, i.e., gateway or access point, in an IoT network to contribute to the NN training process without the need to upload the raw signals to a central server. In other words, federated learning leverages decentralized data sources, facilitating collaborative model training among multiple clients while keeping data stored locally [36]. In this study, the term 'client' refers to an individual receiver, gateway, or access point in an IoT network. Each client connects to a number of IoT end nodes, i.e., transmitters to be identified. The concept of federated learning has been applied in recent RFFI research [9], [25], [37]. For instance, Piva et al. employed a USRP software-defined radio (SDR) to collect signals from 200 radio frequency identification (RFID) tags. They collected three datasets at different distances: 20 cm, 50 cm, and 100 cm, each representing an individual client [25]. Shi et al. aimed to identify up to 100 WiFi 802.11b devices. They conducted several sets of experiments using a spectrum analyzer, and the number of clients was up to ten. The authors in [37] attempted to identify four USRP X310 transmitters. The training dataset was equally divided into 100 subsets to simulate 100 clients.

Although previous studies have demonstrated the feasibility of applying federated learning to RFFI, none of them have considered the practical deployment challenges in a distributed IoT network. In particular, previous federated RFFI systems lack *trainability, scalability, and NN security*. Regarding *trainability*, federated training cannot be performed when the clients are connected to different numbers and groups of transmitters. This is because these clients will have different NN architectures when connected to different numbers of transmitters, making the model aggregation algorithm infeasible, e.g., FedAvg [36]. Concerning *scalability*, a trained NN cannot be used by new clients that are absent during training. When new clients are registered in the IoT network, the federated training has to be performed again, which is inefficient and time-consuming. In terms of *NN security*, previous federated training methods require labeled training data for supervised learning. However, it is impossible to monitor all the clients to ensure that the labels are correct. Attackers can easily modify the MAC addresses as those of legitimate devices to launch label manipulation-based attacks, e.g., label poisoning backdoor attack [38]. Therefore, it is necessary to develop a federated RFFI protocol that overcomes the above limitations.

This paper designed a practical federated RFFI for improved performance by expanding the available training data sources. The proposed approach addresses the constraints in previous federated RFFI studies, which ensures trainability, scalability, and NN security. These features benefit from the unsupervised contrastive training scheme, as well as a three-stage protocol design. A federated LoRa-RFFI testbed is created which comprises 60 commercial-off-the-shelf (COTS) LoRa transmitters and six SDR receivers. It should be noted that the proposed federated RFFI protocol is applicable to any wireless technology, and LoRa serves as a case study in this paper. The contributions are summarized below:

- A federated RFFI protocol is proposed, which involves three stages: federated pre-training, client fine-tuning, and identification. This protocol is well-suited for IoT networks. Each IoT receiver can act as an individual client and contribute to federated pre-training, potentially resolving the data scarcity issue in RFFI. First, the clients perform federated pre-training in an unsupervised fashion and produce an NN-based feature extractor. After that, fine-tuning is carried out on clients to activate their RFFI functionality. More specifically, the feature extractor is connected to a client-specific classifier, and a few labelled signals are used for adjusting the NN parameters. Finally, the received signals can be fed into the fine-tuned classification NN and the device identity can be predicted.
- The proposed protocol guarantees trainability, scalability, and NN security, which are lacking in previous works. Regarding *trainability*, federated training remains feasible even if the clients are connected to various numbers and groups of transmitters. This is accomplished by training a feature extractor rather than a classification NN. In terms of *scalability*, the trained NN is applicable not only to existing clients but also to new clients that are absent during training. This benefits from the three-stage protocol design. More specifically, the fine-tuning stage guarantees excellent performance on individual clients. For *NN security*, the proposed federated RFFI protocol uses unlabeled training data, which makes the NN less vulnerable to label manipulation-based attacks and potential data leakage during the training process.
- Extensive experimental evaluation is carried out using a testbed consisting of 60 COTS LoRa transmitters and six SDR receivers. The 60 transmitters are divided into six groups and separately connected to the SDR receivers, emulating six individual clients. We intentionally ensure that each client connects to a varied number and groups of transmitters, and is equipped with a different SDR receiver. This setup is aligned with a practical, distributed IoT network and is suitable for evaluating federated RFFI protocols. The experimental results demonstrate that the identification accuracy can be improved from 63% to 95% by using federated learning.

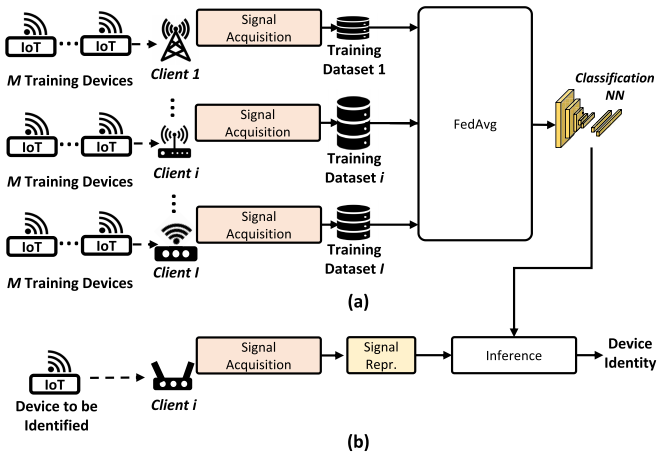


Fig. 1. Overview of a vanilla federated RFFI system. (a) Federated training using the FedAvg algorithm, which involves I clients, i.e., receivers in an IoT network. (b) Identification process at a specific client.

The dataset[1] and code[2] are available online.

The rest of this paper is structured as follows. Section II introduces the motivation for this paper. Section III outlines the details of the proposed federated RFFI protocol. Section IV takes LoRa as a case study to demonstrate how to design a practical federated RFFI system. The case implementation is experimentally evaluated in Section V, and Section VI finally concludes this paper.

II. Motivation

This section summarizes the limitations of previous federated RFFI protocols, followed by a detailed explanation of the motivation for this work.

A. Vanilla Federated RFFI Protocol

This subsection summarizes the federated RFFI systems proposed in previous studies [9], [25], [37]. In contrast to conventional distributed learning methodologies, federated learning does not require the client to upload the training data, i.e., IQ samples, to a central server. This avoids the resource-intensive data uploading process, making it a more optimal approach for RFFI systems.

An overview of a vanilla[3] federated RFFI system is illustrated in Fig. 1, which consists of two stages: federated training and identification. The federated training stage involves I training clients, each connected to M transmitters. Note that each client represents an individual receiver, e.g., gateway, or access point, in an IoT network. The I clients independently capture labeled signals from the transmitters and store them in a local dataset. After that, the well-known FedAvg algorithm is leveraged to train a classification NN in a federated and supervised manner, details of which can be found in [36]. In simple terms, the receivers individually train their local NNs and regularly send them to a cloud server for aggregation. The aggregated classification NN is then dispatched to the training clients for updating the local models. This procedure is repeated over multiple communication rounds to create a well-performing classification NN. In the subsequent identification stage, the received signal is input into the classification NN, which then provides a prediction regarding the identity of the device.

B. Limitations of the Vanilla Federated RFFI Protocol

The vanilla federated RFFI protocol has several major limitations that prevent its application in practical IoT networks:

- **Trainability:** The vanilla federated training cannot be executed when the receivers/clients connect to different numbers of transmitters, while this is actually the most common scenario in practical IoT networks. More specifically, vanilla federated learning relies on model averaging across multiple classification NNs. When clients are connected to different numbers of clients, the local NN architectures are inconsistent due to the varying number of neurons in the final softmax layer. This model heterogeneity restricts effective parameter fusion, making the FedAvg algorithm infeasible.
- **Scalability:** A classification NN trained with the vanilla federated RFFI protocol cannot be utilized by new clients, which compromises its scalability. The vanilla RFFI protocol assumes a closed-set setting, and the trained NN can only identify transmitters included in the training set. However, a new client often connects to a new group of transmitters absent in the training process, making classification NN produced by federated training inapplicable. Moreover, the new client is typically equipped with a different type of receiver. The different receiver hardware characteristics potentially compromise the identification performance as well.
- **NN Security:** The vanilla federated RFFI protocol is based on supervised learning. However, it is impractical to monitor all the clients in an IoT network to ensure the training labels are correct and reliable. In fact, an attacker can easily manipulate the software identifier, e.g., MAC address, of wireless transmitters, and then launch label-flipping or backdoor attacks by poisoning the training labels. Given that attackers can easily change MAC addresses as those of legitimate devices, label manipulation-based attacks pose a significant threat to the security of RFFI systems.

The limitations discussed above motivate the design of a federated RFFI protocol that ensures trainability, scalability, and NN security. Specifically, the protocol should be able to efficiently involve all IoT nodes/edges to perform federated training, and the unsupervised method should be used to prevent label-poisoning attacks from potentially malicious clients. In addition, the trained NN should be able to operate on any new client to improve the practicality.

III. Federated RFFI Protocol

A. System Overview

The proposed federated learning-enhanced RFFI protocol is illustrated in Fig. 2. It consists of three stages, namely federated pre-training, client fine-tuning, and identification.

[1] https://ieee-dataport.org/documents/lorafederatedrffidataset

[2] https://github.com/gxhen/federatedRFFI

[3] In the field of deep learning, the term 'vanilla' is often used to describe an unaltered or standard version of a technique.

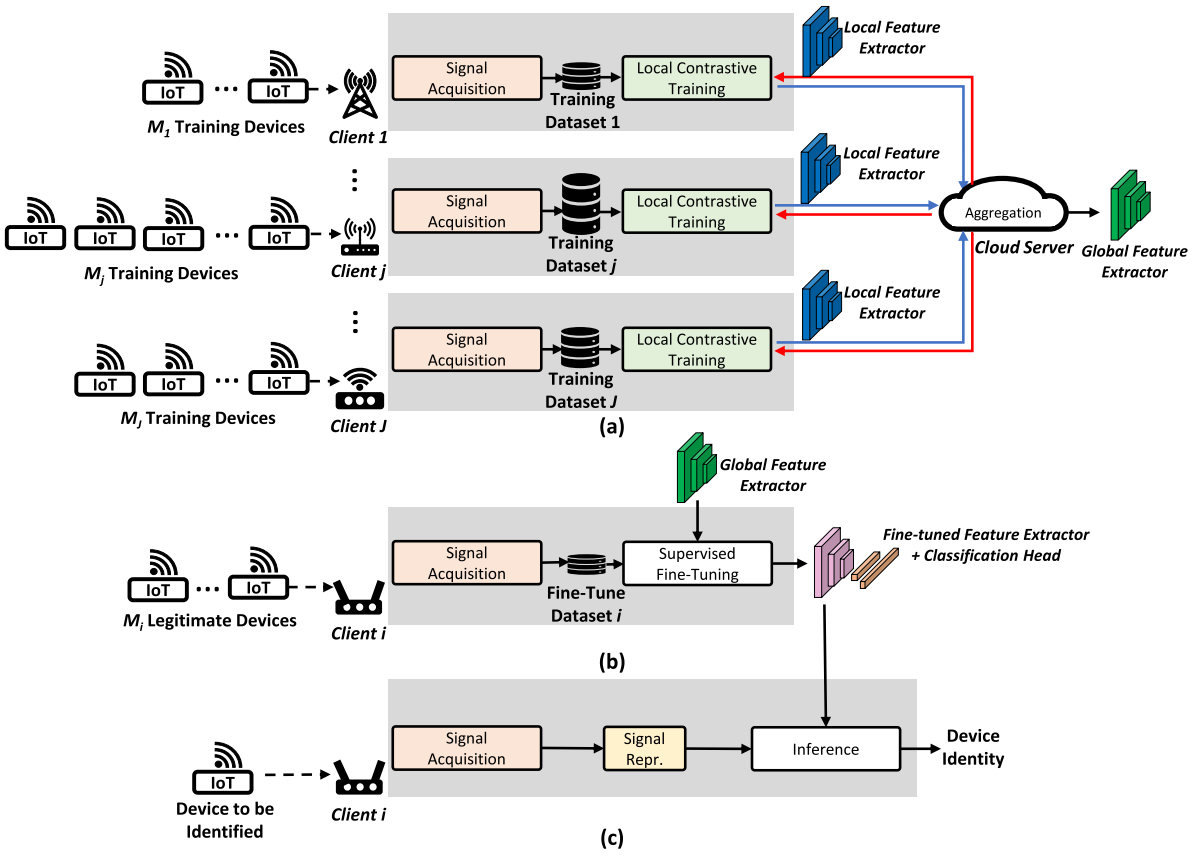


Fig. 2. Overview of the proposed federated RFFI system. (a) Federated pre-training. (b) Client fine-tuning. (c) Identification.

- **Federated Pre-Training** produces an NN-based feature extractor in a federated and unsupervised fashion. More specifically, each client j acquires signals from M_j transmitters it serves and stores them in a training dataset. Note that the number of connected transmitters can be different for each receiver, i.e., $M_1 \neq \ldots \neq M_j \neq \ldots \neq M_J$. This is a more practical setup as each receiver may serve a different number of transmitters in an IoT network. After the signal acquisition is completed, all J clients conduct federated pre-training to produce a feature extractor. In short, each client iteratively trains a local feature extractor with an unsupervised approach, and uploads it to the cloud server after several epochs. The server aggregates all uploaded models into a single one and subsequently dispatches it to the clients. These steps are repeated multiple times until a feature extractor is produced. The input to the feature extractor is the signal representation converted from IQ samples, and the output is the extracted RF fingerprint. By training feature extractors instead of classification NNs, the model architecture can be made independent of the number of connected transmitters, thus solving the issue of model heterogeneity in federated RFFI studies.
- **Client Fine-Tuning** concatenates a pre-trained feature extractor to a classifier and precisely adjusts model weights to enhance the RFFI performance on a specific client. First, we connect a classifier, e.g., several linear layers, to the pre-trained feature extractor to construct a classification NN. Then few labeled signals are collected from M_i transmitters connected to client i to fine-tune its weights. After fine-tuning, the classification NN is capable of predicting the identity of a device with high accuracy.
- **Identification** describes the process of predicting the identity of a device. The receiver i first acquires a signal from the device to be identified. Next, the signal is converted into the appointed signal representation, which is then fed into the fine-tuned classification NN to obtain the predicted device identity.

B. Client Operations in Federated Pre-Training

Any client in an IoT network can contribute to the pre-training process and can therefore implicitly increase the amount of available training data.

1) Signal Acquisition: The client executes signal acquisition algorithms to capture the wireless transmission, i.e., IQ samples, and processes them to meet the RFFI requirements. This includes synchronization, preamble extraction, frequency offset compensation, and power normalization.

- **Synchronization** locates the precise beginning of the received packet.
- **Preamble extraction** refers to that only the preamble part is reserved for the RFFI task. This prevents the NN from learning protocol-specific or payload information.
- **Frequency offset compensation** is employed to calibrate the frequency offset feature of the received signal. Previous studies have demonstrated that oscillator frequency is sensitive to temperature variations [14], [15], [39], thus

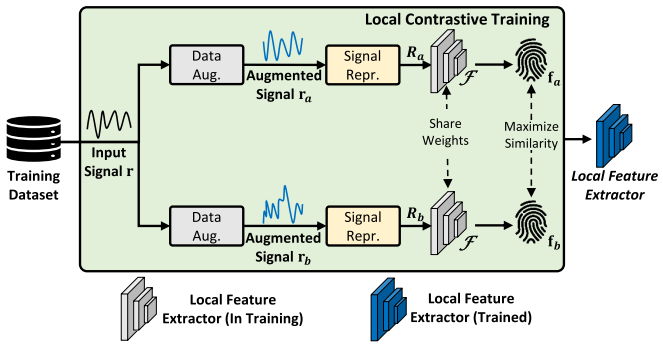


Fig. 3. Local unsupervised contrastive training.

calibrating it can significantly improve system stability. In addition, the frequency offset feature is mimicable, and compensating it can enhance the resilience of the system against spoofing attacks [19], [26], [40].

- **Power normalization** prevents the NN from being interfered with by the variations in received signal strength when making predictions.

As shown in Fig. 2(a), the federated training involves J clients, each of which is connected to M_j wireless transmitters. The receivers execute signal acquisition programs, storing the processed IQ samples into training datasets, given by

$$\mathcal{T}^j = \left\{\mathbf{r}_k^j\right\}_{k=1}^{K^j}, \tag{1}$$

where $\mathcal{T}^j$ denotes the training dataset collected at receiver j and $\mathbf{r}^j$ denotes the received IQ samples.[4] K^j refers to the number of signals in dataset $\mathcal{T}^j$. Unlike previous supervised RFFI protocols, the training dataset does not contain label information in this study. It is highlighted that the $\mathcal{T}^j$ should include as many channel conditions as possible, as this can enhance the transferability and generalization ability of the trained neural network.

2) Local Contrastive Training: After collecting sufficient signals, each receiver j independently trains its local feature extractor $\mathcal{F}^j$ in an unsupervised fashion. More specifically, we developed an unsupervised contrastive learning algorithm in the RFFI context, which has shown outstanding performance in recent studies and is even comparable to supervised approaches [41], [42], [43], [44].

In a nutshell, contrastive training learns feature representation by maximizing the similarity between two augmentation results of the same wireless signal. Specifically, as shown in Fig. 3, we use a channel simulator to augment the input wireless signal $\mathbf{r}$ twice, generating two independent augmented signals, denoted as $\mathbf{r}_a$ and $\mathbf{r}_b$, respectively.[5] The signals are then converted into appropriate signal representations, $\mathbf{R}_a$ and $\mathbf{R}_b$, to assist the RF fingerprint extraction process. Subsequently, the converted signal representations are fed into two shared-weight feature extractors to obtain the RF fingerprints $\mathbf{f}_a$ and $\mathbf{f}_b$. Finally, a contrastive loss $\mathcal{L}_{cntr}$ is defined to guide the parameter-updating process of the feature extractor. The overarching objective function for local unsupervised training is mathematically given by

$$\theta_{\mathcal{F}}^j = \arg\min_{\theta_{\mathcal{F}}^j} \sum \mathcal{L}_{cntr}. \tag{2}$$

Various optimizers can be employed to achieve the objective in (2), such as stochastic gradient descent (SGD), RMSprop, Adam, etc. Designers need to select an appropriate optimizer based on the experimental performance.

Contrastive learning is a self-supervised scheme, which is a sub-category of unsupervised learning. Self-supervised learning has been successfully applied in classification and anomaly detection tasks. As RFFI systems target anomaly detection or classification functions as well [5], self-supervised learning is a suitable approach for training RFFI models. Recent research has applied self-supervised disentangled representation to construct RFFI systems [45], which leverages its implicit data augmentation module to mitigate the data shortage problem.

a) Data augmentation: The data augmentation module, i.e., channel simulator, enables the NN to apply various channel effects to the input signal and learn robust and invariant feature representation. There are many channel simulators available, such as the tapped delay line (TDL) model, the clustered delay line (CDL) model, and the ray tracing model. The designer should model the channel according to the wireless protocol and operating environment, taking into account factors such as signal bandwidth, frequency band, and level of Doppler and multipath effects. Note that the two branches in Fig. 3 share the same channel simulator, but are augmented with different channel impulse responses.

It is worth noting that a well-designed data augmentation module can improve the quality of the learned feature representation [44], i.e., enhanced model performance. Previous studies have demonstrated that data augmentation represents an effective strategy for combating channel effects, as the model transferability is significantly enhanced by augmenting the quantity and diversity of training data [16], [18], [21].

b) Signal representation: The time-domain IQ samples are often converted to other signal representations to be used as NN inputs. This process can enhance identification performance by increasing the efficiency of feature extraction. Various signal representations have been utilized in previous works, such as spectrogram [14], [15], differential constellation trace figure [46], frequency spectrum [47], [48], [49], etc. The design of the signal representation should take into account the physical layer characteristics of the target wireless protocol.

c) Local feature extractor: The converted signal representation is input into a local feature extractor $\mathcal{F}^j$, which is mathematically written as

$$\mathbf{f} = \mathcal{F}^j(\mathbf{R}, \theta_{\mathcal{F}}^j), \tag{3}$$

where $\theta_{\mathcal{F}}^j$ denotes the parameters of the feature extractor. The output $\mathbf{f}$ is a high-dimensional feature representation, i.e., vector, that can be considered as the extracted RF fingerprint.

The architecture of the designed feature extractor depends on the characteristics of input signal representation $\mathbf{R}$.

[4]It is assumed that no collisions occur during the data collection process, i.e., $\mathbf{r}^j$ does not interfere with each other.

[5]Note that $\mathbf{r}_a$ and $\mathbf{r}_b$ are augmented by two independent instances, i.e., channel impulse response, generated from the same channel simulator.

For example, complex-valued NN is suitable for complex time-domain IQ samples [50], whereas CNN is efficient at processing time-frequency domain spectrograms because of their image-like structure [14].

d) Contrastive loss function: The contrastive loss $\mathcal{L}_{cntr}$ encourages the NN to pull positive pairs ($\mathbf{f}_a$ and $\mathbf{f}_b$) in a mini-batch as close as possible while pushing negative pairs ($\mathbf{f}_a$ and feature vectors other than $\mathbf{f}_b$) in a mini-batch farther apart. In short words, the contrastive loss aims to minimize the feature difference between $\mathbf{f}_a$ and $\mathbf{f}_b$, since they are extracted from representations of the same signal. Available contrastive loss functions include the normalized temperature-scaled cross-entropy (NT-Xent) loss, normalized temperature-scaled logistic (NT-Logistic) loss, margin triplet loss, and other recent advances [42], [44].

C. Server Operations in Federated Pre-Training

After all clients have finished the local training process, the feature extractors $\mathcal{F}^j$ will be uploaded to the cloud server and then aggregated into a single global model $\mathcal{F}^{global}$. The aggregation process is mathematically represented as

$$\theta_{\mathcal{F}}^{global} = \sum_{j=1}^{J} \eta \theta_{\mathcal{F}}^{j}, \tag{4}$$

where $\theta_{\mathcal{F}}^{global}$ denotes the parameters of the aggregated feature extractor $\mathcal{F}^{global}$, and $\eta = K^j / \sum_{j=1}^{J} K^j$ is a weighting factor calculated based on the size of the local training dataset $\mathcal{T}^j$. After the aggregation process is finished at the cloud server, $\mathcal{F}^{global}$ is dispatched to each of the J receivers. Subsequently, each receiver restarts the local contrastive training on the basis of the dispatched feature extractor $\mathcal{F}^{global}$.

The above uploading, aggregation, and dispatching steps are iterated until the prescribed number of communication rounds is reached. The federated pre-training finally produces an aggregated feature extractor, $\mathcal{F}_{global}$, which is used for the subsequent client fine-tuning and identification procedures. More specifically, pre-training offers a set of initial parameters for the fine-tuning process, which can considerably enhance system performance since fine-tuning a pre-trained model is typically more efficient than training a model from scratch, i.e., training starts with randomly initialized parameters.

D. Client Fine-Tuning and Identification

The client can activate the RFFI functionality by fine-tuning using the pre-trained feature extractor $\mathcal{F}_{global}$ as a foundation model. In other words, RFFI is a downstream task of federated pre-training. First, client i concatenates a softmax-enabled classification head $\mathcal{C}^i$ to the feature extractor $\mathcal{F}_{global}$ to create a classification NN. Note that the number of neurons in the last layer of the classification head equals M_i, i.e., the number of transmitters connected to client i. The concatenated classification NN is given by

$$\mathcal{N}^i(\cdot, \theta_{\mathcal{N}}^i) = \mathcal{F}^{global}(\cdot, \theta_{\mathcal{F}}^{global}) \circ \mathcal{C}^i(\cdot, \theta_{\mathcal{C}}^i), \tag{5}$$

where the symbol $\circ$ represents function composition and is used to describe multiple models applied in sequence. Note that $\theta_{\mathcal{N}}^i$ is a union of $\theta_{\mathcal{F}}^{global}$ and $\theta_{\mathcal{C}}^i$, i.e., $\theta_{\mathcal{N}}^i = \theta_{\mathcal{F}}^{global} \cup \theta_{\mathcal{C}}^i$. In the forward propagation process, the signal representation $\mathbf{R}$ is fed into the classification NN $\mathcal{N}^i$ and a prediction $\hat{\mathbf{p}}$ is returned

$$\hat{\mathbf{p}} = \mathcal{N}^i(\mathbf{R}, \theta_{\mathcal{N}}^i), \tag{6}$$

where $\hat{\mathbf{p}} = \{p_1, \cdots, p_m, \cdots, p_{M_i}\}$ is a vector containing M_i elements, and p_m represents the probability that the signal is sent from transmitter m.

After creating the classification NN at client i, we will collect a small dataset for fine-tuning, given by

$$\mathcal{D}^i = \left\{ (\mathbf{r}, \mathbf{p})_k^i \right\}_{k=1}^{K^i}, \tag{7}$$

where $\mathcal{D}^i$ denotes the fine-tuning dataset at client i and K^i is the number of fine-tuning samples. $(\mathbf{r}, \mathbf{p})_k^i$ is the k-th signal-label pair collected by receiver i, where $\mathbf{p}$ is the ground-truth label for signal $\mathbf{r}$.

With the classification NN $\mathcal{N}^i$ and dataset $\mathcal{D}^i$, fine-tuning can be subsequently performed in a supervised manner. The objective function for fine-tuning is expressed as

$$\theta_{\mathcal{N}}^i = \arg\min_{\theta_{\mathcal{N}}^i} \sum \mathcal{L}_{ce}(\hat{\mathbf{p}}, \mathbf{p}), \tag{8}$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss that is often used in classification tasks. Optimization algorithms such as Adam and SGD can be employed to achieve this objective. The fine-tuning process also utilizes the data augmentation technique introduced in Section III-B2. More specifically, the signals in a mini-batch are separately fed into a channel simulator to increase the NN's robustness to channel variations and noise.

Note that unlike pre-training where all clients can participate, fine-tuning must be carried out in a controlled environment to ensure that labels in $\mathcal{D}^i$ are correct.

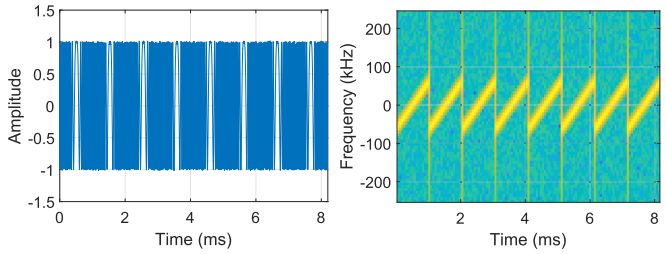
After client i finished fine-tuning, the classification NN can be used to identify a wireless transmitter. As demonstrated in Fig. 2(c), the device to be identified sends a wireless signal to the receiver. The signal is captured, converted to the signal representation, and then input into the classification NN $\mathcal{N}^i$ to acquire the prediction $\hat{\mathbf{p}}$. The device corresponding to the highest probability in $\hat{\mathbf{p}}$ is the final identity predicted by the RFFI system.

IV. Case Study: Federated LoRa-RFFI Protocol

This section takes the LoRa technology as a case study to demonstrate how to design a federated RFFI system. Note that the protocol proposed in Section III can be applied to any wireless technology with appropriate adjustment.

A. LoRa PHY Primer

LoRa is Semtech's proprietary wireless modulation technology. It is derived from the chirp spread spectrum (CSS) technique and encodes information on frequency-varying linear chirps, as shown in Fig. 4(a). As typical non-stationary signals with changing frequency components, LoRa signals are often visualized in the time-frequency domain in the form of spectrograms. Fig. 4(b) presents the spectrogram



(a) (b)

Fig. 4. LoRa's preamble part consists of eight repeated up-chirps. (a) Time-domain waveform (the in-phase branch). (b) Time-frequency domain spectrogram.

of the preamble part in a LoRa packet, which consists of eight repeating unmodulated LoRa symbols. Previous RFFI studies reveal that converting time domain LoRa signals into time-frequency domain representations can improve system performance [14], [15], [16].

B. Client Operations in Federated Pre-Training for LoRa-RFFI

1) LoRa Signal Acquisition: As LoRa is a proprietary protocol owned by Semtech Corporation and there exists no official document outlining the details of its physical layer, most researchers use open-source or self-designed algorithms to capture LoRa transmissions. The LoRa signal acquisition algorithms introduced in [14] are utilized in this work, which includes steps of synchronization, preamble extraction, frequency offset compensation, and power normalization. According to the experimental configurations, each LoRa signal is a vector consisting of 4,096 complex numbers, and its in-phase part is shown in Fig. 4(a).

2) Local Contrastive Training for LoRa-RFFI: As illustrated in Section III-B2 and Fig. 2, each client j trains a local feature extractor in an unsupervised and contrastive manner. Specifically, we augment a time-domain LoRa signal with two independent TDL channel simulators, and then convert the augmentation results into channel-independent spectrograms, respectively. Subsequently, we use the feature extractor to obtain the RF fingerprints of the two augmented results and then compute loss.

a) Data augmentation for LoRa-RFFI: The TDL channel model is utilized in this work. The exponential power delay profile (PDP) is used to model the multipath effect, and the Doppler effect is depicted by the Jakes model. Synthetic Gaussian noise is also added to the signal. The simulation parameter is randomized for each specific LoRa signal. The ranges for RMS delay spread, Doppler frequency, and signal-to-noise ratio (SNR) are [5, 300] ns, [0, 5] Hz, and [0, 80] dB, respectively [16]. The TDL channel simulator is implemented in Python with PyPhysim library.[6]

b) Signal representation for LoRa: The channel-independent spectrogram proposed in [16] serves as the signal representation in this work. It is a time-frequency domain signal representation and is suitable for analyzing the frequency-varying characteristics of the LoRa physical layer signal. The channel-independent spectrogram is generated by dividing neighboring frames in a spectrogram, leveraging the fact that the wireless channel does not change drastically over a short time. Details of how to derive a channel-independent spectrogram can be found in [16]. The window length and overlap are set to 64 and 32 respectively when performing the short-time Fourier transform (STFT). The upper and lower sections of the generated channel-independent spectrogram are cropped by 30%, as these regions are out-of-band noise and lack valuable information. The size of the generated channel-independent spectrogram is 26×126. Note that utilizing channel-independent spectrograms as the signal representation has the potential to address the issue of device heterogeneity in the context of federated training. Specifically, the channel effects are significantly mitigated, thereby reducing the differences in data distribution between clients and facilitating the federated training process.

[6]https://pyphysim.readthedocs.io/en/latest/

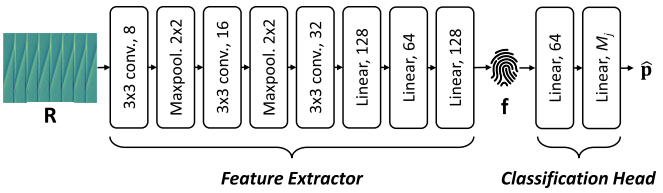


Fig. 5. Architecture of the classification NN, consisting of a feature extractor and a classification head.

c) Local feature extractor for LoRa-RFFI: A CNN is utilized to extract the RF fingerprints, which is depicted in Fig. 5. It consists of three convolutional layers of 8, 16, and 32 3×3 kernels, respectively, two 2×2 maxpooling layers, and three linear layers of 128, 64, and 128 neurons respectively. The outputs of convolutional and linear layers are activated by the ReLU function. The output is a vector containing 128 elements, i.e., the RF fingerprint.

The main reason for using the CNN architecture is that the NN input is a channel-independent spectrogram, which is an image-like signal representation. Previous work has shown that CNNs can efficiently process such type of data [14].

d) Contrastive loss function: The NT-Xent loss is employed in the LoRa-RFFI case study, which has been experimentally shown to achieve superior performance in other machine learning tasks [42]. The loss for a positive pair is mathematically expressed as

$$\ell_{cntr}(\mathbf{f}_a, \mathbf{f}_b) = -\log \frac{e^{sim(\mathbf{f}_a,\mathbf{f}_b)/\lambda}}{\sum_{k=1}^{2\mathcal{B}} \mathbb{I}_{[k\neq a]} e^{sim(\mathbf{f}_a,\mathbf{f}_k)/\lambda}}, \tag{9}$$

where $\mathbb{I}_{[k\neq a]} \in \{0, 1\}$ is an indicator function that equals to one when $k \neq a$. $sim(\cdot,\cdot)$ denotes the similarity function between two representations. λ and $\mathcal{B}$ are the temperature parameter and batch size, respectively. The final loss $\mathcal{L}_{cntr}$ is computed across all the $2\mathcal{B}$ positive pairs in a mini-batch.

In our implementation, the temperature parameter λ and $\mathcal{B}$ are set to 0.05 and 128, respectively, and the function $sim(\cdot,\cdot)$ is defined as the cosine similarity, expressed as

$$sim(\mathbf{f}_a, \mathbf{f}_b) = \frac{\mathbf{f}_a \cdot \mathbf{f}_b}{\|\mathbf{f}_a\| \, \|\mathbf{f}_b\|}, \tag{10}$$

where the operator $\cdot$ denotes the dot product and $\|\cdot\|$ returns the norm of a vector.

The NN is implemented with the PyTorch framework. Each client j randomly splits 10% signals in $\mathcal{T}^j$ for validation and the rest for pre-training. The Adam optimizer is utilized. Validation loss is monitored during local unsupervised training to adjust the learning rate and determine when to stop training. More specifically, the initial learning rate is set to 0.0003, and it decreases by a factor of 0.1 whenever the validation loss remains constant or increases for ten consecutive epochs. The local unsupervised training process stops when the validation loss remains unchanged for 20 consecutive epochs.

C. Server Operations in Federated Pre-Training for LoRa-RFFI

After local contrastive training is finished, all clients upload their local feature extractors to a cloud server, which performs the aggregation process according to (4). This step produces a global feature extractor. Subsequently, the cloud server dispatches the aggregated global feature extractor to all clients so that they can restart the local contrastive training process described in Section IV-B2. The above uploading, aggregation, and dispatching steps are repeated a few rounds before stopping. The effect of the number of communication rounds will be evaluated in Section V-C.

D. Client Fine-Tuning and Identification

After federated pre-training, the global feature extractor can be connected to a classification head to create a classification NN. In our setup, the classification head for client i comprises two linear layers: one with 64 and another with M^i neurons, which is shown in Fig. 5. The outputs of the first and second layers are activated by ReLU and Softmax functions, respectively.

Subsequently, a small amount of labelled data is collected to fine-tune the classification NN. The cross-entropy loss is employed and the TDL channel simulator is also used for augmentation. The fine-tuning process shares almost identical configurations with pre-training, such as the optimizer and learning rate schedule, except for the initial learning rate, which is set to 0.001 during fine-tuning.

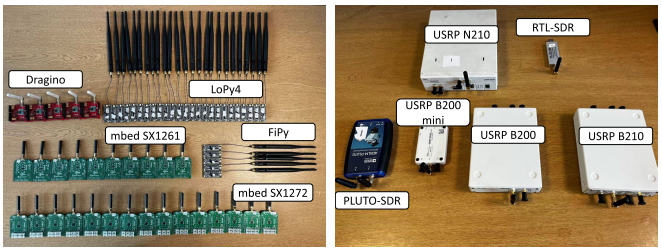
In the identification stage, the device to be identified sends a wireless signal to the receiver. The captured signal is fed into the fine-tuned classification NN and a prediction on the device identity is returned.

V. Experimental Evaluation

A. Experimental Setup

1) Hardware: The testbed comprises six SDR receivers serving as clients, namely PLUTO-SDR, B200-mini, B200, B210, N210 and RTL-SDR. They separately connect to 60 COTS LoRa transmitters, including 25 LoPy4 boards, 15 mbed 1272 LoRa shields, ten mbed LoRa 1261 shields, five FiPy boards, and five Dragino LoRa shields. The LoRa transmitters emit wireless signals at 868.1 MHz. The bandwidth is set to 125 kHz and the spreading factor is seven. The receiver sampling rate is configured as 500 kHz.

The system is evaluated using a PC equipped with an Intel i5 central processing unit (CPU) and an NVIDIA GeForce RTX 4060 graphic processing unit (GPU). Specifically, the GPU handles the training and inference tasks for the NN, while the CPU manages other processes like data augmentation and signal representation conversion.



(a) (b)

Fig. 6. Experimental equipment. (a) 60 COTS LoRa transmitters. (b) Six SDRs act as LoRa Receivers.

TABLE I
Client Settings

Name	Receiver	Connected Tx	# of Tx
Client 1	PLUTO-SDR	LoPy4 1-5	5
Client 2	USRP B200 mini	LoPy4 6-10	10
		mbed 1272 shield 1-5	
Client 3	USRP B200	LoPy4 11-15	15
		mbed 1272 shield 6-10	
		mbed 1261 shield 1-5	
Client 4	USRP B210	LoPy4 16-20	20
		mbed 1272 shield 11-15	
		mbed 1261 shield 6-10	
		FiPy 1-5	
Client 5	USRP N210	LoPy4 21-25	10
		Dragino shield 1-5	
Client 6	RTL-SDR	LoPy4 21-25	10
		Dragino shield 1-5	

2) Client Settings: As introduced in the previous subsection, there are six SDR receivers emulating six individual clients, which are detailed in Table I. They belong to different SDR models and separately connect to a subset of the 60 LoRa transmitters. Among them clients 1-4 are used for federated pre-training and clients 1-6 are used for evaluation.

This experimental setup is more realistic for IoT networks for the following reasons:

- **Clients 1-4 are connected to different groups of transmitters.** In practical IoT networks, the clients often provide services to different groups and numbers of transmitters, and the transmitters in each group are typically from different manufacturers.
- **Clients 1-6 are of different SDR models.** The receivers in an IoT network are likely to be from different manufacturers. Therefore, we deliberately use six different SDR platforms as clients.
- **Clients 5-6 do not participate in pre-training.** It is preferred that the federated RFFI protocol can be applied to a new client that is absent during training. Therefore, we exclude clients 5-6 during pre-training and only use

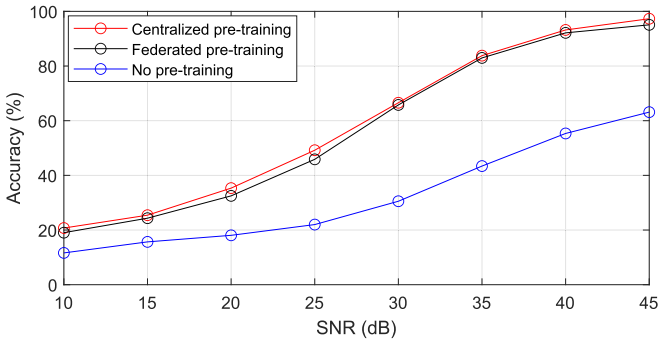


Fig. 7. Performance gains from federated pre-training.

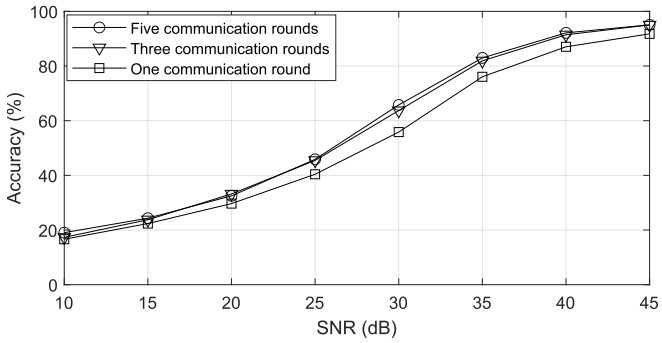


Fig. 8. Effect of the number of communication rounds.

them for fine-tuning and evaluation. This allows us to investigate the identification performance on new clients.

B. Performance Gains Through Federated Pre-Training

Federated pre-training allows numerous clients to collaboratively train a feature extractor, which can effectively benefit the subsequent fine-tuning process. The pre-training, fine-tuning, and identification settings are listed below:

- **Pre-Training**: Clients 1-4 with PLUTO, B200 mini, B200, B210 receivers, respectively. 1,500 signal samples are collected from each transmitter-client pair. Three training strategies are investigated, i.e., centralized pre-training, federated pre-training, and no pre-training. Here, 'centralized pre-training' means that all clients upload their local datasets to the cloud server to form a large-scale dataset and then subsequently perform unsupervised contrastive training. Whereas, the 'no pre-training' strategy refers to skipping the pre-training step and fine-tuning directly from scratch.
- **Fine-Tuning**: Client 5 with a USRP N210 receiver. 200 signal samples are collected from each transmitter.
- **Identification**: Client 5 with a USRP N210 receiver, another 300 signal samples are collected from each transmitter. Synthetic Gaussian noise of various levels is added to the signals to simulate different SNR conditions.

Fig. 7 demonstrates the performance gains through federated pre-training. It can be observed that both centralized and federated pre-training strategies result in significantly higher performance compared to the 'no pre-training' strategy, with an improvement of up to 30%. Furthermore, the results indicate that centralized pre-training slightly outperforms federated pre-training by less than 4%. Despite this marginal increase in accuracy, the cost of requiring all clients to upload their raw signals to a cloud server outweighs the benefits. The excellent identification results also validate the trainability and NN security of the proposed protocol, since the clients are connected to different numbers of transmitters and the pre-training uses unlabeled data.

C. Effect of the Number of Communication Rounds

As introduced in Section III-C, the clients regularly communicate with the cloud server during federated pre-training for aggregation and dispatching. The evaluation settings are as follows:

- **Pre-Training**: Clients 1-4 with PLUTO, B200 mini, B200, B210 receivers, respectively. 1,500 signal samples are collected from each transmitter-client pair. The federated pre-training approach is used, and the aggregated feature extractor is saved after the first, third, and fifth communication rounds.
- **Fine-Tuning**: Client 5 with a USRP N210 receiver. 200 signal samples are collected from each transmitter.
- **Identification**: Client 5 with a USRP N210 receiver, another 300 signal samples are collected from each transmitter. Synthetic Gaussian noise of various levels is added to the signals to simulate different SNR conditions.

The effect of the number of communication rounds is shown in Fig. 8, which demonstrates that more communication rounds lead to improved identification performance. Specifically, the accuracy is enhanced by around 5% when the number of communication rounds increased from one to five. It is also noticed that the improvement becomes marginal after more than three rounds of communication. The results show that there is almost no difference between conducting three or five communication rounds, with an improvement of less than 1%. In our experimental setting, the federated pre-training was terminated after five communication rounds, with rounds 1, 2, 3, 4, and 5 requiring 489, 288, 277, 272, and 211 epochs, respectively. Each epoch costs around 80 seconds. It can be observed that the number of required epochs gradually decreases. In the early rounds, the model is far from the optimal parameter set and requires more epochs to make significant improvements. As the model approaches convergence in later rounds, fewer epochs are needed to refine it.

D. Effect of the Number of Fine-Tuning Signals

Fine-tuning requires correctly labeled signals from legitimate transmitters, which is carried out in controlled environments. Increasing the amount of signal samples improves identification performance but raises the cost of fine-tuning as well. The settings are as follows:

- **Pre-Training**: Clients 1-4 with PLUTO, B200 mini, B200, B210 receivers, respectively. 1,500 signal samples are collected from each transmitter-client pair. The federated pre-training approach is used.
- **Fine-Tuning**: Client 5 with a USRP N210 receiver. The fine-tuning process employs 20, 50, 100, and 200 labeled signal samples, respectively.

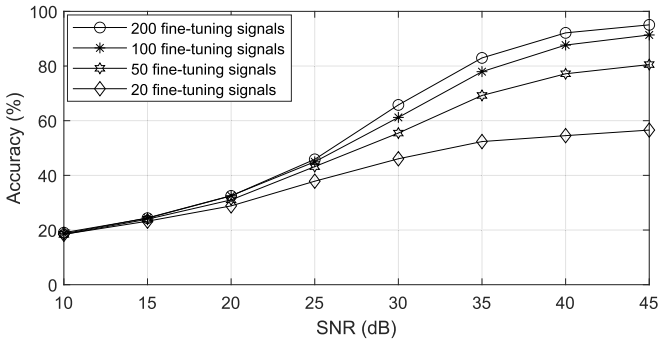


Fig. 9. Effect of the number of fine-tuning signals.

- **Identification**: Client 5 with a USRP N210 receiver, another 300 signal samples are collected from each transmitter. Synthetic Gaussian noise of various levels is added to the signals to simulate different SNR conditions.

The effect of the number of fine-tuning signals is depicted in Fig. 9, which indicates that the more fine-tuning signals, the higher the identification performance. The identification accuracy is acceptable when the number of fine-tuning signal samples approaches 100, i.e. 95.1% at 45 dB. Increasing the number to 200 leads to a further improvement of around 4%. Compared to previous methods that require huge amounts of data to train NNs from scratch, fine-tuning the pre-trained NN requires only 100 signals to achieve comparable performance, significantly reducing the cost of data collection during system deployment. In the experimental setting, the stop condition is reached after approximately 80 epochs, with each epoch requiring approximately three seconds.

E. Performance on Various Clients

The pre-trained feature extractor can be deployed on various clients to enable their RFFI functionality. Given the fact that receivers in an IoT network may come from different manufacturers, it is necessary to evaluate the performance on clients with different receivers to determine if the differences in receiver hardware affect identification accuracy. The evaluation configurations are as follows:

- **Pre-Training**: Clients 1-4 with PLUTO, B200 mini, B200, B210 receivers, respectively. 1,500 signal samples are collected from each transmitter-client pair. The federated pre-training approach is used.
- **Fine-Tuning**: Client 1-6 with six different SDR receivers, 200 signal samples are collected from each transmitter. Note that the fine-tuning signals are not a subset of training signals.
- **Identification**: Client 1-6 with six different SDR receivers, another 300 signal samples are collected from each transmitter.

Table II demonstrates that the RFFI system performs well on all six clients, with accuracy consistently above 90%. We found that the identification performance is excellent regardless of whether the receiver participated in pre-training or not. Despite the fact that clients 5 and 6, i.e., N210 and RTL-SDR, are absent during the pre-training process, the identification accuracy remains over 95%. The accuracy of Client 1 is comparatively lower than that of the other clients. This is presumed to be due to the fact that the transmitters connected to Client 1 are all of the same model, i.e., 5 LoPy4, and thus have bit-similar hardware characteristics. This is the most challenging situation for RFFI applications, However, despite this, the accuracy remains above 90%, which demonstrates an excellent identification capability. A special note is that USRP N210 and RTL-SDR are the most and least expensive of the six SDR receivers. This confirms that RFFI performance remains unaffected by the quality of the receiver hardware. The excellent performance on various clients validates the scalability of the federated RFFI protocol.

TABLE II
PERFORMANCE ON VARIOUS CLIENTS

	Involved During Pre-training				Not Involved	
Name	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6
SDR	PLUTO	B200 mini	B200	B210	N210	RTL
Acc.	91.27%	97.20%	99.04%	97.20%	97.53%	99.97%

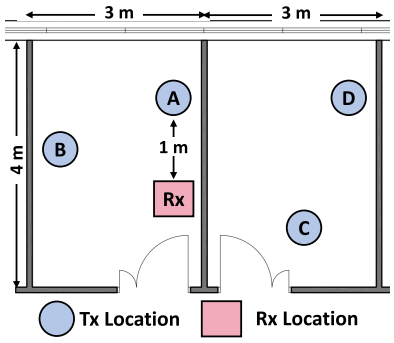


Fig. 10. Floor plan and Tx/Rx locations.

F. Performance at Different Locations

The RFFI system should be robust to location variations considering the mobility of IoT end nodes. To achieve this, we simulate channel effects, i.e., multipath and Doppler effects, during data augmentation and convert the collected IQ samples to a channel-independent spectrogram as the signal representation, as detailed in Section IV-B2. The RFFI system is evaluated in a real office environment with transmitters placed at different locations. The floor plan is given in Fig. 10 and the settings are provided below:

- **Pre-Training**: Clients 1-4 with PLUTO, B200 mini, B200, B210 receivers, respectively. 1,500 signal samples are collected from each transmitter-client pair. The federated pre-training approach is used.
- **Fine-Tuning**: Client 5 with a USRP N210 receiver. 200 signal samples are collected from each transmitter. The transmitters are placed at Location A when collecting fine-tuning signals.
- **Identification**: Client 5 with a USRP N210 receiver, another 300 signal samples are collected from each transmitter. The transmitters are in turn placed at Location A to Location D during data collection.

The identification results are given in Table III. It can be observed that the system performance reaches 97% when the

TABLE III
PERFORMANCE AT DIFFERENT LOCATIONS

	LOS		NLOS	
Transmitter Location	A	B	C	D
Accuracy	97.00%	89.83%	91.13%	89.20%

fine-tuning and identification datasets are collected at the same location, i.e., Location A. When the transmitters move to Locations B, C, and D, the accuracy drops by less than 8% and the final identification results still remain around 90%. These results demonstrate that the designed RFFI system is relatively robust to location variations.

VI. CONCLUSION

This paper proposed a federated RFFI protocol that is practical in the actual wireless context. It consists of three stages: federated pre-training, client fine-tuning, and identification. First, the federated pre-training produces an NN-based feature extractor in an unsupervised contrastive manner. Subsequently, the clients connect several linear layers to the feature extractor to create a classification NN, and then fine-tune its parameters to activate its RFFI functionality. After fine-tuning, the classification NN can predict the identity of the transmitter by analyzing the received physical layer signal. A federated LoRa-RFFI testbed was created to evaluate the proposed protocol, involving 60 COTS LoRa transmitters and six SDR receivers. The experimental results demonstrated that the federated RFFI can improve the accuracy from 63% to 95%, which benefits from the expanding of training data. It is concluded that federated learning is a promising technique to mitigate the data-hungry dilemma during RFFI development.

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. 138406–138446, 2020.

[2] 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, Sep. 2016.

[3] 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.

[4] 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. 10th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jul. 2017, pp. 58–63.

[5] J. Zhang, G. Shen, W. Saad, and K. Chowdhury, "Radio frequency fingerprint identification for device authentication in the Internet of Things," *IEEE Commun. Mag.*, vol. 61, no. 10, pp. 110–115, Oct. 2023.

[6] W. Yan, T. Voigt, and C. Rohner, "RRF: A robust radiometric fingerprint system that embraces wireless channel diversity," in *Proc. 15th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, May 2022, pp. 85–97.

[7] X. Qi, A. Hu, and T. Chen, "Lightweight radio frequency fingerprint identification scheme for V2X based on temporal correlation," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1056–1070, 2024.

[8] X. Yang and D. Li, "LED-RFF: LTE DMRS-based channel robust radio frequency fingerprint identification scheme," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1855–1869, 2024.

[9] J. Shi, H. Zhang, S. Wang, B. Ge, S. Mao, and Y. Lin, "FedRFID: Federated learning for radio frequency fingerprint identification of WiFi signals," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 154–159.

[10] M. Foruhandeh, A. Z. Mohammed, G. Kildow, P. Berges, and R. Gerdes, "Spotr: GPS spoofing detection via device fingerprinting," in *Proc. 13th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jul. 2020, pp. 242–253.

[11] J. Smailes, S. Köhler, S. Birnbach, M. Strohmeier, and I. Martinovic, "Watch this space: Securing satellite communication through resilient transmitter fingerprinting," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2023, pp. 608–621.

[12] 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, Jan. 2019.

[13] A. Al-Shawabka et al., "Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 646–655.

[14] 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, Aug. 2021.

[15] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRA using spectrogram and CNN," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[16] 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.

[17] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, "RFAL: Adversarial learning for RF transmitter identification and classification," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 2, pp. 783–801, Jun. 2020.

[18] M. Cekic, S. Gopalakrishnan, and U. Madhow, "Wireless fingerprinting via deep learning: The impact of confounding factors," in *Proc. 55th Asilom. Conf. Signals Syst. Comput.*, 2021, pp. 677–684.

[19] 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, Aug. 2019.

[20] T. Jian et al., "Radio frequency fingerprinting on the edge," *IEEE Trans. Mobile Comput.*, vol. 21, no. 11, pp. 4078–4093, Nov. 2022.

[21] 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, Oct. 2020.

[22] N. Soltani, G. Reus-Muns, B. Salehi, 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. 15518–15531, Dec. 2020.

[23] 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.

[24] 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. 22nd Int. Symp. Theory, Algorithmic Found., Protocol Design Mobile Netw. Mobile Comput.*, Jul. 2021, pp. 1–26.

[25] 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. 22nd Int. Symp. Theory, Algorithmic Found., Protocol Design Mobile Netw. Mobile Comput.*, Jul. 2021, pp. 1–20.

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

[27] A. Elmaghbub and B. Hamdaoui, "LoRa device fingerprinting in the wild: Disclosing RF data-driven fingerprint sensitivity to deployment variability," *IEEE Access*, vol. 9, pp. 142893–142909, 2021.

[28] S. Hanna, S. Karunaratne, and D. Cabric, "WiSig: A large-scale WiFi signal dataset for receiver and channel agnostic RF fingerprinting," *IEEE Access*, vol. 10, pp. 22808–22818, 2022.

[29] R. Xie et al., "A generalizable model-and-data driven approach for open-set RFF authentication," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4435–4450, 2021.

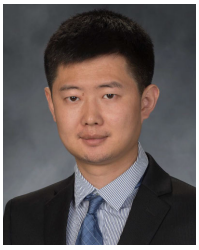
[30] S. Rajendran and Z. Sun, "RF impairment model-based IoT physical-layer identification for enhanced domain generalization," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1285–1299, 2022.

[31] G. Shen, J. Zhang, A. Marshall, R. Woods, J. Cavallaro, and L. Chen, "Towards receiver-agnostic and collaborative radio frequency fingerprint identification," *IEEE Trans. Mobile Comput.*, vol. 23, no. 7, pp. 7618–7634, Jul. 2024.

[32] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. F. Moura, "A deep learning approach to IoT authentication," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[33] 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.
[34] G. Shen, J. Zhang, and A. Marshall, "Deep learning-powered radio frequency fingerprint identification: Methodology and case study," *IEEE Commun. Mag.*, vol. 61, no. 9, pp. 1–7, Sep. 2023.
[35] (2023). *Labeled Faces in the Wild (LFW) Home*. Accessed: Nov. 6, 2023. [Online]. Available: http://vis-www.cs.umass.edu/lfw/
[36] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54, 2017, pp. 1273–1282.
[37] C. Zhang, S. Dang, J. Zhang, H. Zhang, and M. A. Beach, "Federated radio frequency fingerprinting with model transfer and adaptation," in *Proc. IEEE Conf. Comput. Commun. Workshops*, May 2023, pp. 1–6.
[38] R. Jha, J. Hayase, and S. Oh, "Label poisoning is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 36, 2023, pp. 71029–71052.
[39] S. D. Andrews, "Extensions to radio frequency fingerprinting," Ph.D. dissertation, Dept. Elect. Comput. Eng., Virginia Tech, Blacksburg, VA, USA, 2019.
[40] K. Merchant and B. Nousain, "Enhanced RF fingerprinting for IoT devices with recurrent neural networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 590–597.
[41] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6210–6219.
[42] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1597–1607.
[43] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.
[44] J. Gui et al., "A survey on self-supervised learning: Algorithms, applications, and future trends," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 2, pp. 1–20, Jun. 2024.
[45] R. Xie, W. Xu, J. Yu, A. Hu, D. W. K. Ng, and A. L. Swindlehurst, "Disentangled representation learning for RF fingerprint extraction under unknown channel statistics," *IEEE Trans. Commun.*, vol. 71, no. 7, pp. 3946–3962, Jul. 2023.
[46] 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, Feb. 2019.
[47] X. Wang, P. Hao, and L. Hanzo, "Physical-layer authentication for wireless security enhancement: Current challenges and future developments," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 152–158, Jun. 2016.
[48] T. Ohtsuji, T. Takeuchi, T. Soma, and M. Kitsunezuka, "Noise-tolerant, deep-learning-based radio identification with logarithmic power spectrum," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[49] S. Andrews, R. M. Gerdes, and M. Li, "Crowdsourced measurements for device fingerprinting," in *Proc. 12th Conf. Security Privacy Wireless Mobile Netw.*, 2019, pp. 72–82.
[50] 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 Eur. Symp. Secur. Privacy (EuroSP)*, Sep. 2020, pp. 322–338.

Guanxiong Shen (Member, IEEE) received the B.Eng. degree from Xidian University, Xi'an, China, in 2019, and the Ph.D. degree from the University of Liverpool, U.K., in 2023. He is currently an Associate Professor with Southeast University, Nanjing, China. His research interests include the application of artificial intelligence in wireless technologies, with a particular focus on the wireless physical layer. He served as the Co-Chair for the IEEE GLOBECOM and the ICC Workshop on machine learning and deep learning for wireless security.

Junqing Zhang (Member, IEEE) 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, U.K., in 2016. From February 2016 to January 2018, he was a Post-Doctoral Research Fellow with Queen's University Belfast. From February 2018 to October 2022, he was a Tenure Track Fellow and then a Lecturer (Assistant Professor) with the University of Liverpool, U.K., where he has been a Senior Lecturer (Associate Professor) since October 2022. His research interests include the Internet of Things, wireless security, physical layer security, key generation, radio frequency fingerprint identification, and wireless sensing. He was a recipient of U.K. EPSRC New Investigator Award.



Xuyu Wang (Member, IEEE) received the B.S. degree in electronic information engineering and the M.S. degree in signal and information processing from Xidian University, Xi'an, China, in 2009 and 2012, respectively, and the Ph.D. degree in electrical and computer engineering from Auburn University, Auburn, AL, USA, in August 2018. He is currently an Assistant Professor with the Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL, USA. His research interests include wireless sensing, the Internet of Things, wireless localization, smart health, wireless networks, and deep learning. He received the NSF CRII Award in 2021. He was a co-recipient of the ACM FAcct 2023 Best Paper Award, the 2022 Best Journal Paper Award of the IEEE ComSoc eHealth Technical Committee, the IEEE INFOCOM 2022 Best Demo Award, the IEEE ICC 2022 Best Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the IEEE GLOBECOM 2019 Best Paper Award, the IEEE ComSoc MMTC Best Journal Paper Award in 2018, the IEEE PIMRC 2017 Best Student Paper Award, and the IEEE SECON 2017 Best Demo Award.

Shiwen Mao (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Polytechnic University in 2004. He is currently a Professor and the Earle C. Williams Eminent Scholar, and the Director of the Wireless Engineering Research and Education Center, Auburn University. His research interests include wireless networks, multimedia communications, and smart grids. He received the IEEE ComSoc MMTC Outstanding Researcher Award in 2023, the SEC 2023 Faculty Achievement Award for Auburn, the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019, Auburn University Creative Research and Scholarship Award in 2018, and the NSF CAREER Award in 2010, and several service awards from IEEE ComSoc. He was a co-recipient of the 2022 Best Journal Paper Award of the IEEE ComSoc eHealth Technical Committee, the 2021 Best Paper Award of *Digital Communications and Networks* (Elsevier/KeAi), the 2021 IEEE Internet of Things Journal Best Paper Award, the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the 2018 Best Journal Paper Award and the 2017 Best Conference Paper Award from IEEE ComSoc MMTC, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He was a co-recipient of the Best Paper Award from IEEE GLOBECOM 2023 (two), 2019, 2016, and 2015, IEEE ICC 2022 and 2013, and IEEE WCNC 2015, and the Best Demo Award from IEEE INFOCOM 2024, IEEE INFOCOM 2022, and IEEE SECON 2017. He is the Editor-in-Chief of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, a Member-at-Large of the IEEE Communications Society Board of Governors, and the Vice President of Technical Activities of the IEEE Council on Radio Frequency Identification (CRFID).