Jammer classification with Federated Learning

Peng Wu, Helena Calatrava, Tales Imbiriba, Pau Closas
Northeastern University
Dept. of Electrical & Computer Eng.
Boston, MA (USA)
{wu.p, calatrava.h, talesim, closas}@northeastern.edu

2023 IEEE/ION Position, Location and Navigation Symposium (PLANS) | 978-1-6654-1772-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PLANS53410.2023.10140124

***Abstract*—Jamming signals can jeopardize the operation of GNSS receivers until deying its operation. Given their ubiquity, jamming mitigation and localization techniques are of crucial importance, for which jammer classification is of help. Data-driven models have been proven useful in detecting these threats, while their training using crowdsourced data still poses challenges when it comes to private data sharing. This article investigates the use of federated learning to train jamming signal classifiers locally on each device, with model updates aggregated and averaged at the central server. This allows for privacy-preserving training procedures that do not require centralized data storage or access to client local data. The used framework FedAvg is assessed on a dataset consisting of spectrogram images of simulated interfered GNSS signal. Six different jammer types are effectively classified with comparable results to a fully centralized solution that requires vast amounts of data communication and involves privacy-preserving concerns.**

***Index Terms*—Jamming detection, machine learning, distributed inference, neural networks, federated learning.**

I. Introduction

GNSS jamming signals are L-band spectrum interferences that can overpower a GNSS receiver until denying its operation [1], [2]. A wide variety of jammers can be found in the online market at very cheap prices, which makes human-made intentional jamming signals a threat [3], [4]. In addition, signals do not need to be malicious to have a jamming effect, where multiple examples exist of legitimate waveforms that can pose a threat to GNSS receivers, such as continuous wave (CW) interferences produced by damaged electronics and signals emitted by Distance Measurement Equipment (DME) technology conceived for aircraft navigation [5]. Jamming sources are placed on Earth or, in the case of drone jammers, in the proximity of the Earth's surface. As a consequence of the path-loss attenuation given by the large distance between Earth and GNSS satellites, jamming interferences are received with remarkably higher power than the useful GNSS signal, which can lead to performance disruption in areas with a radius of several kilometers [6]. In the literature, it is suggested that jamming is the main cause of GNSS-based service outages [7] and, consequently, we consider that protection against this kind of attack is a desirable feature in GNSS receivers [8], [9].

Jammer classification can be of great help to classical Interference Classification (IC) techniques, which are formulated as an estimation problem where the jamming signal is detected and estimated, often with a parametric model [10]. As the aim of these techniques is to first reconstruct the interference, the knowledge of its type or class is key to speed up the algorithm. For instance, if knowing that a CW interference is threatening a receiver, it would only be required to estimate the interference central frequency in order to reconstruct its waveform and implement an IC measure. It is also relevant to highlight that by performing jamming classification, the task of detection is explicitly taken care of. In the vast majority of previous GNSS studies regarding protection against jamming interferences, the focus is on its detection [11], mitigation [12], and localization [13]. Nevertheless, and according to [7], little effort had been dedicated to the classification of jamming signals until recent publications, besides some work in the context of radar systems such as the Machine Learning (ML) jamming prediction algorithm proposed in [14]. In [7], they propose a Support Vector Machine (SVM) and a Convolutional Neural Network (CNN) based classifiers for the purpose of jammer classification, which they treat as an image classification problem. They suggest that with a small amount of training data, it is possible to achieve classification accuracy above 90%, being the accuracy of jamming detection close to 99%. The use of multivariate time-series approaches can also lead to an increase in classification accuracy in jammer classification techniques, according to the work presented in [15], which makes use of state-of-the-art ML techniques.

Most studies on GNSS integrity rely on synthetic data since data collection in the presence of jamming signals can represent a tremendous effort. This is especially so if different interference types and received power values are desired. Besides recreating effects such as the ones introduced by multipath reflections can be difficult. The use of real GNSS interfered data is, however, of great interest when it comes to training data-driven classifiers, as well as effectively assessing their performance. An option to collect real GNSS data is to resort to traditional crowdsourcing approaches, where clients record data and share it with a central unit that is in charge of training the classifier. Nevertheless, crowdsourcing has some concerns about user privacy, as it requires that the users involved in the scenario send their data directly to a centralized server. Aimed at solving this limitation, Federated Learning (FL) has recently attracted great interest due to its privacy-protecting nature and the efficient use of resources by harnessing the processing power of edge devices [16]. FL is a promising solution that enables many clients to jointly train machine learning models

This work has been partially supported by the National Science Foundation under Award ECCS-1845833.

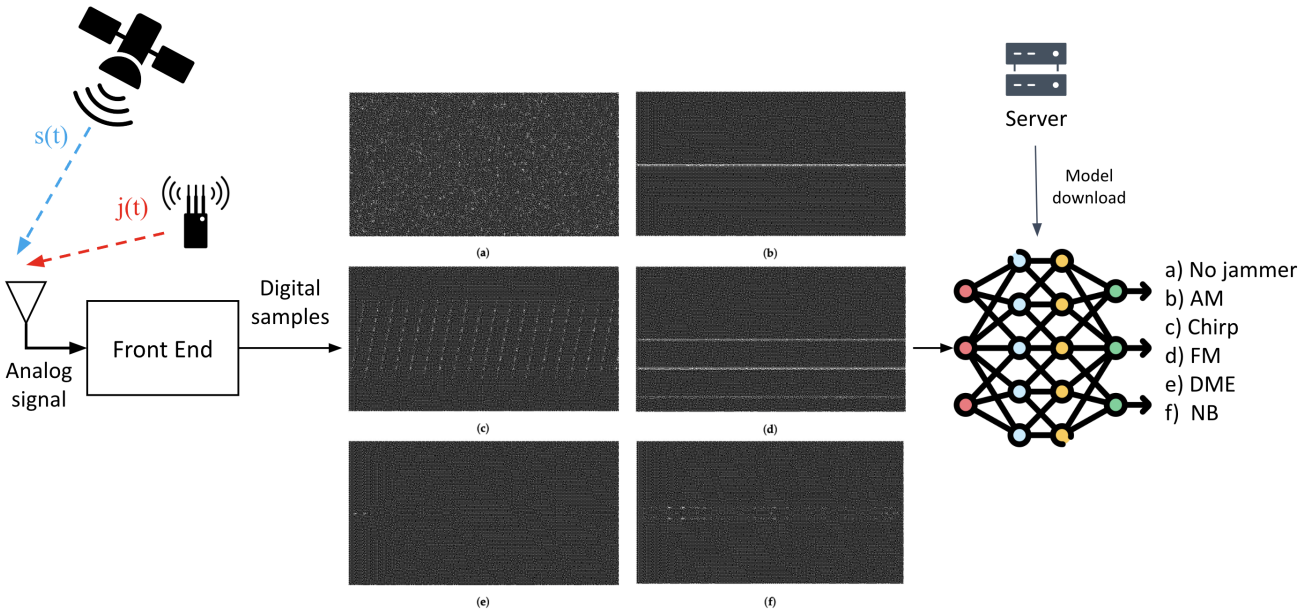

Fig. 1: System diagram of the considered jamming signal classification: a receiver downloads a pretrained model from the server, which can be either i) trained on locally available data and sent back to the server for fusion with other models; or ii) used to perform jamming classification results based on local data. Monochrome spectrogram images of the six jammer types available in the used dataset [7] are shown, namely (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) corresponds to clean signal (no interference).

while maintaining local data decentralization. Collaboration between users in distributed scenarios has been proven useful in GNSS interference management tasks [17]. With FL, instead of exchanging data and conducting centralized training, each party sends its model to the server, which updates a joint model and sends the global model back to the parties. Since their original data is not exposed, FL is an effective way to address privacy issues [18].

In this paper, we aim at training jamming signal classifiers using privacy-preserving strategies that can cope with crowdsourcing data collection strategies. Our overall goal is to obtain a Neural Network (NN) based global model capable of classifying different jamming signals as depicted in Figure 1. To preserve client privacy while leveraging crowdsourcing data collection strategies we exploit FL approaches as shown in Figure 2 where model parameters are shared with clients allowing for local classification of jamming signals and avoiding data sharing. In the proposed framework we assume the possible existence of C different jamming types while the FL approach is performed over a network with M collaborative users. We study the FL-based jamming classifier under different data distribution scenarios. In the first scenario clients' data is independent and identically distributed (IID), that is, all clients observe a similar amount of instances from all C classes. In the second, and more challenging, scenario clients observe data that is unbalanced towards different classes. Working with non-IID data poses several challenges that are common in realistic scenarios, given that not all clients have access to all available types of data. In the context of this work, this is the case when not all participating users observe the same classes of jammers.

The remainder of this paper is organized as follows. Section II provides a description of the satellite signal model and targeted jammer types. The used FL technique is derived in Section III, while the experimental setup and results can be found in Section IV. Finally, Section V concludes the paper.

II. System Model

For the purpose of this article, the analog baseband equivalent of the received GNSS signal can be modeled 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, typically modeled as an additive white Gaussian noise (AWGN) 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 [4]. Accurate knowledge of $j(t)$ allows for prompt reaction to a jamming threat, either for its localization [19] or mitigation. Related to the latter, IC techniques aim to estimate the waveform of $j(t)$ so that it can be reconstructed and directly subtracted from $r(t)$. As it has been previously mentioned in Section I, identifying the type of waveform of $j(t)$ can be useful for several purposes, including its reconstruction and mitigation through IC techniques.

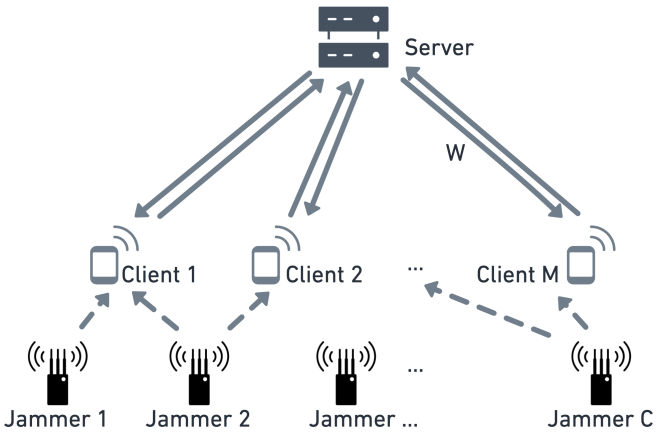

Fig. 2: Federated learning framework to train jamming signal classifiers: M collaborative clients receive the parameters of the classifier from a server; these clients retrain the model based on local data; and upload their updated classifier to the server in charge of fusing the results. This process does not require exchange of actual data or positions from the clients, thus preserving their privacy.

Jammers can be classified according to several features such as the type of device by which they are broadcasted, their frequency spectrum, and their number of antennae [3]. In this paper, we are targeting the same jammer types as in [7], given that we are using the dataset they provide and using their results as a benchmark. The aim of our research is to use the FL technique explained in Section III for the classification of the following jammer types. This classification is mostly performed according to their behavior in the frequency domain.

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 [7], we are not considering Wide Band (WB) jammers given that it is very difficult to detect their presence when analyzing spectrogram images. Jammers of types 1) to 5) have narrow spectrums which overpower the signal of interest buried in noise. We would also like to point out that the classification strategy, proposed in Section III, is able to perform the detection task, as the absence of interference can be properly identified. The five waveform expressions of $j\,(t)$ for each jammer type presented in the list above can be found in [7], and are not explicitly used in both training or testing of the proposed FL solution.

While AM and FM jammers target pre-fixed frequencies, other jammers such as chirp jammers sweep over different frequency bands. Consequently, feature extraction approaches based on spectral analysis, such as spectrograms, of the signals are suitable for distinguishing different jammer types. This is so since the short-time Fourier transform allows the time-frequency localization of the interference signal. In [7], they successfully approached jammer classification as an image classification problem, where spectrograms of the received signal $r(t)$ were treated as images. More precisely, the spectrograms are computed on the discrete-time version of $r(t)$ in 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 $n \in \mathbb{Z}$.

III. Federated Learning Methodology

A significant number of FL algorithms have been discussed in different areas [20], [21], especially in the field of image classification. One *de facto* approach for FL is Federated Averaging (FedAvg) [18], which fuses the model parameters by a weighted sum. According to previous studies [22], [23], the learning effectiveness of standard FL methods is compromised under non-IID data settings.

In this section, we aim on leveraging FL strategies to learn a unique global model capable of making accurate predictions on data available to different clients. More precisely, we consider the setup depicted in Fig. 2, where M collaborative clients aim at training a global classification model (e.g. a neural network) such that class posteriors

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

where $\mathbf{y} \in \mathbb{R}^C$ is the vector of class posteriors with elements $p(y = \ell|\mathbf{X})$, with $\ell \in \{\text{AM, Chirp, FM, DME, NB, NO}\}$, $\mathbf{h} : \mathbf{X} \mapsto \mathbf{h}(\mathbf{X})$ is the NN 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 [7] 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 datasets $\mathcal{D}_i = \{\mathbf{y}_n^{(i)}, \mathbf{X}_n^{(i)}\}_{n=1}^{L_i}$, $i \in \{1, \ldots, M\}$.

Mathematically, the training process can be formulated as the minimization of a loss function:

$$\min_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega}) \text{ where } \mathcal{L}(\boldsymbol{\omega}) = \sum_{i=1}^{M} \mathcal{F}_i(\boldsymbol{\omega}) \tag{3}$$

where $\mathcal{L}(\boldsymbol{\omega})$ is the global loss functional, while $\mathcal{F}_i : \mathbb{R}^d \to \mathbb{R}, \boldsymbol{\omega} \mapsto \mathcal{F}_i(\boldsymbol{\omega})$ are local loss functions.

Among the different strategies to solve (3) we highlight the conventional FL approach: FedAvg, which considers a single global objective, along with other variants such as FedProx, [24], which consider adding a regularization term to the objective function, to prevent overfitting, and MOON [25], which use a contrastive loss term to control the local model drifts away from the global model. However, our practical experience with these three methods, when applied to the jammer classification problem at hand, is that they perform very similarly with FedAvg presenting a slight advantage. Thus, we report results only for the FedAvg in our experiments in Section IV.

FedAvg is a *de facto* approach for FL in which local models are trained locally. The clients then upload local trained model parameters to a cloud server, which is in charge of fusing it

Algorithm 1 FedAvg Algorithm

Input: number of clients M; the architecture of local models $\mathbf{h}$ with initial $\boldsymbol{\omega}_0$; local loss functions $\mathcal{F}_i$; data $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$; number of iterations T; number of local epochs E; learning rate η;
for $t = 1, \ldots, T$ **do**
  for $i \in M$ **do**
    $\boldsymbol{\omega}_{t+1}^i \leftarrow$ solution of (5) using local data $\mathcal{D}_i$ for E epochs with learning rate η
    Upload local model parameters $\boldsymbol{\omega}_{t+1}^i$ to server.
  end for
  Update global model parameters $\boldsymbol{\omega}_{t+1}$ with equation (6) and send it to local clients.
end for
Output: $\boldsymbol{\omega}_T$

(e.g. weights in a neural network) to compute a unique global model. Using the cardinality of the local data $|\mathcal{D}_i|$ as a metric of model reliability, (3) is modified as:

$$\mathcal{L}(\boldsymbol{\omega}) = \sum_{i=1}^{M} \frac{N_i}{N} \mathcal{L}_i(\boldsymbol{\omega}) \tag{4}$$

where $\mathcal{L}_i(\boldsymbol{\omega}) = \frac{1}{N_i} \sum_{n \in \mathcal{D}_i} f_n(\boldsymbol{\omega})$, $f_n(\boldsymbol{\omega})$ is the loss of the prediction using sample n from the dataset $\mathcal{D}_i$. $\mathcal{D}_i$ is the data partition for client i, N_i is the number samples available to the i-th client in $\mathcal{D}_i$, and $N = N_1 + \cdots + N_M$ is the total number of data points.

The optimization in (4) is solved iteratively through multiple rounds of local optimization in the clients and fusion in the server. First, at iteration t each client updates its model parameters solving:

$$\boldsymbol{\omega}_{t+1}^i = \arg\min_{\boldsymbol{\omega}} \mathcal{L}_{i,t}(\boldsymbol{\omega}) \tag{5}$$

where the index t in $\mathcal{L}_{i,t}(\boldsymbol{\omega})$ indicates that the local parameters were initialized using the fused global parameters, $\boldsymbol{\omega}_t$, from the previous iteration. Secondly, a *global update equation* is used, where the global model parameter is updated by averaging the locally updated models from each device as:

$$\boldsymbol{\omega}_{t+1} = \sum_{i=1}^{M} \frac{N_i}{N} \boldsymbol{\omega}_{t+1}^i \tag{6}$$

where $\boldsymbol{\omega}_{t+1}$ is the updated global model, and the sum is over all locally updated models $\boldsymbol{\omega}_{t+1}^i$ from each device. The implementation details of FedAvg are shown in Algorithm 1.

IV. Experiments

This section presents a set of experiments to show the applicability of federated learning to train, in a distributed manner, a jammer classifier able to achieve performances closer to those from a classifier trained on a centralized node with access to all local datasets. We first describe the dataset used, then how it is employed in a distributed learning scheme, how the model was configured, and finally the obtained results.

A. Data preprocessing

The dataset provided by the authors in [7], which is available in open access at https://zenodo.org/record/3370934, is used to conduct the following experiments. It contains 61800 available *.bmp* monochrome spectrogram images with 512×512 pixel resolution, binary scale and 600 DPI. To compute the spectrograms, simulated GNSS signal interfered by the aforementioned jammer types (see Section II) is processed. In [7], the authors used 6000 images for training (1000 for each jammer class), 1800 images for validation and 54000 for testing.

In order to optimize computational resources and expedite the training process we performed some data preprocessing, approach that is usually employed in machine learning context. Particularly, in this paper, we utilized both the training and validation datasets (validation step is often omitted from the experimentation process, unless performing hyperparameter tuning), the combined dataset was then split into a 75% train and 25% test division. Additionally, to further enhance the process, image resolution was reduced from 512×512 to 256×256 pixels through the use of bilinear interpolation techniques. Additionally, once all the data was preprocessed, it was normalized in order to facilitate the training phase.

B. Federated data setting

Two different data settings were investigated. First, the case of an IID setting, wherein all clients received similar data distributions, that is, a similar amount of samples from each class. For the experiments, we uniformly split the data into 20, 30, and 40 clients, in order to examine how client numbers may influence the results. This split resulted in approximately 65, 43, and 32 samples per client for 20, 30, and 40 clients, respectively.

The second set of experiments was for a non-IID setting, where the focus is on having an unbalanced distribution of the class labels for training. To generate non-IID splits of the dataset, we followed the approach in [20], where client data is sampled using a Dirichlet distribution. Specifically, for a given client i, we defined the probability of sampling data from a 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 advantage of this approach is that the imbalance level can be flexibly changed by adjusting the concentration parameter $\beta_{i,j}$. In this paper, the concentration parameter $\beta_{i,j}$ is set to a relatively small value of 0.1, allowing for a more unbalanced partitioning. This is evident when inspecting the distribution of data points among clients, where many clients only contain a few labels. This can be observed in Figure 3, providing a snapshot of the number of samples per class for each client when $M = 20$ clients. This leads to an unequal partition of the data, with some clients containing a disproportionately large or small percentage of certain class labels.

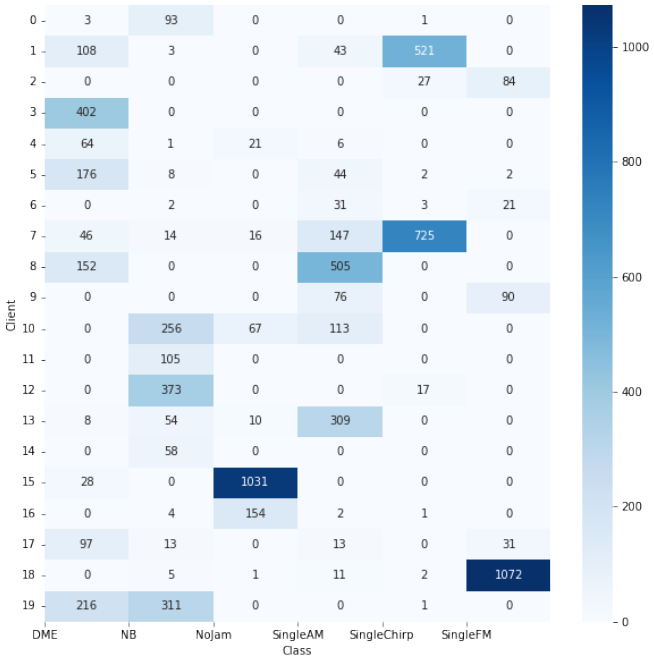

Fig. 3: Number of data points per classes for each of the $M = 20$ clients.

C. Model setting

The authors in [26] employed a convolutional neural network (CNN) for the task of training a classifier based on the full dataset $\mathbf{D}$. This solution becomes the baseline in our results, where the same CNN architecture is considered, while it is trained using the FL framework described earlier. In particular, the architecture of the CNN consisted of one convolutional layer, one pooling layer, and one fully connected layer with a ReLU activation function. The convolution layer utilized 16 filters of size $12 \times 12 \times 1$, with a learning rate of 0.01, and an SGD optimizer [27] was used. The last layer is softmax layer to produce classification results. Plus, Cross-entropy is used for cost function.

D. Results

Figure 4 shows the accuracy of federated averaging algorithms with 400 communication rounds under an IID data setting. The accuracy of the centrally trained model was used as benchmark (around 93.4% accuracy). The figure also compares the accuracy when different numbers of clients M were used. As expected, when a small number of clients were used, better results were achieved when compared to a larger number of clients. Intuitively, fewer clients have more data available and better train local models. Nevertheless, the results show high accuracy results for the tested number of clients.

The confusion matrix in Figure 5 reveals that each jammer class achieves relatively high accuracy, with the DME jammer type and the clean signal (i.e., type NoJam) providing the highest accuracies (over 99%). The classifier is able to detect

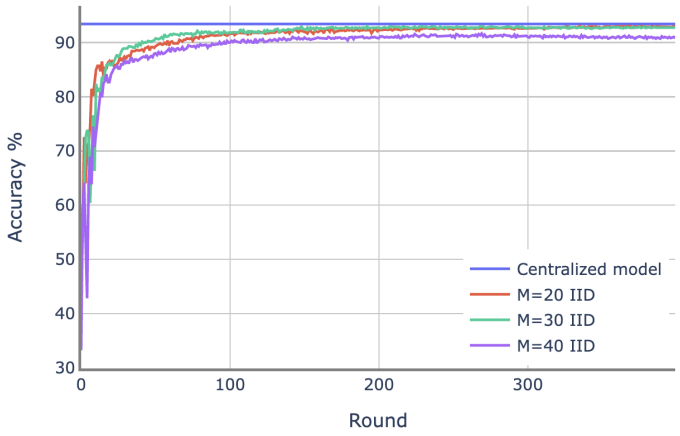

Fig. 4: Accuracy FedAvg in 400 rounds under IID data setting.

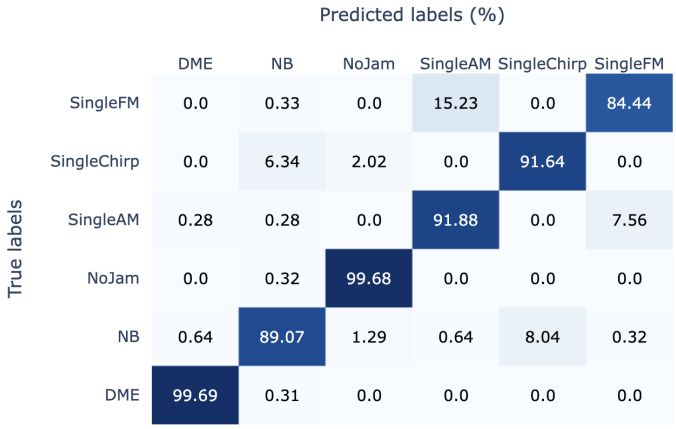

Fig. 5: Confusion matrix after FedAvg training under the IID data setting for $M = 20$ clients.

the absence of interference, as spectrogram (a) from Figure 1 notably differs from the rest. This is 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, as 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. Regarding DME (or pulsed) interferences, they are only active during their duty cycle. If the duty cycle is short with respect to the window duration of the short-time Fourier transform, the resulting spectrum shows an almost clean image with a few magnitude peaks, which notably differs from the spectra of other jammer types. The SingleFM and NB jammer types achieved less than 90% accuracy. If inspecting the confusion matrix non-diagonal elements, it can be seen that it is difficult for the classifier to distinguish between the SingleAM and SingleFM types, as they all span one or two narrow bands of the signal spectrum. The SingleFM spectrogram is equivalent to the SingleAM spectrogram with an additional band. It is also difficult for the classifier to distinguish between the NB and SingleChirp interferences. A possible explanation is that

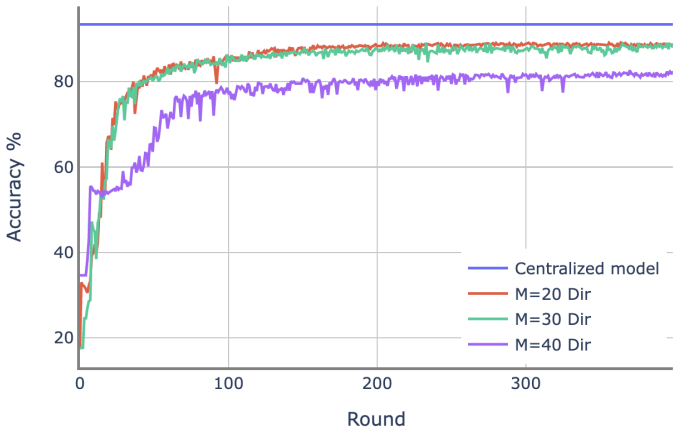

Fig. 6: Accuracy FedAvg in 400 rounds under Dirichlet data setting

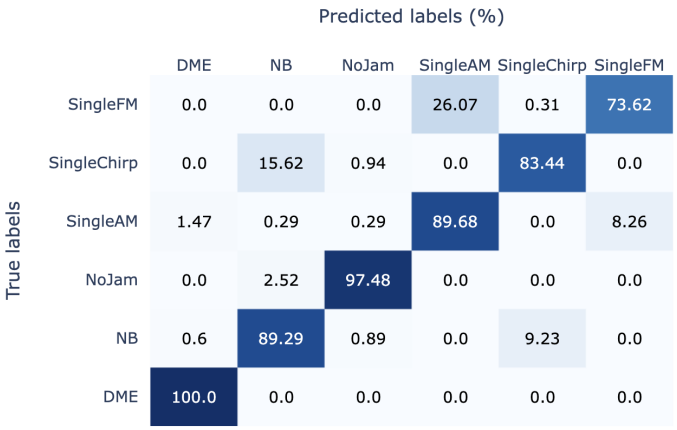

Fig. 7: Confusion matrix after FedAvg training under the non-IID data setting for $M = 20$ clients.

both of them have a lower magnitude in their spectra due to being more spread. This makes their spectrogram images look blurry when compared to the ones from the SingleAM and SingleFM types.

Figure 6 illustrates the accuracy of the FedAvg algorithm under a non-IID data setting, for different numbers of clients and compared to the accuracy of the global benchmark. The results show that the accuracies of the different client numbers are lower than the results of the homogeneous IID data setting, indicative of increased difficulty in learning with the heterogeneity of the data. Moreover, the comparison of different clients is similar to that of the IID data setting: the more clients there were, the lower their accuracy. It is also noticed that when the number of clients was 40, it took more communication rounds to converge than when smaller numbers of clients were considered.

Figures 7 and 8 show the confusion matrix for 20 and 40 clients under the non-IID, Dirichlet data setting. The conclusion still holds that the DME jammer type and clean signal were the easiest to classify, with accuracies of 100%

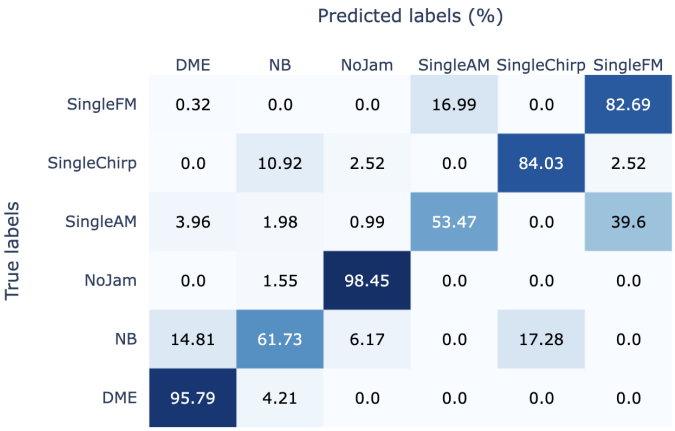

Fig. 8: Confusion matrix after FedAvg training under the non-IID data setting for $M = 40$ clients.

and 97.48% for $M = 20$ and 95.79% and 98.45% for $M = 40$ clients. For $M = 40$, very low accuracies were achieved with the NB and SingleAM jammer types, while the worst accuracy was achieved with the SingleFM jammer type for $M = 20$. As in Figure 5, when inspecting the non-diagonal elements, it can be seen how it was difficult for the classifier to distinguish between SingleAM and SingleFM types, and also between NB and SingleChirp types. For $M = 40$, given that the performance was worse due to having a higher number of clients (implying less local data), it was also difficult for the classifier to distinguish between NB and DME signals. Nevertheless, for a high number of clients (i.e., $M = 40$), accuracies above 80% were obtained 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 the ones obtained with the benchmark training process: the centralized classification algorithm proposed in [7]. In their results, the DME (or pulsed) interference and clean signal also provided the highest accuracy. Also, their confusion matrices showed the classifier difficulty when it comes to distinguishing SingleAM and SingleFM interferences, and also NB and SingleChirp interferences. Our obtained accuracies for $M = 20$ when classifying the DME and NB types exceed the accuracy provided by the benchmark CNN. Consequently, we have shown that the proposed federated learning framework allows to obtain comparable results to the ones offered by state-of-the art centralized classification algorithms while preserving user data privacy and security.

V. Conclusion

This paper demonstrates the efficacy of FL in the context of GNSS jamming classification using the FedAvg, which would allow the successful implementation of a crowdsourcing scheme where real data is gathered without compromising user privacy. Results are provided for spectrogram image classification of simulated GNSS signal under the threat of six different

jammer types. Although classification accuracy results are high under certain configurations for all the studied jammer types, DME and clean signal provide the highest accuracies (above 99%). On the other hand, it is difficult for the classifier to distinguish between AM and FM, as well as between NB and Chirp jammer types. The FL framework performance has been successfully compared to the one provided by the benchmark centralized classification algorithm in [7], showing that it is possible to work in a collaborative scenario without observing a relevant performance drop while preserving user protection. Experimental results showed that i) it is more difficult to learn non-IID data than IID data; and that ii) having fewer data on the local clients decreases the performance of the results.

REFERENCES

[1] M. G. Amin, P. Closas, A. Broumandan, and J. L. Volakis, "Vulnerabilities, threats, and authentication in satellite-based navigation systems [scanning the issue]," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1169–1173, 2016.

[2] Y. J. Morton, F. van Diggelen, J. J. Spilker Jr, B. W. Parkinson, S. Lo, and G. Gao, *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications*. John Wiley & Sons, 2021.

[3] D. Borio, F. Dovis, H. Kuusniemi, and L. L. Presti, "Impact and detection of gnss jammers on consumer grade satellite navigation receivers," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1233–1245, 2016.

[4] R. Morales-Ferre, P. Richter, E. Falletti, A. de la Fuente, and E. S. Lohan, "A survey on coping with intentional interference in satellite navigation for manned and unmanned aircraft," *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 249–291, 2020.

[5] H. Li, D. Borio, and P. Closas, "Dual-Domain Robust GNSS Interference Mitigation," Miami, Florida, Oct. 2019, pp. 991–1002. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=16991

[6] R. Mitch, "Signal characteristics of civil GPS jammers," *24th Int. Tech. Meet. Satell. Div. Inst. Navig. 2011, ION GNSS 2011*, vol. 3, p. 1907, 2011.

[7] R. M. Ferre, A. D. L. Fuente, and E. S. Lohan, "Jammer classification in GNSS bands via machine learning algorithms," *Sensors (Switzerland)*, vol. 19, no. 22, pp. 5–7, 2019.

[8] F. Dovis, *GNSS interference threats and countermeasures*. Artech House, 2015.

[9] S. Thombre, M. Z. H. Bhuiyan, P. Eliardsson, B. Gabrielsson, M. Pattinson, M. Dumville, D. Fryganiotis, S. Hill, V. Manikundalam, M. Pölöskey *et al.*, "GNSS threat monitoring and reporting: Past, present, and a proposed future," *The Journal of Navigation*, vol. 71, no. 3, pp. 513–529, 2018.

[10] D. Borio and P. Closas, "A Fresh Look at GNSS Anti-Jamming," *Inside GNSS*, pp. 54–61, 09 2017.

[11] Y. Arjoune, F. Salahdine, M. Islam, E. Ghribi, and N. Kaabouch, "A novel jamming attacks detection approach based on machine learning for wireless communication," March 2020.

[12] D. Borio, H. Li, and P. Closas, "Huber's Non-Linearity for GNSS Interference Mitigation †," *Sensors*, vol. 18, no. 7, p. 2217, Jul. 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/7/2217

[13] L. Strizic, D. M. Akos, and S. Lo, "Crowdsourcing GNSS Jammer Detection and Localization," January 2018, pp. pp. 626–641.

[14] G.-H. Lee, J. Jo, and C. H. Park, "Jamming Prediction for Radar Signals Using Machine Learning Methods," *Security and Communication Networks*, vol. 2020, pp. 1–9, Jan. 2020. [Online]. Available: https://www.hindawi.com/journals/scn/2020/2151570/

[15] J. M. Voigt, "Classification of GNSS Jammers using Machine Learning : Multivariate Time Series and Image Classification Based Approaches," Master's thesis, University of Agder, 2021, accepted: 2021-10-19T12:18:20Z Publication Title: 59. [Online]. Available: https://uia.brage.unit.no/uia-xmlui/handle/11250/2823900

[16] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[17] M. Nicola, G. Falco, R. Morales Ferre, E.-S. Lohan, A. De La Fuente, and E. Falletti, "Collaborative solutions for interference management in gnss-based aircraft navigation," *Sensors*, vol. 20, no. 15, p. 4085, 2020.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[19] A. Nardin, T. Imbiriba, and P. Closas, "Crowdsourced localization of jammers with an augmented path loss model," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2023.

[20] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[21] P. Wu, T. Imbiriba, J. Park, S. Kim, and P. Closas, "Personalized federated learning over non-iid data for indoor localization," in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2021, pp. 421–425.

[22] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv e-prints*, pp. arXiv–1909, 2019.

[23] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-IID data," in *Eighth International Conference on Learning Representations (ICLR)*, 2020.

[24] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[25] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 713–10 722.

[26] R. Morales Ferre, A. de la Fuente, and E. S. Lohan, "Jammer classification in GNSS bands via machine learning algorithms," *Sensors*, vol. 19, no. 22, p. 4841, 2019.

[27] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.