Received: 4 March 2024 | Revised: 16 January 2025 | Accepted: 18 January 2025

DOI: 10.33012/navi.688

ORIGINAL ARTICLE

Federated Learning of Jamming Classifiers: From Global to Personalized Models

Peng Wu | **Helena Calatrava** | **Tales Imbiriba** | **Pau Closas**

[1] Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA

Correspondence
Peng Wu
Department of Electrical and Computer Engineering at Northeastern University, Boston, MA, USA.
Email: wu.p@northeastern.edu

Funding Information
This research was supported by the National Science Foundation under Awards ECCS-1845833 and CCF-2326559.

Abstract

Jamming signals can jeopardize and ultimately prevent the effective operation of global navigation satellite system (GNSS) receivers. Given the ubiquity of these signals, jamming mitigation and localization techniques are of crucial importance, and these techniques can be enhanced with accurate jammer classification methods. Although data-driven models have proven useful for detecting jamming signals, training these models using crowdsourced data requires sharing private data and may therefore compromise user privacy. This article explores the use of federated learning to locally train jamming signal classifiers on each device, with model updates aggregated and averaged at a central server. This approach ensures user privacy during model training by removing the need for centralized data storage or access to clients' local data. The personalized federated learning strategies employed in this study are also tested on non-independent and identically distributed data sets composed of spectrogram images from interfered GNSS signals. In addition, this article discusses the effect of model quantization, which is used to effectively reduce communication costs, as well as a fusion strategy for personalized federated learning schemes in which multiple classifiers are available.

Keywords

distributed inference, federated learning, Jamming detection, machine learning, neural networks.

1 | INTRODUCTION

Global navigation satellite system (GNSS) jamming signals are L-band spectrum interferences that can overpower a GNSS receiver and prevent its effective operation (Amin et al., 2016; Morton et al., 2021). A wide variety of jammers can be found in the online market at cheap prices, which makes intentional, human-made jamming signals a threat such that to national security and safe navigation (Borio et al., 2016; Morales-Ferre et al., 2020). In addition, signals do not need to be malicious to have a jamming effect: even legitimate waveforms, including the continuous wave interferences produced by damaged electronics and the signals emitted by Distance Measurement Equipment technology conceived for aircraft navigation, can interfere with GNSS receivers (Li et al., 2019). Jamming sources are located on Earth or, in the case of drone jammers, near Earth's surface. Because of path-loss

 Licensed under CC-BY 4.0 © 2025 Institute of Navigation

attenuation in GNSS signals due to the large distance between Earth and GNSS satellites, jamming interferences are often received with remarkably higher power than the useful GNSS signal, which can lead to localized performance disruption over a radius of several kilometers (Mitch et al., 2011). Jamming has been suggested to be the main cause of GNSS-based service outages (Morales Ferre et al., 2019), making protection against this kind of attack a desirable feature in GNSS receivers (Dovis, 2015; Thombre et al., 2018).

One such form of protection involves jammer classification, which can enhance classical interference cancellation techniques. In general, interference cancellation techniques are formulated as an estimation problem where the jamming signal is detected and estimated, often with a parametric model (Borio & Closas, 2017). Because the aim of these techniques is to first reconstruct the interference, knowledge of the type or class of interference can accelerate the algorithm. For example, if the algorithm knows that a continuous wave interference is threatening a receiver, it would only need to estimate the interference's central frequency in order to reconstruct its waveform and implement a cancellation measure. Furthermore, jamming classification techniques inherently involve detecting the interference. Most previous GNSS studies regarding protection against jamming interferences have focused on detecting (Arjoune et al., 2020), mitigating (Borio et al., 2018), and localizing the interference (Strizic et al., 2018), with little effort dedicated to the classification of jamming signals (Morales Ferre et al., 2019) until more recent publications (Chen et al., 2024; Mehr & Dovis, 2022). The notable exception is some previous work in the context of radar systems, such as the machine learning jamming prediction algorithm proposed by Lee et al. (2020).

Recent studies have focused more explicitly on potential approaches to classifying jamming signals. For example, Morales Ferre et al. (2019) proposed Support Vector Machine- and Convolutional Neural Network-based classifiers for the purpose of jammer classification, which they treated as an image classification problem. Their classifiers achieved nearly 99% accuracy at detecting a jamming incident and over 90% classification accuracy when differentiating among a variety of jamming types (namely those considered in this article) after being trained on a set of 600 images per class. According to Voigt (2021), the use of multivariate time-series approaches can also increase the accuracy of jammer classification techniques. More recent studies (Closas et al., 2024; Mehr & Dovis, 2024) in GNSS interference classification have prominently utilized machine learning techniques. For example, Mehr & Dovis (2022) demonstrated the effectiveness of Convolutional Neural Networks for jammer classification, while Chen et al. (2024) explored a compound neural network model. Residual Neural Networks have also been employed (Brieger et al., 2022; Zengyuan et al., 2023). Mehr et al. (2023) and van der Merwe et al. (2024) contributed classifiers based on other machine learning techniques, highlighting the growing popularity and success of these methods in GNSS interference classification.

Most studies of GNSS integrity rely on synthetic data because data collection in the presence of jamming signals is difficult. This difficulty is further compounded in studies that wish to use different interference types and received power values. However, some interference effects, such as the ones introduced by multipath reflections, can be difficult to recreate in synthetic data sets. Thus, despite the challenges of obtaining real GNSS interference data, the use of real data has the potential to significantly improve the training and assessment of data-driven classifiers. One option for collecting real GNSS data involves traditional crowdsourcing approaches, where clients record data and share it with a central unit that trains the classifier. Nevertheless, crowdsourcing raises concerns about user privacy because it requires that users send their data directly to a centralized server.

Aimed at addressing these concerns, Federated Learning (FL) has recently attracted great interest because it protects user privacy and efficiently uses resources by harnessing the processing power of edge devices (Niknam et al., 2020). FL is a promising solution that enables many clients to jointly train machine learning models while maintaining local data decentralization. Such collaboration between users in distributed scenarios has proven useful in GNSS interference management tasks (Jiang et al., 2024; Nicola et al., 2020). Instead of exchanging data and conducting centralized training, each party in a FL system sends its model to the server, which then updates a joint model and sends the global model back to the parties. Because the original user data is not exposed, FL effectively addresses privacy issues (McMahan et al., 2017).

FL has proven highly beneficial for jamming signal classification in GNSS applications, particularly for protecting the privacy of collaborating users and reducing the amount of data being exchanged. In the traditional crowdsourcing-based alternative, in which users send snapshot data to a server in charge of centrally training the classifier, intercepted data snapshots could be used to compute the position of the user. These snapshots therefore reveal confidential user information, which is generally undesirable for most users but especially problematic in contexts like military or other sensitive applications. In contrast, FL allows for model training without the need to share raw data, thereby preserving privacy. Additionally, the bandwidth required for such transmission in a crowdsourcing framework can be very large (as we will see from the typical data sets that are employed in training such models), making this approach impractical from a communication perspective. FL reduces this communication overhead by transmitting only model updates rather than large amounts of raw data to a central server. Finally, GNSS receivers are typically spread across various geographical locations, each experiencing different jamming events. Decentralized, FL-based methods leverage this geographic distribution to create more robust and generalized models. In other words, by learning from diverse environments, the models enhance their performance and robustness.

Despite the advantages of FL, one common problem is the challenge of non-independent and identically distributed (IID) data, which diminishes learning effectiveness. Non-IID data refers to the scenario where the data distribution across edge devices differs significantly, posing unique challenges for FL algorithms. Various approaches have been developed to address this issue. One such approach, called personalized FL, has garnered significant attention because it tailors models to each client's local data distribution. There are several ways to achieve this customization, including local fine-tuning (Ben-David et al., 2010; Wang et al., 2019), meta-learning (Fallah et al., 2020; Jiang et al., 2019), transfer learning (Li & Wang, 2019), model mixture methods (Deng et al., 2020), and pair-wise collaboration methods (Huang et al., 2021).

Quantization is another crucial aspect of FL (Krishnamoorthi, 2018). Specifically, quantization enables efficient communication and reduces computational costs by representing model parameters with lower precision. By quantizing the weights or gradients, we significantly reduce the amount of data that needs to be transmitted during the aggregation process. This downsizing is especially beneficial in FL scenarios where communication resources are limited and bandwidth is a constraint (Lang & Shlezinger, 2022; Reisizadeh et al., 2020). Many quantization methods have been proposed, such as Uniform Quantization (Widrow et al., 1996), Non-uniform Quantization (Baskin et al., 2021), Stochastic Quantization (Damgaard & Hüffel, 1987), Vector Quantization (Gray, 1984) and Quantization-Aware Training (Jacob et al., 2018). In general, integrating

quantization into FL algorithms can enhance scalability and privacy while maintaining reasonable model accuracy.

Here, we continue our preliminary (Wu et al., 2023) and more recent work (Deng et al., 2024) towards training jamming signal classifiers using privacy-preserving strategies that can cope with crowdsourcing-based data collection. In this extended study, we specifically investigate (i) the challenge of non-IID datasets, with the goal of developing solutions based on personalized FL strategies, and (ii) the impact of model quantization in the communication process. Our overall objective is to obtain a Neural Network-based global model capable of classifying different jamming signals, as shown in Figure 1. To preserve client privacy while leveraging crowdsourcing data collection strategies, we exploit FL approaches in which model parameters are shared with clients, thereby allowing local classification of jamming signals while avoiding data sharing (Figure 2).

In our proposed framework, we assume the possible existence of C different jamming types and perform our FL approach over a network with M collaborative users. We then study the FL-based jamming classifier under two different data distribution scenarios. In the first scenario, clients' data is IID; that is, all clients observe a similar amount of interference instances from all C classes. In the second, non-IID scenario, clients observe data that is unbalanced towards different classes. Although working with non-IID data poses several challenges, non-IID data is common in real-world scenarios given that not all clients have access to all available types of data. In the context of this work, non-IID data arises when not all participating users observe the same classes of jamming signals. We therefore investigate different techniques for addressing the challenges of non-IID data in the context of jammer classification. Rather than employing a single global model

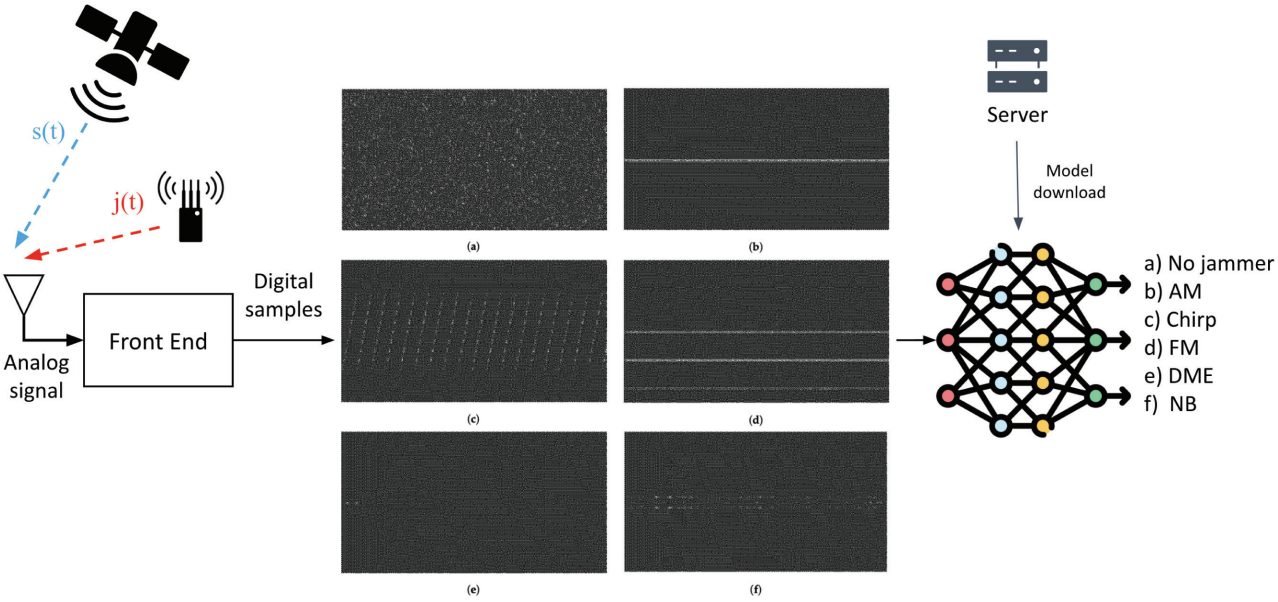


FIGURE 1 System diagram of the jamming signal classifier considered herein. First, a receiver downloads a pre-trained model from the server, which can be either *i*) trained on locally available data and sent back to the server for fusion with other models; or *ii*) used to perform jamming classification on local data. Monochrome spectrogram images of the six jammer types available in the test data set from Morales Ferre et al. (2019) are shown: (b) Amplitude Modulated (AM), (c) chirp, (d) Frequency Modulated (FM), (e) Pulsed or Distance Measurement Equipment (DME), and (f) narrow band (NB) jammers. Class (a) shows a clean signal (no interference).

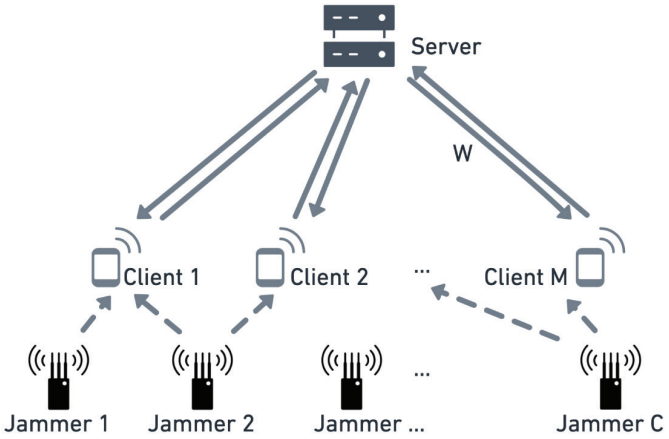


FIGURE 2 Federated learning framework for training jamming signal classifiers. First, M collaborative clients receive the parameters of the classifier from a server. These clients retrain the model based on their local data and then upload their updated classifier to the server in charge of fusing the results. This process does not require the exchange of actual user data or positions, thus preserving their privacy.

distributed among clients, we use a framework where each client maintains a personalized model. This approach enables clients to more accurately represent their unique data distributions while simultaneously benefiting from the collective knowledge derived from other clients. Finally, we investigate the effects of quantization techniques on the transmission of parameters between clients and the server and demonstrate the influence of quantization bit depth on the performance of various FL algorithms.

The remainder of this paper is organized as follows. In Section 2, we describe the satellite signal model and targeted jammer types. Our FL technique is then derived in Section 3, and the experimental setup and results are described in Section 4. Finally, Section 5 concludes the paper.

2 | SYSTEM MODEL

For the purpose of this article, we model the analog baseband equivalent of the received GNSS signal as

$$r(t) = s(t) + j(t) + w(t), \tag{1}$$

where $s(t)$ contains the useful GNSS satellite signals and $w(t)$ represents sources of randomness, such as thermal noise, which are typically modeled as an additive white Gaussian noise process. The term $j(t)$ represents the signal waveform generated by a jamming source as measured at the receiver. Several waveforms are possible for $j(t)$ depending on the type of jammer (Morales-Ferre et al., 2020). Accurate knowledge of $j(t)$ allows for prompt reaction to a jamming threat, either for its localization (Nardin et al., 2023) or mitigation. For mitigation, interference cancellation techniques aim to estimate the waveform of $j(t)$ so that it can be reconstructed and directly subtracted from $r(t)$.

Jammers can be classified according to their characteristic features, including the type of device by which they are broadcast, their frequency spectrum, and their number of antennae (Borio et al., 2016). In this paper, we target the same jammer types as in Morales Ferre et al. (2019), given that we use their data set of jammer

signals and benchmark our results against theirs. Overall, the aim of our research is to use the FL technique explained in Section 3 for the classification of the following jammer types:

1. Amplitude Modulated (AM);
2. Chirp;
3. Frequency Modulated (FM);
4. Pulsed or Distance Measurement Equipment (DME);
5. Narrow Band (NB) jammers; and
6. No interference.

As in Morales Ferre et al. (2019), we do not consider wideband jammers given that their presence is difficult to detect when analyzing spectrogram images. All five jammers considered here have narrow spectra that overpower the signal of interest, which becomes buried in noise. Note that our classification strategy, proposed in Section 3, simultaneously performs the task of detecting interference because the absence of interference can be properly identified. The waveform expressions $j(t)$ for each of the five jammer types listed above can be found in Morales Ferre et al. (2019) but are not explicitly used for training or testing the FL solution proposed here.

Our classification strategy mostly relies on the behavior of these five jammer types in the frequency domain. While AM and FM jammers target pre-fixed frequencies, others, such as chirp jammers, sweep over different frequency bands. Feature extraction approaches based on spectral analysis of the signals, such as their spectrograms, are therefore suitable for distinguishing different jammer types. Specifically, the short-time Fourier transform allows for the time-frequency localization of the interference signal. (Morales Ferre et al., 2019) successfully approached jammer classification as an image classification problem, where spectrograms of the received signal $r(t)$ were treated as images. In their approach, the spectrograms are computed on the discrete-time version of $r(t)$ in Equation (1), which, at an appropriate sampling rate $f_s = 1/T_s$, would be modeled as $r[n] = s[n] + j[n] + w[n]$, where $t = nT_s$ for $n \in \mathbb{Z}$.

3 | FEDERATED LEARNING METHODOLOGIES

Many different FL algorithms have been discussed for different applications (Li, Sahu, Talwalkar et al., 2020; Park et al., 2022; Wu et al., 2021) but especially in the field of image classification. One *de facto* approach for FL is Federated Averaging (FedAvg) (McMahan et al., 2017), which fuses the model parameters by a weighted sum. According to previous studies (Hsu et al., 2019; X. Li et al., 2020), the learning effectiveness of standard FL methods is compromised when using non-IID data. In this section, we explore two FL strategies. The first leverages FL to develop a unique global model capable of making accurate predictions on data from various clients, whereas the second focuses on learning personalized models for individual clients with the goal of achieving higher accuracy on their local data sets.

Global FL. In the first strategy, we consider the setup depicted in Fig. 2, where M collaborative clients train a global classification model (e.g., a neural network) such that:

$$\mathbf{y} = \mathbf{h}(\mathbf{X}; \boldsymbol{\omega}) \tag{2}$$

where $\mathbf{y} \in \mathbb{R}^C$ is the vector of class posteriors with elements $p(y = \ell \mid \mathbf{X})$, with $\ell \in \{\text{AM, Chirp, FM, DME, NB, NO}\}$. Here, $\mathbf{h} : \mathbf{X} \mapsto \mathbf{h}(\mathbf{X})$ is the neural network classifier parameterized by $\boldsymbol{\omega} \in \mathbb{R}^{N_\omega}$, and $\mathbf{X} \in \mathbb{R}^{T_w \times N}$ is the spectrogram of the received GNSS signal $\mathbf{r}[n]$. See Morales Ferre et al. (2019) for more details regarding the construction of the spectrogram data. In this contribution, we assume that the data $\mathcal{D}$ is composed of M disjoint data sets $\mathcal{D}_i = \{\mathbf{y}_n^{(i)}, \mathbf{X}_n^{(i)}\}_{n=1}^{L_i}, i \in \{1, \ldots, M\}$.

The training process for this model can be formulated as the minimization of a loss function:

$$\min_{\boldsymbol{\omega}} \left\{ \mathcal{L}(\boldsymbol{\omega}) = \sum_{i=1}^{M} \mathcal{F}_i(\boldsymbol{\omega}) + \lambda \mathcal{A}(\boldsymbol{\omega}, \boldsymbol{\omega}_g) \right\} \tag{3}$$

where $\boldsymbol{\omega}_g$ is the initial weight, or, if the model is trained recursively, the aggregated weight from the global model of the previous round. $\mathcal{L}(\boldsymbol{\omega})$ is the global loss function, and $\mathcal{F}_i : \mathbb{R}^d \to \mathbb{R}, \boldsymbol{\omega} \mapsto \mathcal{F}_i(\boldsymbol{\omega})$ are local loss functions. The regularizer term $\lambda \mathcal{A}(\boldsymbol{\omega}, \boldsymbol{\omega}_g)$ helps prevent the model from drifting away from the global model and protects against overfitting. Setting $\lambda = 0$ represents the conventional FedAvg approach to FL (McMahan et al., 2017), and if $\lambda \neq 0$, the regularizer can be the L-2 norm, resulting in FedProx (Li, Sahu, Zaheer, et al., 2020). The $\mathcal{A}$ term can be more complex, as demonstrated in model-contrastive FL (MOON) (Q. Li et al., 2021), which incorporates a contrastive loss term for control.

Personalized FL. For personalized FL, each client has its own personalized model which is shared among clients to cross-pollinate the local data without sharing it. The objective here is to learn multiple models, for which we can mathematically define the loss function:

$$\min_{\boldsymbol{\omega}} \left\{ \mathcal{L}(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_M) = \sum_{i=1}^{M} \mathcal{F}_i(\boldsymbol{\omega}_i) + \lambda \mathcal{A}(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_M) \right\} \tag{4}$$

where $\boldsymbol{\omega} = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_i, \ldots, \boldsymbol{\omega}_M]$. When $\boldsymbol{\omega}_i = \boldsymbol{\omega}_g - \alpha \nabla \mathcal{F}_i(\boldsymbol{\omega}_g)$ and $\lambda = 0$, the model reduces to local fine-tune, meta-learning, or transfer learning with different strategies and cases (Tan et al., 2023). In the case where $\lambda \neq 0$ and $\mathcal{A}$ represents the L-2 norm $||\boldsymbol{\omega}_i - \boldsymbol{\omega}_g||$, this function is referred to as Ditto (T. Li et al., 2021). More complex $\mathcal{A}$ terms can be employed to observe the relationships among different models. This approach typically involves pairwise collaboration methods that calculate the similarity between models and determine the fusion weight for each model, as proposed by FedAMP (federated attentive message passing) (Huang et al., 2021).

Because our goal is to employ FL approaches for jamming classification, not to compare different FL methods, we choose several popular FL methods that have already shown promising results across diverse applications. In this section, we 1) discuss the practical problem of quantizing the shared parameters before transmitting them, which affects the performance of the models through the different FL strategies; and 2) introduce a fusion strategy to make a classification decision based on a set of personalized models, which results from a probabilistic interpretation of the local classifiers.

3.1 | Quantization in Federated Learning

In FL schemes, the shared information needs to be quantized before being transmitted in order to reduce the communication bandwidth. Here, we briefly discuss quantization, which we then experimentally investigate in Section 4

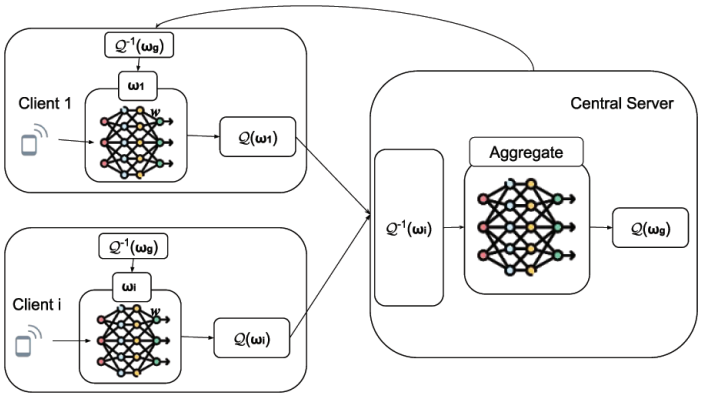


FIGURE 3 In FL schemes, quantization of shared data happens during transmission from clients to the server and from the server back to clients.

The quantization operation and dequantization formulas in an FL scheme are as follows:

$$\mathcal{Q}_N(x) = \left\lfloor \frac{x}{s} \right\rceil + z \tag{5}$$

where $\mathcal{Q}_N(x)$ is the N-bit quantized version of a variable x, $\lfloor \cdot \rceil$ is the round operator, s is the scale, and z is the zero point. When designing the quantizer, the choice of scale and zero point is crucial. These values are typically chosen based on the range of the input signal values in order to minimize information loss. For instance, if the range of values in a signal is $[a, b]$, the scale and zero point can be computed as:

$$s = \frac{b-a}{2^N - 1} \tag{6}$$

$$z = \frac{\lfloor 0 - a \rceil}{s} \tag{7}$$

Once the quantized information is received, it can be re-quantized for processing. For instance, in the FL scheme in Fig. 3, the server re-quantizes model parameters before fusing them to some $N' > N$ number of bits. This re-quantization can be achieved mathematically by:

$$\mathcal{Q}_{N'}^{-1}(q) = s \times (q - z) \tag{8}$$

where $\mathcal{Q}_{N'}^{-1}(q)$ is the N'-bit re-quantized version of $q = \mathcal{Q}_N(x)$. The scale and zero point are computed as before but using N' instead of N.

For our work, the model parameters are quantized only for communication purposes, namely when they are downloaded from the server to clients and uploaded from clients to the server. During the local training and server aggregation, the weights are re-quantized to float values. The details of this re-quantization are shown in Algorithm 1, where $\mathcal{Q}(\boldsymbol{\omega})$ represents the quantized vector parameters.

A generic FL algorithm operates as follows: given M clients and an initial model $\boldsymbol{\omega_0}$, each client receives the current model from the server at each of the T iterations where the process is repeated. Clients then update their model parameters based on their local data and send the quantized model parameters back to the server. The server aggregates these local parameters into either a single global model (in the case of general or global learning) or into M personalized models (i.e., one for

ALGORITHM 1
General FL Algorithm

Input: number of clients M; the architecture of local models $\mathbf{h}$ with initial $\boldsymbol{\omega}_0$ (if personalized models, then $\boldsymbol{\omega}_0 = [\boldsymbol{\omega}_0^1, \ldots, \boldsymbol{\omega}_0^M]$); local loss functions $\mathcal{F}_i$; data $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$; and number of iterations T;
for $t = 0, \ldots, T-1$ **do**
  Server: send $\mathcal{Q}(\boldsymbol{\omega}_t)$ to each client
  for $i \in M$ **do**
    Update the $\boldsymbol{\omega}_{t+1}^i \leftarrow$ Equation (3) or (4)
    Upload local model parameters $\mathcal{Q}(\boldsymbol{\omega}_{t+1}^i)$ to server.
  end for
  Server: aggregate $\boldsymbol{\omega}_{t+1}^i$ as $\boldsymbol{\omega}$ with different FL algorithms
end for
Output: $\boldsymbol{\omega}_T$

each client in the case of personalized learning). The aggregated parameters are then sent back to the respective clients for the next iteration of training. Detailed steps for this generic FL approach are outlined in Algorithm 1.

3.2 | Fusion Strategy used in Federated Learning

Some personalized models, like FedAMP, can only learn from data that has a similar distribution as their local training data. This requirement creates a challenge when there is no prior knowledge of which model to use for new data points. In these cases, a fusion scheme is needed to combine the M available models and generate a reasonable estimate (Wu et al., 2024).

In a scheme similar to FedAMP, the $i \in \{1, \ldots, M\}$ classifier (i.e., each of the personalized models) provides a categorical posterior distribution for c, the jammer class, which can take any of the $j \in \{1, \ldots, L\}$ labels based on the current data $\mathbf{y}$ and the corresponding model $\mathcal{M}_i$. In other words:

$$p(c \mid \mathcal{M}_i, \mathbf{y}) = \prod_{j=1}^{L} p_{j|i}^{[c=j]}, \tag{9}$$

where $p_{j|i}$ denotes the probability of the j-th label given the i-th classifier, and $[c = j]$ is an indicator function that returns 1 if $c = j$ and 0 otherwise. The a priori class probability $p(c)$ is categorical and defined by the probabilities $p_{j|0}$, which, in the equiprobable case, result in $p_{j|0} = 1/L, \forall j$. The optimal fusion rule is provided by the joint posterior distribution, which Pastor et al. (2021) showed is proportional to:

$$\begin{aligned} p\left(c \mid \mathcal{M}_{1:M}, \mathbf{y}\right) &\propto \frac{\prod_{i=1}^{M} p\left(c \mid \mathcal{M}_i, \mathbf{y}\right)}{p(c)} = \frac{\prod_{i=1}^{M}\prod_{j=1}^{L} (p_{j|i})^{[c=j]}}{p(c)} \\ &= \prod_{j=1}^{L} \left(\frac{p_{j|1} \cdot p_{j|2} \cdots p_{j|M}}{p_{j|0}^{M-1}} \right)^{[c=j]}, \end{aligned} \tag{10}$$

Here, the M different models are conditionally independent given the c. The resulting joint distribution is categorical, from which the maximum a posteriori probability can be readily obtained to predict the class c.

4 | EXPERIMENTS

This section presents a series of experiments in which we show the applicability of FL to train, in a distributed manner, a jammer classifier. Our classifier performs comparably to a classifier trained on a centralized node with access to all local data sets. In the following sections, we describe the data set used for our experiments, how this data set is employed in a distributed learning scheme, how the model was configured, and the results obtained from our experiments.

4.1 | Data Pre-processing

We used the data set provided by Morales Ferre et al. (2019), which is available open-access at https://zenodo.org/record/3370934. This data set contains 61800 *.bmp* monochrome spectrogram images with 512 × 512 pixel resolution, binary scale, and 600 DPI. The spectrograms were computed from simulated GNSS signals affected by interference from the aforementioned jammer types (see Section 2). Morales Ferre et al. (2019) used 6000 images for training (1000 for each jammer class), 1800 images for validation, and the remaining 54000 images for testing.

To optimize computational resources and expedite the training process, we pre-processed the data following an approach typical in machine learning contexts. First, we used both the training and validation data sets, as the validation step is often omitted from the experimentation process unless performing hyperparameter tuning. The combined dataset, which includes both training and validation data, was divided into 75% for training and 25% for testing. To further enhance the training process, image resolution was reduced from 512 × 512 to 256 × 256 pixels using of bilinear interpolation techniques. Finally, once all the data were preprocessed, the pixel values were normalized to the range [–1, 1] using mean 0.5 and standard deviation 0.5 to facilitate the training.

4.2 | Federated Data Setting

We investigated two different data settings. First, in the IID setting, all clients received similar data distributions (i.e., a similar number of samples from each jammer class). For these experiments, we uniformly split the data into groups of 20, 30, and 40 clients to examine how client numbers may influence the results. This split resulted in approximately 65, 43, and 32 samples per client for the 20-, 30-, and 40-client scenarios, respectively.

Second, we considered a non-IID setting in which the distribution of training samples from different jammer classes was unbalanced across clients. To generate non-IID splits of the data set, we followed the approach by T. Li, Sahu, Talwalkar, & Smith (2020), in which client data is sampled using a Dirichlet distribution. In brief, we defined the number of clients and classes, and then, for a given client i, we defined the probability of sampling data from each label $j \in \{1, \ldots, C\}$ as the vector $(p_{i,1}, \ldots, p_{i,C}) \sim \mathrm{Dir}(\boldsymbol{\beta})$, where $\mathrm{Dir}(\cdot)$ denotes the Dirichlet distribution and $\boldsymbol{\beta} = (\beta_{i,1}, \ldots, \beta_{i,C})^{\top}$ is the concentration vector parameter. The concentration parameter $\boldsymbol{\beta}$ is used to generate proportions for distributing each class's data points among the clients. These proportions then determine how many data points of that class each client will receive. By adjusting $\boldsymbol{\beta}$, we control the level of skewness in the data distribution, with smaller $\boldsymbol{\beta}$ values leading to more uneven, non-IID distributions.

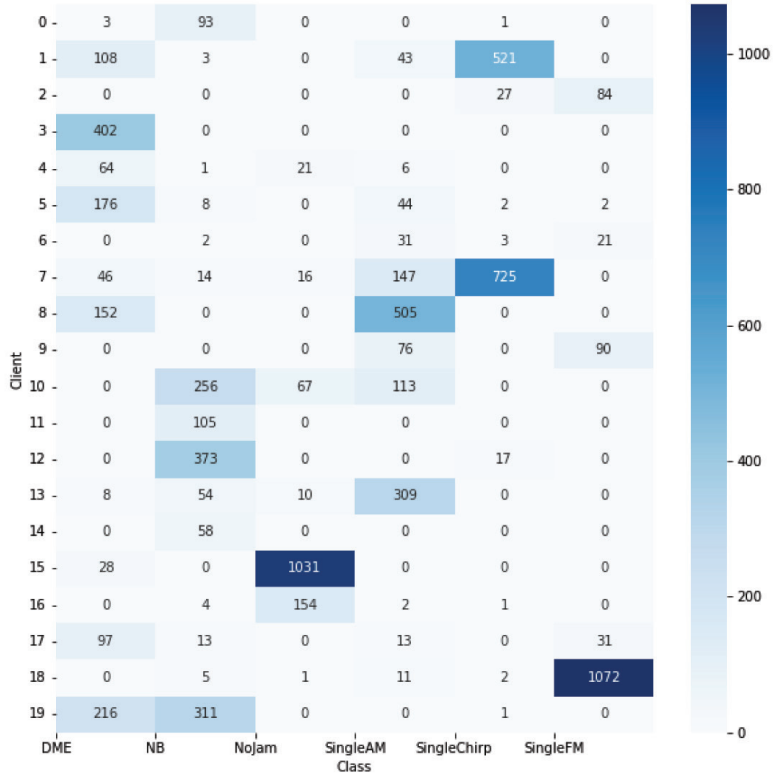


FIGURE 4 Number of data points per class for each of the $M = 20$ clients.

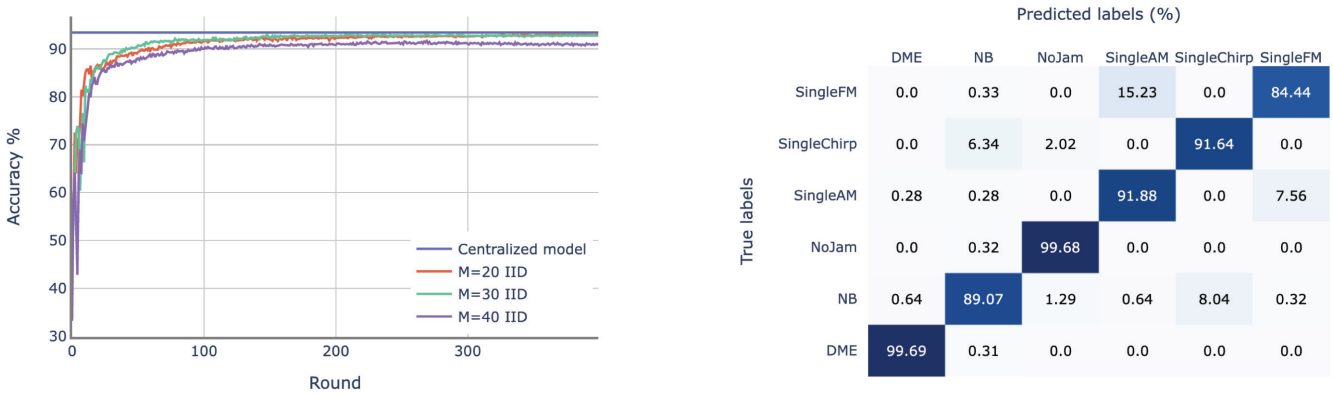
The advantage of this approach is that the imbalance level can be flexibly changed by adjusting $\beta_{i,j}$. For our analysis, we set the concentration parameter $\beta_{i,j}$ to a relatively small value of 0.1, thereby creating a more unbalanced partitioning. This imbalance is evident in the distribution of data points among clients, as many client data sets only contain a subset of the six labels. For example, Figure 4, shows the number of samples per class for each client when $M = 20$ clients, and some clients contain a disproportionately large or small percentage of certain class labels.

4.3 | Model Setting

Morales Ferre et al. (2019) employed a convolutional neural network (CNN) to train a classifier based on their full data set **D**. Their solution serves as the benchmark for our results, which rely on the same CNN architecture to train a classifier using the FL framework described earlier. This CNN consisted of one convolutional layer, one pooling layer, and one fully connected layer with a ReLU activation function. The convolution layer used 16 filters of size $12 \times 12 \times 1$, a learning rate of 0.01, and a stochastic gradient descent optimizer (Ruder, 2016). The last layer was the softmax layer to produce classification results. Cross-entropy was used for the cost function.

4.4 | IID and Non-IID Experiments

Figure 5(a) shows the accuracy of federated averaging algorithms over 400 communication rounds in an IID data setting. The final accuracy of the centrally trained



(a) (b)

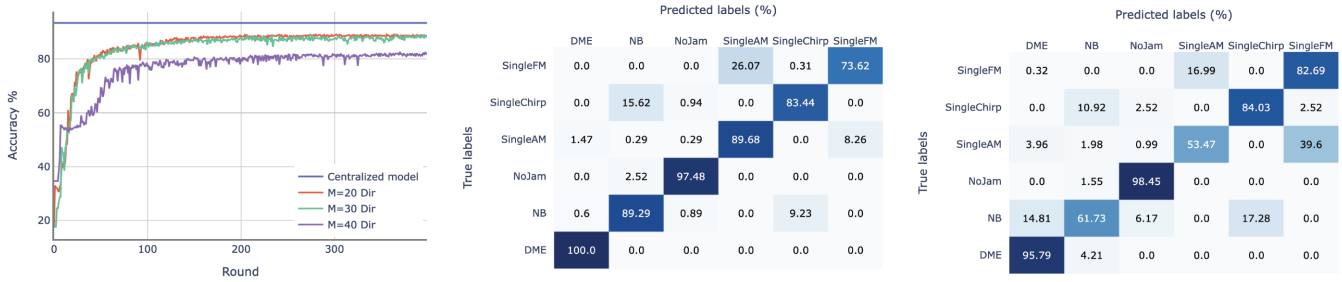
FIGURE 5 Example of FedAvg in 400 rounds under IID data setting. (a) Accuracy. (b) Confusion matrix of FedAvg for $M = 20$ clients.

model (approximately 93.4%) was used as a benchmark for subsequent tests. This figure also compares the accuracy achieved with different numbers of clients M. As expected, better results were achieved when a small number of clients were used. With a fixed amount of data, fewer clients means that each client has access to a larger share of the data, thus enabling better training of local their models. Nevertheless, high accuracy was achieved for all tested numbers of clients.

The corresponding confusion matrix in Figure 5(b) reveals that each jammer class is identified with relatively high accuracy, with the DME jammer type and the clean signal ("NoJam") achieving the highest accuracies (over 99%). The classifier is therefore able to accurately detect the absence of interference, as the interference-free spectrogram in Figure 1(a) differs notably from the others. This unique spectrogram arises because the spectrum of a clean signal contains the signal of interest buried in Gaussian noise, which pollutes the whole spectrogram. On the other hand, because jamming signals are received with dramatically higher power than the satellite signal of interest, the noise $w(t)$ cannot be observed in spectrograms (b)-(f) from Figure 1. In contrast, the SingleFM and NB jammer types were classified with less than 90% accuracy. The results from Figure 1 suggest that the classifier struggled to distinguish between SingleAM and SingleFM interference, which both span only one or two narrow bands of the signal spectrum. Indeed, the SingleFM spectrogram is equivalent to the SingleAM spectrogram with an additional band. The classifier also struggled to distinguish between the NB and SingleChirp interferences, both of which have a lower magnitude in their spectra due to being more spread. This spread makes their respective spectrogram images look blurry relative to spectrograms from SingleAM and SingleFM interference.

Figure 6(a) illustrates the corresponding accuracy of the FedAvg algorithm for different numbers of clients in a non-IID data setting, again compared to the accuracy of the centrally trained model as a benchmark. The results show that the accuracies for different client numbers are lower than the corresponding accuracies in the homogeneous IID data setting, indicating the increased difficulty of learning with heterogeneous data. Consistent with the IID data setting, increasing the number of clients reduced the algorithm's overall accuracy. Moreover, with 40 clients, the algorithm took more communication rounds to converge than when smaller numbers of clients were considered.

Figures 6(b) and 6(c) show the corresponding confusion matrices for 20 and 40 clients under the non-IID, Dirichlet-distributed data setting. As in the IID data setting, the DME jammer type and clean signal were the easiest to classify, with



(a) (b) (c)

FIGURE 6 Example results from FedAvg across 400 rounds with a non-IID data set (a) Accuracy of different number of clients. (b) Confusion matrix for $M = 20$ clients. (c) Confusion matrix for $M = 40$ clients.

accuracies of 100% and 97.48% for $M = 20$ and 95.79% and 98.45% for $M = 40$ clients, respectively. For $M = 40$, classification accuracies were low for the NB and SingleAM jammer types, and for $M = 20$, the worst accuracy was achieved with the SingleFM jammer type. As in Figure 5(b), inspecting the non-diagonal elements reveals that the classifier specifically struggled to distinguish between SingleAM and SingleFM interference and between NB and SingleChirp interference. For $M = 40$, where performance was already worse due to the higher number of clients (implying less local data), the classifier also struggled to distinguish between NB and DME signals. Nevertheless, even with a high number of clients (i.e., $M = 40$), the classifier achieved accuracies above 80% with the DME, clean signal, SingleChirp, and SingleFM jammer types. For a lower number of clients (i.e., $M = 20$), all jammer types could be classified with an accuracy above 80%.

As a final remark, the results presented in this section are comparable to those obtained with the benchmark training process: the centralized classification algorithm proposed by Morales Ferre et al. (2019). In their results, classification accuracy was also highest for the DME (or pulsed) interference and the clean signal. Their confusion matrices likewise showed that their classifier struggled to distinguish SingleAM from SingleFM interferences and NB from SingleChirp interferences. Moreover, our obtained accuracies for $M = 20$ when classifying the DME and NB types exceed the accuracies achieved by the benchmark neural network. Our proposed FL framework therefore allows us to obtain results comparable to those from state-of-the-art centralized classification algorithms while also preserving user data privacy and security.

4.5 | Comparison of FL Algorithms

Despite achieving comparable results to the benchmark, the above FL algorithm nevertheless yields unfavorable outcomes with low accuracy when applied to non-IID datasets. In this section, we compare four different FL algorithms to assess their performance in non-IID data settings. This evaluation encompasses several metrics. First, we consider accuracy (Acc), which was discussed earlier, and we also employ the macro F-1 (F1) score to account for variations in sample and class distribution across different clients. Furthermore, we evaluate the classification accuracy for new data points that lack prior information from any specific clients. The corresponding metrics for these data points are the server accuracy (S-Acc) and the server F1 score. Of the four FL algorithms we consider, FedAvg, FedProx, and Ditto learn a global model automatically. On the other hand, FedAMP only

TABLE 1
Results of different FL algorithms. "Acc" denotes Accuracy, "F1" refers to the macro F1 score, "S-Acc" represents Server Accuracy, and "S-F1" signifies the Server F1 score. The numbers following these abbreviations indicate the quantization bits, such as "Acc-8" for Accuracy with 8 bits.

Data	Algorithms	Acc	F1	S-Acc	S-F1	Acc-8	F1-8	S-Acc-8	S-F1-8	Acc-4	F1-4	S-Acc-4	S-F1-4
(40, 0.1)	FedAvg	82.84	82.75	**82.84**	**82.75**	69.86	50.96	**69.86**	**50.96**	35.92	10.55	**35.92**	10.55
	FedProx	73.38	73.04	73.38	73.04	59.69	43.35	59.69	43.35	28.97	14.12	28.97	**14.12**
	FedAMP	**94.61**	81.92	75.29	74.7	**96.4**	**86.83**	60.49	44.37	**79.24**	**58.22**	21.42	5.88
	Ditto	93.63	**83.3**	72.47	72.8	95.2	84.45	67.17	40.97	53.02	35.68	34.63	8.57
(20, 0.1)	FedAvg	85.55	**85.21**	**85.55**	**85.21**	74.56	**73.07**	**74.56**	**73.07**	33.09	21.49	**33.09**	**21.49**
	FedProx	60.95	55.22	60.95	55.22	57.41	51.29	57.41	51.29	18.59	7.44	18.59	7.44
	FedAMP	**95.34**	71.74	61.71	56.2	**93.87**	69.97	59.91	54.01	**71.39**	**37.33**	17.16	4.88
	Ditto	95.29	71.99	71.29	68.37	93.41	65.64	62.49	57.26	30.76	24.37	28.4	16.34
(20, IID)	FedAvg	91.50	91.40	91.50	91.40	**89.7**	**89.66**	**89.7**	**89.66**	30.07	24.77	30.07	24.77
	FedProx	**91.63**	**91.55**	**91.63**	**91.55**	83.6	83.51	83.6	83.51	33.88	26.73	33.88	26.73
	FedAMP	90.51	90.23	89.95	89.85	87.94	87.68	87.4	87.31	36.04	30.08	40.09	**35.92**
	Ditto	88.47	87.65	90.26	90.16	85.49	84.78	86.17	86.12	**46.30**	**35.92**	**44.24**	35.49

uses personalized models, which require a fusion strategy. Moreover, we assess how the number of quantization bits affects the performance of these algorithms. We evaluate performance using both 8-bit and 4-bit settings, as the difference from the original 32-bit configuration is marginal for 16-bit encoding, while the 2-bit scheme yields unsatisfactory results.

The results for the various FL algorithms are presented in Table 1. Based on the results in the IID data setting (bottom rows), personalized models like FedAMP do not demonstrate a significant advantage over centralized methods such as FedAvg and FedProx, which consistently achieved superior outcomes. Even so, the performance differences among all algorithms were relatively minor. However, in the non-IID data setting, personalized FL algorithms clearly outperform centralized algorithms with respect to accuracy. Notably, FedAMP consistently delivers the highest accuracy across all data configurations, maintaining superior performance even with 4-bit quantization, where its accuracy exceeds 70%. In contrast, other approaches generally yield accuracies at or below 50% at 4-bit quantization.

Ditto also demonstrates commendable performance, with the unquantized and 8-bit quantization cases achieving similar performance as FedAMP, but for the 4-bit quantization case, Ditto loses the ability to match the high level of accuracy achieved by FedAMP. Even so, Ditto usually achieves higher server accuracy and server F1 score than FedAMP. This discrepancy arises because Ditto learns a global model alongside the personalized models, which provides it with greater robustness. In contrast, FedAMP relies solely on personalized models, and server accuracy is derived from the fusion of the personalized models.

With respect to server accuracy and the F1 score, centralized FL (i.e., FedAvg) is the strongest approach. As shown in Table 1., FedAvg consistently achieves top-tier server accuracy and F1 scores across various quantization scenarios. Conversely, FedProx does not perform as well under our experimental settings, suggesting a potential requirement for further hyperparameter optimization. Such optimization falls outside the scope of this study.

Finally, our analysis reveals that quantization significantly influences the training process: lower bit precision resulted in poorer outcomes. Nevertheless, the results at 8-bits remain commendably robust, especially within personalized FL frameworks. This finding offers valuable insights for the design of future FL systems.

5 | CONCLUSION

This paper demonstrates the efficacy of selected FL algorithms in the context of GNSS jamming classification. These algorithms would allow the successful implementation of a crowdsourcing scheme in which real data is gathered without compromising user privacy. We provide results of spectrogram image classification for simulated GNSS signals affected by six different jammer types. Under certain FL configurations, classification accuracies are high for all the studied jammer types, though DME and clean signals were consistently classified with the highest accuracies (above 99%). Conversely, the classifier could struggle to distinguish between AM and FM and between NB and Chirp jammer types. Nevertheless, the FL framework proposed herein performed favorably relative to the benchmark centralized classification algorithm in Morales Ferre et al. (2019), showing that it is possible to work in a collaborative scenario that protects user privacy without causing performance to drop. Our experimental results specifically showed that *i*) it is more difficult to learn non-IID data than IID data; *ii*) assuming the total number of data points is the same, having more clients each with fewer data points decreases classifier performance; *iii*) personalized FL algorithms are more effective at handling non-IID data; and *iv*) choosing a different quantization bit number can reduce communication costs while still maintaining good performance. Future research in this area will include collecting real-world data to investigate various practical non-IID scenarios. These scenarios include cases where clients are situated at varying distances from jammer sources and where clients are situated across diverse environments where the signal characteristics differ despite being of the same type. Finally, we aim to explore and develop various FL algorithms to enhance classification efficiency while maintaining performance and privacy (Wu, 2024). These algorithms could involve the use of different quantization methods (Almanifi et al., 2023) and differential privacy techniques (Yin et al., 2021).

REFERENCES

Almanifi, O. R. A., Chow, C.-O., Tham, M.-L., Chuah, J. H., & Kanesan, J. (2023). Communication and computation efficiency in federated learning: A survey. *Internet of Things*, *22*, 100742. https://doi.org/10.1016/j.iot.2023.100742

Amin, M. G., Closas, P., Broumandan, A., & Volakis, J. L. (2016). Vulnerabilities, threats, and authentication in satellite-based navigation systems [scanning the issue]. *Proceedings of the IEEE*, *104*(6), 1169–1173. https://doi.org/10.1109/JPROC.2016.2550638

Arjoune, Y., Salahdine, F., Islam, M. S., Ghribi, E., & Kaabouch, N. (2020). A novel jamming attacks detection approach based on machine learning for wireless communication. *Proc. of the 2020 International Conference on Information Networking (ICOIN),* Barcelona, Spain, 459–464. https://doi.org/10.1109/ICOIN48656.2020.9016462

Baskin, C., Liss, N., Schwartz, E., Zheltonozhskii, E., Giryes, R., Bronstein, A. M., & Mendelson, A. (2021). UNIQ: Uniform noise injection for non-uniform quantization of neural networks. *ACM Transactions on Computer Systems*, *37*(1–4), 1–15. https://doi.org/10.1145/3444943

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, *79*(1–2), 151–175. https://doi.org/10.1007/s10994-009-5152-4

Borio, D., & Closas, P. (2017). A fresh look at GNSS anti-jamming. *Inside GNSS*, 54–61. https://insidegnss.com/a-fresh-look-at-gnss-anti-jamming/

Borio, D., Dovis, F., Kuusniemi, H., & Presti, L. L. (2016). Impact and detection of GNSS jammers on consumer grade satellite navigation receivers. *Proceedings of the IEEE*, *104*(6), 1233–1245. https://doi.org/10.1109/JPROC.2016.2543266

Borio, D., Li, H., & Closas, P. (2018). Huber's non-linearity for GNSS interference mitigation †. *Sensors*, *18*(7). https://doi.org/10.3390/s18072217

Brieger, T., Raichur, N. L., Jdidi, D., Ott, F., Feigl, T., van der Merwe, J. R., Felber, W. (2022). Multimodal learning for reliable interference classification in GNSS signals. *Proc. of the 35th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2022),* Denver, CO, 3210–3234. https://doi.org/10.33012/2022.18586

Chen, H., Chen, H., Lei, Z., Zhang, L., Li, B., Zhang, J., & Wang, Y. (2024). Compound jamming recognition based on a dual-channel neural network and feature fusion. *Remote Sensing*, *16*(8). https://doi.org/10.3390/rs16081325

Closas, P., Ortega, L., Lesouple, J., & Djurić, P. M. (2024). Emerging trends in signal processing and machine learning for positioning, navigation and timing information: Special issue editorial. *EURASIP Journal on Advances in Signal Processing*, *2024*(84). https://doi.org/10.1186/s13634-024-01182-8

Damgaard, P. H., & Hüffel, H. (1987). Stochastic quantization. *Physics Reports*, *152*(5), 227–398. https://doi.org/10.1016/0370-1573(87)90144-X

Deng, M., Luo, R., & Yao, Z. (2024). GNSS interference signal classification based on federated learning. *Proc. of the 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall),* Washington, DC, 1–7. https://doi.org/10.1109/VTC2024-Fall63153.2024.10757822

Deng, Y., Kamani, M. M., & Mahdavi, M. (2020). Adaptive personalized federated learning. *arXiv preprint,* arXiv:2003.13461. https://doi.org/10.48550/arXiv.2003.13461

Dovis, F. (2015). *GNSS interference threats and countermeasures.* Artech House. https://ieeexplore.ieee.org/document/9100186

Fallah, A., Mokhtari, A., & Ozdaglar, A. (2020). Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, *33*, 3557–3568. https://proceedings.neurips.cc/paper_files/paper/2020/file/24389bfe4fe2eba8bf9aa9203a44cdad-Paper.pdf

Gray, R. (1984). Vector quantization. *IEEE ASSP Magazine*, *1*(2), 4–29. https://doi.org/10.1109/MASSP.1984.1162229

Hsu, T.-M. H., Qi, H., & Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv e-prints*, arXiv:1909.06335. https://doi.org/10.48550/arXiv.1909.06335

Huang, Y., Chu, L., Zhou, Z., Wang, L., Liu, J., Pei, J., & Zhang, Y. (2021). Personalized cross-silo federated learning on non-IID data. *Proceedings of the AAAI Conference on Artificial Intelligence 35*(9), 7865–7873. https://doi.org/10.1609/aaai.v35i9.16960

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proc. of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* Salt Lake City, UT, 2704–2713. https://doi.org/10.1109/CVPR.2018.00286

Jiang, M., Ye, Z., Xiao, Y., & Gou, X. (2024). Federated transfer learning aided interference classification in GNSS signals. *Proc. of the 2024 IEEE/CIC International Conference on Communications in China (ICCC),* Hangzhou, China, 1988–1993. https://doi.org/10.1109/ICCC62479.2024.10681801

Jiang, Y., Konečný, J., Rush, K., & Kannan, S. (2019). Improving federated learning personalization via model agnostic meta learning. *arXiv preprint,* arXiv:1909.12488. https://doi.org/10.48550/arXiv.1909.12488

Krishnamoorthi, R. (2018). Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint,* arXiv:1806.08342. https://doi.org/10.48550/arXiv.1806.08342

Lang, N., & Shlezinger, N. (2022). Joint privacy enhancement and quantization in federated learning. *Proc. of the 2022 IEEE International Symposium on Information Theory (ISIT),* Espoo, Finland, 2040–2045. https://doi.org/10.1109/ISIT50566.2022.9834551

Lee, G.-H., Jo, J., & Park, C. H. (2020). Jamming prediction for radar signals using machine learning methods. *Security and Communication Networks*, *2020*(1), 2151570. https://doi.org/10.1155/2020/2151570

Li, D., & Wang, J. (2019). FedMD: Heterogenous federated learning via model distillation. *arXiv preprint,* arXiv:1910.03581. https://doi.org/10.48550/arXiv.1910.03581

Li, H., Borio, D., & Closas, P. (2019). Dual-domain robust GNSS interference mitigation. *Proc. of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019),* Miami, FL, 991–1002. https://doi.org/10.33012/2019.16991

Li, Q., He, B., & Song, D. (2021). Model-contrastive federated learning. *Proc. of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Nashville, TN, 10708–10717. https://doi.org/10.1109/CVPR46437.2021.01057

Li, T., Hu, S., Beirami, A., & Smith, V. (2021). Ditto: Fair and robust federated learning through personalization. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (Vol. 139, 6357–6368). PMLR. https://doi.org/10.48550/arXiv.2012.04221

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, *37*(3), 50–60. https://doi.org/10.1109/MSP.2020.2975749

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* (Vol. 2, 429–450). https://doi.org/10.48550/arXiv.1812.06127

Li, X., Huang, K., Yang, W., Wang, S., & Zhang, Z. (2020). On the convergence of FedAvg on non-IID data. *Proc. of the 2020 International Conference on Learning Representations (ICLR).* https://doi.org/10.48550/arXiv.1907.02189

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. y. (2017). Communication-efficient learning of deep networks from decentralized data. In A. Singh & J. Zhu (Eds.), *Proceedings of the 20th international conference on artificial intelligence and statistics* (Vol. 54, 1273–1282). PMLR. https://doi.org/10.48550/arXiv.1602.05629

Mehr, I. E., & Dovis, F. (2022). Detection and classification of GNSS jammers using convolutional neural networks. *Proc. of the 2022 International Conference on Localization and GNSS (ICL-GNSS),* Tampere, Finland, 1–6. https://doi.org/10.1109/ICL-GNSS54081.2022.9797030

Mehr, I. E., & Dovis, F. (2024). A deep neural network approach for classification of GNSS interference and jammer. *IEEE Transactions on Aerospace and Electronic Systems*, 1–18. https://doi.org/10.1109/TAES.2024.3462662

Mehr, I. E., Minetto, A., & Dovis, F. (2023). A navigation signals monitoring, analysis and recording tool: Application to real-time interference detection and classification. *Proc. of the 36th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2023),* Denver, CO, 3878–3887. https://doi.org/10.33012/2023.19391

Mitch, R. H., Dougherty, R. C., Psiaki, M. L., Powell, S. P., O'Hanlon, B.W., Bhatti, J. A., & Humphreys, T. E. (2011). Signal characteristics of civil GPS jammers. *Proc. of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2011),* Portland, OR, 1907–1919.

Morales Ferre, R., de la Fuente, A., & Lohan, E. S. (2019). Jammer classification in GNSS bands via machine learning algorithms. *Sensors*, *19*(22). https://doi.org/10.3390/s19224841

Morales-Ferre, R., Richter, P., Falletti, E., de la Fuente, A., & Lohan, E. S. (2020). A survey on coping with intentional interference in satellite navigation for manned and unmanned aircraft. *IEEE Communications Surveys Tutorials*, *22*(1), 249–291. https://doi.org/10.1109/COMST.2019.2949178

Morton, Y. J., van Diggelen, F., Spilker Jr, J. J., Parkinson, B. W., Lo, S., & Gao, G. (2021). *Position, navigation, and timing technologies in the 21st century: Integrated satellite navigation, sensor systems, and civil applications*. John Wiley & Sons. https://doi.org/10.1002/9781119458449

Nardin, A., Imbiriba, T., & Closas, P. (2023). Crowdsourced jammer localization using APBMs: Performance analysis considering observations disruption. *Proc. of the 2023 IEEE/ION Position, Location and Navigation Symposium (PLANS),* Monterey, CA, 511–519. https://doi.org/10.1109/PLANS53410.2023.10140023

Nicola, M., Falco, G., Morales Ferre, R., Lohan, E.-S., De La Fuente, A., & Falletti, E. (2020). Collaborative solutions for interference management in GNSS-based aircraft navigation. *Sensors*, *20*(15), 4085. https://doi.org/10.3390/s20154085

Niknam, S., Dhillon, H. S., & Reed, J. H. (2020). Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, *58*(6), 46–51. https://doi.org/10.1109/MCOM.001.1900461

Park, J., Moon, J., Kim, T., Wu, P., Imbiriba, T., Closas, P., & Kim, S. (2022). Federated learning for indoor localization via model reliability with dropout. *IEEE Communications Letters*, *26*(7), 1553–1557. https://doi.org/10.1109/LCOMM.2022.3170878

Pastor, F., García-González, J., Gandarias, J. M., Medina, D., Closas, P., García-Cerezo, A. J., & Gómez-de Gabriel, J. M. (2021). Bayesian and neural inference on LSTM-based object recognition from tactile and kinesthetic information. *IEEE Robotics and Automation Letters*, *6*(1), 231–238. https://doi.org/10.1109/LRA.2020.3038377

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., & Pedarsani, R. (2020). FedPAQ: A communication efficient federated learning method with periodic averaging and quantization. In S. Chiappa & R. Calandra (Eds.), *Proceedings of the twenty third international conference on artificial intelligence and statistics* (Vol. 108, 2021–2031). PMLR. https://doi.org/10.48550/arXiv.1909.13014

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint,* arXiv:1609.04747. https://doi.org/10.48550/arXiv.1609.04747

Strizic, L., Akos, D. M., & Lo, S. (2018). Crowdsourcing GNSS jammer detection and localization. *Proc. of the 2018 International Technical Meeting of the Institute of Navigation,* Reston, VA, 626–641. https://doi.org/10.33012/2018.15546

Tan, A. Z., Yu, H., Cui, L., & Yang, Q. (2023). Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(12), 9587–9603. https://doi.org/10.1109/TNNLS.2022.3160699

Thombre, S., Bhuiyan, M. Z. H., Eliardsson, P., Gabrielsson, B., Pattinson, M., Dumville, M., others (2018). GNSS threat monitoring and reporting: Past, present, and a proposed future. *The Journal of Navigation*, *71*(3), 513–529. https://doi.org/10.1017/S0373463317000911

van der Merwe, J. R., Franco, D. C., Feigl, T., & Rügamer, A. (2024). Optimal machine learning and signal processing synergies for low-resource GNSS interference classification. *IEEE Transactions on Aerospace and Electronic Systems*, *60*(3), 2705–2721. https://doi.org/10.1109/TAES.2023.3349360

Voigt, J. M. (2021). *Classification of GNSS jammers using machine learning: Multivariate time series and image classification based approaches* [Master's thesis, University of Agder]. https://hdl.handle.net/11250/2823900

Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., & Ramage, D. (2019). Federated evaluation of on-device personalization. *arXiv e-prints*, arXiv:1910.10252. https://doi.org/10.48550/arXiv.1910.10252

Widrow, B., Kollar, I., & Liu, M.-C. (1996). Statistical theory of quantization. *IEEE Transactions on Instrumentation and Measurement*, *45*(2), 353–361. https://doi.org/10.1109/19.492748

Wu, P. (2024). *Bayesian data fusion for distributed learning* [Doctoral thesis, Northeastern University]. Northeastern University Library. https://doi.org/10.17760/D20659818

Wu, P., Calatrava, H., Imbiriba, T., & Closas, P. (2023). Jammer classification with federated learning. *Proc. of the 2023 IEEE/ION Position, Location and Navigation Symposium (PLANS),* Monterey, CA, 228–234. https://doi.org/10.1109/PLANS53410.2023.10140124

Wu, P., Imbiriba, T., Elvira, V., & Closas, P. (2024). Bayesian data fusion with shared priors. *IEEE Transactions on Signal Processing*, *72*, 275–288. https://doi.org/10.1109/TSP.2023.3343564

Wu, P., Imbiriba, T., Park, J., Kim, S., & Closas, P. (2021). Personalized federated learning over non-IID data for indoor localization. *Proc. of the 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC),* Lucca, Italy, 421–425. https://doi.org/10.1109/SPAWC51858.2021.9593115

Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Transactions on Computer Systems*, *54*(6). https://doi.org/10.1145/3460427

Zengyuan, J., Xiaoying, Z., Siyuan, T., Xueqing, Z., & Jibo, W. (2023). Jamming identification based on inverse residual neural network with integrated time-frequency channel attention. *Journal of Signal Processing*, *39*(2), 343–355. https://doi.org/10.16798/j.issn.1003-0530.2023.02.015

How to cite this article: Wu, P., Calatrava, H., Imbiriba, T., & Closas, P. (2025). Federated learning of jamming classifiers: From global to personalized models. *NAVIGATION, 72*(1). https://doi.org/10.33012/navi.688