This is the author's version of a paper which has been accepted for publication in IEEE Internet of Things Journal.

Deep Federated Fractional Scattering Network for Heterogeneous Edge Internet-of-Vehicle Fingerprinting: Theory and Implementation

Tiantian Zhang, *Graduate Student Member, IEEE*, Dongyang Xu, *Member, IEEE*, Jing Ma, *Student Member, IEEE*, Ali Kashif Bashir, *Senior Member, IEEE*, Maryam M. Al Dabel, Hailin Feng, *Member, IEEE*

***Abstract*—With the rapid development of distributed edge intelligence (DEI) within Internet of vehicle (IoV) network, it is required to support heterogeneous rapid, reliable and lightweight authentication which prevents eavesdropping, tampering and replay attacks. Radio Frequency Fingerprinting (RFF), which leverages unique and tamper-proof hardware characteristics, is an emerging deep learning based physical layer technology poised to achieve excellent authentication within DEI enhanced heterogeneous IoV. However, centralized collection of critical datasets will bring severe privacy concerns as well as huge communication overheads towards resources-constrained IoV nodes. In this paper, we propose a deep federated fractional scattering fingerprinting network (FFSFNet) which amalgamates fractional wavelet scattering and federated learning to achieve excellent identification. Particularly, we first exploit fractional wavelet scattering to extract RFF characteristics from non-stationary waveform, eliminate redundancies and enhance interpretability. To improve the training efficiency and privacy protection capability, we design a novel federated framework, which not only completes distributed training, reduces overhead but also protects privacy. Furthermore, we conducted a comprehensive comparative analysis of different model quantization schemes and validated the proposed scheme with field programmable gate array (FPGA) accelerators. Experimental results demonstrate that the proposed FFSFNet can maintain excellent identification performance with only 5.08% of original samples. The model size and inference latency can be effectively improved by quantization with limited degradation. Moreover, the identification testing accuracy of FFSFNet can eventually converge to 99.4% with 0.64ms inference latency per sample.**

***Index Terms*—Distributed edge intelligence, radio frequency fingerprinting, fractional wavelet, scattering network, heterogeneous federated learning.**

I. Introduction

This work was supported in part by the National Key R&D Program of China under Grant 2022YFB2902203; in part by the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University under Grant 2023D13. (*Corresponding author: Dongyang Xu.)*

Tiantian Zhang, Dongyang Xu and Jing Ma are with School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China. Dongyang Xu is also with National Mobile Communications Research Laboratory, Southeast University, China (emails: tiantianzhang@stu.xjtu.edu.cn; xudongyang@xjtu.edu.cn; poppy-01@stu.xjtu.edu.cn).

Ali Kashif Bashir is with the Department of Computing and Mathematics, E-154, John Dolton, Chester Street, M15 6H, Manchester Metropolitan University, Manchester, United Kingdom. (emails: dr.alikashif.b@ieee.org).

Maryam M. Al Dabel is with Department of Computer Science and Engineering, College of Computer Science and Engineering, University of Hafr Al Batin, Saudi Arabia (emails: maldabel@uhb.edu.sa).

Hailin Feng is with College of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou, China. (emails: hlfeng@zafu.edu.cn).

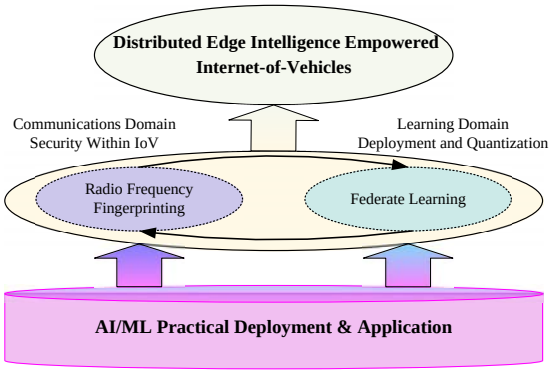


Fig. 1. The architecture of distributed edge intelligence empowered Internet-of-Vehicles.

IN recent years, the rapid advancement of the Internet of Vehicles (IoV) has facilitated seamless connectivity among smart vehicles, road infrastructure, network facilities and users. This intelligent integration of IoV enables comprehensive environmental perception, enhances transportation efficiency, and reduces road accidents, paving the way for next generation of intelligent transportation systems [1]–[4]. The intelligent IoV network encompasses a vast array of heterogeneous vehicles and roadside units (RSUs) nodes that generate substantial volumes of critical information [5], [6]. Furthermore, the promising IoV architecture, based on cloud and edge intelligence, enables distributed artificial intelligent (AI) among smart vehicles and RSUs to process related data and computations locally. Given the vast amount of raw data generated within IoV networks, AI plays a crucial role in applications such as assisted driving, autonomous driving, intelligent traffic management and vehicle-road collaboration [7]. Centralized data collection through wireless communications is essential in IoV networks, while it raises significant privacy concerns and imposes substantial transmission overheads.

To tackle with the issues above, the framework of distributed edge intelligence (DEI) is developed in which edge servers cooperate with a number of edge clients to jointly train effective AI models for various IoV applications while preserving clients' privacy [8]. To achieve this paradigm, two big challenges are required to respectively deal with in communications and learning domain. As shown in Fig. 1, in *communications domain*, it is required to guarantee rapid and reliable wireless access links during IoV communications be-

tween heterogeneous nodes. If rapid and reliable coordination of information among various communication nodes within vehicles can be maintained, a secure, efficient and environmentally friendly I[illegible]ed [9], [10]. However, tr[illegible]tion protocols which rel[illegible]ing and susceptible to wireless deception and intrusion attacks. Furthermore, as various IoV nodes utilize diverse communication protocols, a vast num[illegible]inuously generated and [illegible]3]. Rapid and reliable [illegible]omputing resources through DEI poses significant challenges in the intelligent IoV netw[illegible] federated learning related [illegible]information, limited com[illegible]rth. The collaboration tasks within DEI framework requires the efficient transmission of crucial information, while also addressing concerns such as data privacy, computing resource allocation and transmission overhead [15]–[17]. Meanwhile, reliable, rapid, and lightweight authentication services are essential for facilitating the exchange of critical information across various scenarios within IoV networks [18]–[20].

Radio frequency fingerprinting (RFF) which offers high reliability and low latency identification through unique and tamper-proof hardware features can be utilized to establish the security link within IoV networks [21]–[25]. Although RFF can provide secure and reliable authentication services between heterogeneous nodes in IoV networks, traditional RFF relies on centralized training, which requires massive shared data and computing resources. Within learning domain, federated learning (FL) is proposed to guarantee the data privacy and coordinate computing resources [26]–[29]. However, the data redundancy and Insufficient interpretability greatly limits the widespread application of the DEI [30]. From the perspective of DEI empowered IoV, distributed FL can handle the privacy concerns, reduce the transmission overhead and coordinate computing resources. Furthermore, RFF can provide reliable and rapid authentication services to protect the critical information from attacking. As illustrated in Fig. 2, the architecture of deep federated fractional scattering network is designed for the heterogeneous IoV fingerprinting. This framework allows diverse local nodes which includes vehicles nodes, RSUs, and user to utilize their extensive datasets for local model training, thereby eliminating transmission overhead between nodes. Moreover, traditional models are typically deployed in 32-bit float-point format which always requires huge memory and computing resources.

To address high memory usage and energy consumption in the deployment of practical deep neural networks (DNNs), a comprehensive survey of quantization concepts and methods for DNNs is provided in [31]. A remarkable quantization scheme can be applied to the recommendation models in production environments based on low-precision hardware [32]. To mitigate the accuracy loss associated with quantization after model training, a novel quantization scheme is designed to solve high dynamic range, zero overflow, diverse normalization, and limited model parameters [33]. Besides, the trade off between delay and accuracy in model quantization still is

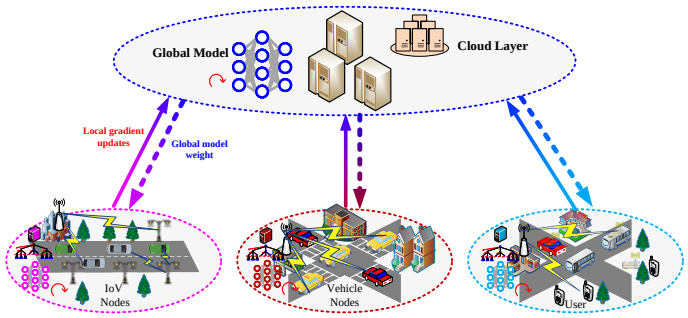


Fig. 2. The architecture of deep federated fractional scattering network for heterogeneous Internet-of-Vehicles fingerprinting.

a challenging problem. The computing accelerator based on FPGA can utilize model prune, quantization and multi-core parallel computing to achieve efficient model inference for edge application [34]. Considering the energy consumption of federated learning, the gradient sparsity, weight quantization and pruning can be utilized to improve the efficiency of federated models which are deployed on 5G terminals [35]. Without model quantization optimization, these models remain large, necessitating high memory and causing high inference latency which complicates the model's deployment in resource-constrained IoV network, presenting significant challenges. In summary, how to establish an excellent RFF with distributed edge intelligence for providing rapid and reliable authentication ability embracing the following three challenging aspects:

1) Privacy and transmission overhead concerns: The conventional centralized learning scheme imposes significant demands on computing resources and raises substantial privacy concerns. Designing an effective training framework to efficiently ensure privacy protection and facilitate training on resource-limited nodes has emerged as a challenging problem.
2) Data redundancy and insufficient interpretability: The extensive raw datasets from various nodes contains significant redundancy. Effectively extracting the RFF characteristics embedded within non-stationary waveform poses substantial challenges. Moreover, traditional model training suffers from its black box operational mode constraining the practical deployment and application.
3) Model practical deployment limitations: The optimization and deployment of high-performance, cost-effective models have become pressing concerns. Traditional post-training model deployment encounters challenges related to memory, performance, and latency, all of which must be addressed within IoV networks.

A. Related Work

With the rapid evolution of intelligent IoV, the traditional high-level encryption-based authentication schemes becoming vulnerable to fake and deception attacks, making them inadequate for critical applications such as autonomous driving, vehicle-road collaboration, and human-machine interac-

tion. A novel physical layer authentication scheme which leverages unique hardware features and facilitated by deep learning can offer secure, reliable, and lightweight access authentication services through DEI within IoV networks. DEI enables distributed machine learning among smart vehicles and roadside units (RSUs) at the IoV network edge, close to the data sources. By deploying models on data-source side, DEI significantly reduces data transmission overhead and ensures timely feedback during data generation, thereby enhancing the operational efficiency of the entire intelligent IoV. However, each distributed model trains solely on local data, lacking the ability to utilize global data. Moreover, distributed RFF requires substantial datasets for model learning. Designing an efficient network model is crucial for its effective deployment. Since DEI is often deployed at the edge, lightweight deployment and efficient inference of RFF models are key constraints for its application. In summary, efficiently designing and deploying RFF models within IoV networks under a distributed architecture remains a significant challenge. Furthermore, model quantization emerges as an essential technology for efficient deployment, reducing model size and inference latency with limited performance losses.

Despite the challenges discussed above are still exist, researchers have made significant strides in addressing these issues. To address the signal length diversity and robustness of RFF, authors in [36] proposed a novel network capable of handling variable-length signals, enabling efficient feature extraction of diverse samples. Considering the non-stationary RFF application, authors in [37] proposed a deep learning based RFF scheme for massive LoRa nodes identification. Authors in [38] proposed a deep learning-based RFF that exploits physical-layer hardware impairments as unique features for devices identification. To contend with mixed time-varying distortion challenges, authors in [39] involved spectral cyclic shift division to suppress interference. Considering the inevitable presence of unmanned aerial vehicle (UAV), authors in [40] extracted features from preamble signals and constructed a distributed model for UAV identification. To address the data collection under various channel conditions, authors in [41] proposed different data augmentations for transmitter and receiver according to the availability of datasets. The authors in [42] reviewed the distance-based classifier and automatic feature extractor. Then, combined with deep learning to form hybrid RFF schemes. The authors in [43] provided the tutorial of building closed-set and open-set RFF system, and created the testbed to publicly provide the collected datasets online.

Traditional RFF schemes demand extensive datasets, significant computational resources for training and interpretability. Authors in [44] have developed translation-invariant operators, and a scattering propagator was introduced to capture the non-linear characteristics. Furthermore, considering the translation invariance of wavelet scattering networks, the authors in [45] confirmed their ability to preserve high-frequency characteristics. Authors in [46] defined the temporal deformation and local translation invariance which enhanced the representation of Mel-scale frequency cepstral coefficients (MFCC) with multi-order scattering coefficients. Considering the time shift invariance of signals, authors in [47] introduced a time-frequency scattering transformation to achieve multiscale energy decomposition. Authors in [48] proposed fractional wavelet scattering network that efficiently extracts non-stationary medical texture features. To improve the model's interpretability, authors in [49] proposed a scattering network based on fractional wavelets, the energy conservation, deformation stability and other properties have been proven. Authors in [50] involved deep fractional scattering to extract RFF features of LoRa preamble, demonstrating that it can efficiently handle RFF features under non-stationary conditions. Furthermore, considering the reliance of distributed learning towards computational resources, authors in [51] proposed a hybrid architecture that combines fractional scattering networks with FL to ensure the privacy and efficiency. Therefore, efficient RFF features extraction methods can achieve the removal of redundant and improve the learning effectiveness of FL framework.

In typical IoV networks, heterogeneity is a defining characteristic. FL can effectively harnesses distributed intelligence nodes, encompassing computational resources and limited storage capacity, as emphasized by [52], [53]. It is crucial for client models to effectively update the server model to achieve the desired performance and convergence, as discussed in [54]. To improve the learning efficiency, authors in [55] proposed an adaptive gradient update strategy which achieves dynamic optimization during the training. To realize optimal model, authors in [56] employed dynamic updates weights to improve the FL efficiency. Furthermore, authors in [57] advocated for dynamic optimization through a strategic combination of local iterations and global aggregation. In their study [58], authors proposed a RFF scheme that leverages data augmentation to discern large-scale nodes. Authors in [59] developed RFF FL model to identify Wi-Fi samples. Results confirmed that proposed scheme achieves competitive accuracy compared to centralized training.

Currently, traditional RFF schemes cannot be directly applied to heterogeneous IoV network to provide rapid, reliable and lightweight authentication services due to the limited storage, computing resources and privacy. Model quantization is a crucial technology for reducing computing and memory requirements. Authors in [60] proposed a scheme that using integer weights for model inference. Experimental results indicated that proposed scheme successfully balances accuracy and inference latency. Different models have its own specific structural, authors in [61] proposed a customized quantization strategy for different layers. Authors in [63] explored the application of various quantization strategies in lightweight distributed semantic communication. Experimental results revealed that by pruning and quantizing, a compression rate of nearly 40 times can be achieved almost without losing performance. In summary, DEI and RFF can be utilized to authenticate heterogeneous nodes within IoV networks. However, existing traditional models lack interpretability and require huge training samples. The presence of redundant information significantly hampers the efficiency of learning. Consequently, efficiently extracting RFF characteristics, enhancing interpretability and privacy protection have become paramount issues within DEI framework. Furthermore, tradi-

tional model deployment imposes high demands on computing resources and memory, leading to significant inefficiencies and increased inference latency. Consequently, implementing privacy-protected, explainable, and efficient d istributed RFF within DEI-enabled intelligent IoV systems has become a significant challenge.

B. Contribution

To address the aforementioned challenges, we proposed FFSFNet which amalgamates fractional scattering and federated learning to achieve excellent identification performance within IoV network. Moreover, FFSFNet can significantly reduce redundancy and improve learning efficiency under the DEI framework, while enhancing the interpretability, deployment feasibility and privacy. Besides, model quantization can optimize the size while keeping performance loss within a controllable range. Furthermore, reducing its dependence on memory, inference latency, and greatly promotes its practical deployment within IoV network. Specifically, the main contributions of this paper are summarized as follows:

1) To mitigate data privacy concerns and reduce the substantial communication overhead associated with centralized data collection, storage, and sharing. Within DEI framework, federated learning is introduced to jointly train effective models for identification while preserving privacy. To further enhance learning efficiency, a novel residual network has been designed, capable of attaining remarkable identification with only 0.579M parameters.
2) The redundant information presents a significant challenge to the DEI efficiency. We develop novel FFSFNet to extract the multi-scale RFF characteristics embedded in non-stationary waveform. This brings a substantial redundantly reduction, thereby significantly improving learning efficiency, diminishing reliance on computing resources and enhancing model interpretability during the training process.
3) Furthermore, to tackle memory and computing resource constraints during practical deployment, we conducted a comprehensive comparative analysis of various quantization schemes and validated with FPGA accelerator. Experimental results demonstrate that FFSFNet can maintain up to 99.4% identification accuracy by only utilizing about 5.08% of original samples among 35 different nodes within heterogeneous IoV network. Model quantization can effectively reduce model size and inference latency with minimal performance degradation.

C. Organization

The remainder of our paper is organized as follows. In Section II, the signal structure, preprocessing and basic principles of wavelet scattering network are discussed. Then, the architecture of proposed deep fractional wavelet scattering network is discussed in Section III. Furthermore, the proposed deep federated fingerprinting framework based on fractional wavelet scattering network is presented in Section IV-A. In Section V, we introduce the detailed experiment setup and remarkable experimental results. Finally, the representative conclusion is drawn in Section VI.

II. System Model

A. Signal Structure and Preprocessing for Heterogeneous IoV Network

In this section, we explore the waveform characteristics of two distinct signals that have been gathered: LoRa which utilizes the chirp modulation and orthogonal frequency division multiplexing (OFDM). Our emphasis will be specifically directed towards an in-depth investigation of the mechanism involved in RFF characteristics generation within these two different waveform. Firstly, a complete LoRa signal consists of three parts: preamble, delimiter and payload symbols which are modulated by the linearly frequency. Then, the modulated signal is processed by digital-to-analog converter and a matched power amplifier. At time t it can be represented as:

$$x_L(t) = Ae^{j2\pi\left(\omega_{min}+\frac{1}{2}\tau t+\Delta\omega\right)t} \quad (0 \le t \le T), \tag{1}$$

where A represents the amplitude, ω_{min} is the minimum operating frequency which equals to $-\frac{B}{2}$. Additionally, B represents the operating bandwidth and the working band range can be denoted as $[-\frac{B}{2}, \frac{B}{2}]$. Besides, T denotes the symbol duration and $\tau = \frac{B}{T}$ represents the frequency sweep rate. Furthermore, $\Delta\omega$ denotes the specific frequency offset. For ease of representation, we discretize the original signal and the received can be expressed as:

$$y_L\left(n\right) = H_L * I_L\left(x_L(n)\right) + N_L, \tag{2}$$

where H_L represents the channel, I_L denotes the hardware impairments and N_L represents the channel noise. OFDM employs multi-carrier modulation, allocating multiple orthogonal sub-carriers in frequency domain to achieve high throughput transmission. The k-th data symbol resulting from bit stream modulation is denoted as $F(i)$. Then, the inverse fast Fourier transform (IFFT) of the frequency signal can be expressed as:

$$x_O(n) = \frac{1}{M}\sum_{i=0}^{M-1} F(i)e^{j2\pi in/M} \qquad 0 \le n \le M-1, \tag{3}$$

where n denotes the discrete samples index. Furthermore, frequency and phase mismatches between transmitter and receiver lead to CFO and phase offset. The received signal can be expressed as:

$$y_O(n) = H_O * D_O(x_O\left(n\right)) + N_O, \tag{4}$$

where D_O denotes the non-linear RF characteristics, H_O represents the wireless channel, N_O denotes the system noise which also satisfies the Gaussian distribution. In summary, despite significant variations in modulation and transmission methods between LoRa and OFDM, they pass through the similar RF modules and all the corresponding RFF impairments characteristics have been embedded into the waveform.

B. The Basic Principles of Wavelet Scattering Network

In numerous situations, the characteristics of signal exhibit significant variations across time and spatial dimensions. These variations are often compounded by noise and deformation, which pose challenges to effective RFF features extraction and identification. The wavelet transform can be

utilized to its variable scaling and multi-resolution analysis. Specifically, for input signal $x(t)$ that are continuous over a finite time interval, the wavelet transform can be defined as

$$W_x(a,b) = \frac{1}{\sqrt{a}} \int_{\mathbb{R}} x(t)\psi^*(\frac{t-b}{\sqrt{a}})dt, \tag{5}$$

where a and b serve to adjust the wavelet's scale and its temporal shift along the time axis t, respectively. The mother wavelet $\psi(t)$, upon undergoing transformations via a scaling parameter j and a rotation parameter z, yields a collection of wavelet sets characterized by varied scales and orientations as follows

$$\psi_\lambda(u) = 2^{2j}\psi(2^j z^{-1}u), \tag{6}$$

where $\lambda = 2^j z$ signifies the composite parameter in the wavelet transform, encapsulating both scale and rotation details. Wavelet filter banks excel in dissecting and seizing both information and energy across various scales and directions from a given signal. The convolution between these filters and $x(t)$ fundamentally can be defined as a process of features extraction.

$$\mathcal{R}'[\lambda]x = |x(t) * \psi_\lambda|, \tag{7}$$

where $\mathcal{R}'[\lambda]$ denotes the modulus operator. Besides, $*$ represents the convolution operation. Subsequently, through the convolution of the wavelet modulus coefficients with the scale function $\phi(t)$, we are able to distill the low-frequency elements of the signal. This process yields the translation-invariant scattering wavelet coefficients.

$$\mathcal{O}'[\lambda]x = |x(t) * \psi_\lambda| * \phi(t), \tag{8}$$

where $\phi(t)$ denotes the scale function and $\mathcal{O}'[\lambda]$ denotes the related calculation process. Considering that the low-frequency part reflects the large-scale geometric characteristics of the signal, it exhibits strong invariance to local changes such as translation, rotation, and scaling. Therefore, we introduce a low-pass filter $\phi(t)$ to extract the low-frequency components of the signal, ensuring translational invariance and deformation stability of the features.

During the wavelet scattering transform, operations involving modulus tend to emphasize low-frequency characteristics while inadvertently diminishing high-frequency details. To mitigate the loss of high frequency characteristics, the transform employs iterative steps at elevated levels, incorporating additional modulus operations and low-pass filtering. Throughout these iterations, the signal undergoes dispersion across various scales and orientations via distinct paths. Upon reaching the k layer, this calculation process can be denoted as follows

$$\begin{aligned}
\mathcal{O}'_{k-1}x(u,\lambda_1,...,\lambda_{k-1}) &= |...||x * \psi_{\lambda_1}| * \psi_{\lambda_2}|... * \psi_{\lambda_{k-1}}| \\
&\quad * \phi_J(t) \\
&= \mathcal{R}'_{k-1}x(u,\lambda_1,...,\lambda_{k-1}) * \phi_J(t), \\
\mathcal{R}'_k x(u,\lambda_1,...,\lambda_k) &= |...||x * \psi_{\lambda_1}| * \psi_{\lambda_2}|... * \psi_{\lambda_k}| \\
&= \mathcal{R}'_{k-1}x(u,\lambda_1,...,\lambda_{k-1}) * \psi_{\lambda_k},
\end{aligned} \tag{9}$$

where $\mathcal{R}'_k$ denotes the high frequency coefficients calculation path of k layer. It can be found from (9) that the scattering network establishes a unique form of hierarchical convolutional network by sequentially employing complex wavelet operators and modulus operations. This approach ensures that the translation-invariant coefficients $\mathcal{O}'_k$ which can be calculated layer by layer. Concurrently, the wavelet modulus coefficients $\mathcal{R}'_k$, can be relayed to the subsequent layer of the network for additional processing, thereby enhancing the accurate and robustness of feature extraction.

III. The Architecture of Deep Fractional Scattering Network

A. The Basic Principles of Fractional Wavelet Transform

The fractional Fourier transform (FRFT) facilitates the transformation of time series signals into the Fourier domain. For a d-dimensional signal, the FRFT is mathematically defined as follows:

$$X_\alpha(\tau) = \mathcal{F}^\alpha\{x(t)\}(\tau) \equiv \int_{\mathbb{R}^d} x(t)\mathcal{Q}_\alpha(\tau,t)dt, \tag{10}$$

where $\mathcal{Q}_\alpha(\tau,t) = \prod_{i=1}^d \mathcal{Q}_{\alpha_i}(\tau_i,t_i)$, and

$$\mathcal{Q}_{\alpha_i}(\tau_i,t_i) = \begin{cases} P_{\alpha_i} e^{j\frac{\tau_i^2+t_i^2}{2}\cot\alpha_i - jt_i\tau_i\csc\alpha_i}, & \alpha_i \neq n\pi, \\ \delta(t_i-\tau_i), & \alpha_i = 2n\pi, \\ \delta(t_i+\tau_i), & \alpha_i = (2n-1)\pi, \end{cases} \tag{11}$$

where α_i denotes the fractional rotation angle. Besides, $P_{\alpha_i} = \sqrt{\frac{1-j\cot\alpha_i}{2\pi}}$ represents the corresponding fractional sacle factors and $n \in \mathbb{Z}$. Specifically, when the fractional rotation angle $\alpha_i = \pi/2$, the FRFT simplifies into the classical Fourier transform (FT). Therefore, FRFT can obtain the fractional components of input signal, offering a flexibility surpassing that of conventional FT. However, the FRFT obscures the time-varying aspects of signals by integrating across the entire time axis, thus failing to capture the spectrum information within localized time windows. Fortunately, fractional wavelet transform (FRWT) can maintain the local time-frequency characteristics of signals. For any signal $x(t) \in \mathbb{R}^d$, the d-dimensional FRWT can be defined as

$$W_x^\alpha(\lambda,t) = \int_{\mathbb{R}^d} x(\tau)\psi^*_{\alpha,\lambda,t}(u)du, \tag{12}$$

where the d-dimensional fractional wavelet kernel function is detailed as

$$\psi_{\alpha,\lambda,t}(u) = \frac{1}{\sqrt{|\lambda|^d}}\psi\left(\frac{u-t}{\lambda}\right)e^{-j\frac{u\Omega_0 u - t\Omega_0 t}{2}}, \tag{13}$$

where λ and t denote the scale and time shift, respectively. The FRWT across various scales equivalent to the application of band-pass filters. Notably, when $\alpha = \pi/2$, the FRWT convert into the traditional wavelet transform (WT). The rotation angle α is a key parameter in the fractional wavelet characteristics. An appropriate α can improve the distinguishability of RFF scattering coefficients. During the experiment, this critical hyperparameter $\alpha = \pi/4$ was selected through multiple parameter evaluations to extract the scattering coefficients. This parameter affects the characteristics of fractional wavelet basis function and also impacts the scattering coefficients which

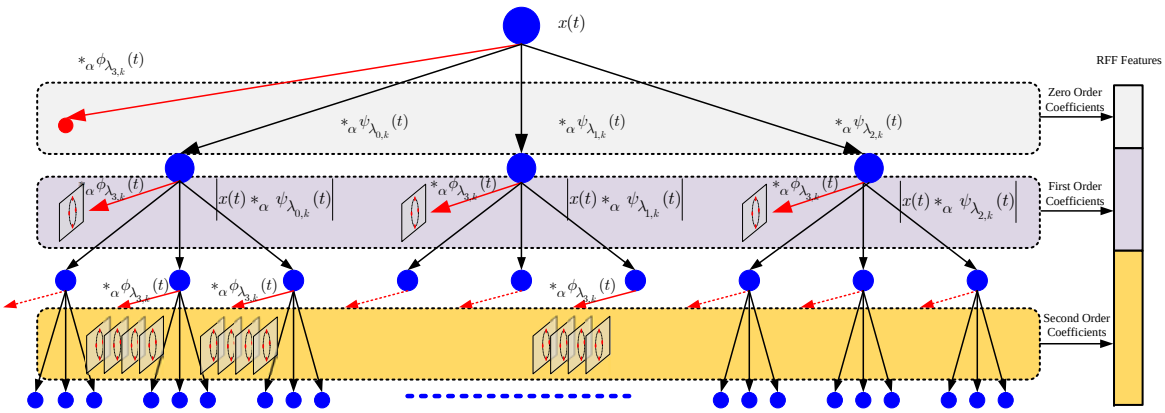


Fig. 3. The architecture of fractional wavelet scattering network.

contain RFF features. Therefore, it is essential to perform grid search or Bayesian parameters selection optimization in practical applications. This ensures that the fractional wavelet scattering network can effectively represent RFF features, thereby improving the model's performance.

B. The Architecture of Fractional Wavelet Scattering Network

In the analysis of the multidimensional $x(t)$, we introduced FRWT to capturing the non-stationary signal's variation characteristics across distinct orientations. This enhancement entails two pivotal steps: firstly, the orientation of each individual fractional wavelet $\psi(t)$ is modified using a rotation factor z_n; secondly, scale 2^q is utilized to appropriately scale the rotated wavelets. By integrating the dual principles of directionality and scaling, the signal can be accurately analyzed in multiple directions with various scales. The discrete fractional directional wavelets can be denoted as $\psi_{\alpha,\lambda_{q,n},t}(u)$, where $\lambda_{q,n} = 2^q z_n$ represents the combination of rotation and scale adjustment. Therefore, the calculation process of FRWT can be equivalent to a special set of filters which can decompose the signal into detailed and the general parts. More specifically, the signal's low-frequency components of $x(t)$ can be obtained with 2^J and the related calculation process can be denoted as

$$\mathcal{Q}_{\alpha,J}x(t) = x(t) *_\alpha \phi_{2^J}(t), \tag{14}$$

where $\mathcal{Q}_{\alpha,J}x(t)$ represents the basic general low-frequency components. Besides, $\phi_{2^J}(t)$ denotes the fractional scale function endowed with low-pass attributes and can be defined by

$$\phi_{\lambda_J}(t) \equiv \frac{1}{|\lambda_J|^d}\phi\left(-\frac{t}{\lambda_J}\right), \tag{15}$$

Furthermore, the signal's high-frequency components which obtained with scale $2^q \leq 2^J$ represents the detailed characteristics and can be denoted as

$$W_{\alpha,x}(\lambda_{q,n},t) = x(t) *_\alpha \psi_{\lambda_{q,n}}(t), \quad 1 \leq n \leq N, \tag{16}$$

where $\psi_{\lambda_{q,n}}(t) \triangleq \frac{1}{|\lambda_{q,n}|^d}\psi^*\left(-\frac{t}{\lambda_{q,n}}\right)$ encapsulates the fractional wavelet's extraction capability for high-frequency characteristics. Besides, $W_{\alpha,x}(\lambda_{q,n},t)$ represents the high-frequency detailed characteristics. The potential minor translations within the signal complicates its practical application. To address the issue and maintain relative translational invariance, the modulus operation is proposed and can be defined as

$$\mathcal{R}[\lambda]x(t) = |W_{\alpha,x}(\lambda_{q,n},t)| = |x(t) *_\alpha \psi_{\lambda_{q,n}}(t)|, \tag{17}$$

where $\mathcal{R}$ denotes the complete modulus operation. Furthermore, the nonlinear coefficients are subsequently filtered by $\phi_{2^J}(t)$ to harness non-zero translational invariance which can be represented as

$$\begin{aligned}\mathcal{O}[\lambda]x(t) &= |W_{\alpha,x}(\lambda_{q,n},t)| *_\alpha \phi_{2^J}(t)\\ &= |x(t) *_\alpha \psi_{\lambda_{q,n}}(t)| *_\alpha \phi_{2^J}(t),\end{aligned} \tag{18}$$

where $\mathcal{O}$ denotes the calculation operator with multi-scale filters. It can be found that the structure of fractional wavelet scattering transform is extremely similar to deep convolutional networks. By conceptualizing this structure as a fractional wavelet scattering network, we can find that it not only inherits the hallmark features of traditional scattering networks, such as translational invariance and stability to deformations, but also suitable for non-stationary signal analysis. As shown in Fig. 3, the every path of FFSFNet can be denoted as $p^{(k)} = (p_1^{(k)}, p_2^{(k)}, ..., p_k^{(k)})$. Where the k represents the corresponding maximum path length and the coefficients of kth can be denoted as

$$\begin{aligned}&\mathcal{R}^\alpha[p^{(k)}]x(t) = \mathcal{R}^\alpha[p_k^{(k)}]\cdots\mathcal{R}^\alpha[p_2^{(k)}]\mathcal{R}^\alpha[p_1^{(k)}]x(t)\\ &= \left|\ldots\left|\left|x(t) *_\alpha \psi_{p_1^{(k)}}(t)\right| *_\alpha \psi_{p_2^{(k)}}(t)\right|\ldots *_\alpha \psi_{p_k^{(k)}}(t)\right|,\end{aligned} \tag{19}$$

where $p_i^{(k)} = 2^{q_i} z_n (q_i \leq J, 1 \leq i \leq k)$ represents the scale and rotation parameters of the fractional wavelets. Following the $\mathcal{R}^\alpha[p^{(k)}]x(t)$, applying the low-pass filter $\phi_{2^J}(t)$ yields the k th level output coefficients $\mathcal{O}^\alpha[p^{(k)}]x(t)$ as

$$\mathcal{O}^\alpha[p^{(k)}]x(t) = \mathcal{R}^\alpha[p^{(k)}]x(t) *_\alpha \phi_{2^J}(t), \tag{20}$$

where k denotes the layer of fractional wavelet scattering network and $\mathcal{O}^\alpha[0]x(t) = x(t) *_\alpha \phi_{2^J}(t)$ represents the low frequency characteristic of input signal. Fig. 3 illustrates the architecture of a fractional wavelet scattering network with three different levels. Within this architecture, the original input signal $x(t)$ goes through a series of level-wise processing, generating fractional order wavelet scattering coefficients $\mathcal{O}^\alpha[p^{(k)}]x(t)$ for different levels $k = 0, 1, 2$ which

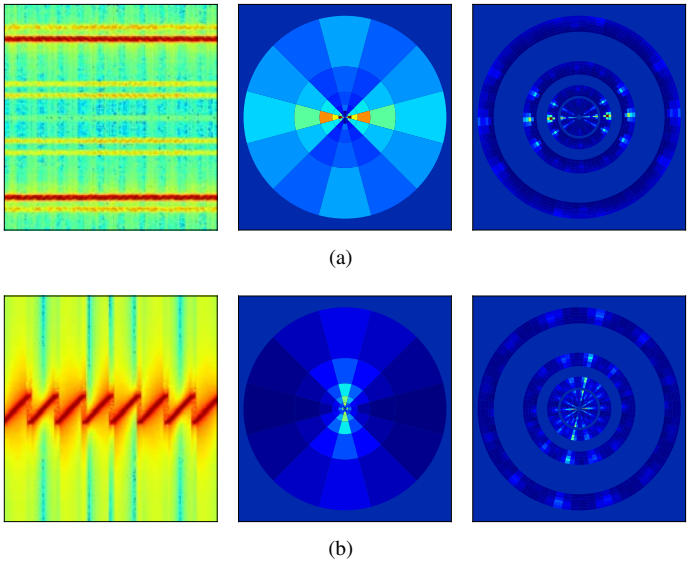


Fig. 4. The STFT and various fractional coefficients of different scattering layers. (a) Type1-6 nodes: STFT, 1-th fractional wavelet coefficients, 2-th fractional wavelet coefficients (left, middle, right); (b) Type7 nodes: STFT, 1-th fractional wavelet coefficients, 2-th fractional wavelet coefficients (left, middle, right).

constitute the output of scattering network and form a sample. Furthermore, the fractional wavelet transmission coefficients $\mathcal{R}^{\alpha}[p^{(k)}]x(t)$ from one layer serving as the input for subsequent layer. Specifically, the first layer outputs the fractional scattering coefficients $\mathcal{O}^{\alpha}[P^{(0)}]x(t)$ and the fractional wavelet transmission coefficients $\mathcal{R}^{\alpha}[P^{(1)}]x(t)$ can be calculated by

$$\begin{aligned} &\mathcal{O}^{\alpha}[P^{(0)}]x(t) = \mathcal{R}^{\alpha}[P^{(0)}]x(t) *_{\alpha} \phi_{2^J}(t) = x(t) *_{\alpha} \phi_{2^J}(t) \\ &\mathcal{R}^{\alpha}[P^{(1)}]x(t) = \{\left|x(t) *_{\alpha} \psi_{p_1^{(1)}}(t)\right|\}_{p_1^{(1)} \in \Omega}, \end{aligned} \tag{21}$$

where $\Omega = \{\lambda_{q,n} = 2^q z_n | 2^q \leqslant 2^J, 1 \leq n \leqslant N\}$ encompasses all feasible scale-orientation combinations of the fractional wavelets. For k-th layer of network, both the fractional wavelet transmission coefficients $\mathcal{R}^{\alpha}[P^{(k)}]x(t)$ and the scattering coefficients from the preceding layer $\mathcal{O}^{\alpha}[P^{(k-1)}]x(t)$ to be output are denoted as

$$\begin{aligned} &\mathcal{R}^{\alpha}[P^{(k)}]x(t) = \mathcal{R}^{\alpha}[p_k^{(k)}]\mathcal{R}^{\alpha}[P^{(k-1)}]x(t), \forall l^{(k)} \in \Omega^k \\ &\mathcal{O}^{\alpha}[P^{(k-1)}]x(t) = \mathcal{R}^{\alpha}[P^{(k-1)}]x(t) *_{\alpha} \phi_{2^J}(t), \end{aligned} \tag{22}$$

Through this approach, the network can process the signal in a hierarchical manner, extracting the signal's fractional wavelet scattering coefficients which contains the related RFF features layer by layer. According to the principle of energy conservation, as the number of layers m in the fractional wavelet scattering network increases, the energy of network transmission signal will gradually decrease and eventually approach zero [49]. Energy is retained in the scattering coefficient of the different layers' output. It can be observed from (22) that when the number of network layers n is greater than or equal to m, the energy $\mathcal{O}^{\alpha}[P^{(k-1)}]x(t)$ of the fractional scattering network will also tend to zero. This indicates that the depth of fractional scattering network can be controlled within a certain range, and the signal's energy and information loss can be considered negligible.

As illustrated in Fig. 4, the short-time Fourier transform (STFT) and the fractional wavelet scattering coefficients across various layers are presented. Specifically, the STFT spectrum for the first type of node is displayed on the left in Fig. 4(a), highlighting the evolution of signal frequency over time. Here, the intensity of the color signifies the energy magnitude. The middle diagram of Fig. 4(a) showcases the distribution of first-level scattering coefficients, represented as a circular ring divided into different sectors which denotes a specific frequency range. The right diagram presents the distribution of the second layer scattering coefficients, offering a more detailed frequency segmentation. Similarly, Fig. 4(b) presents the STFT and scattering coefficients for another type nodes in an analogous fashion. In the first layer of the scattering network, frequencies organize into a circular structure at a distinct scale 2^{j_1}, termed as binary ring. Where the j_1 represents the scale factor in the fractional wavelet, which is primarily responsible for scaling the fractional wavelet basis function to varying degrees. This configuration is subdivided into multiple sectors, each characterized by a unique rotational angle z_1 which denotes the different rotation directions and fractional wavelets $\psi(u)$ can be rotated by different rotation factors z_1. Besides, Ω represents different quadrant regions, indicating different frequency domain positions, and is related to the scale factor 2^{j_1}, as shown in Fig. 4 for different dyadic annuli $\Omega[2^{j_1}z_1]$. For exploring the second layer, it not only referred to the scale and direction of the primary level, but also further divided the fan shapes of the primary level. Furthermore, these quadrants are divided into rotating sectors according to the angle vector, forming a more refined sector which can be denoted as $\Omega[2^{j_1}z_1, 2^{j_2}z_2]$. In summary, the scattering network can be utilized to refine frequency partitioning from various levels, while capturing the multi-scale nature of complex signals.

C. The Deformation Stability of Fractional Wavelet Scattering Network

It should be emphasized that wireless signals are inevitably affected by noise during transmission which may cause certain deformations and impact the RFF characteristics extraction. It is imperative for fractional wavelet scattering network to accommodate these fractional deformation which can be defined as

$$\mathcal{T}_{\varepsilon}^{\alpha}x(t) = x(t - \varepsilon(t))e^{-j\varepsilon(t)\left(t - \frac{\varepsilon(t)}{2}\right)\cot\alpha}, \tag{23}$$

where $\mathcal{T}_{\varepsilon}^{\alpha}$ represents the corresponding operator and $\varepsilon(t)$ denotes the fractional deformations. If $\alpha = \pi/2$, the deformations will degenerate into traditional forms.

$$\mathcal{T}_{\varepsilon}x(t) = x(t - \varepsilon(t)), \tag{24}$$

Furthermore, under the assumption of existing a constant $\mathcal{L}$ and any signal $x(t) \in L^2(\mathbb{R}^d)$ satisfying deformation gradient $|| \bigtriangledown \varepsilon||_{\infty} \leq 1/(2d)$, the specific error caused by fractional deformations can be limited within an acceptable range and can be denoted as

$$\|\mathcal{O}^{\alpha}[P]\mathcal{T}_{\varepsilon}^{\alpha}x - \mathcal{O}^{\alpha}[P]x\|_2 \leq \mathcal{L}\Gamma_{\alpha}(\varepsilon)\,\|\mathcal{R}^{\alpha}[P]x\|_2\,, \tag{25}$$

where $\Gamma_\alpha(\varepsilon)$ quantifies the deformation constraint and defined as following

$$\begin{aligned}\Gamma_\alpha(\varepsilon) \triangleq & 2^{-J}\|\varepsilon\|_\infty + \left(J\|\nabla\varepsilon\|_\infty + \left\|\nabla^2\varepsilon\right\|_\infty\right)\delta_{\alpha-\frac{\pi}{2}} \\ & + \left(\|\nabla\varepsilon\|_\infty + 2\sqrt{2}\left(1-\delta_{\alpha-\frac{\pi}{2}}\right)\right.,\end{aligned} \tag{26}$$

where $\delta_{\alpha-\frac{\pi}{2}}$ denotes a adjustment factor which only contains two different factors [0,1]. When the $\alpha = \pi/2$, the $\delta_{\alpha-\frac{\pi}{2}} = 1$ and all other cases are 0. Stability is crucial for fractional wavelet scattering networks, especially when dealing with deformed signals. By controlling negative effects within specified limits, fractional wavelet scattering network demonstrate robustness against subtle deformation.

IV. The Proposed Efficient Deep Federated Fingerprinting Framework

A. The Architecture of Deep Federated Framework for Fingerprinting

In the heterogeneous architecture of the IoV, we categorize a myriad of intelligent nodes into distinct groups, illustrated in Fig. 2. Each group can be treat as an operational unit, inherently unable to engage in centralized data collection and model training. To address this, federated learning is involved to achieve the distributed learning among the different nodes and without aggregating the data to any central server. Within this architectural paradigm, the federated averaging (FedAvg) scheme serves as a core strategy and can be characterized as:

$$\min_w \ f(w) = \sum_{i=1}^{N} p_i F_i(w) = \mathbb{E}_i[F_i(w)], \tag{27}$$

where p_i represents the probability of client i being selected. Besides, $F_i()$ is the local objective function defined on client i and N is the number of clients. Furthermore, w is the global model weight value, each client performs computing tasks locally to optimize its local objective function $F_i(w)$, while contributing to the global model.

However, the various groups of clients, as depicted in Fig. 2 face varying computational and resource constraints due to hardware and network differences. The FedAvg algorithm, requiring consistent local updates from all clients, overlooks the limitations of less capable nodes. We address this by introducing the FedProx, which adds a proximal term $\frac{\sigma}{2}\|w-w^t\|^2$, penalizing deviations from the global model to ensure that local updates remain closely aligned with the global.

$$\min_w y_i(w;\ w^t) = F_i(w) + \frac{\sigma}{2}\|w-w^t\|^2, \tag{28}$$

where $\frac{\sigma}{2}\|w-w^t\|^2$ penalizes the discrepancy between client's model w and the global model w^t. Besides, σ is a non-negative parameter adjusting the penalty level which can control the distance between local and global model. In our practical implementation, we set $\mu = 0.5$, determined through a grid search method. The additional proximal term $\|w-w^t\|^2$ can effectively limit the impact of deviated local updates, thereby reducing fluctuations in global model performance. Although this proximal term may lead to a temporary decrease in model accuracy for some clients, as their updates are no longer solely optimized based on their own data, it helps maintain the stability and consistency of the global model from a global perspective. More importantly, by suppressing inconsistent local updates, the proximal term significantly improves the learning efficiency of the system when handling distributed heterogeneous data, thereby optimizing the overall performance of the federated learning system. To address the local optimization, we introduce the θ_i^t-inexact solution, allowing clients to aim for an approximate rather than an exact optimal solution. Specifically, in tackling the local problem (28), it is sufficient to find an θ_i^t-inexact solution $\hat{w}$, that it satisfies the following criteria:

$$\left|\left|\nabla y_i(\hat{w};\ w^t)\right|\right| \leq \theta_i^t \left|\left|\nabla y_i(w;\ w^t)\right|\right|, \tag{29}$$

where θ_i^t acts as a relaxation factor within the range $[0,1]$, setting the tolerable limit for gradient inaccuracy. The proposed FedProx schemes proceeds as follows: the server randomly selects K clients from a total of N and forms a subset Ω_t. The server then sends the current global weights w^t to these selected clients. Each client i employs stochastic gradient descent (SGD) to update the model locally. Then, the updated model weights can be denoted as

$$w_i^{t+1} = w_i^t - \lambda_1 \nabla y_i(w;\ w^t), \tag{30}$$

where $\nabla y_i(w;\ w^t) = \nabla F_i(w) + \sigma(w - w^t)$ denotes the gradients update, λ_1 represents the client learning rate and w_i^{t+1} is an θ_i^t-inexact solution that satisfies (29). Upon completion of the local updates, each client sends their updated model gradient ∇_i^t back to the server. Subsequently, the server aggregates these gradients by calculating their average from all clients. This aggregated gradient is then utilized to update the global model's gradient.

$$\nabla_t = \frac{1}{K}\sum_{i\in\Omega_t} \nabla_i^t, \tag{31}$$

where ∇_t denotes the global updates. The server proceeds to update the weights by employing the Adam optimization.

$$w^{t+1} = w^t + \lambda_2 \frac{v_t}{\sqrt{z_t}+\varepsilon}, \tag{32}$$

where λ_2 represents the global learning rate, and ε is a very small fixed values to avoid division by zero. Momentum v^t and second moment z^t can be expressed as:

$$\begin{aligned} v^t &= \xi_1 v^{t-1} + (1-\xi_1)\nabla_t, \\ z^t &= \xi_2 z^{t-1} + (1-\xi_2)\nabla_t^2, \end{aligned} \tag{33}$$

where ξ_1 and ξ_2 are the decay rates of the gradient and its square respectively. Considering the practical performance, we employed the recommended values $\xi_1 = 0.9$ and $\xi_2 = 0.999$ during our experiment [62]. In this way, the proposed scheme can coordinate model training on multiple devices and ultimately summarize a globally optimized model efficiently.

B. The Architecture of Designed Models

As depicted in Fig. 5, we introduce ResNet-1d alongside other benchmark models. In Fig. 5(a), the ResNet-1d architecture initiates by convolving the input with 64 linear filters of size 7×1, aiming to capture feature information across

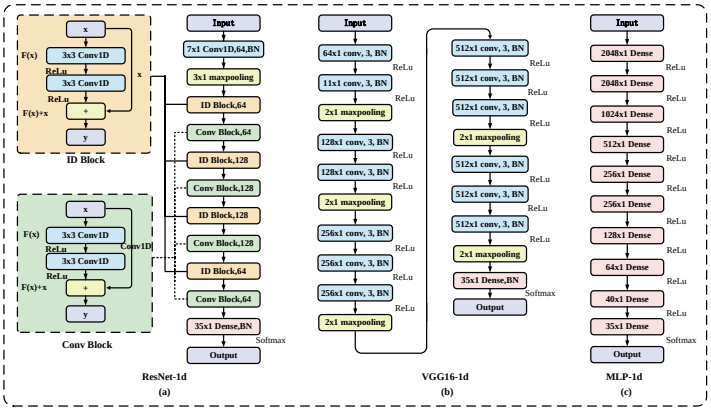


Fig. 5. The architecture of proposed ResNet-1d and other benchmark models.

various temporal scales. Furthermore, increasing the filter size (17×1 or 70×1) can cover a longer span, potentially capturing features over a longer scale and better capturing lower frequency information. However, this may overlook some high-frequency details, causing the model to miss subtle features. Additionally, larger filters require more computing resources and significantly longer training times. Subsequently, this architecture incorporates two distinct residual structures: the ID Block and the Conv Block. The ID Block comprises two convolutional layers of size 3×3 and an additional additive unit that directly merges the input with the output. In contrast, the Conv Block employs our additive module, which combines the convolutional result of the input with output. Each Conv Block is followed by a ReLU activation function to mitigate common issues related to gradient vanishing. If Conv Blocks and ID Blocks are not alternately cascaded, the model may become deeper but harder to train, posing a risk of gradient vanishing or exploding. Alternating ConvBlocks and IDBlocks balances feature extraction and critical information transmission, improving training effectiveness and model performance.

As illustrated in Fig. 5(b), the VGG16-1d also embraces one-dimensional data input and adeptly handles samples via 5 sets of convolution layers with varying depths. Commencing from the input layer, the network initiates feature extraction employing a sequence of convolutions with progressively increasing kernel sizes. ReLU activation and batch normalization layers are applied to augment the model's non-linearity and stability. Ultimately, the 35x1 fully connected and softmax layer are utilized to output the final decision. As shown in Fig. 5(c), the multilayer perceptron (MLP-1d) consists of straightforward yet versatile fully connected neural network model comprising 10 distinct Dense layers. The model initiates from a higher dimension size, progressively diminishing to 35, enabling it to acquire intricate RFF features from the input. Furthermore, ReLU activation layer is applied to introduce nonlinearity, while the Softmax layer is utilized to produce the ultimate classification probabilities.

C. Network Model Quantification and Acceleration

With the advancement of AI and models deployment, DEI has emerged as a critically important potential application architecture. However, the deployment of models at the edge

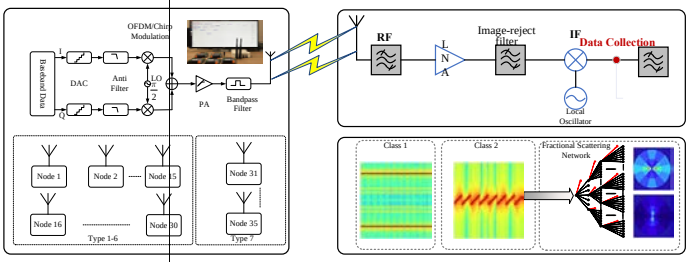


Fig. 6. The architecture of typical data collection system which is utilized to acquire the large-scale RF raw data.

is hindered by constraints such as limited storage, memory, power consumption and latency, as these models typically demand substantial computing resources. Model quantization represents a pivotal technique in addressing these challenges. It reduces the model size and power consumption while maintaining performance by converting the model's high-precision floating-point weights into lower precision floating-point or fixed-point. However, careful optimization is necessary during the quantization process to minimize accuracy loss, which adds complexity to the procedure. The activation function enhances the model's ability to capture nonlinear characteristics, crucial for learning complex functions. Thus, accurately quantizing the activation function is essential to preserve the accuracy. Model quantization comprises two primary components: weight and activation quantization. Given the distinct architecture of distributed training, this study employs a post-quantization training strategy. The global model's weights w^t initially maintain 32-bit floating-point and will be converted into an $\bar{m}$-bit integer.

$$\tilde{\mathbf{w}}_{i,j,n}^t = \mathcal{R}\left(S_{w,\bar{m}}\left(\mathbf{w}_{i,j,n}^t - \min\left(\mathbf{w}^t\right)\right)\right), \tag{34}$$

where $S_{w,\bar{m}}$ represents the corresponding scaling factor, which determines the quantization of floating-point precision to the corresponding $\bar{m}$-bits integer and can be annotated as

$$S_{w,\bar{m}} = \frac{2^{\bar{m}} - 1}{\max\left(\mathbf{w}^t\right) - \min\left(\mathbf{w}^t\right)}, \tag{35}$$

To optimize quantization accuracy of the activation function, one could increase the bit width of the activation functions, followed by re-quantizing these results to a predetermined bit width. Moreover, the presence of outliers in the activation function's output may expand the quantization range excessively and degrade the quantization accuracy within the effective range. To address this issue, using an empirical moving average (EMA) can effectively manage these outliers, ensuring more stable quantization outcomes [63]. In summary, model quantization is designed to minimize model size, reduce memory requirements, and enhance inference efficiency. However, it is important to note that these benefits may be accompanied by potential reduction of the model accuracy, impacting inference performance to some extent.

V. Experiment Setup and Results Analysis

A. Experiment Setup

In this paper, we have developed a data collection system specifically engineered to capture large-scale RF raw samples.

TABLE I
LABELS OF IOV NODES.

Nodes	Label
	Training Dataset
Type1 Nodes	1, 2, 3, 4, 5
Type2 Nodes	6, 7, 8, 9, 10
Type3 Nodes	11, 12, 13, 14, 15
Type4 Nodes	16, 17, 18, 19, 20
Type5 Nodes	21, 22, 23, 24, 25
Type6 Nodes	26, 27, 28, 29, 30
Type7 Nodes	31, 32, 33, 34, 35

This system collects various raw signal samples from 35 different nodes, as detailed in Table I. As depicted in Fig. 6, the datasets includes two different classes of terminals. The first class of terminals includes type 1-6, each with five distinct nodes. As illustrated in Fig. 6, includes a series of components integral to signal processing: IQ modulator, filter, digital-to-analog converter (DAC) and power amplifier (PA). As the signal passes through these different modules, RF impairments are involved to the original waveform. To ensure thorough data collection, parameters are meticulously configured prior to channel filter selection. Upon receipt of a physical uplink shared channel signal from the terminal, the receiver captures demodulation reference signal symbols. The intermediate frequency of the collected signal is set at 140 MHz, with additional wireless subcarriers spaced around this central frequency. Complete frequency information is achieved with a sampling rate of 122.88 MHz, covering a frequency range from 17.7575 to 19.3625 MHz, where each subcarrier spans a bandwidth of 15 kHz. The second class of terminals which includes type7 consisting of five distinct LoRa nodes. The complete LoRa packet structure comprises three main components: preamble, start frame delimiter (SFD) and effective data. The preamble is crucial for synchronizing and marking the start of frame, although it does not convey any substantive information. Despite the expectation that LoRa devices have identical preamble structures, minor hardware variations can cause slight, often unintentional, errors in the preambles of different nodes. At the receiver's end, the GNU Radio's file sink module is utilized to extract these preambles from the LoRa frames received by a USRP B210 [1].

Furthermore, the extensive raw samples gathered from the 35 different nodes of these two class is processed by a fractional wavelet scattering network to create the RFF datasets for subsequent analysis. Each node includes 5000 training samples and 200 testing samples. During our experiment, we have configured the scattering network parameters scale factors J, rotation factors Q, fractional factors a, b and corresponding network layers k according to the performance evaluation. Considering the practical deployment, the network layers is set as $k = 2$. As for the model parameters configuration, the learning rate is set to 0.0001, the batch size is 150, and the number of training epochs is 200 for centralized model learning. Within the distributed federated learning framework,

[1]https://github.com/tapparelj/gr-lora_sdr

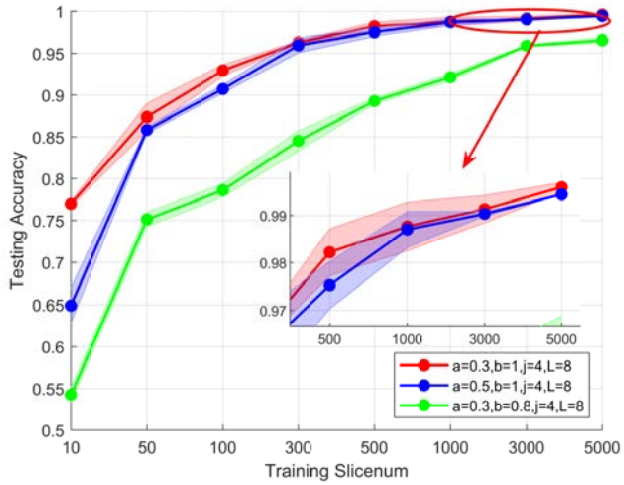


Fig. 7. The accuracy performance of different training samples and a, b with ResNet-1d $j = 4, L = 8$.

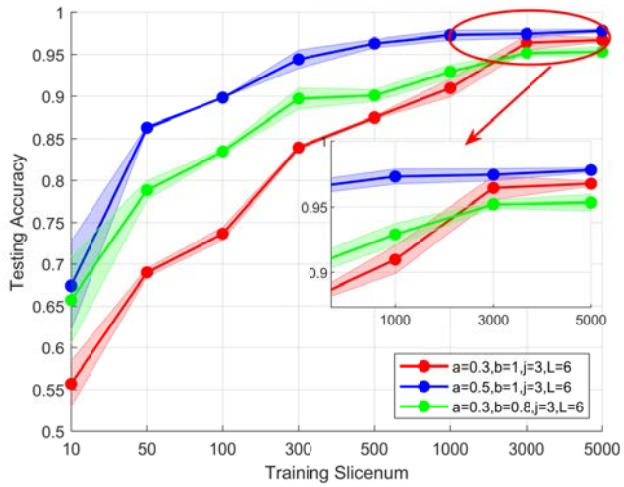


Fig. 8. The accuracy performance of different training samples and a, b with ResNet-1d $j = 3, L = 6$.

the learning rate of clients is set as 0.0001, and the global learning rate is set to 0.9. Ultimately, all those experiments were conducted on a Supermicro server equipped with 8 NVIDIA RTX 3090 Ti GPUs and 512 GB of RAM. To maximize the utilization of computing resources, we utilized 2 GPUs in parallel for each training session to complete the model's training and testing.

B. The Performance Impacts of Different Fractional Scattering Parameters

The parameterization of fractional wavelet scattering networks mainly involves the settings of wavelet filters and the optimal fractional orders. The experimental results, as illustrated in Fig. 7 and Fig. 8, highlight the effects of adjusting the scale size J and rotational angle L. In practical applications, the basic principle for selecting these two parameters is based on the length and directional characteristics of the input signal.

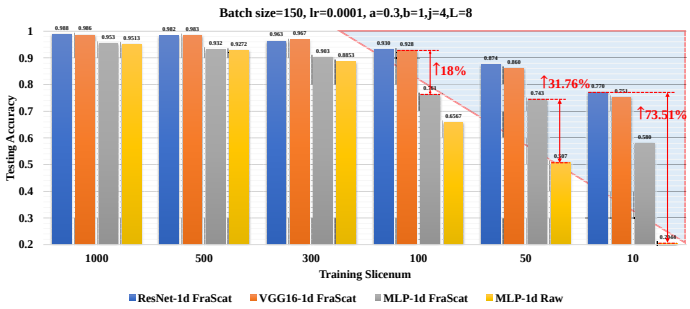


Fig. 9. The testing accuracy of different models and training samples with configuration of $a = 0.3, b = 1, j = 4, L = 8$.

Specifically, Fig. 7 employs settings of $J = 4$ and $L = 8$, while Fig. 8 corresponding to $J = 3$ and $L = 6$. These results demonstrate how the volume of training samples affects accuracy and evaluate the impact of different fractional order parameters including J, L, a and b. The findings confirm that testing accuracy decreases with a reduction in the number of training samples, regardless of the parameter configurations. Particularly, with the fractional parameters $J = 4$, $L = 8$, $a = 0.3$ and $b = 1$ yielded the best performance and higher stability, indicated by a smaller variance in the shadow area. In contrast, with the settings of $J = 3$ and $L = 6$, while the trend in testing accuracy was similar to that of $J = 4$, $L = 8$, various combinations of fractional parameters showed slightly different performances. Notably,the accuracy differences among different combinations of a and b are minimal with larger training samples. Furthermore, compared to Fig. 7, the larger variance shadow area in Fig. 8 indicates more significant result volatility. While larger J and L will bring larger network and scattering coefficients which may not bring better performance. It is crucial to select parameters according to the signal characteristics, optimal performance and computational efficiency. In conclusion, the experimental results emphasize that the optimal parameterization for the fractional orders $a = 0.3$ and $b = 1$, along with wavelet filter settings of $J = 4$ and $L = 8$, allows the wavelet scattering network to extract RFF features with higher separability, thereby achieving superior performance in practical application.

As illustrated in Fig. 9, we have conducted representative experiments to validate the identification accuracy performance of different models trained with varying numbers of slicenum. It can be found that when the training slicenum size is large, there is only little difference in testing accuracy among all models. However, as the training slicenum decreases, the performance gap between models begins to significantly increase. MLP-1d Raw exhibits the largest performance fluctuation range, with testing accuracy decreasing from 95.13% to 20.44%, indicating a high dependency on the training samples quantity. In contrast, MLP-1d FraScat, which undergoes fractional-order wavelet transform for feature extraction, significantly improves its performance with a reduction in sample quantity, especially with 50 training slicenums, where its testing accuracy exceeds MLP-1d Raw by 31.76%. VGG16-1d FraScat and ResNet-1d FraScat demonstrate better stability and higher testing accuracy. Particularly, ResNet-1d

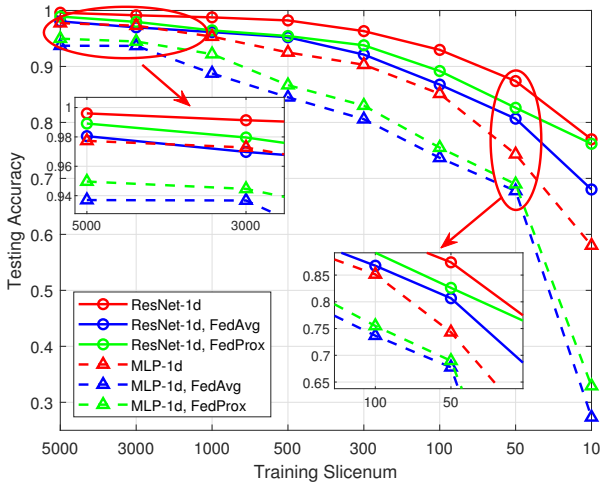
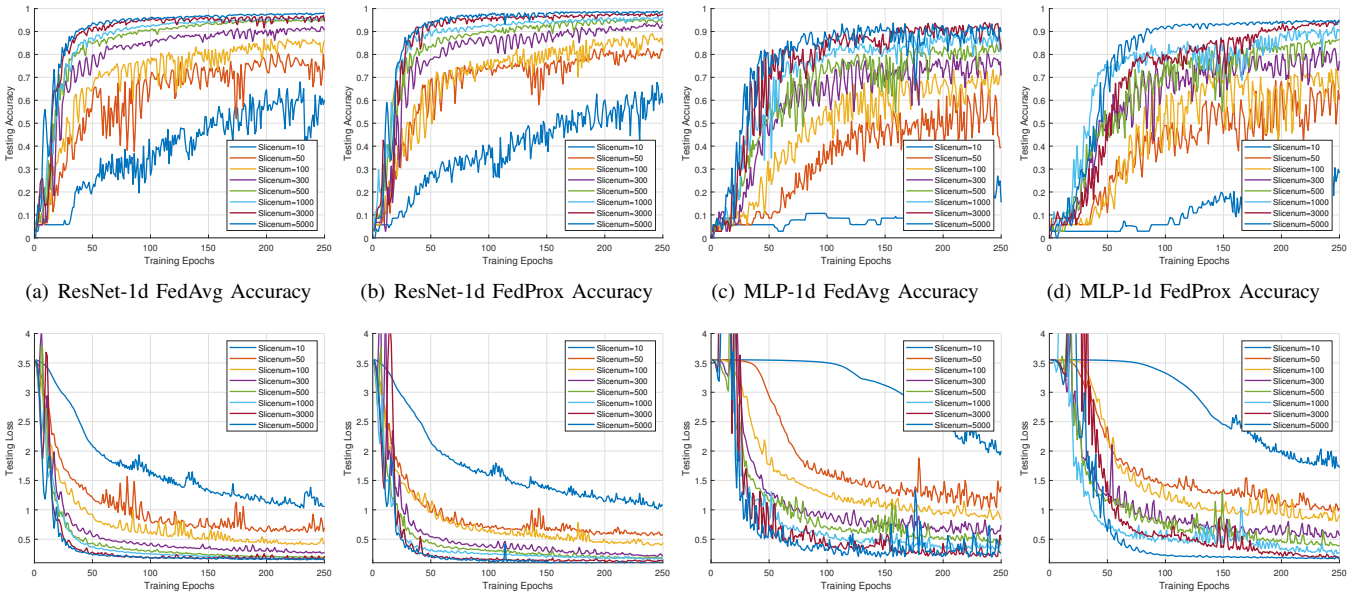


Fig. 10. The average testing accuracy of different federated learning schemes with 5000 training samples equal.

FraScat exhibits stronger learning efficiency and adaptability to the slicenum decreasing, surpassing MLP-1d Raw and MLP-1d FraScat by a significant margin, with a improvement of 73.51% with 10 training samples. Notably, ResNet-1d FraScat utilizes only 11.23% of the parameters compared to VGG16-1d FraScat, and as training slicenum decreases, the discrepancy in accuracy with the other three models gradually widens, indicating its powerful generalization and efficient learning ability. In summary, through comparative analysis of the performance of different models in response to changes in training sample quantity, ResNet-1d FraScat consistently maintains the highest testing accuracy in all scenarios, particularly demonstrating outstanding learning capability, efficient parameter utilization, and strong generalization ability when handling small-scale datasets.

C. The Performance Analysis of Different Federated Learning Schemes and Models

As illustrated in Fig. 10, the average testing accuracy of ResNet-1d and MLP-1d under different federated learning frameworks are presented. It can be found that with the decrease of training slicenum, the average testing accuracy of two models under centralized training and federated learning shows a downward trend, but the centralized training maintains an advantage in testing accuracy compared with federated learning in both MLP-1d and Resnet-1d models. Besides, we can see that the average testing accuracy of FedProx is higher than that of FedAvg. FedProx enhances model convergence stability and accuracy by introducing a proximal term based optimization on top of the FedAvg update strategy. Notably, as the training slicenum decreases, ResNet-1d shows a smaller accuracy decrease, while MLP-1d experiences a substantial decline. Particularly, the average testing accuracy of MLP-1d under both federated learning schemes experiences a significant drop compared to centralized training. ResNet-1D can effectively extract deep-level features of datasets with fewer samples, thereby achieving better identification accuracy, and



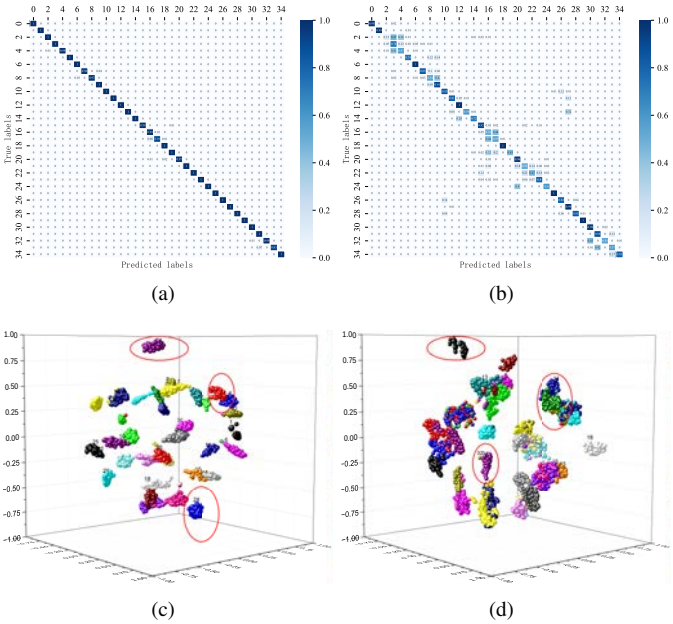
(a) ResNet-1d FedAvg Accuracy (b) ResNet-1d FedProx Accuracy (c) MLP-1d FedAvg Accuracy (d) MLP-1d FedProx Accuracy

(e) ResNet-1d FedAvg Loss (f) ResNet-1d FedProx Loss (g) MLP-1d FedAvg Loss (h) MLP-1d FedProx Loss

Fig. 11. The testing performance of various federate learning schemes via different number of training samples.

demonstrating superior generalization capabilities under federated learning.

As depicted in Fig. 11, the testing performance of ResNet-1d and MLP-1d networks under two federated learning framework are illustrated. We present the real-time performance of the aggregated models in terms of testing accuracy and loss after each round of updates. Fig. 11(a)-(d) show steady increase in testing accuracy of the aggregated models with the increase of training epochs, while Fig. 11(e)-(h) demonstrate the gradual decrease in the loss of the aggregated models. Fig. 11(a), (b) display the real-time testing accuracy of ResNet-1d under the two federated learning schemes, while Fig. 11(c), (d) show the real-time testing accuracy of MLP-1d under the same methods. It can be observed that despite the poor training performance of both federated learning schemes with 10 samples, FedProx generally outperforms FedAvg in terms of accuracy across different network models. Fig. 11(a), (c) illustrate the real-time testing accuracy of the two different network models under the FedAvg learning scheme, whereas Fig. 11(b), (d) depict the same under FedProx framework. Notably, ResNet-1d exhibits faster convergence and higher testing accuracy across different slicenum, with a smaller decline in testing accuracy observed for smaller slicenum. ResNet-1d demonstrates superior feature extraction capabilities, leading to better learning performance and generalization abilities.

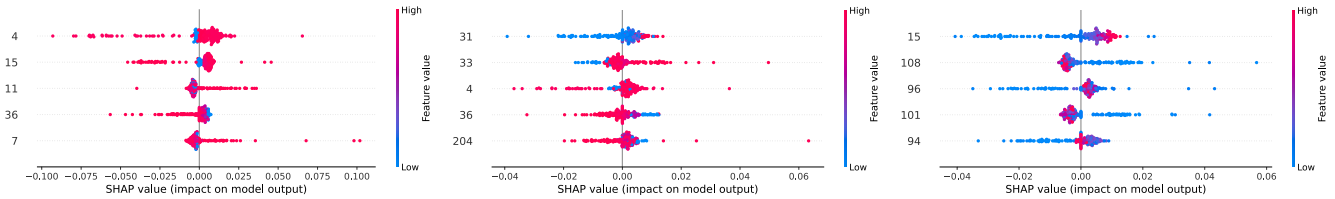


(a) (b)

(c) (d)

Fig. 12. Visualization of the confusion matrix and the t-SNE based RFF features with ResNet-1d FedProx. (a) Confusion matrix with 5000 training samples. (b) Confusion matrix with 1000 training samples. (c) The corresponding normalized t-SNE based RFF features of (a). (d) The corresponding normalized t-SNE based RFF features of (b).

As shown in Fig. 12, the confusion matrix and t-distributed stochastic neighbor embedding (t-SNE) based RFF features with ResNet-1d across different training slicenums are presented. It can be observed from Fig. 12(a) that there only a few nodes occurs errors with ResNet-1d and 5000 heterogeneous training slicenums. On the contrary, with the training slicenums decrease to 1000, the identification accuracy for nodes labeled as 2 is only 0.12 and the overall accuracy decreases significantly compared to the performance of 5000 training slicenums. Fig. 12(c)(d) visualize the RFF features extracted by ResNet-1d after nonlinear dimensionality reduction using t-SNE which is designed to explore high-dimensional structures, for training slicenum of 5000 and 1000, respectively. t-SNE is based on random neighborhood embedding and commonly utilized in manifold learning for dimensionality reduction. The core idea of t-SNE is to define a probability distribution over data samples in the high-dimensional space that represents the similarity between different samples. Then, the Kullback-Leibler (KL) divergence is employed to minimize



(a) (b) (c)

Fig. 13. Visualization of the SHAP value via different model, parameters and training samples. (a) ResNet-1d, 5000 samples; (b) ResNet-1d, 10 samples; (c) VGG16-1d, 5000 samples.

the divergence between these different distributions. t-SNE requires careful optimization of several parameters, including perplexity, early exaggeration, learning rate, iterations, initialization, and random state. Among these, perplexity is related to the number of nearest neighbors used in manifold learning algorithms, with larger datasets necessitating higher perplexity values. The learning rate helps the cost function avoid local minima during global optimization. The random state determines the random seeds used during initialization, influencing the local minimum and potentially leading to slightly different final results. Consequently, random seed are generally fixed during t-SNE execution. In Fig. 12(c), distinct clusters are formed for each type of nodes, distinguished by different colors and shapes, indicating the model's high identification capability. This suggests that ResNet-1d effectively captures and distinguishes RFF features of different nodes with 5000 training samples. In contrast, as shown in Fig. 12(d), with a reduced training slicenum of 1000, the boundaries between clusters become blurred, and features of different nodes are more mixed, indicating a decrease in the model's feature discrimination performance. This is consistent with the results of confusion matrix. Under different sample sizes, devices with identification errors are often misclassified as adjacent nodes, suggesting that even with insufficient samples, the model maintains a certain identification logic. Particularly, the differentiation between different nodes is well demonstrated, thanks to their significant differences in RFF features.

Fig. 13 illustrated the characteristics of SHAP values obtained through different model parameters and training samples. It identifies the five most influential fractional-order scattering coefficients on the model's output along the y-axis and quantifies their SHAP values on the x-axis which indicates the extent of each coefficient's impact on the model's predictions. Furthermore, we utilize different color to delineates the nature of impact: red for positive and blue for negative influences. Specifically, Fig. 13(a) and (b) present the SHAP values of coefficients within the ResNet-1d model, trained with 10 and 5000 samples, respectively. It is clear that fractional scattering coefficients numbered 4 and 36 maintain significant SHAP values regardless of the training samples, underscoring their essential role in the decision-making process of the ResNet-1d model. Conversely, Fig 13(c) demonstrates that the SHAP values for VGG16-1d model with 5000 training samples, revealing distinct key features and impacts, attributable to differences in model architecture and training parameters. These results underscore the pivotal role of fractional scattering coefficients in elucidating the interpretability of models architecture for specific tasks.

D. The Performance Analysis of Different Model Quantization Schemes and Practical Evaluation Based on FPGA Accelerator

Traditionally, model deployment applications predominantly utilize 32-bit float point precision on universal server platforms to deliver external services. However, within the DEI framework, constraints such as limited computational resources, energy usage, and storage capacity preclude the support of full precision model operations. Current heterogeneous edge computing nodes often employ customized ARM or FPGA units to expedite edge computing tasks. Consequently, adapting traditional, comprehensive models for deployment on these distributed edge nodes to offer immediate services poses a significant challenge in the distributed computing landscape.

Quantization-aware training (QAT) and post-training quantization (PTQ) are two distinct approaches to achieve the model quantization. PTQ is an efficient method for quantizing without requiring retraining. It converts the model weights and activation functions obtained after training from floating-point numbers to lower precision. PTQ mainly includes three different types: float16, dynamic range, and integer quantization. PTQ allows the quantization process to be conducted independently after training, without affecting the training process. This makes PTQ particularly suitable for scenarios requiring rapid deployment or model updates, as it quickly reduces the model's storage and computational requirements without retraining. Additionally, PTQ does not require labeled training data, thereby reducing the complexity and cost of data preparation. QAT embeds quantization constraints within the model's training phase, enabling tailored quantization processes for different layers to more effectively address the accuracy challenges inherent in model quantization. For instance, pseudo quantization nodes may be introduced to analyze the distribution characteristics of model data and provide feedback on quantization-induced accuracy loss. Such nodes simulate the accuracy loss associated with lower bit quantization, incorporating this loss into the network model and feeding it into the loss function. This allows the optimizer to specifically target and minimize this loss during the training process. In this process, all calculations, including forward and backward propagation and pseudo-quantization node calculations, are performed using floating-point arithmetic. The model is quantized into the true integer format only after training is

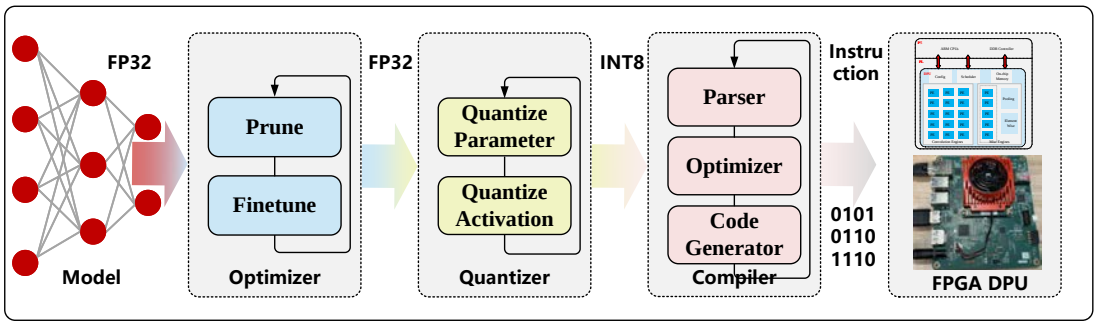


Fig. 14. The architecture of quantified model deployment including optimizer, quantizer and compiler.

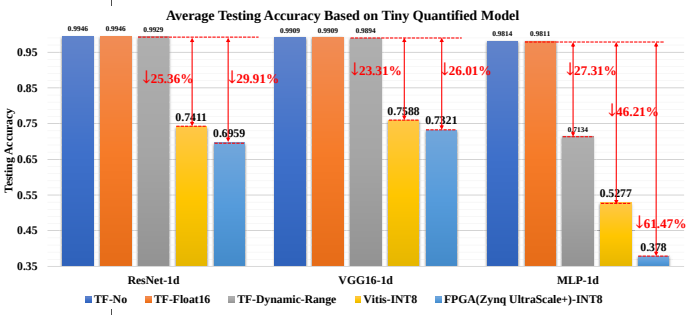


Fig. 15. The average testing accuracy of different network models and quantization schemes with 5000 training samples.

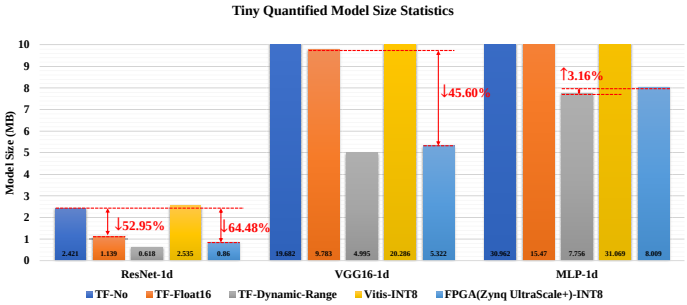


Fig. 16. The model size comparison of different quantization schemes.

completed. Although QAT can better adapt to the losses caused by quantization, it has a higher computational cost, requiring more training time and resources, and relies on complete labeled training data. This dependence on labeled data can be a limiting factor in situations where data acquisition is challenging. As depicted in Fig. 14, PTQ involves pruning and fine-tuning the model to optimize its structure after the model learning convergence has been achieved. This process includes quantization parameters configuration for different layers to produce the final int8 quantization model. To deploy this quantized model on data processing unit (DPU) accelerator, parser is utilized to optimize and generate object code in the .xmodel format which is suitable for deployment on the DPU cores. Moreover, to improve the accuracy and robustness of quantized model, a calibration dataset is employed to capture activation statistical data. Given the computational complexity and distributed nature of federated learning, our paper presents a construction and comprehensive evaluation of PTQ model.

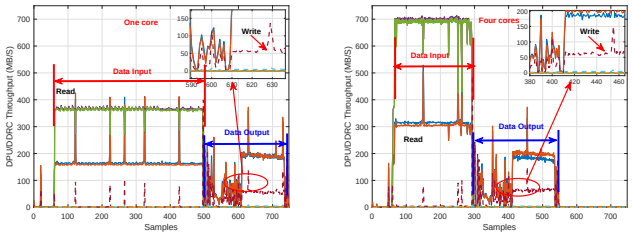
Given the variability in accuracy losses associated with different quantization techniques, we performed a comprehensive comparative analysis of how various quantization methods impact the performance of different models. As illustrated in Fig. 15, we evaluated the performance disparities among different models with 5000 samples across five different quantization schemes: TF No, TF-float16, TF-dynamic-range, Vitis-int8, and FPGA-int8. It is clear that the accuracy degradation from TF-float16 is minimal across three different models, with virtually no performance deterioration observed. As a result, this approach effectively reduces the model size while preserving near-original model performance. To facilitate dynamic adjustment of model quantization range during optimization, we implemented the TF-dynamic-range in our experiments. As shown in Fig. 15, the experimental results indicate that dynamic optimization preserves high identification accuracy for the ResNet-1d and VGG16-1d models.

However, as for the MLP-1d model, Vitis-int8 and TF-dynamic-range quantization strategy may lead to a significant decline of approximately 71.34% and 27.31% compared to TF-No. The primary challenge with dynamic quantization, particularly evident in the MLP-1d model, is its dependence on an appropriate dynamic range. The absence of a suitable batch normalization layer in the MLP-1d model impedes effective global optimization during the dynamic quantization process. To deploy these models on a DPU, we employed Vitis tools to achieve int8 quantization and optimization for model deployment towards FPGA. It can be found that identification performance significantly declined across all models with int8 quantization. This decline primarily stems from the limited parameter bit width during the optimization process and the inability to customize dynamic quantization for individual layers. Specifically, when models are deployed on FPGA, the hardware optimization process exacerbates accuracy performance degradation, particularly in MLP-1d models. Thus, for the int8 quantization scheme, it is crucial to perform global optimization based on parameter distribution and to execute customized layer-specific optimizations.

As shown in Fig 16, the corresponding model sizes under different quantization schemes are statistically analyzed. Overall, the ResNet-1d corresponding model size is only about 2.42MB, which is much smaller than other models. The main reason is that ResNet-1d model contains fewer parameters.

TABLE II
THE NUMBER OF DIFFERENT MODEL PARAMETERS.

Model	Total Params	Trainable Params	Non-trainable Params
ResNet-1d	579183	575913	3270
VGG16-1d	5157359	5148841	8518
MLP-1d	7916911	7,916,911	0



(a) One DPU core mode. (b) Four DPU cores mode.

Fig. 17. The DPU/DDRC throughput via different DPU scheduling schemes.

Compared with the original model, the size of TF-float16 quantization model decreased by approximately 52.95%, while the INT8 for FPGA deployment can compress the model to 0.86 MB, bringing about 64.48% reduction. As for VGG16-1d, due to the inclusion of massive parameters, bring a larger model size. However, after the quantization with FPGA-INT8, the model size is reduced to 5.322MB which is about 45.60% compared to the original. The MLP-1d model contains massive parameters in the fully connected layer, but due to the low efficiency of extracting RFF features, the final identification performance is not ideal. Those TensorFlow (TF) based quantization methods leverage optimized tools provided by TF , bring a smaller quantized model size compared to the Vitis-INT8 quantization scheme, despite having a same bit width. The main reason is that due to the efficient storage mechanisms within TF, which outperform the Vitis-ai framework. The Vitis-ai quantizes convolutional, activation layers and retains redundant information, leading to larger model sizes. In contrast, the FPGA-INT8 quantization scheme builds upon Vitis-INT8 by incorporating additional model analysis, optimization, and code stream compilation, enabling compatibility with the FPGA DPU cores. This further optimization significantly reduces the quantized model size. Therefore, the mdoel size of the quantized model is influenced by quantization and storage optimization strategies. Comparing with Fig. 15, it can be found that the performance constraints of the model include not only the parameters and quantization method of the model, but also the structural characteristics of model.

As demonstrated in Table II, it presents the parameter statistics of three distinct models, encompassing the total number, trainable, and non-trainable parameters. Furthermore, ResNet-1d, VGG16-1d and and MLP-1d comprises 579183, 515359 and 791691 parameters, respectively. Besides, Table II also reveals that both ResNet-1d and VGG16-1d feature non-trainable parameters, typically associated with fixed operations within the model. Conversely, all parameters in MLP-1d are trainable. Despite the simplistic structure of MLP-1d, the massive parameters leading to heightened computational complexity and training costs.

To further investigate the scheduling performance bottleneck of the DPU across different operational modes, throughput is assessed for various DDR controller (DDRC) channels linked to the DPU's underlying acceleration core. As depicted in Fig. 17, allocating all testing samples to different DPU cores for inference via distinct DDRC channels after they are received through external channel. Fig. 17(a) demonstrates that within single-core operation mode, the model retrieves testing samples externally and transfers it to the respective core. Owing to the limitations of single-core processing, data retrieval rates are slower. Once the inference calculations are complete, the results are transmitted via the DDRC channel. Contrastingly, the four-cores mode utilizes different DDRC configuration at the reading stage, achieving considerably faster speeds compared to single-core mode. Additionally, the FPGA system's computing resources and bandwidth scheduling mechanism are likely highly optimized. Regardless of the number of DPU cores utilized, these mechanisms effectively coordinate the computation and data transmission of each core, thereby minimizing resource waste and imbalance. Once all inference tasks are completed, the system begins scheduling and outputting the final corresponding results. Although the calculation process benefits from the acceleration provided by multiple DPU cores, the output of results is primarily governed by the DDRC and its scheduling, rather than the number of DPU cores. As a result, the data output latency of the data remains largely unaffected by the number of active DPU cores. In summary, the model significantly accelerates both data reading and inference processes in the multi-cores parallel mode. However, the computing latency reflects an approximate doubling of overall efficiency of four-cores mode.

Model inference latency is a critical determinant for the practical application, influenced by factors such as the numbers of parameters, model architecture, and the implementation of parallel computing. Table III presents the computed inference latency for various models, tested on samples under different quantization schemes and record the corresponding latency. Additionally, the FPGA-int8 scheme is deployed to the KV260 platform for evaluating the model inference latency. It can be found from Table III that quantization typically can achieve effective reduction of the inference latency compared to original model with the notable exception of the dynamic quantization method. Dynamic quantization exhibits a considerable increase in inference latency, attributable to its complex optimization calculations. The Vitis-int8 scheme significantly reduces inference latency across all models due to the deep compression and optimized calculations process. However, the model quantization also brings significant reduction towards identification accuracy. As for FPGA-int8 quantization, all models demonstrate improvements in latency performance to varying extents, correlated with the parameters size and structure of the model. Additionally, four-cores can markedly

TABLE III
THE AVERAGE PER SLICE TESTING LATENCY OF QUANTIFIED MODEL.

Model	Quantization	Latency (s)
ResNet-1d	TF-No	0.00125
	TF-Float16	0.00123
	TF-Dynamic-Range	0.08751
	Vitis-INT8	0.00098
	FPGA-INT8 (4 core)	**0.00064**
	FPGA-INT8 (1 core)	**0.00121**
VGG16-1d [37]	TF-No	0.00404
	TF-Float16	0.00388
	TF-Dynamic-Range	0.44814
	Vitis-INT8	0.00075
	FPGA-INT8 (4 core)	**0.00110**
	FPGA-INT8 (1 core)	**0.00166**
MLP-1d [38]	TF-No	0.00148
	TF-Float16	0.00153
	TF-Dynamic-Range	0.00054
	Vitis-INT8	0.00030
	FPGA-INT8 (4 core)	**0.00120**
	FPGA-INT8 (1 core)	**0.00180**

reduce the inference latency, achieving approximately 50% improvement with ResNet-1d. In summary, DPU based edge computing nodes substantially reduce inference latency without severely compromising model performance. This advancement is crucial for deploying models in critical tasks within future intelligent IoV networks.

VI. CONCLUSION

In this paper, we propose a novel FFSFNet framework which combines federated learning with fractional wavelet scattering networks to efficiently extract RFF features, reducing data redundancy and significantly enhancing model interpretability. Simultaneously, the incorporation of federated learning ensures privacy, reduces communication overhead and computational resources requirements through DEI across numerous resource-limited IoV nodes. To address memory and computing resource limitations, we conducted a comprehensive analysis of quantization schemes and validated with FPGA DPU accelerator. Extensive experiments on practical datasets, involving 35 different heterogeneous nodes, assessed the performance of FFSFNet against various models and quantization schemes. Furthermore, ResNet-1d consistently achieved the remarkable testing accuracy under centralized training, outperforming VGG16-1d and MLP-1d, particularly with fewer samples. Under the federated learning framework, ResNet-1d FedProx achieves approximately 99% accuracy benefiting from the adaptive proximal term. The ResNet-1d model with float16 quantization can maintain 99.46% without any decreasing while reducing model size by 52.95%. However, although the model size can be reduced by 64.48% with FPGA int8 quantization scheme, it also causes about 29.91% accuracy degradation. Furthermore, model inference latency with FPGA acceleration is reduced to 1.21ms with single-core mode and 0.64ms with multi-core parallel mode, outperforming GPU servers mode. Notably, the superior performance of the FFSFNet and its quantization scheme offers a promising distributed edge intelligence-based RFF framework for intelligent IoV.

REFERENCES

[1] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 1–14, 2023.

[2] Y. Ni, L. Cai, J. He, A. Vinel, Y. Li, H. Mosavat-Jahromi, and J. Pan, "Toward reliable and scalable internet of vehicles: Performance analysis and resource management," *Proc. IEEE*, vol. 108, no. 2, pp. 324–340, 2020.

[3] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, 2020.

[4] K. Yu, L. Tan, S. Mumtaz, S. Al-Rubaye, A. Al-Dulaimi, A. K. Bashir, and F. A. Khan, "Securing critical infrastructures: Deep-learning-based threat detection in iiot," *IEEE Commun. Mag.*, vol. 59, no. 10, pp. 76–82, 2021.

[5] H. Li, K. Ota, and M. Dong, "Learning iov in 6g: Intelligent edge computing for internet of vehicles in 6g wireless communications," *IEEE Wireless Commun.*, vol. 30, no. 6, pp. 96–101, 2023.

[6] Z. Guo, Y. Shen, A. K. Bashir, M. Imran, N. Kumar, D. Zhang, and K. Yu, "Robust spammer detection using collaborative neural network in internet-of-things applications," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9549–9558, 2021.

[7] N. H. Chu, D. N. Nguyen, D. T. Hoang, Q.-V. Pham, K. T. Phan, W.-J. Hwang, and E. Dutkiewicz, "Ai-enabled mm-waveform configuration for autonomous vehicles with integrated communication and sensing," *IEEE Internet Things J.*, vol. 10, no. 19, pp. 727–743, 2023.

[8] M. I.-C. Wang, C. H.-P. Wen, and H. J. Chao, "Hierarchical cooperation and load balancing for scalable autonomous vehicle routing in multi-access edge computing environment," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 6959–6971, 2023.

[9] B. Häfner, V. Bajpai, J. Ott, and G. A. Schmitt, "A survey on cooperative architectures and maneuvers for connected and automated vehicles," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 1, pp. 380–403, 2022.

[10] S. Aoki, C.-W. Lin, and R. Rajkumar, "Human-robot cooperation for autonomous vehicles and human drivers: Challenges and solutions," *IEEE Commun. Mag.*, vol. 59, no. 8, pp. 35–41, 2021.

[11] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6g internet of things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, 2022.

[12] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6g:Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, 2022.

[13] L. Zong, D. Qiao, H. Wang, and Y. Bai, "Sustainable cross-regional transmission control for the industrial augmented intelligence of things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 214–223, 2023.

[14] S. Shen, C. Yu, K. Zhang, and S. Ci, "Adaptive artificial intelligence for resource-constrained connected vehicles in cybertwin-driven 6g network," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 269–278, 2021.

[15] F. Liu, Z. Chen, and B. Xia, "Data dissemination with network coding in two-way vehicle-to-vehicle networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2445–2456, 2016.

[16] S. Lu, "Modeling isotropic traffic flow under vehicle-to-vehicle communication: A kinetic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 237–248, 2022.

[17] S. Aoki and R. Rajkumar, "Cyber traffic light: Safe cooperation for autonomous vehicles at dynamic intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 519–534, 2022.

[18] 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. 893–909, 2021.

[19] L. Y. Paul, G. Verma, and B. M. Sadler, "Wireless physical layer authentication via fingerprint embedding," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 48–53, 2015.

[20] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid rf fingerprint extraction and device classification scheme," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 349–360, 2018.

[21] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: Vulnerabilities, attacks, and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, 2019.

[22] M. Köse, S. Taşcioğlu, and Z. Telatar, "Rf fingerprinting of iot devices based on transient energy spectrum," *IEEE Access*, vol. 7, pp. 715–726, 2019.

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

[24] S. A. Chaudhry, J. Nebhen, A. Irshad, A. K. Bashir, R. Kharel, K. Yu, and Y. B. Zikria, "A physical capture resistant authentication scheme for the internet of drones," *IEEE Commun. Standards Mag.*, vol. 5, no. 4, pp. 62–67, 2021.

[25] J. Liu, Y. Sun, F. Xu, K. Yu, A. K. Bashir, and Z. Liu, "Iis: Intelligent identification scheme of massive iot devices," in *Proc. IEEE Annu. Comput., Softw., Appl. Conf., COMPSAC*, Virtual, Online, Spain, July. 2021, pp. 1623–1626.

[26] P. Huang, D. Li, and Z. Yan, "Wireless federated learning with asynchronous and quantized updates," *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2393–2397, 2023.

[27] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu, "Practical and private (deep) learning without sampling or shuffling," in *Proc. Mach. Learn. Res.*, Virtual, Online, July. 2021, pp. 5213–5225.

[28] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[29] C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang, and L. Gao, "An efficient and reliable asynchronous federated learning scheme for smart public transportation," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6584–6598, 2023.

[30] Q. Zhang, H. Sun, X. Gao, X. Wang, and Z. Feng, "Time-division isac enabled connected automated vehicles cooperation algorithm design and performance evaluation," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2206–2218, 2022.

[31] B. Rokh, A. Azarpeyvand, and A. Khanteymoori, "A comprehensive survey on model quantization for deep neural networks in image classification," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 6, nov 2023. [Online]. Available: https://doi.org/10.1145/3623402

[32] Z. Deng, J. Park, P. T. P. Tang, H. Liu, J. Yang, H. Yuen, J. Huang, D. Khudia, X. Wei, E. Wen, D. Choudhary, R. Krishnamoorthi, C.-J. Wu, S. Nadathur, C. Kim, M. Naumov, S. Naghshineh, and M. Smelyanskiy, "Low-precision hardware architectures meet recommendation model inference at scale," *IEEE Micro*, vol. 41, no. 5, pp. 93–100, 2021.

[33] J. Lee, Y. Kwon, S. Park, M. Yu, J. Park, and H. Song, "Q-hyvit: Post-training quantization of hybrid vision transformers with bridge block reconstruction for iot systems," *IEEE Internet of Things Journal*, pp. 1–1, 2024.

[34] X. Zhang, G. Xiao, M. Duan, Y. Chen, and K. Li, "Appq-cnn: An adaptive cnns inference accelerator for synergistically exploiting pruning and quantization based on fpga," *IEEE Transactions on Sustainable Computing*, pp. 1–14, 2024.

[35] D. Shi, L. Li, R. Chen, P. Prakash, M. Pan, and Y. Fang, "Toward energy-efficient federated learning over 5g+ mobile devices," *IEEE Wireless Communications*, vol. 29, no. 5, pp. 44–51, 2022.

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

[37] 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.

[38] 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. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 165–178, 2019.

[39] J. He, S. Huang, S. Chang, F. Wang, B.-Z. Shen, and Z. Feng, "Radio frequency fingerprint identification with hybrid time-varying distortions," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2023.

[40] G. Reus-Muns and K. Chowdhury, "Classifying uavs with proprietary waveforms via preamble feature extraction and federated learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 7, pp. 6279–6290, 2021.

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

[42] J. Zhang, G. Shen, W. Saad, and K. Chowdhury, "Radio frequency fingerprint identification for device authentication in the internet of things," *IEEE Commun. Mag.*, pp. 1–7, 2023.

[43] G. Shen, J. Zhang, and A. Marshall, "Deep learning-powered radio frequency fingerprint identification: Methodology and case study," *IEEE Commun. Mag.*, pp. 1–7, 2023.

[44] S. Mallat, "Group invariant scattering," *Comm. Pure Appl. Math.*, vol. 65, no. 10, pp. 1331–1398, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21413

[45] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, 2013.

[46] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4114–4128, 2014.

[47] J. Andén, V. Lostanlen, and S. Mallat, "Joint time–frequency scattering," *IEEE Trans. Signal Process.*, vol. 67, no. 14, pp. 3704–3718, 2019.

[48] L. Liu, J. Wu, D. Li, L. Senhadji, and H. Shu, "Fractional wavelet scattering network and applications," *IEEE Rev. Biomed. Eng.*, vol. 66, no. 2, pp. 553–563, 2018.

[49] J. Shi, Y. Zhao, W. Xiang, V. Monga, X. Liu, and R. Tao, "Deep scattering network with fractional wavelet transform," *IEEE Trans. Signal Process.*, vol. 69, pp. 4740–4757, 2021.

[50] T. Zhang, P. Ren, D. Xu, and Z. Ren, "Dfsnet: Deep fractional scattering network for lora fingerprinting," in *Pro. IEEE Glob. Commun. Conf.*, Virtual, Online, Brazil, Dec. 2022, pp. 4897–4902.

[51] T. Zhang, D. Xu, and P. Ren, "Dffnet: Deep federated radio fingerprinting based on fractional wavelet scattering network," in *2023 International Wireless Communications and Mobile Computing (IWCMC)*, 2023, pp. 1346–1351.

[52] X. Yan, Y. Miao, X. Li, K.-K. R. Choo, X. Meng, and R. H. Deng, "Privacy-preserving asynchronous federated learning framework in distributed iot," *IEEE Internet Things J.*, vol. 10, no. 15, pp. 281–291, 2023.

[53] P. He, C. Lan, A. K. Bashir, D. Wu, R. Wang, R. Kharel, and K. Yu, "Low-latency federated learning via dynamic model partitioning for healthcare iot," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 10, pp. 4684–4695, 2023.

[54] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous networks," 2018, *arXiv:abs/1812.06127*. [Online]. Available: http://arxiv.org/abs/1812.06127

[55] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive Federated Optimization," 2020, *arXiv:2003.00295*.

[56] M. Chahoud, H. Sami, A. Mourad, S. Otoum, H. Otrok, J. Bentahar, and M. Guizani, "On-demand-fl: A dynamic and efficient multicriteria federated learning client deployment scheme," *IEEE Internet Things J.*, vol. 10, no. 18, pp. 822–834, 2023.

[57] H. Zhang, K. Zeng, and S. Lin, "Fedur: Federated learning optimization through adaptive centralized learning optimizers," *IEEE Trans. Signal Process.*, vol. 71, pp. 2622–2637, 2023.

[58] 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. Int. Symp. Mobile Ad Hoc Networking Comput.*, New York, NY, USA, July. 2021, pp. 41-–50.

[59] 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 *Pro. IEEE Glob. Commun. Conf.*, Virtual, Online, Brazil, Dec. 2022, pp. 154–159.

[60] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," 2018, *arXiv:1712.05877*.

[61] I. Hubara, Y. Nahshan, Y. Hanani, R. Banner, and D. Soudry, "Improving post training neural quantization: Layer-wise calibration and integer programming,"2020, *arXiv:2006.10518*.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[63] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, 2021.