Classifying UAVs With Proprietary Waveforms via Preamble Feature Extraction and Federated Learning

Guillem Reus-Muns and Kaushik Roy Chowdhury, *Senior Member, IEEE*

***Abstract*—Small unmanned aerial vehicles (UAVs) are deployed in a number of different emerging market segments as well as for recreational hobby flying. Driven by their ubiquitous availability, a large number of manufacturers offer UAV models in different form factors, control and load carrying capacities. This paper proposes a deep learning method to detect the type/model of the UAV using the transmitted RF signals, even when these signals follow proprietary medium access protocols whose headers cannot be decoded. The main contributions are as follows: (i) We show how the preamble portion of the packet is better suited for learning subtle protocol-specific differences, instead of randomly selecting any subset of the transmitted packet, (ii) we propose a pre-processing scheme that generates cross-correlation feature maps to enhance the classification accuracy, (iii) we develop a deep convolutional neural architecture that can be trained in data collected from static scenarios and then tested in practical hovering conditions with 98.2% accuracy of UAV model classification, demonstrating robustness to channel variations, and (iv) we extend this model towards a federated learning paradigm where sensors send individually trained models back to a central controller that combines them, without any appreciable loss of accuracy. Our evaluations are performed on an 8.9 GB dataset collected from static and flying UAVs, which we also release as part of the technical contributions of the work.**

***Index Terms*—UAV classification, deep learning, federated learning.**

I. Introduction

UNMANNED aerial vehicles (UAVs) are ushering in a revolutionary transformation in telecommunications, agriculture, surveillance and construction sectors. Recent market analysis suggests that the UAV market will increase to 63.6 billion USD by 2025 [1]–[3]. However, the easy availability of small form factor UAVs, driven by business and hobby needs, raises concerns over malicious UAV usage. Some examples of such undesirable activity include privacy intrusion [4] and weaponized attacks [5]–[7]. Thus, developing competent UAV identification mechanisms that can detect the arriving UAVs in flight will help to take preemptive measures. The overarching goal of this paper is to detect the make/model of the hovering UAV using its RF control and data signaling, assuming a general case that the UAVs transmit proprietary waveforms that cannot be decoded. A combination of passive and active methods have been previously proposed to detect and classify UAVs, both in the RF and non-RF domains [3], [8], [9].

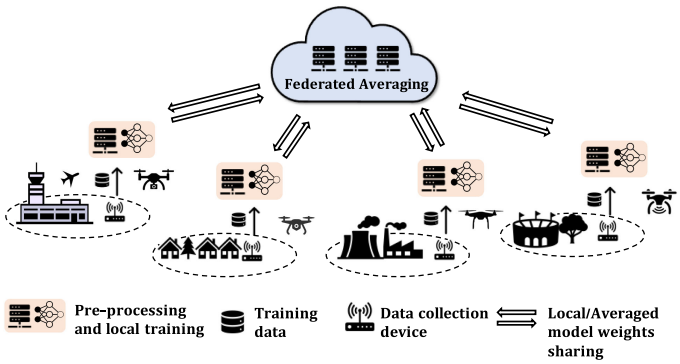


Fig. 1. A set of distributed devices train independent UAV classification models using their locally collected data and the cross-correlation based feature extraction. All models are combined into one at the cloud through Federated Averaging and shared back to each independent device for continuous training and testing.

In-flight UAVs and the ground-based radio controllers (RCs) are in constant communication to exchange control and telemetry signals. Thus, using these RF signals is an opportunity to passively identify unique signatures for each UAV. Our key insight here is that even when proprietary communication protocols are used, physical layer (PHY) frames always contain specific preambles at the beginning of each frame. These preambles are used to carry out time/frequency synchronization and channel estimation tasks. While the preambles themselves may serve as discriminating features for different UAV models, we choose to operate at the I/Q sample-level, with no *a priori* knowledge of the transmission rate, modulation scheme, or bit pattern that composes each preamble. However, since the preambles occur repeatedly in transmitted packets and are more robust to wireless channel effects, we wish to leverage its inherent cross-correlation properties. Towards this aim, we generate a feature map using the preambles and show how this enables successful UAV make/model detection, which we define as the task of *classification*. Since the format of the transmitted frame is unknown, we try to coarsely estimate the location of the preamble within the received I/Q samples, partition the entire stream into different slices, and analyze the cross-correlation properties for each slice independently. We note that our objective is not to detect a specific emitter of the same model UAV (what is known as RF fingerprinting), but instead, to detect a specific model (e.g.,

Manuscript received October 1, 2020; revised February 15, 2021; accepted April 22, 2021. Date of publication May 17, 2021; date of current version July 20, 2021. This work was supported by the NSF award CNS 1923789. The review of this article was coordinated by Dr. Yue Gao. *(Corresponding author: Guillem Reus-Muns.)*

The authors are with the Institute for the Wireless Internet of Things, Northeastern University, Boston, MA 02115 USA (e-mail: greusmuns@coe.neu.edu; krc@ece.neu.edu).

Digital Object Identifier 10.1109/TVT.2021.3081049

a DJI M100 versus a DJI Mavic Pro 2). We use deep learning for learning the model ID, specifically, 3D convolutional neural networks (CNN), which have proven to be remarkably successful in a number of image processing tasks [10]–[12], and more recently for RF fingerprinting [13]–[18].

Given that new UAV models are continuously entering the market, it is impractical to assume that a central dataset is always updated with signal collections from these new models. Thus, training a single classifier over the entire dataset poses a challenge. Furthermore, users are not incentivized to provide such data to the central entity, given that the signals transmitted by UAVs may contain audiovisual data or location information in the form of GPS coordinates. As the collected data format is I/Q samples, it is difficult to extract out the coordinates before sharing with other third parties. Thus, users may be more comfortable allowing signal collection by a local, trusted entity, with assurance that the dataset itself will not be ported elsewhere. Hence, we consider a federated learning scenario for training our model [19], where independent neural networks are trained independently at local entities with local data. These trained architectures are then combined in the cloud, while the raw data always remains at the local level [20].

Finally, we consider an extreme case of non-IID (Independent and Identically Distributed) data, where data from a single class, i.e., UAV model, is only available at a very small number of data collection locations. This assumption is relevant in cases where a particular model enters a country-specific market, is less widespread in use due to supply/distribution issues, or is simply a new entrant in the market. We start from the approach in [21], where a small IID set is shared with all users. We train each model locally using a weighted categorical cross-entropy to compensate for the non-IID data. Indeed, we see an accuracy improvement of up to 68% compared to the baseline approach in [21].

Our contributions in context of classification of generalized UAV make/models are summarized as follows:

- We propose a novel pre-processing scheme that generates a cross-correlation-based feature map in order to highlight the preamble properties for the classification task. (Section IV-A). Furthermore, we empirically demonstrate the benefits of using PHY preambles for UAV classification (Sec. III-C).
- We consider a federated learning scenario with non-IID data across users, which combines the data-sharing approach in [21] with a weighted categorical cross-entropy loss. Our experiments show faster convergence time during training and higher accuracy during testing (Sec. V).
- We thoroughly evaluate our approach through experimental data under both similar and totally different channel conditions to showcase its robustness. We comprehensively study the cost-benefits of federated learning during performance evaluation. Finally, we generate a dataset of raw IQ samples from stock UAV transmissions deploying 4 UAV models. To support independent investigations beyond this work, we make this dataset publicly available.

The paper is organized as follows: Sec. II summarizes the relevant related work in the field. Sec. III analyzes the downlink/uplink UAV technology, describes the dataset and motivates the use of PHY preamble data for classification tasks. Sec. IV describes the cross-correlation based approach and tackles the new device detection problem. Sec. V evaluates the approach using federated learning and improves its performance for non-IID data. We conclude the paper in Sec. VI.

II. Related Work

In this section, we review the state-of-the-art in UAV detection/classification, with a focus on passive RF-based methods. Also, we summarize the work in federated learning with non-IID data.

A. Passive RF

Most off-the-shelf UAVs continuously exchange data and control with an RC or other ground-based controllers. We provide experimental evidence and examples of such downlink communication in Sec. III-A. Since these links are bidirectional, it is also possible to similarly leverage uplink RC transmissions. Different UAV classification approaches have been proposed that leverage RC and/or UAV signals.

The authors in [26] propose a system based on three hand-picked features from the received RF signals and use an artificial neural network classifier to detect the presence/absence of UAVs. This is a precursor to the *classification* problem, which we denote as a *detection* problem. Similarly, a UAV detection solution based on Gaussian Mixture Model (GMM) is proposed in [27]. Another approach is to detect the presence of UAVs by passively eavesdropping on the RF communications and examining the physical characteristics of the drone, such as vibrations in its physical frame [24].

UAV manufacturers often customize RF technology for their uplink/downlink signaling, and waveforms may differ in terms of the frame interval or signal strength. Early attempts for UAV classification use such features [28], [29]. Customized headers or preambles containing vendor-specific information can be leveraged to distinguish the make of UAVs at the physical layer or at the MAC layer via the MAC address. Such features are exploited for classification in [23], where the authors extract a Hash Fingerprint from preamble data and develop a classifier using a distance-based support vector data description (SVDD). However, this approach is only tested indoors and shows low performance under Gaussian noise.

UAV classification based on the uplink RC transmissions is proposed in [22], where multiple signals are captured with a spectrum analyzer and processed via wavelet decomposition. In addition, the authors implement and compare the performance of multiple classification algorithms. A similar scenario is considered in [25], this time under the interference of Wi-Fi/Bluetooth signals. However, in realistic scenarios, the RC maybe far away from the actual flight path of the UAV, and thus, capturing the RC signals for analysis might be unfeasible. In addition, multiple UAVs coexisting in the same spectrum may employ FHSS (as seen in the popular DJI family of UAVs), which makes it impossible to isolate signals from different sources. As we explain in Sec. III-A, our scenario involves a UAV selecting a distinct, non-overlapping downlink channel. Hence, we design a downlink-based UAV classifier in this paper.

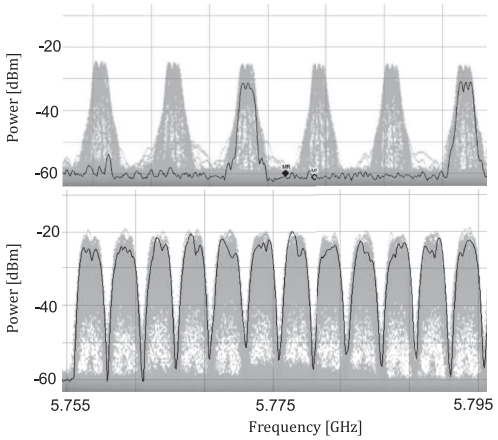


Fig. 2. FHSS pattern captured from a DJI M600 RC (top) and a DJI Mavic Pro 2 (bottom).

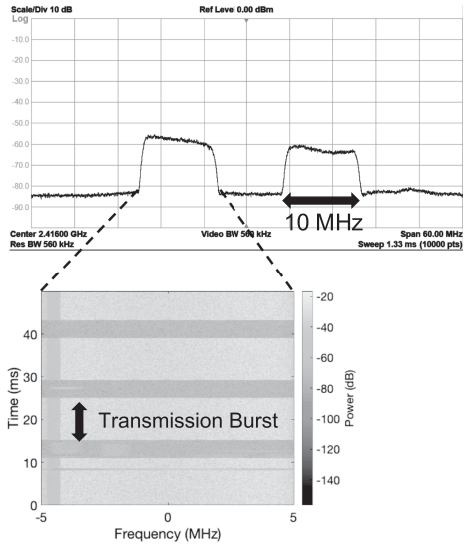


Fig. 3. Downlink transmission of two DJI M600 UAVs (top). Spectrogram visualization of the downlink bursty transmissions (bottom).

B. Federated Learning for Non-IID Data

Federated Learning, first introduced by Google, proposes a privacy-preserving approach were multiple models are trained locally at different devices and combined at the cloud [19], [30], [31].

Instead of transmitting potentially sensitive data to a centralized entity (cloud), privacy is maintained by training independent models at each edge device, which only share the updated model weights.

Many novel solutions have been proposed in the field [32]–[34].

Non-IID data poses an added challenge to federated learning, since assuming IID data across different devices may become unrealistic in many scenarios. The authors in [35] proposed an algorithm that provides faster convergence than federated averaging, even for non-IID data. [36] proposed a method where a generative model is used at each edge device to generate extra data under a non-IID setting. [21] proposed a data-sharing strategy to tackle the non-IID training problem, where a small portion of the locally obtained dataset is shared with all the users in order to reduce the weight divergence. We adopt this approach in this paper to demonstrate highly accurate UAV classification, even with class imbalances across the different data collection entities.

III. Preliminary Experiments

A. COTS UAV Signal Analysis

We first study RF signaling on several UAV models belonging to the DJI family, to analyze the uplink/downlink signal characteristics. We note that these waveforms are all proprietary, so no decoding is possible. We gather measurements in a static environment using the Tektronix RSA507 A spectrum analyzer.

1) Uplink: Several models from the DJI-made COTS UAVs employ FHSS for the RC to UAV uplink. For instance, in Fig. 2 we show the RF spectrum of captured transmissions from two different DJI controllers in the 5.725-5.825 GHz band. The RC uses spread spectrum of 70 MHz bandwidth with hops of 5 MHz (M600) and 2 MHz (Mavic Pro 2). The 2.4 GHz band is also utilized, where we observe a bandwidth of up to 80 MHz and a minimum hopping distance of 2 MHz between subcarriers. These numerical values are of relevance for the discussion conducted in the following subsections.

2) Downlink: UAVs communicate periodically with the RC to report telemetry data, battery level or relay application traffic such as video streaming. Based on the application, such transmissions may vary in bandwidth and transmission periodicity. Consider next the characterization of the downlink of a DJI M600 UAV: These devices transmit in the 2.401GHz-2.481 GHz band, choosing one out of eight 10 MHz channels, based on a proprietary interference analysis algorithm. In Fig. 3, we show a spectrum visualization of two DJI 600 transmissions. In contrast to what happens in the uplink, where different UAVs share the same band, we observe that the downlink transmissions do not overlap. Moreover, we also observe that the UAV accesses the medium at a fixed periodicity ($\approx$50 Hz), as we show in the spectrogram in Fig. 3. Based on our experiments, the DJI M100 and DJI Phantom 3 employ the same downlink technology. The DJI Mavic Pro 2 may also use channels of up to 20 MHz, if required, to support its video streaming capabilities.

B. Data Collection

We create our own dataset with 4 different COTS DJI UAV models, with more than one UAV of the same model in some cases: 2 DJI M100, 2 DJI M600, 1 DJI Phantom 3 and 1 DJI Mavic Pro 2. As mentioned in Sec. II and summarized in Table. I, other UAV classification works used data from FHSS uplink transmissions under the assumption that only a single UAV is present at a time. In contrast, we collect downlink data only, where different UAVs transmit in different bands to avoid mutual interference (Sec. III-A2). Thus, collecting data for every independent UAV is possible under multi-UAV scenarios. However, multiple FHSS transmissions from independent UAVs

TABLE I
SUMMARY OF RELATED WORK ON PASSIVE RF UAV CLASSIFICATION/DETECTION

Work	Data Collection	Static/Hovering	Multi-UAV	Source	Results
[22]	I/Q samples from different UAV RCs. Only indoors scenario is considered.	Static	No	UAV Uplink	97.1% classification accuracy
[23]	I/Q samples from different UAV/WLAN routers	Both	No	UAV Downlink and WLAN router	UAV detection. 86.5% rejection rate for 802.11b.
[24]	I/Q samples from a UAV-mounted SDR	Hovering	No	UAV-mounted SDR	96.7% Accuracy, 95.9% Precision and 97.3% Recall.
[25]	I/Q samples from different UAV RCs under WiFi and Bluetooth interference. Only indoors scenario is considered.	Static	No	UAV Uplink	98.13% accuracy using a kNN classifier.
[26]	RF Data for UAV and non-UAV presence is collected at different distances. Both indoors and outdoors are considered.	Hovering	No	UAV Uplink/Downlink	UAV vs non-UAV classification accuracy of up to ≈97%.
[27]	Data for both RC and UAV RF transmissions in an indoors lab scenario. The authors used an Agilent9404 oscilloscope and a directional antenna to collect data.	Static	No	UAV Uplink/Downlink	97.48% UAV signal detection accuracy.
This paper	UAV downlink stock transmission collected indoors and outdoors	Both	Yes	UAV Downlink	99.97%

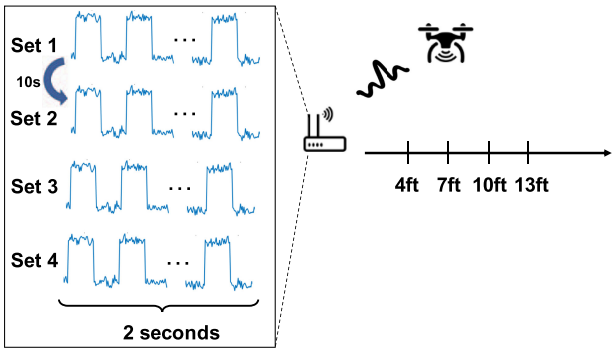


Fig. 4. Data collection overview. The process represented in this figure is repeated for each UAV and distance.

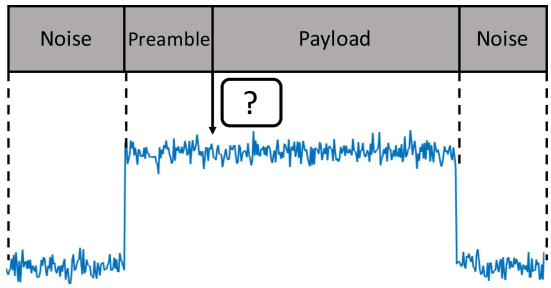


Fig. 5. Received frame structure. The exact location of the preamble is unknown due to the proprietary MAC protocol.

coexist in the same band and only single-UAV scenarios can be considered.

We use an Ettus USRP B210 and MATLAB running on an Ubuntu machine for data collection. We fly each UAV individually at different distances from the receiver (4 ft, 7 ft, 10 ft, 13 ft) and collect multiple streams of IQ samples for each UAV and location. The software defined radio (SDR) is tuned to receive only in the specific channel where the UAV is transmitting, as described in Sec. III-A2. As shown in Fig. 4, we collect 4 sets of 2 seconds of data for each UAV-location pair. Each set is collected with a separation of ∼10 seconds. As presented in Fig. 3 and explained in Sec. III-A2, downlink communication occurs in periodically. Thus, each set contains a number of interleaved short periods of UAV-transmitted frames and noise. This data collection process, represented in Fig. 4, is repeated for each different UAV and distance. In order to capture data under different channel conditions, we collect two different datasets following the previously described methodology. The first dataset is collected in an indoors/lab scenario, where the UAVs are static on the ground. The second dataset is collected outdoors, with the UAVs hovering in the air. These two different datasets test the robustness of our approach under different wireless channel conditions.

C. PHY Preamble for Classification

Without loss of generality, we assume that preambles are added at the start of physical layer (PHY) frames to achieve synchronization and perform channel estimation tasks, amongst others. Since there is no standard globally adopted for all UAVs, most vendors use their own proprietary protocol. Our hypothesis is that these preambles differ sufficiently to enable accurate UAV classification. As shown in Fig. 5, the preambles in a PHY frame are always located at the very start.

Next, we aim to validate the hypothesis that using the PHY preamble increases UAV classification accuracy, as opposed to using any set of contiguous IQ samples randomly selected from the payload. To do so, we empirically compare the classification accuracy of both approaches. The classification accuracy is computed as the quotient between the number of correctly classified inputs and the number of elements in the dataset.

We use the data of 2 different UAV models (2 DJI M100 and 2 DJI M600), i.e., four UAVs in total. To guarantee the deep

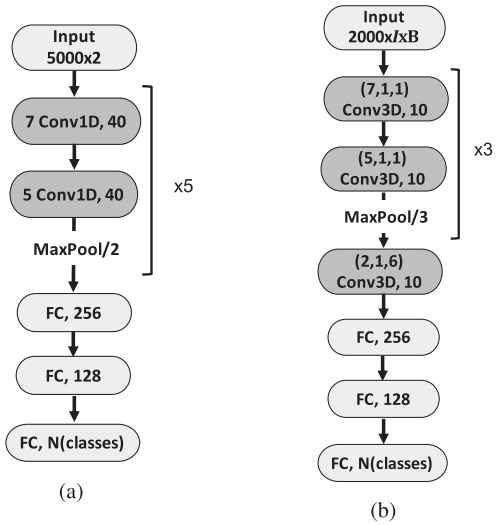


Fig. 6. Baseline CNN (a) and Cross-Correlation CNN (b).

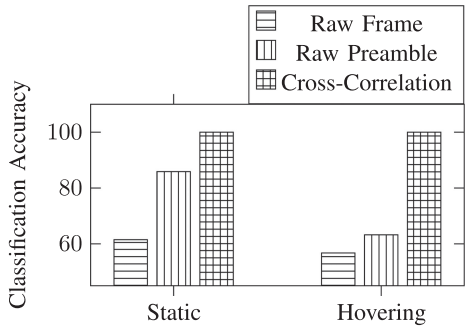


Fig. 7. Classification accuracy for the 2 UAV case.

learning model learns UAV-model-specific instead of device-dependant features, we test the performance of our model with different UAVs used for testing. In particular, one of each DJI M100/M600 are used for training and different DJI M100/M600 are used for testing. In order to study the potential effect of the UAV hovering, we consider two different scenarios where (i) the UAVs are placed at a static location on the ground and (ii) the UAVs are hovering in the air. Given that the PHY frame structure is defined by the proprietary protocol, the exact length and location of the preambles present is unknown. For this work, we subtract the first 5 k IQ samples of every frame. We justify the choice of this value later in Sec. IV-A1.

We train a neural network with the architecture shown in Fig. 6(a). The complex-valued IQ samples are split and fed into the neural network in two different columns, as shown in Fig. 9. This experiment is conducted for the two scenarios mentioned earlier in this subsection. In Fig. 7, we show the obtained accuracy for each studied setup.

We observe that when the UAVs are static, an accuracy of 61.5% and 85.88% is obtained for the cases pertaining to using payload IQ samples and preamble IQ samples, respectively. Moreover, the performance drops considerably under hovering conditions, with the corresponding accuracy values falling to 56.75% and 63.24%.

Thus, we identify that (i) there is meaningful information in the preambles transmitted by the UAVs that enhance the classification accuracy, and (ii) the intrinsic stochastic properties of the wireless channel under UAV hovering conditions dominate the potential discriminating features of the preamble. In the next section, we propose a cross-correlation based approach that exploits the preamble features independent of the wireless channel conditions.

IV. Preamble-Based UAV Classification

The wireless channel changes in sub-second intervals under-ing hovering conditions. This, in turn, lowers the probability of correct UAV classification. Here, we propose a classification method based on the cross-correlation properties intrinsic in the PHY preamble, which is present in every transmitted frame.

A. Cross Correlation Based Classification

We wish to obtain discriminating features that can be generalized for different data collection scenarios, and ultimately, facilitate UAV classification in unknown deployment environments. Since the exact location of each preamble within a frame is unknown, we propose to create a feature map based on the cross-correlation of different slices within the first *N* samples of each frame.

We collect one set of *N* samples for each of the UAVs i, $i = 1, 2, \ldots, I$, referred as p_i, which becomes the reference for computing the cross-correlation (R) for every received frame.

First, we define the cross-correlation as:

$$\mathrm{R}_{\mathrm{xy}}[m] = \mathrm{E}\{x_{n+m} y_n^*\} \tag{1}$$

where $-\infty < n < \infty$, E{} denotes the expected value operator, the asterisk denotes the conjugate and x and y are two random sequences. As we are working with finite length sequences, R can be expressed as:

$$\mathrm{R}_{\mathrm{xy}}[m] = \begin{cases} \sum_{n=0} x_{n+m} y_n^* & m \geq 0 \\ R_{yx}^*[-m] & m < 0 \end{cases} \tag{2}$$

Then, we define the vector $C_{xy}[m]$ as:

$$C_{xy}[m] = \mathrm{R}_{\mathrm{xy}}[m - L], \qquad m = L-1, L, \ldots, 2L-1. \tag{3}$$

where L is the length of each sequence. In this work, we investigate the properties of the cross-correlation between a sequence of captured IQ samples, c_j, and a set of pre-stored sequences $p_i, \forall i$, for classification purposes. However, the location of such discriminating features is unknown, which may be embedded in certain sub-sequences within these preambles.

Without loss of generality, let us define p_i^b as a sub-sequence of p_i with length $L < N$ and starting index b, $0 \leq b < N - L$. Similarly, be also define c_j^b as a sub-sequence of c_i. The sub-indices i and j indicate the UAV model or class:

$$i, j = \{M100, M600, Phantom3, Mavic2Pro\}$$

1) Sequence Length L: We investigate $C[m]$ for different p_i^b and c_i, in order to set the sequence length N that will be subtracted from the beginning of each received frame. In particular, we define the peak-to-average ratio for the cross-correlation

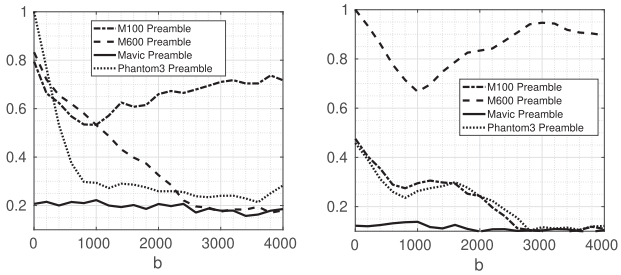


Fig. 8. $\bar{\gamma}_{p_i^b c_j}[b]$ with j=*M100* (a) and j=*M600* (b). (a) $j = M100$. (b) $j = M600$.

between two sequences as γ:

$$\gamma_{p_i^b c_j}[b] = \frac{\max\{C_{p_i^b c_j}[m]\}}{\mu_{p_i^b c_j}} \tag{4}$$

$$\mu_{p_i^b c_j} = \frac{1}{(2L-1)} \sum_m C_{p_i^b c_j}[m] \tag{5}$$

$$\bar{\gamma}_{p_i^b c_j}[b] = \frac{\gamma_{p_i^b c_j}[b]}{\gamma_0} \tag{6}$$

$$\gamma_0 = \max(\gamma_{p_i^b c_j}) \quad \forall i \tag{7}$$

where $\mu_{p_i^b c_j}$ is the average cross-correlation between p_i^b and c_j ($C_{p_i^b c_j}[m]$), $\gamma_{p_i^b c_j}[b]$ is the peak-to-average cross-correlation ratio between the sequences p_i^b and c_j. For visualization purposes, $\gamma_{p_i^b c_j}[b]$ is normalized ($\bar{\gamma}_{p_i^b c_j}[b]$) to values between 0 and 1 after applying the normalization factor γ_0. For ease of notation, we express $\bar{\gamma}_{p_i^b c_j}$ as $\bar{\gamma}_{ij}$, where i and j indicate the UAV model for p and c, respectively. Then, $\bar{\gamma}_{ij}$ represents the normalized peak-to-average ratio for the cross-correlation between p_i^b and c_j. Intuitively, $\bar{\gamma}$ is used as a relative measure to quantify how big the correlation peak is in comparison to the whole cross-correlation sequence. While the generated cross-correlation feature space does not rely on simple peak detection, we define $\bar{\gamma}_{ij}$ to quantitatively measure whether the two sequences have meaningful cross-correlation properties. $\bar{\gamma}_{ij}[b]$ is a sequence of $\bar{\gamma}_{ij}$ values for different starting indices b for the sequence p_i^b. Thus, we will set N to the smallest value such that meaningful cross-correlation properties are captured, which we express as:

$$N = b_l + L \ni \bar{\gamma}_{ij}[b_l] \simeq \bar{\gamma}_{ij}[b_h], \quad \forall b_l \ll b_h, \quad \forall i, j \tag{8}$$

where b_l is the smallest b value such that $\bar{\gamma}_{ij}$ stays constant even if b increases.

We present in Fig. 8 the evolution of $\bar{\gamma}_{ij}[b]$ over b for sequences captured for different UAV models (M100, M600). The length of the sequence p_i^b is set to $L = 2000$. We observe how $\bar{\gamma}_{ij}[b]$ varies distinctly with b for different i, j pairs. The curves present multiple inflection points due to the different cross-correlation properties of the sub-sequences present within the analyzed sequences. Thus, each peak corresponds to two matching sub-sequences, in contrast to low $\bar{\gamma}_{p_i^b c_j}[b]$ values, which we interpret as two sub-sequences with low cross-correlation properties. However, we observe that for $b >$3000, $\bar{\gamma}_{ij}[b]$ stabilizes and therefore, no meaningful cross-correlation information is available. Since the condition in (8) is fulfilled for $b >$3000, we set N to 5000 for the rest of the paper.

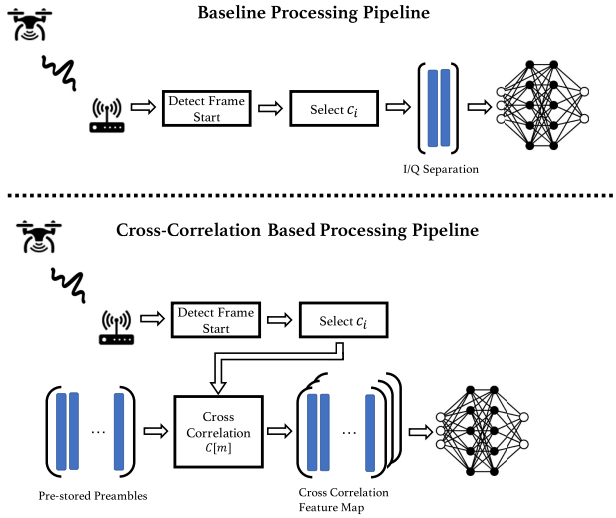


Fig. 9. Processing pipeline for baseline and cross-correlation based approaches.

2) Generating Feature Maps: Next, we empirically investigate the cross-correlation properties for different p_i^b-c_i pairs in order to motivate its importance for our classification task. In Fig. 10, we plot different cases of $\mathrm{C_{p_i^b c_j^b}}$, with $i = \{$*M100*, *M600*, *Phantom3*, *Mavic2Pro*$\}$ and $j = \{$*Phantom3* (10(a)–10(d)), *M100* (10(e)–10(h))$\}$. As expected, we observe how $C[m]$ shows different shapes based on what sub-frames are used. In some cases, certain cross-correlation contours can only be observed for specific p_i^b and c_i^b combinations.

For instance, $\mathrm{C_{p_i^1 c_{Phantom3}^1}}$ shows a very similar pattern, with peaks around $m = 5000$, for all UAVs expect the *Mavic2Pro*. In contrast, $\mathrm{C_{p_i^{3000} c_{Phantom3}^{3000}}}$ with $i =$ *M600* (Fig. 10(b)) is clearly distinguishable from the $i = \{M100, Phantom3\}$ cases, with clear correlation peaks for $m = 5000$ (Fig. 10(a) and Fig. 10(c), respectively). A similar analysis can be done for $\mathrm{C_{p_i^1 c_{M100}^1}}$, where certain features seem to be only present for specific cross-correlation sub-sequence pairs.

However, finding the specific value pairs for i and b that achieve the best classification performance is not straight forward. In addition, different UAVs might require different values for these variables. Thus, we propose a general framework that aims to generate a cross-correlation-based feature map that will highlight all the potential classification features, regardless of prior knowledge of the UAVs under study.

In Fig. 11, we show how we generate such feature maps. For a certain incoming sequence c_j, we generate a number of sub-sequences c_j^b with length L and a sliding window ω. Similarly, a previously stored sequence p_i is chopped into sub-sequences p_i^b. Then, each sub-sequence c_i^b is cross-correlated with every sub-sequence p_1^b following Eq. (3), obtaining $C_{p_1^b c_j^b}[m]$. We concatenate all the computed $C[m]$ into a matrix with dimensions $L \times B$, with $B = \frac{N-L}{\omega}$. The same process is repeated for all

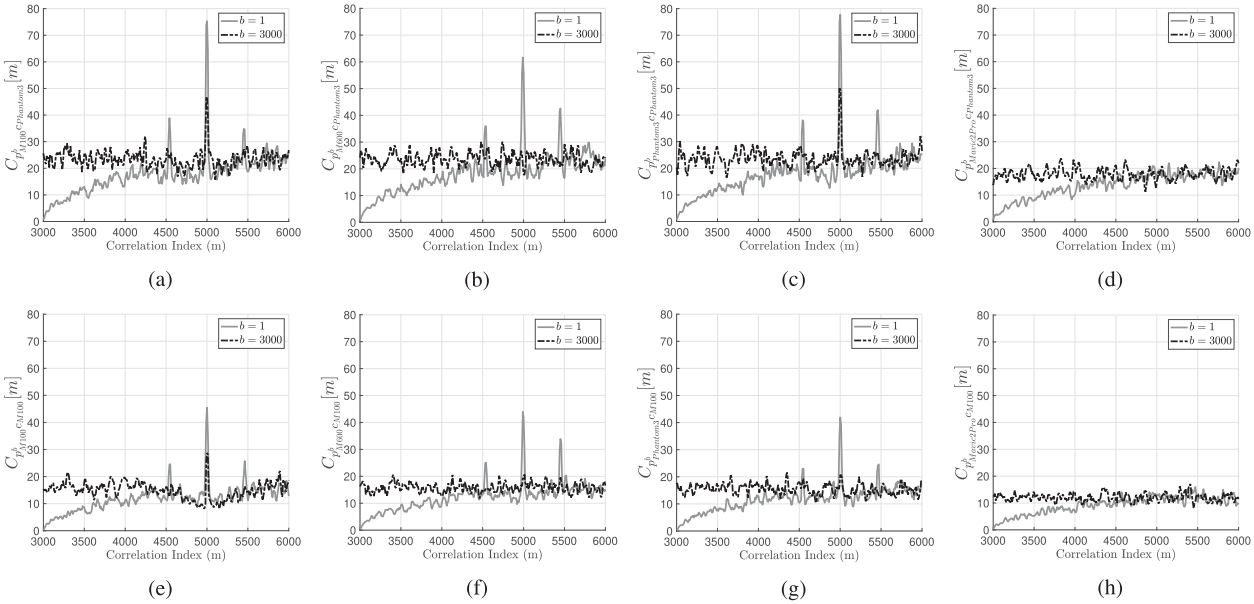


Fig. 10. $C_{p_i^b c_j^b}$ with $j =$ *Phantom3* (a)–(d) and $j =$ *M100* (e)–(h). Cross-correlation across different preambles show different properties, whose features are leveraged for classification purposes.

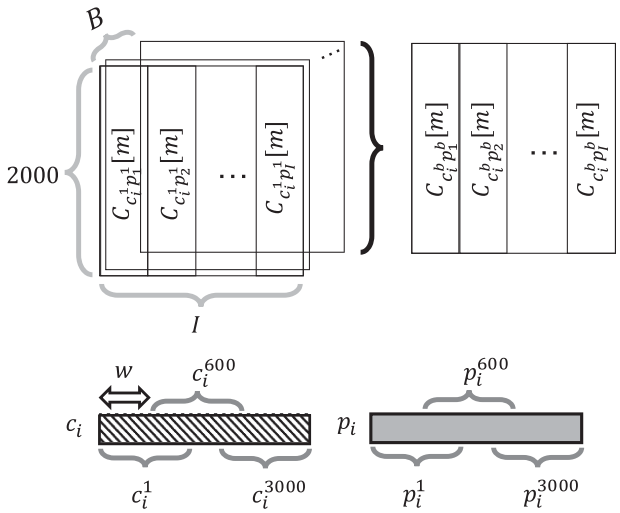


Fig. 11. Cross-correlation feature map.

i values, ultimately generating a 3D matrix of dimension $L \times B \times I$.

B. Evaluation

We train a neural network with the architecture presented in Fig. 6(b) with generated feature maps represented as a 3D matrix. We design the architecture in Fig. 6(b) using 3D convolutional layers under the premise that different $C[m]$ do not share any spatial features. Thus, the filters of size (F,1,1) search for relevant features across each independent cross-correlation. It is not until the last convolutional layer where the size of the filters allows the different computed features for each UAV to be combined, before the fully-connected layers. We use PyTorch for our implementation and train the model with Adam optimizer using a learning rate of 0.001. The hyperparameters of the model, such as filter size, number of filters and the depth of the model are chosen carefully through cross validation.

TABLE II
CLASSIFICATION ACCURACY RESULTS

Training	Testing	Classification Accuracy
Static	Static	100%
Static	Hovering	98.2%
Hovering	Static	100%
Hovering	Hovering	99.97%

In Fig. 9 we show the processing pipeline. The receiver device collects the raw IQ samples used to compute the feature map, as shown in Fig. 11, which we use as the input to the neural network. As part of the comparison study in Sec. III-C, we train a classifier on the same dataset with only 2 UAVs and we show the results obtained in Fig. 7. Our method is able to achieve near-perfect accuracy for both the static (indoor) and the hovering (outdoor) scenarios.

In order to study the robustness of our approach, we extend the evaluation to 4 different UAV models with the two datasets described in Sec. III-B. We partition them into train, validation and testing datasets, following an 80/10/10 distribution. Furthermore, we empirically evaluate the robustness of our method under unseen channel conditions, we train two models independently with static and hovering data. Next, we test each model with data collected under both seen and unseen scenarios. Thus, we evaluate the performance of the static-data-trained model with both static and hovering data, separately. Similarly, we use static and hovering data to test the model trained with hovering data only. We present the results in Table II. We observe that the worst performing case (static training, hovering testing),

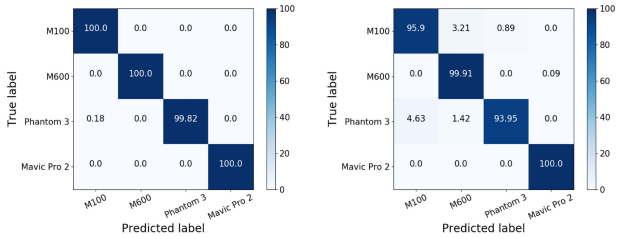


Fig. 12. Confusion matrix after training and testing on hovering data (a), training on static data and testing on hovering data (b).

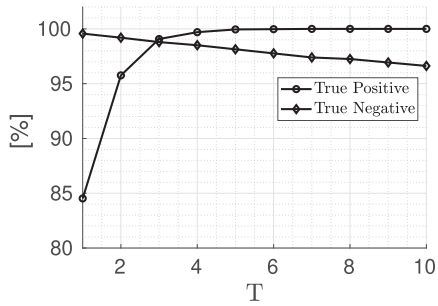


Fig. 13. Anomaly detection rate over T.

achieves an accuracy of 98.2%, whereas we see near-perfect accuracy for other scenarios. These results reveal that a model trained with data collected in a controlled environment can be subsequently deployed in more realistic scenarios. The channel conditions and the noise level in these unseen environments are likely completely different. We observe near-perfect accuracy for most of the training-testing combinations, except for the static-hovering case, where we expect a small accuracy reduction due to the unseen hovering conditions during training.

In Fig. 12 we show the confusion matrices for two different training-testing pairs, which supports our claim of channel-independent and generalized UAV classification.

C. New UAV Detection

If a trained network is tested with data from a new UAV, it inevitably classifies the input unit as one of the known classes. To discriminate this prediction from a "correct" old device prediction, we consider the Euclidean distance to the predictions for set of known devices in order to label each unknown test unit as "old" or "new". This problem is commonly referred in the literature as *novelty detection*.

1) Overview of the Approach: Distance-based metrics for novelty detection have been widely explored in the research community [37]–[41], where a data point is considered to be from an unknown class if it is far from its neighbours. While multiple distance metrics have been explored, Euclidean distance is the most popular choice for univariate and multivariate continuous attributes. To use the distance-based approach for new UAV detection, the softmax scores s for a set with known labels (old) are computed and kept as a reference. Assume the reference set has K samples each indexed with k = 1,.., K-1. Then, the Euclidean distance to the closest point within the reference set (d_t) is computed for each incoming datapoint with softmax scores s_t, which can be expressed as:

$$d_t = \min\left(\sqrt{\sum_{k=0}^{k-1}(s_k - s_t)^2}\right) \tag{9}$$

Additionally, we average the metric d_t from multiple independent predictions ($\bar{d_T}$) to take a single novelty detection decision:

$$d_T = \frac{1}{T}\sum_{t=1}^{T} d_t \tag{10}$$

where T is the number of averaged samples. Notice that that the transmission rate for UAVs is typically around 50 Hz (Sec. III-A2). Thus, collecting enough data for computing multiple predictions is feasible in fractions of a second. Next, the computed distance metric is used to discriminate whether a certain signal belongs to the known set of UAVs or is a new unknown device. The threshold is selected though exhaustive exploration for multiple values.

$$y_t = \begin{cases} new\ \bar{d_T} > thresh \\ old\ otherwise \end{cases} \tag{11}$$

2) Evaluation: We test the "new" UAV detection approach on the same dataset described in Sec. III-B. We use the trained model in Sec. IV-B and collect data for a fifth UAV (DJI Air 2), which will be used as the new device.

In this case, the new UAV detection accuracy is defined as the number of correctly detected "new" devices ($y_t = new$), divided by the size of the dataset. Without loss of generality, we select the *DJI Air 2* as the new device, and the other four as old devices. Additionally, we evaluate the performance with the number of averaged samples T in Fig. 13. By taking a single sample (T = 1), we obtain a new device detection accuracy (true positive) of 84.52% with less than 0.5% of old devices incorrectly classified as new (false positive). This shows that $> 99.5\%$ of the samples are correctly classified as old UAVs. Furthermore, we achieve a new device detection accuracy of 99.7% and a true negative rate of 98.51% with T = 4.

V. Impact of Federated Learning

A. Scenario Description

The work described in the previous sections proposes a solution to the UAV classification problem under the assumption that the complete dataset is available at a centralized entity during training, and could only be deployed across different locations/devices during the testing phase. However, sharing data with potential personal or private information poses multiple privacy concerns.

To avoid sharing raw IQ samples between data collection devices and a central entity, we next consider a setup where the raw data is kept at each data collection receiver. Moreover, each receiver trains a UAV classification model locally and only will share the weights of this model with the centralized entity.

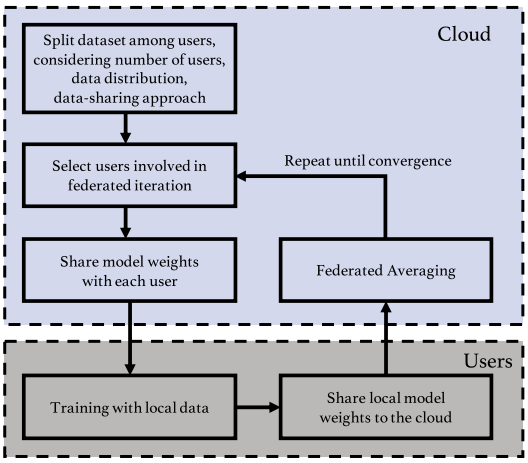


Fig. 14. Federated learning flowchart.

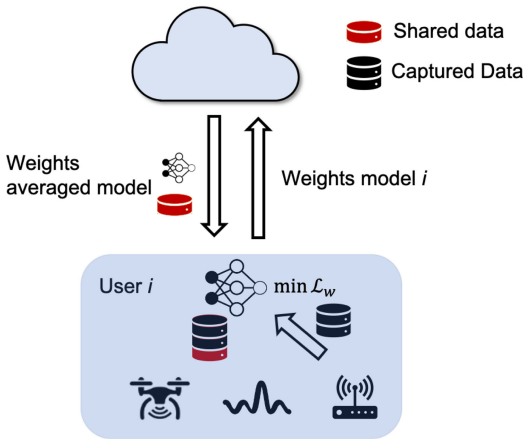


Fig. 15. Federated Learning with Data-Sharing and Weighted Loss ($\mathcal{L}_w$). During training, each user computes the class representation and weights the loss accordingly.

Notice that this ensures that the data itself will not be shared outside the data collection device. This system architecture is commonly referred as federated learning. Typically, federated learning involves a *cloud* entity and multiple *users*, and its operation can be described into three steps:

First, the central entity or cloud shares a copy of the model with the latest weight updates with all the users. We interchangeably use the term 'user' with a RF data collection device with computation resources to train a neural network in the following part of the paper. Second, each user updates the received model independently using its own data and shares it back to the central entity.

Finally, the different models trained by each user are combined into a process commonly referred as *federated averaging* [31].

The above three steps are collectively referred to as *federated iteration*. We summarize the federated learning process in Fig. 14. Furthermore, we define the following parameters:

- *Local Epochs:* Number of training epochs computed before the model weights are shared back with the central entity for model averaging.
- *Dataset Distribution:* defines whether the user-stored data is IID or non-IID.
- *Number of Users:* Total number of users.
- *Percentage of Users:* Percentage of the total number of users active per federated iteration.

In the rest of this section, we describe the challenges of working with a non-IID dataset for federated learning. We describe and evaluate our approach to improve accuracy and speed of convergence.

B. UAV Classification With Non-IID Data

The class distribution across devices is a critical factor that impacts the performance of a model or its convergence time during training. For instance, the authors in [21] analyze the weight divergence of a trained model under different non-IID conditions.

We consider a scenario where a certain user/device is able to collect data from a single UAV model and does not have full access to the data collected by other devices. Thus, this represents the most extreme non-IID case, since every locally stored dataset at each device only contains data from a single class. As summarized in Sec. II-B, different approaches have been proposed in the literature to facilitate training under non-IID conditions. Here, similarly to the work in [21], we consider that the centralized entity shares a small part of the dataset across all different users in order to we relax the non-IID constraint and facilitate the learning.

Then, we combine the data-sharing strategy with a weighted loss based on the uneven class distribution [42], [43], which scales the loss of each training unit according to the class distribution. We note that while both techniques have been proposed previously by the machine learning community, they have never been jointly investigated in a federated learning setting. We summarize our overall approach in Fig. 15.

We choose the categorical cross-entropy as the loss function, which has been widely adopted by the research community and is a standard choice for multi-label classification tasks [44]–[46]. Its weighted version ($\mathcal{L}_w$) can be expressed as:

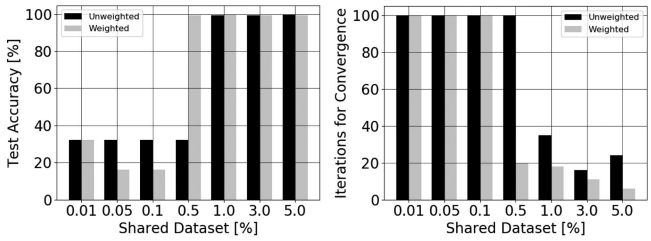
$$\mathcal{L}_w = -\sum_{i=1}^{I} \bar{w_i} y_i \log \hat{y_i} \tag{12}$$

where I is the number of classes, i is the class index, y and $\hat{y}$ are the one-hot vector representation of the true and predicted class, respectively. One-hot encoding is a vector representation of the class label i, whose all I elements take value 0 except the ith one, that takes value 1. $\bar{w_i}$ the normalized weight for class i, which is computed as:

$$\bar{w_i} = \frac{w_i}{\sum_{i=1}^{I} w_i} \tag{13}$$

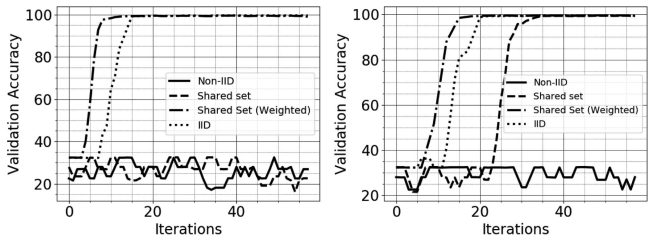
$$w_i = \frac{D_i}{\sum_{i=1}^{I} D_i} \tag{14}$$

where D_i represents the number of elements for class i. Next, we show that by employing a weighted categorical cross entropy



(a) (b)

Fig. 16. Test accuracy (a) and convergence iterations (b) for different shared dataset percentages of the overall data. The weighted categorical cross-entropy approach shows superior performance than simple unweighted training. These experiments were run for 40 federated users, 10% users and 2 local epochs per federated iteration.



(a) (b)

Fig. 17. Validation Accuracy vs Iterations for 20 users (a) and 40 users (b).

loss function with the weights defined as in Eq. 13, we improve both performance and speed of convergence under non-IID settings.

C. Evaluation

We adopt the data-sharing approach in [21] to relax the non-IID constraint. Notice that this approach does not conflict with the privacy issue that federated learning tries to tackle, since only a small percentage of the overall dataset is shared. For instance, data collected at a disclosed location can be used as the shared set, while the rest of the collected data would remain private at each independent user/device.

Next, we analyze how the percentage of total shared data and the weighted categorical cross entropy impacts the system performance. While we aim to optimize the speed of convergence and the classification accuracy, the size of the shared dataset introduces an overhead in memory usage and network bandwidth that should be minimized.

For evaluation purposes, we implement a federated learning framework using Python and PyTorch. The overall dataset is divided among all the users, following a certain data distribution (IID, Non-IID, etc). In every federated iteration, a certain percentage of users (*Percentage of Users*) update the weights of their models for a number of epochs, defined as *Local Epochs*. After every federated iteration, the users that recently updated their models share their weights back with the central entity for federated averaging. The users involved in each federated iteration are chosen randomly. If the data-sharing approach is used, a small percentage of the dataset is shared among all users.

In Fig. 16, we observe that by sharing a 0.5% of the overall dataset, in combination with the weighted loss approach, we achieve a testing accuracy of 99.28%. In contrast, training does not converge for the same data-sharing strategy if the weighting term w in (12) is not applied (unweighted). Thus, the combination of both techniques provides a performance improvement over the standalone method in [21]. Similarly, we also observe an improvement in convergence time, which we analyze in terms of the required federated iterations for convergence. As shown in Fig. 16(b), the number of required iterations decreases when the shared dataset size increases. Additionally, the weighted loss approach improves the convergence time for all the cases, for as much as 51.4% with 1% of shared data.

In Fig. 17, we analyze the validation accuracy per iteration under different data distributions (IID, non-IID) and training strategies (shared set, weighted loss) for 20 and 40 users (Fig. 17(a)–17(b), respectively). We observe how the weighted loss approach shows the best speed of convergence, even better than the IID case. This is expected, given that the IID dataset is split by randomly sampling from it, which causes some classes to be more represented than others at certain devices, increasing the number of iterations needed for convergence. Also as expected, the non-IID case does not converge, since the weight divergence in each independent model makes it impossible to obtain good performance through the classic federated averaging. We observe a disparity in performance on the data-sharing case for different number of users. With 40 users, the same accuracy as the IID with the weighted case is achieved, although it takes a longer number of iterations. In contrast, when only 20 users are involved in training, the accuracy does not exceed $\approx$32%. This is justified, given that the local share of data is bigger for a smaller number of users, since the data is split evenly among all the users. Then, for the non-IID case with 40 users, the shared data size is proportionally smaller than the single-class locally-stored data, which is not enough to compensate for the class bias. For instance, if 1% of data is shared across users and there are 20 users in total, each device will store 4.95% of the total data. Considering that this will be only data from the same UAV and that the commonly shared dataset will also contain data from that single UAV, that class will be $\approx$21 times more represented than the other ones. Such an outcome will impact the neural network with a severe bias towards that particular class. By doubling the number of users, this ratio is halved, which facilitates training under such severe non-IID conditions. Similarly, increasing the shared dataset size is expected to have an equivalent effect.

In summary, we achieve an accuracy of 99.28% and 99.51% on the UAV classification task while sharing as little as 0.5% and 1% of the overall data among all the devices, respectively.

VI. Conclusion

In this paper, we addressed the UAV classification problem using proprietary RF waveforms. We proposed a method that leverages from the cross-correlation properties intrinsic in the PHY preambles and we empirically demonstrate its superior performance over other approaches that work directly with raw IQ samples. Additionally, we show its resiliency to varying

channel conditions, showcasing the *train once deploy anywhere* paradigm. We achieve an overall accuracy of 99.97% when training and testing with hovering UAV data and 98.2% when the model is trained from static UAV transmissions instead. Moreover, we consider a federated learning scenario where the dataset is collected at different locations/devices and cannot be shared due to privacy reasons. Thus, the whole dataset is not accessible at a centralized entity. Furthermore, we assume the class representation at each device to be unbalanced (non-IID) and we propose a weighted categorical cross-entropy approach to counteract the class imbalances. We thoroughly evaluate its performance and show how it improves over the unweighted data sharing strategy. We obtain an accuracy as high as 99.51%, with a marginal decrease in performance compared to the centralized training approach. In summary, our UAV classification approach mitigates security and safety concerns, resulting in enhanced protection with privacy via our federated learning training approach.

REFERENCES

[1] R. Levick, "Drone industry just beginning to take off," *Forbes Mag.*, vol. 15, 2018, https://www.forbes.com/sites/richardlevick/2018/05/15/drone-industry-just-beginning-to-take-off/?sh=1144655372bc.

[2] "Global drone service market report 2019," 2020. [Online]. Available: https://markets.businessinsider.com/news/stocks/global-drone-service-market-report-2019-market-is-expected-to-grow-from-usd-4-4-billion-in-2018-to-usd-63-6-billion-by-2025-at-a-cagr-of-55-9-1028147695

[3] V. Marojevic, I. Guvenc, R. Dutta, M. L. Sichitiu, and B. A. Floyd, "Advanced wireless for unmanned aerial systems: 5 g standardization, research challenges, and aerpaw architecture," *IEEE Veh. Technol. Mag.*, vol. 15, no. 2, pp. 22–30, Jun. 2020.

[4] "Privacy concerns raised by drone," Accessed: Nov. 15, 2017. [Online]. Available: https://www.dronethusiast.com/drone-laws/

[5] "Use of weaponized drones in terrorism," Accessed: Nov. 15, 2017. [Online]. Available: https://www.washingtonpost.com/world/national-security/use-of-weaponized-drones-by-isis-spurs-terrorism-fears/2017/02/21/9d83d51e-f382-11e6-8d7-263470bf0401story.html?utm term=.8eb8a9251214

[6] A. Solodov, A. Williams, S. Al Hanaei, and B. Goddard, "Analyzing the threat of unmanned aerial vehicles (UAV) to nuclear facilities," *Secur. J.*, vol. 31, no. 1, pp. 305–324, 2018.

[7] B. Hubbard, P. Karasz, and S. Reed, "Two major saudi oil installations hit by drone strike, and US blames Iran," *The New York Times* Sep. 2019, pp. 1–4, Accessed: Sep. 15, 2019. [Online]. Available: https://www. nytimes. com/2019/09/14/world/middleeast/saudi-arabia-refineries-drone-attack.html

[8] M. Z. Anwar, Z. Kaleem, and A. Jamalipour, "Machine learning inspired sound-based amateur drone detection for public safety applications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2526–2534, Mar. 2019.

[9] Z. Kaleem and M. H. Rehmani, "Amateur drone monitoring: State-of-the-art architectures, key enabling technologies, and future research directions," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 150–159, Apr. 2018.

[10] D. Maturana and S. Scherer, "Voxnet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 922–928.

[11] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, 2017, Art. no. 67, Multidisciplinary Digital Publishing Institute.

[12] J. Huang and S. You, "Point cloud labeling using 3D convolutional neural network," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 2670–2675.

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

[14] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "Oracle: Optimized radio classification through convolutional neural networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2019, pp. 370–378.

[15] F. Restuccia *et al.*, "DeepRadioID: Real-time channel-resilient optimization of deep learning-based radio fingerprinting algorithms," in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2019, pp. 51–60.

[16] A. Gritsenko, Z. Wang, T. Jian, J. Dy, K. Chowdhury, and S. Ioannidis, "Finding a 'new' needle in the haystack: Unseen radio detection in large populations using deep learning," in *Proc. IEEE Int. Symp. Dynamic Spectr. Access Netw.*, 2019, pp. 1–10.

[17] T. Jian *et al.*, "Deep learning for RF fingerprinting: A massive experimental study," *IEEE Internet Things Mag.*, vol. 3, no. 1, pp. 50–57, Mar. 2020.

[18] N. Soltani, G. Reus-Muns, B. Salehihikouei, J. Dy, S. Ioannidis, and K. Chowdhury, "RF fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15518–15531, Dec. 2020.

[19] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

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

[21] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[22] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-uav detection and classification from rf fingerprints using machine learning techniques," in *Proc. IEEE Aerosp. Conf.*, 2019, pp. 1–13.

[23] Z. Shi, M. Huang, C. Zhao, L. Huang, X. Du, and Y. Zhao, "Detection of lSSUAV using hash fingerprint based SVDD," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–5.

[24] P. Nguyen *et al.*, "Matthan: Drone presence detection by identifying physical signatures in the drone's RF communication," in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl., Serv.*, 2017, pp. 211–224.

[25] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Detection and classification of uavs using RF fingerprints in the presence of wi-fi and bluetooth interference," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 60–76, 2019.

[26] H. Zhang, C. Cao, L. Xu, and T. A. Gulliver, "A UAV detection algorithm based on an artificial neural network," *IEEE Access*, vol. 6, pp. 24720–24728, 2018.

[27] C. Zhao, M. Shi, Z. Cai, and C. Chen, "Detection of unmanned aerial vehicle signal based on Gaussian mixture model," in *Proc. IEEE 12th Int. Conf. Comput. Sci. Educ.*, 2017, pp. 289–293.

[28] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective RF-based detection of drones," in *Proc. 2nd Workshop Micro Aerial Veh. Netw., Syst., Appl. Civilian Use*, 2016, pp. 17–22.

[29] H. Li, G. Johnson, M. Jennings, and Y. Dong, "Drone profiling through wireless fingerprinting," in *Proc. IEEE 7th Annu. Int. Conf. CYBER Technol. Automat., Control, Intell. Syst.*, 2017, pp. 858–863.

[30] J. Konečný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," 2015, *arXiv:1511.03575*.

[31] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016, *arXiv:1602.05629*.

[32] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[33] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.

[34] K. Bonawitz *et al.* "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*.

[35] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-IID data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[36] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data," 2018, *arXiv:1811.11479*.

[37] V. Chandola, S. Boriah, and V. Kumar, "Understanding categorical similarity measures for outlier detection," Univ. Minnesota, Minneapolis, MN, Rep. TR 08–008, 2008.

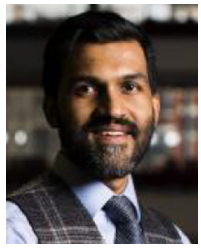
[38] V. Hautamaki, I. Karkkainen, and P. Franti, "Outlier detection using k-nearest neighbour graph," in *Proc. 17th Int. Conf. Pattern Recognit.*, 2004, vol. 3, pp. 430–433.

[39] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *Proc. SIAM Int. Conf. Data Mining. SIAM*, 2008, pp. 243–254.

[40] D. G. Stork, R. O. Duda, P. E. Hart, and D. Stork, *Pattern Classification*. Wiley-Interscience Publication, Hoboken, NJ, USA, 2001.

[41] J. Zhang and H. Wang, "Detecting outlying subspaces for high-dimensional data: The new task, algorithms, and performance," *Knowl. Inf. Syst.*, vol. 10, no. 3, pp. 333–355, 2006.
[42] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, p. 27, 2019.
[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[44] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, vol. 1, no. 2.
[45] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[46] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," 2017, *arXiv:1702.05659*.

Guillem Reus-Muns received the B.Sc. degree in telecommunications engineering from the Polytechnic University of Catalonia, Barcelona, Spain, and the M.Sc. degree in electrical and computer engineering in 2018 from Northeastern University, Boston, MA, USA, where he is currently working toward the Ph.D. degree. His current interests include mobile communications, networked robotics, machine learning for wireless communications, and spectrum access.



Kaushik Roy Chowdhury (Senior Member, IEEE) received the M.S. degree from the University of Cincinnati, Cincinnati, OH, USA, in 2006 and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA, in 2009. From 2009 to 2015, he was an Assistant Professor with Northeastern University, Boston, MA, USA, where he is currently an Associate Professor with Electrical and Computer Engineering Department. His current research interests include deep learning for wireless sensing and spectrum access, networked robotics, wireless RF energy harvesting/transfer, and IoT applications for intra/on-body communication. He was the winner of the Presidential Early Career Award for Sci- entists and Engineers (PECASE) in 2017, ONR Director of Research Early Career Award in 2016, and the NSF CAREER Award in 2015.