Federated Radio Frequency Fingerprinting with Model Transfer and Adaptation

Chuanting Zhang[†], Shuping Dang[†], Junqing Zhang[‡], Haixia Zhang[§], and Mark A. Beach[†]

[†]Department of Electrical and Electronic Engineering, University of Bristol, United Kingdom
[‡]Department of Electrical Engineering and Electronics, University of Liverpool, United Kingdom
[§]School of Control Science and Engineering, Shandong University, Jinan, China

Abstract—The Radio frequency (RF) fingerprinting technique makes highly secure device authentication possible for future networks by exploiting hardware imperfections introduced during manufacturing. Although this technique has received considerable attention over the past few years, RF fingerprinting still faces great challenges of channel-variation-induced data distribution drifts between the training phase and the test phase. To address this fundamental challenge and support model training and testing at the edge, we propose a federated RF fingerprinting algorithm with a novel strategy called model transfer and adaptation (MTA). The proposed algorithm introduces dense connectivity among convolutional layers into RF fingerprinting to enhance learning accuracy and reduce model complexity. Besides, we implement the proposed algorithm in the context of federated learning, making our algorithm communication efficient and privacy-preserved. To further conquer the data mismatch challenge, we transfer the learned model from one channel condition and adapt it to other channel conditions with only a limited amount of information, leading to highly accurate predictions under environmental drifts. Experimental results on real-world datasets demonstrate that the proposed algorithm is model-agnostic and also signal-irrelevant. Compared with state-of-the-art RF fingerprinting algorithms, our algorithm can improve prediction performance considerably with a performance gain of up to 15%.

Index Terms—Radio frequency (RF) fingerprinting, federated learning, network security, model transfer and adaptation.

I. INTRODUCTION

Wireless communications have been connecting massive Internet of things (IoT) devices and transforming our everyday life with numerous exciting applications including smart homes, smart cities, connected healthcare, etc. However, due to the broadcast nature of wireless transmissions, IoT devices always suffer from security vulnerabilities and face many attack vectors, such as phishing and spoofing [1]. Device identification is crucial for protecting wireless networks since access requests can be denied if malicious devices or users are identified. Traditional device identification schemes are mainly based on techniques stemming from cryptography, such as encryption using public or private keys to protect higher-layer information security and to achieve identity recognition. On the other hand, these techniques could fail if the key is compromised. Thus, new types of security mechanisms in the lower layer are needed to enhance wireless network security.

Radio frequency (RF) fingerprinting is an emerging physical layer security technique that helps with device identification by exploiting hardware impairments that are hidden in the electromagnetic waves of the transmitter [2]–[4]. Recent research has found that every transmitter has its unique RF fingerprint resulting from imperfections of analog components, which are non-reproducible by attackers [5], [6].

Implementing RF fingerprinting using machine learning algorithms has been the mainstream due to their effective non-linear relationship modeling between input features and device identity. The earlier works on RF fingerprinting focus mainly on using handcrafted features to carry out classification, relying heavily on domain knowledge and human labor to extract features [7]. What is even worse, the significance of these features is non-deterministic and task-dependent. This dilemma can be greatly eased thanks to the advancement of deep learning techniques, in particular, deep neural networks, making automatic feature learning possible. The work in [8] adopted a convolutional neural network (CNN) framework to identify the type of protocol in use and the specific radio transmitter. After this, research works based upon CNN for RF fingerprinting have flourished [9]–[14]. In [9], the authors proposed DeepRadioID, another CNN framework for device identification that is robust against different channel conditions with the help of a carefully optimized digital finite input response filter. Similarly, to overcome the performance degradation caused by the randomness of wireless channels, other solutions, such as metric learning [10], carrier frequency offset compensation [11], and short-time Fourier transformation [13], were also explored in the existing literature. Specifically, these solutions were verified for different application scenarios, including unmanned aerial vehicle communication networks [14] and LoRa networks [12]. Experimental results on large-scale datasets demonstrate that considerable performance gains can be harvested by using the CNN architecture compared with traditional learning algorithms [15], [16].

Though many studies exist nowadays, RF fingerprinting still faces several challenges hindering its practical implementation. First, most of the works mentioned above focus on centralized learning and require data to be transferred to a server before processing, which is obviously inappropriate when data privacy and protection are mandatory [17], [18]. Second, one of the fundamental challenges of RF fingerprinting is the data distribution drift between the training and testing phases, which is generally caused by environments. The drift could lead to

a significant mismatch between training and testing data and degrade prediction performance considerably. How to solve this challenge is still yet to be resolved. Last but not least, training a deep CNN is time-consuming due to the requirement of adapting a huge number of network parameters. Designing an effective and efficient CNN architecture specifically for RF fingerprinting is also challenging.

This paper tries to address the above challenges and proposes a federated RF fingerprinting algorithm based on a strategy termed model transfer and adaptation (MTA). We design a CNN framework based on the dense connectivity [19] with careful calibration, aiming to reduce model complexity. Besides, with MTA, our model can quickly adapt to the new data distribution using a limited number of data samples and requires no complex pre-processing of data. Our main contributions are summarized as follows:

- We propose an RF fingerprint algorithm based upon the federated learning paradigm, which does not require raw data collected in a centralized manner, thus guaranteeing data privacy.
- We introduce dense connectivity into RF fingerprinting and design a novel CNN framework that is capable of considerably reducing model complexity.
- To overcome the data distribution drift and the data mismatch problem, we propose MTA, a strategy that can fully leverage the knowledge of a learned model from one channel condition and quickly adapt it to new channel conditions requiring only a limited number of data samples. MTA is model-agnostic and can be applied to most learning algorithms.
- We verify our algorithm on real-world datasets, and the results demonstrate that our algorithm is superior to state-of-the-art RF fingerprinting methods.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

In this paper, we consider the RF fingerprinting problem in future edge networks, as shown in Fig. 1. There are M separate edge cloud units that connect to a trusted central server. Each edge cloud unit possesses both communication and computing abilities and serves a group of user devices. With the consideration of privacy preservation, raw data transferring among edge cloud units is not permitted. Meanwhile, each edge unit $m \in \mathcal{M} = \{1, 2, \ldots, M\}$ is deployed with a prediction model which is parameterized with w. This model performs prediction whenever a user device makes access requests to the base station and judges if the device is from a legitimate user group or not. Generally speaking, each edge cloud unit m has a local and private dataset denoted as $D_m = \{(\mathbf{x}_{m,i}, y_{m,i})\}_{i=1}^{N_m}$ with $\mathbf{x}_{m,i} \in \mathbb{R}^{C \times H}$ being features extracted from in-phase and quadrature (IQ) samples and $y_{m,i}$ the corresponding device identity. C indicates the number of types of input adopted, and H represents the dimension of the input. We detail the construction of $\mathbf{x}_{m,i}$ in Section III-A.
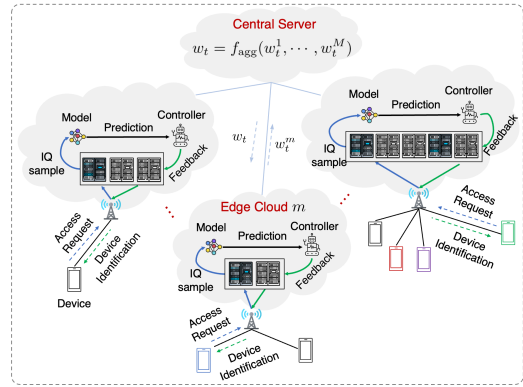

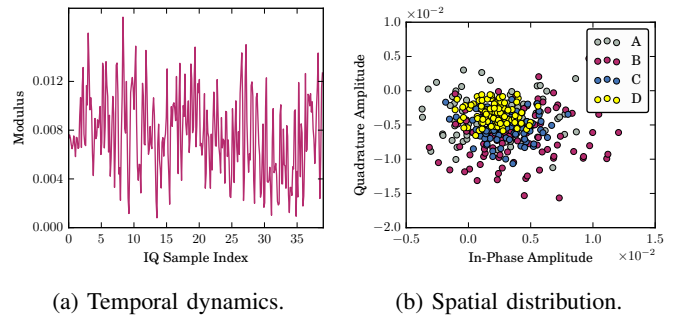Fig. 1: System model of the proposed federated RF fingerprinting.

(a) Temporal dynamics.   (b) Spatial distribution.

Fig. 2: Visualization of IQ samples in the time and space domains.

B. Problem Formulation

Given a prediction model and each edge cloud unit's private dataset D_m, the prediction problem can be formulated as

$$\arg\min_w \left\{ f(w) = \frac{1}{M} \sum_{m=1}^{M} f_m(w) \right\}, \quad (1)$$

where $f_m(\cdot)$ denotes the local objective at the edge cloud unit m, which can be explicitly expressed as

$$f_m(w) = \frac{1}{N_m} \sum_{i=1}^{N_m} \ell(\mathsf{d}_{m,i}; w), \quad (2)$$

where $\mathsf{d}_{m,i} \in D_m$ denotes the input data samples selected from D_m, and $\ell(\cdot)$ is a customized loss function quantifying the accuracy of the prediction returned by the model and will be detailed in Section III-B. In this paper, (1) is expected to be solved collaboratively by the M edge cloud units as shown in Fig. 1.

III. CNN FRAMEWORK DESIGN FOR RF FINGERPRINTING

A. Training Sample Construction

Given a stream of received IQ samples $\mathbf{s}$, whose value at time slot t is given by $s_t = I_t + jQ_t$, where I_t and Q_t are two real values denoting the in-phase and quadrature amplitudes of the RF signal. A sample IQ stream is illustrated in Fig. 2.
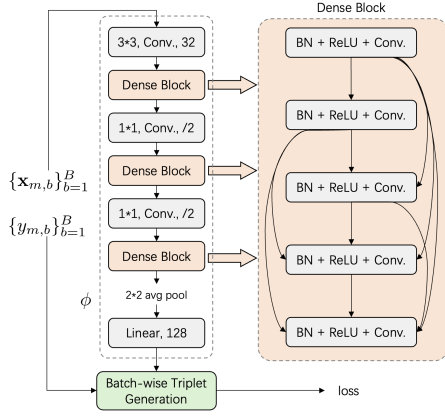
Fig. 3: Proposed framework for RF fingerprinting.

By using the sliding window scheme, we first generate a series of data segments with the window size H, then extract the in-phase branch (I_t), quadrature branch (Q_t), and the corresponding modulus branch ($\sqrt{I_t^2 + Q_t^2}$), and treat them as input features to our prediction model. Thus, the types of input features (C) are 3. In this paper, H is set to 1024 since we empirically found that larger values yield no performance improvements yet raise computational complexity.

B. CNN Framework with Dense Connectivity

Compared with traditional RF fingerprinting systems that adopt residual connectivity for learning RF features, we resort to dense connectivity [19] for enhancing feature reuse and propagation between different layers. As shown in Fig. 3, our proposed CNN framework mainly consists of three convolutional layers, three dense blocks, and a linear layer, followed by a batch-wise triplet generation module. For convolutional layers, their kernel sizes vary according to functional differences and so does the corresponding output feature mapping relation. Each dense block comprises L layers with each layer l being a composite function $\mathcal{F}_l(\cdot)$ that sequentially performs operations of batch normalization (BN), ReLU activation, and convolution. For the l-th layer of a dense block, the output feature mapping can be expressed as

$$\mathbf{x}_l = \mathcal{F}_l(\mathbf{x}_0 \oplus \mathbf{x}_1 \oplus \cdots \oplus \mathbf{x}_{l-1}), \qquad (3)$$

where $\mathbf{x}_0$ denotes the initial input to a dense block, and $\oplus$ is the concatenation operation. It should be noted that concatenation operation can greatly facilitate feature propagation and reuse between different layers, mitigating the gradient vanishing and exploding problems as an additional benefit.

Since a dense block has many layers and each layer will transfer its feature mapping relation to all its subsequent layers, the model complexity increases exponentially and the training process could be quite time-consuming. To mitigate this drawback, we set the kernel size of the convolutional layer between dense blocks to 1×1 and also reduce the feature mapping relation to half of its input. We denote the framework as a function ϕ, then for batch input $\mathbf{x}_{m,b} \in D_m$, the output

---

Algorithm 1: Federated RF Fingerprinting

Input: IQ samples $\{D_m\}_{m=1}^M$; training round R; learning rate η; initialized global model w_0

Output: Global model: w

1 **for** $r = 0, 1, \cdots, R$ **do**
2   Select a subset of edge cloud units $\mathcal{M}_r$
    ▷ run on edge cloud units in parallel
3   **for** $m \in \mathcal{M}_r$ **do**
4   Receive global model and $w_r^m \leftarrow w_r$;
5   **for** *each local epoch* $k = 0, 1, \cdots$ **do**
6   **for** *each batch* $\mathcal{B}$ *in* D_m **do**
7          $w_r^m \leftarrow w_r^m - \eta \nabla f(\mathcal{B}; w_r^m)$;
8      $w_{r+1}^m \leftarrow w_r^m$ and upload it to the central server;
    ▷ run on server side
9   Local model aggregation:
    $w_{r+1} = \frac{1}{|\mathcal{M}_r|} \sum_{m \in \mathcal{M}_r} w_{r+1}^m$;

---

feature vectors from the final linear layer can be written as $\{\phi(\mathbf{x}_{m,b})\}_{b=1}^B$.

After obtaining the feature vectors, we construct B triplets $\{\mathbf{a}_{m,b}, \mathbf{p}_{m,b}, \mathbf{n}_{m,b}\}_{b=1}^B$ based on the label information $\{y_{m,b}\}_{b=1}^B$. Specifically, $\mathbf{a}_{m,b}$ is associated with Anchor, and its value is exactly $\phi(\mathbf{x}_{m,b})$; $\mathbf{p}_{m,b}$ is relevant to Positive which is sampled from $\{\phi(\mathbf{x}_{m,b})\}_{b=1}^B$, and its corresponding label is the same as $\mathbf{a}_{m,b}$. Similarly, $\mathbf{n}_{m,b}$ pertains to Negative, which is also sampled from $\{\phi(\mathbf{x}_{m,b})\}_{b=1}^B$, and its corresponding label is different from $\mathbf{a}_{m,b}$. With the defined triplets, the loss function is computed by

$$\ell(\mathbf{a}_{m,b}, \mathbf{p}_{m,b}, \mathbf{n}_{m,b}) = \max\{d_{ap} - d_{an} + \alpha, 0\}, \qquad (4)$$

where α is a preset threshold that controls the flexibility; d_{ap} and d_{an} represent the Anchor-Positive and Anchor-Negative distances, respectively, which are defined as

$$\begin{aligned} d_{ap} &= ||\mathbf{a}_{m,b} - \mathbf{p}_{m,b}||_2^2, \\ d_{an} &= ||\mathbf{a}_{m,b} - \mathbf{n}_{m,b}||_2^2. \end{aligned} \qquad (5)$$

Local model parameter w can be obtained by optimizing the above loss function.

C. Federated Training

Having data prepared and the RF fingerprinting framework constructed, we can proceed with the model training. The training procedure is summarized in Algorithm 1 and generally consists of three steps:

- A subset edge cloud units $\mathcal{M}_t$ is selected and the central server broadcasts the global model w_r to each edge cloud unit of $\mathcal{M}_t$ at the beginning of the r-th round training;
- Based upon the received global model, each edge cloud unit updates w_r using its local and private data. The one-step update rule can be written as

$$w_{r+1}^m \leftarrow w_r - \eta \nabla f(\mathcal{B}_r^m; w_r), \qquad (6)$$

where η is the local learning rate, and $\mathcal{B}_r^m$ is the batch input data. The local model will be uploaded to the central server once the updating process is complete.

- The central server aggregates local models to yield a new global model by the following relation:

$$w_{r+1} = \frac{1}{|\mathcal{M}_r|} \sum_{m \in \mathcal{M}_r} w_{r+1}^m. \tag{7}$$

These three steps will be iterated until the loss function converges or the training process reaches the maximum round. After that, we obtain the final model parameter w and use it to make the prediction.

D. Model Transfer and Adaptation

In the last subsection, we explained how we obtain model global w. When deploying the model in practical environments to conduct production, its performance will be significantly challenged when the environment drifts since the data samples from the new environment could be rather different from those used to train the model. To alleviate this problem, we propose a strategy called MTA based on the idea of transfer learning [20]. The MTA strategy works as follows: when the system encounters a different environment with new data, we do not retrain the model from scratch. Instead, we transfer the learned model w to this new environment and use it as the initialized global model. Consequently, we can benefit from this well-designed initialization and easily adapt the global model using a limited number of IQ samples for only several rounds, producing an updated global model w'. Then we use w' to identify the labels of new IQ samples.

E. Model Prediction

Once new data $\mathbf{x}$ comes, we input it into the model and obtain its feature $\phi(\mathbf{x})$. The distances between $\phi(\mathbf{x})$ and each class's feature vector will be calculated and compared. The label of $\phi(\mathbf{x})$ will be predicted as the one with the smallest distance from it.

IV. EXPERIMENTAL EVALUATION

A. Setup

1) Dataset and Parameter Settings: We test the performance of our algorithm on a public benchmark dataset released in [10]. The dataset includes four devices' three different kinds of IQ samples pertaining to WiFi, 4G LTE, and 5G NR. The four devices (denoted as A, B, C, and D) are USRP X310 transmitters and the receiver is a USRP B210 platform. The signal samples are random payloads simulated using MATLAB toolboxes and are logged on two different days with different channel environments. A visualization of the IQ samples has been shown in Fig. 2, from which we can notice that different devices' IQ samples are coupled with each other, and simple linear models could not effectively distinguish them. In our simulations, the first day's IQ samples will be used as the training dataset. For the second day's IQ samples, we select the first ρ samples to perform the MTA strategy and test the performance on the rest of the IQ samples.

Both the training and test datasets have 12000 samples. All data is distributed into 100 edge cloud units to imitate a federated learning scenario. The model is trained locally using the Adam optimizer for 10 epochs with a batch size 10. We set the learning rate to 0.0001 and the global round to 50. In each global round, we suppose 10% of edge cloud units are involved in training due to different traffic loads.

2) Proposed and Baseline Algorithms: For our algorithm, each dense block has 6 convolutional layers. There are two versions of our algorithm sharing the same network architecture, i.e., Proposed-Basic (w/o MTA strategy) and Proposed-MTA (w/ MTA strategy). We compare our algorithms with the following baselines.

- Federated MLP: An architecture that has two linear layers with hidden sizes of 512 and 4.
- Federated CNN. An architecture that has two convolutional layers and two linear layers. The hidden sizes are 16, 64, 512, and 4, respectively.
- Federated ResNet. An architecture that has four convolutional blocks and one linear layer. Each block consists of four convolutional layers with feature maps of 32, and the last linear layer has a hidden size of 4.
- Centralized ResNet. This algorithm has exactly the same architecture as the above ResNet but is trained in a centralized way.

Unless otherwise stated, the kernel sizes of hidden layers and the activation functions in both our algorithms and the baselines are 3×3 and ReLU, respectively.

B. Overall Prediction Performance

Table I presents a detailed performance comparison between our proposed algorithms and those baselines. The prediction performance is measured in terms of accuracy, and a greater value indicates better prediction. We can observe from Table I that our RF fingerprinting algorithm achieves the highest prediction accuracy on all datasets of different signal samples, compared with all baselines regardless of centralized training or federated training. There is a considerable performance improvement in our results over the second-best ones. For example, the obtained accuracy of 0.9343 on the 4G signal and 0.9033 on the hybrid signal (a mix of 4G, 5G, and WiFi signals) samples are much higher than the accuracy of 0.8257 and 0.7692 pertaining to the centralized ResNet and the federated ResNet, respectively. Even without using the MTA strategy, our designed CNN framework with dense connectivity still achieves better predictions than baselines. The results indicate that approximately 10% to 15% improvements can be achieved by our proposed algorithm. A more noticeable observation is that our algorithm has roughly 80,000 parameters, which are less than those of the baselines. In a nutshell, the results presented in Table I demonstrate the effectiveness of dense connectivity and the MTA strategy for RF fingerprinting.

C. Per-Device Prediction Performance

The previous subsection reports the overall prediction performance for different types of signal samples while this

TABLE I: Performance comparison between the proposed algorithm and baselines.

		Centralized ResNet	Federated MLP	Federated CNN	Federated ResNet	Proposed-Basic	Proposed-MTA
Notes	# of Parameters	157.54 K	1.58 M	1.19 M	157.54 K	79.95 K	
	Privacy-Preserved	✗	✓	✓	✓	✓	
Signals	4G	0.8257	0.7110	0.7895	0.7245	<u>0.8683</u>	**0.9343**
	5G	0.8375	0.7123	0.7725	0.7448	**0.9105**	<u>0.9100</u>
	WiFi	<u>0.9690</u>	0.7205	0.8213	0.7688	0.7508	**0.9800**
	Hybrid	0.7268	0.6923	0.7553	0.7692	<u>0.7721</u>	**0.9003**

(a) Centralized ResNet (4G).   (b) Proposed-Basic (4G).   (c) Proposed-MTA (4G).

(d) Centralized ResNet (Hybrid).   (e) Proposed-Basic (Hybrid).   (f) Proposed-MTA (Hybrid).
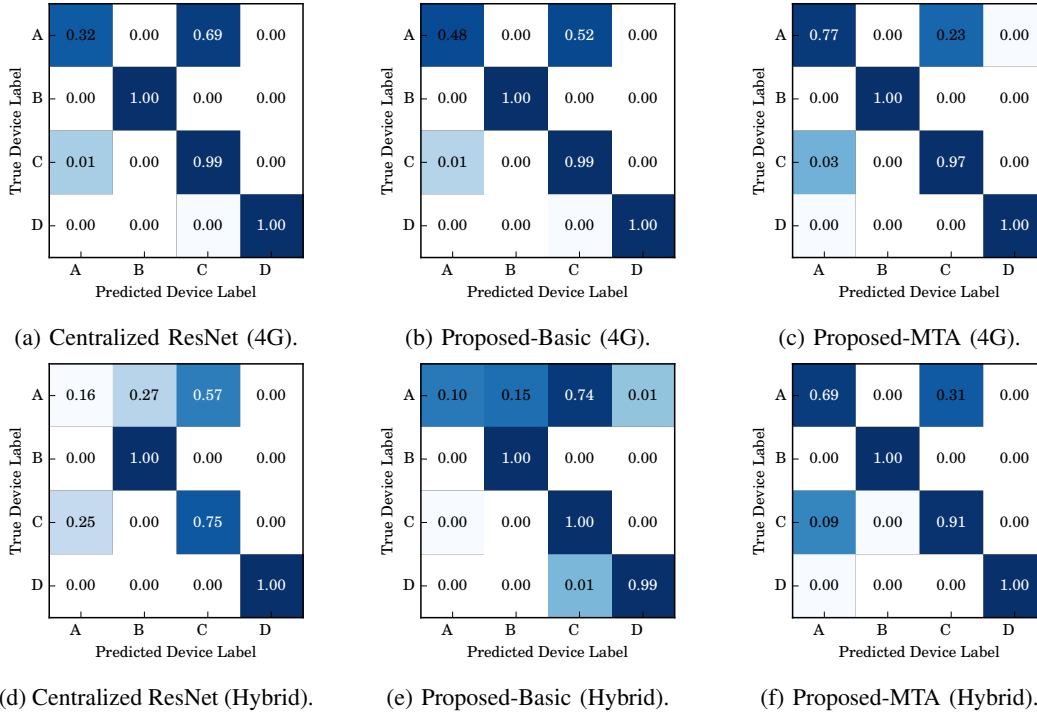
Fig. 4: Confusion matrix of centralized ResNet and the proposed algorithms for various signal datasets.

section presents the prediction results for each device. We only compare our proposed algorithm with federated ResNet on the 4G and hybrid signal datasets for simplicity.

The obtained results are displayed in Fig. 4. Both federated ResNet and the proposed algorithm perform well for devices B, C, and D on the 4G signal samples, achieving almost 100% accuracy. But federated ResNet's accuracy for device A is only 32%, which may be because device A is quite similar to device C. However, the obtained results of the proposed algorithm for device A are 48% (w/o MTA) and 77% (w/ MTA). For the hybrid signal dataset, a similar phenomenon can be observed, and we omit the repeated explanations here for simplicity. Through in-depth analysis and comparison of Fig. 4, we can substantiate that our proposed algorithm, in particular with the MTA strategy, consistently outperforms the baseline algorithms in terms of per-device prediction.

D. Generalization Ability of MTA

Training a model on one channel condition and testing it on another will degrade the prediction performance considerably due to the data distribution mismatch between different environments, as shown in Fig. 5. The training accuracies of federated CNN and ResNet are up to 0.9, but their test accuracies are lower than 0.8. This phenomenon inspired us to propose the MTA strategy. To demonstrate the generalization ability of our MTA strategy, we apply it to all algorithms and compare the prediction performance with and without MTA in Fig. 6a. It is evident that the MTA strategy can consistently boost the corresponding algorithm's performance. Taking the Federated ResNet algorithm as an example, its accuracy with respect to the 4G signal dataset and the hybrid signal dataset is improved from 0.7245 and 0.7692 to 0.8255 and 0.7847, respectively. In addition, Fig. 6b displays the prediction performance of federated ResNet and our proposed algorithm when increasing the number of data samples used in the MTA strategy. We can observe that the accuracy gradually increases when more data samples are available for model adaptation.
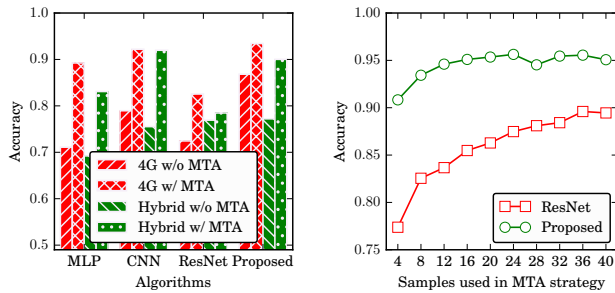
V. CONCLUSION

This paper investigated the RF fingerprinting problem in the context of federated learning for edge networks. We proposed
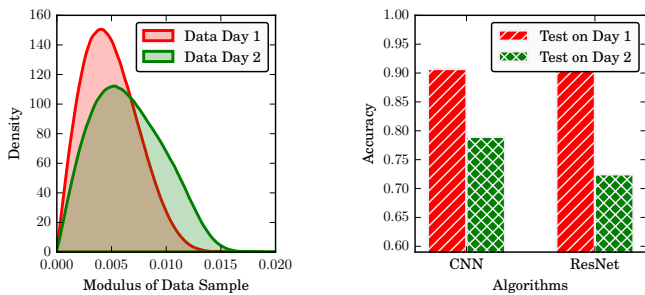
(a) Data distribution mismatch.

(b) Performance degradation.

Fig. 5: Illustration of data distribution mismatch between different days and its impacts on prediction performance.

(a) Predictions w/ and w/o MTA.

(b) Predictions versus ρ.

Fig. 6: Generalization ability of MTA.

a novel CNN framework by introducing dense connectivity into RF fingerprinting. We found that because of time-varying wireless environments, there could exist data distribution drifts between training and testing datasets, further leading to data distribution mismatches and degraded prediction performance. Accordingly, we designed an MTA strategy to mitigate this challenge and verified that the prediction performance can be considerably improved. More importantly, our federated training involved no data transferring among different edge cloud units, thus the privacy and protection of IQ dataset were guaranteed. Possible future research directions include further reducing the communication workload between the edge cloud and central server by introducing model compression and enhancing the security of model transferring by adopting differential privacy techniques.

REFERENCES

[1] J. Zhang, C.-H. Chang, C. Gu, and L. Hanzo, "Radio frequency fingerprints vs. physical unclonable functions - are they twins, competitors or allies?" *IEEE Network*, pp. 1–9, 2022.

[2] M. Nair, T. Cappello, S. Dang, V. Kalokidou, and M. A. Beach, "RF fingerprinting of LoRa transmitters using machine learning with self-organizing maps for cyber intrusion detection," in *Proc. IEEE/MTT-S International Microwave Symposium*, 2022, pp. 491–494.

[3] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of radio frequency fingerprinting techniques," *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222–233, 2020.

[4] Y. Xing, A. Hu, J. Zhang, L. Peng, and G. Li, "On radio frequency fingerprint identification for DSSS systems in low SNR scenarios," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2326–2329, 2018.

[5] S. U. Rehman, K. W. Sowerby, and C. Coghill, "Analysis of impersonation attacks on systems using RF fingerprinting and low-end receivers," *Journal of Computer and System Sciences*, vol. 80, no. 3, pp. 591–601, 2014.

[6] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[7] I. O. Kennedy, P. Scanlon, F. J. Mullany, M. M. Buddhikot, K. E. Nolan, and T. W. Rondeau, "Radio transmitter fingerprinting: A steady state frequency domain approach," in *Proc. IEEE 68th Vehicular Technology Conference*, 2008, pp. 1–5.

[8] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 146–152, 2018.

[9] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "DeepRadioID: Real-time channel-resilient optimization of deep learning-based radio fingerprinting algorithms," in *Proc. Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 51–60.

[10] G. Reus-Muns, D. Jaisinghani, K. Sankhe, and K. R. Chowdhury, "Trust in 5G open RANs through machine learning: RF fingerprinting on the POWDER PAWR platform," in *Proc. IEEE GLOBECOM*, 2020, pp. 1–6.

[11] 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.

[12] 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.

[13] 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.

[14] 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. 15 518–15 531, 2020.

[15] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "Deep learning for RF fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

[16] 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, 2020.

[17] 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 International Conference on Artificial Intelligence and Statistics*, vol. 54, 20–22 Apr 2017, pp. 1273–1282.

[18] 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. Twenty-Second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2021, p. 41–50.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE CVPR*, 2017, pp. 4700–4708.

[20] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, 2019.