Weighted Average Consensus Algorithms in Distributed and Federated Learning

Bernardo Camajori Tedeschini, *Member, IEEE*, Stefano Savazzi, *Member, IEEE*, and Monica Nicoli, *Senior Member, IEEE*

***Abstract*—The exponential growth of the Internet of Things (IoT) has created an essential demand for Distributed Machine Learning (DML) systems. In this context, Federated Learning (FL) allows IoT devices to collaboratively train models while maintaining data ownership and privacy. Despite the evident advantages, FL faces practical challenges such as client selection and adaptation to heterogeneous data distributions. Recently, consensus-driven algorithms have been proposed to enable efficient and scalable FL without a central coordinating entity. Weighted Average Consensus (WAC) tools, primarily used in distributed signal processing, fail to address FL-specific challenges. The paper proposes a new family of server-less FL algorithms optimized to exploit WAC techniques. In particular, we propose an evolution of the centralized Federated Adaptive Weighting (FedAdp) method and present three distinct WAC schemes specifically designed for non-Independent and Identical Distributed (IID) data. Each scheme has a unique aggregation part that optimizes the weights of the clients' local models. The performances are evaluated in a real-world IoT system, analyzing their convergence properties in the context of heterogeneous client populations. Results show that the proposed algorithms outperform vanilla consensus FL up to 56% of accuracy and they are resilient to both label and sample data skewness.**

***Index Terms*—FL over networks, weighted average consensus, non-independent and identical distributed, edge computing.**

I. Introduction

THE rapid growth of the Internet of Things (IoT) and the diffusion of devices with increasing computational capabilities created a significant need for Distributed Machine Learning (DML) [1]. Federated Learning (FL) is a DML approach that allows devices to collaboratively train models while keeping their data local, thus preserving privacy and security [2], [3], [4], [5]. In vanilla FL framework, a central entity, i.e., Parameter Server (PS), coordinates the learning process among the participating devices, or clients, by aggregating their locally computed model updates. This technique is particularly relevant in 5G/6G networks [6], [7], [8], where devices can efficiently communicate and share data, enabling a wide range of applications such as autonomous driving, smart cities, and advanced healthcare services [9], [10], [11], [12].

While FL offers undoubted advantages, it also faces several challenges that must be addressed to ensure robustness, efficiency, and security across various application domains [7], [13], [14]. These challenges include client-selection [15], [16], [17], [18], [19], which involves determining the optimal set of clients to participate in the training process; energy consumption [20], [21], [22], [23], as IoT devices often have limited battery life, calling for the design of energy-efficient FL algorithms; and incentive mechanisms [24], which foster cooperation among clients by rewarding them for their contributions. Additionally, Over-The-Air (OTA) computations and communications [25], [26], [27] require efficient techniques for data transmission and model aggregation while minimizing latency and bandwidth.

Alongside the aforementioned issues, security remains a major concern in FL [28], [29], [30], as the system is vulnerable to various attacks such as data poisoning and model manipulation. Another aspect to consider is online learning [31], [32], [33], which necessitates algorithms capable of adapting to dynamic and evolving data distributions. Furthermore, achieving fast-convergence [34], [35] and time-efficient asynchronous FL [36], [37], [38], [39] is crucial for practical implementation, particularly in scenarios with limited connectivity or highly dynamic environments. Addressing these various aspects is essential for unlocking the full potential of FL and ensuring the successful deployment of FL-based systems across a wide range of applications.

A. Related Works

One of the main concerns in FL is the non-Independent and Identical Distributed (IID) data distribution among clients [40]. Indeed, in case of data heterogeneity, traditional methods like Federated Averaging (FedAvg), which rely on local Stochastic Gradient Descent (SGD), may struggle to achieve convergence when the participating devices execute an excessive number of local updates or have skewed data distributions. To overcome this limitation, adaptive learning rate strategies [41], [42], [43] and smart clustering or pooling techniques [44], [45], [46] have been introduced in the last years.

Of particular interest are the works of FedProx [47] and its variants [35], [48] which employ an inexact proximal point update for local optimization, i.e., penalizing the deviation of the local model from the PS global one. While FedProx is

Received 12 October 2024; accepted 6 January 2025. Date of publication 16 January 2025; date of current version 24 February 2025. This work was supported by a Ph.D. grant from the ministry of the Italian government Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR). Recommended for acceptance by Dr. Hoang Dinh. *(Corresponding author: Bernardo Camajori Tedeschini.)*

Bernardo Camajori Tedeschini and Monica Nicoli are with the Politecnico di Milano, 20133 Milan, Italy (e-mail: bernardo.camajori@polimi.it; monica.nicoli@polimi.it).

Stefano Savazzi is with the Consiglio Nazionale delle Ricerche (CNR), 20133 Milan, Italy (e-mail: stefano.savazzi@cnr.it).

Digital Object Identifier 10.1109/TNSE.2025.3528982

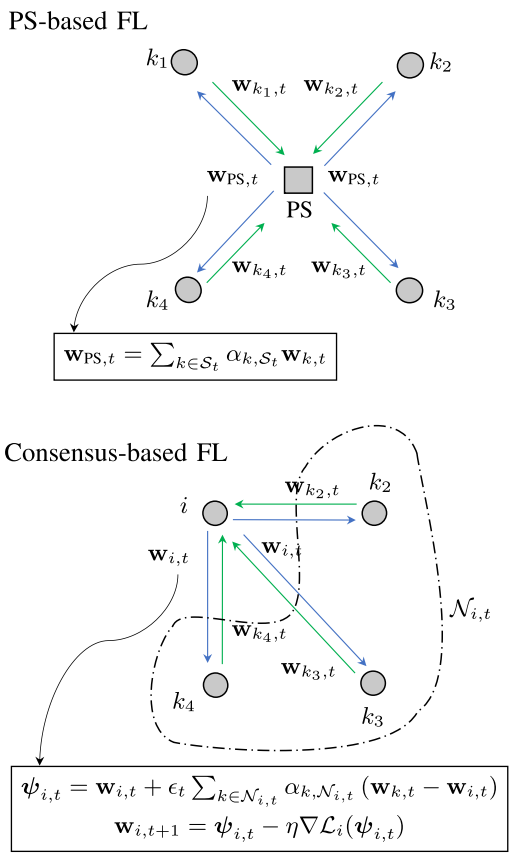


Fig. 1. Comparison of PS and consensus-based architecture in a network with four clients. The aggregation weights $\alpha_{k,\mathcal{S}_t}$ and $\alpha_{k,\mathcal{N}_{i,t}}$ are computed and applied in the first case by the PS and in the second case by each individual client (node i in the figure).

performed individually by each client, other algorithms focused on developing ad-hoc PS aggregation weighting to regulate the impact of each local model in the global update. An example can be found in [49], where the authors developed a Federated Adaptive Weighting (FedAdp) algorithm which employs PS aggregation weights based on the inner product between local gradients and the global gradient. Indeed, this metric can be used as a dis/similarity measure to estimate the contribution of the local model. Intuitively, the more the local and global gradients are orthogonal, the less the local model will positively contribute to the global aggregation.

Recently, a distributed version of FL, known as consensus-based FL [50], [51], [52], [53], has been proposed to address some of the challenges associated with PS-based FL, i.e., with centralized approaches [54]. As depicted in Fig. 1, differently from PS-based FL, in consensus-FL, clients collaboratively train models while also reaching a consensus on the model updates without a central coordinating entity, which leads to a more efficient and scalable learning. These techniques have evolved from conventional distributed maximum likelihood estimation based on consensus [55], where individual nodes depend exclusively on their local data and the information shared via connections with nearby nodes to update their local approximations. In the simplest version of consensus, i.e., Averaged Consensus (AC) [56], the model parameters are updated in a synchronized fashion and with constant or absence of weighting aggregation. Similar to AC, consensus-FL faces challenges such as asynchronous and fast convergence [57], [58], as well as reducing the carbon footprint [59]. To address these issues, consensus-FL algorithms, such as Gossip FL [60], Consensus-driven Federated Averaging (CFA) [61], [62], and dynamic layer selection [63], [64], have been proposed.

In distributed estimation, smarter Weighted AC (WAC) has been proposed [65] to extend the continuous-time approach [66] to a general discrete-time vector-parameter estimation problem. Recently, attempts have been made to apply WAC algorithms to FL under non-IID data distributions and network dynamics. For example, in [67], the authors used simulated clients with label-skewness (each client possessing data from only one or a few classes) but did not explore sample-skewness scenarios or conduct experiments involving real devices. Similarly, in [68], the focus was on label-skewness with simulations on time-varying topologies, but again, sample-skewness and real-world network dynamics were not considered. Moreover, these approaches typically employ conventional CFA algorithms, where the weighting is based solely on the number of samples in the local datasets. Such methods do not account for real network dynamics and do not exploit the learning contribution of each node when determining the appropriate weights for model aggregation. Given the relevance of applications like massive-IoT networks [52], vehicular communications [69], and industrial networks [70], where non-IID data distribution and network dynamics are common, the adoption of WAC in FL systems to handle these challenges is clearly a research direction to explore.

B. Contribution

The design of adaptive weighting algorithms for PS-based FL has been extensively studied in the literature and main challenges can be considered well understood. On the other hand, transferring algorithms optimized for conventional FL architectures to a fully decentralized platform (with no physical PS server) is challenging and partially addressed. The current literature shows a lack of consensus-based algorithms specifically designed for non-IID frameworks, since solutions developed for centralized FL contexts can not be directly applied to fully-distributed setups. To move a step forward in this direction, this paper proposes WAC algorithms designed for decentralized FL setups and under heterogeneous client populations assumptions. The proposed solutions extend popular FL techniques, such as FedAdp, broadening their scope of applicability in distributed contexts, where the adaptive aggregation of model parameters and the optimization strategy are performed client-side without a coordinating PS-part.

In summary, the main contributions are as follows.

- We make a comparative analysis on PS and consensus-based FL, highlighting the key similarities and differences, and investigating the main schemes for centralized weighted FL;
- We propose three different solutions for achieving weighted FL in decentralized network architectures, which mainly differ for the aggregation part according to the local contribution of the neighbors (based on a virtual PS, on

a selected client or on the local retained model). To the best of our knowledge, this is the first attempt to extend conventional WAC techniques to fully-decentralized FL with consensus;
- We analyze the convergence properties and evaluate the FL performance on a real platform consisting of heterogeneous IoT devices. The heterogeneity is taken into account by introducing asymmetries in both samples and label distributions, namely sample and label skewness, and assessed through different models and datasets complexities.

C. Paper Organization

The structure of this paper is as follows. Section II provides an overview of PS and consensus-based FL, together with a description of FL for non-IID local distributions. In Section III, we describe the weighted consensus algorithms, i.e., WAC for FL, detailing the main steps and convergence properties of the proposed algorithms. Section IV presents details about the dataset and FL platform, both simulated and with real IoT devices, followed by a discussion of the numerical results obtained with different non-IID data characterizations. Finally, Section V concludes the paper.

II. System Model

In this section, we first describe the vanilla PS-based FL tools and the consensus-based solutions for server-less architectures. Next, we discuss the main existing categories of FL algorithms for non-IID data.

A. From PS-Based to Server-Less FL Driven by Consensus

In the framework of FL, we consider a network that includes one PS and a collection of K clients denoted as $\mathcal{K} = \{1, \dots, K\}$. For notation purposes, throughout the paper, we will indicate with subscripts i and k the indices for the clients and neighbors, respectively. Each client possesses its own distinct dataset $\mathcal{D}_i$ with a size $D_i = |\mathcal{D}_i|$. The ultimate loss of the FL procedure is to achieve a global Deep Learning (DL) model that minimizes a loss function $\mathbf{w}_{\text{PS}} = \text{argmin}_{\mathbf{w}} \mathcal{L}(\mathbf{w})$, where $\mathcal{L}(\mathbf{w}) = \frac{1}{K}\sum_{i=1}^{K} \frac{D_i}{\sum_{i' \in \mathcal{K}} D_{i'}} \mathcal{L}_i(\mathbf{w}, \mathcal{D}_i)$, $\mathcal{L}_i$ represents the local cost determined by client i utilizing the data batches $\mathcal{D}_i$. An iterative process, involving a *local model optimization* step followed by an *aggregation* step executed on the PS, is used to obtain the global model.

At time (i.e., federated round) $t = 1, \dots, N_{\text{FL}}$, a set $\mathcal{S}_t \subseteq \mathcal{K}$ of clients is chosen to carry out the training procedure. Clients are required to generate local models via optimization in the FL process, typically employing supervised and gradient-based techniques, e.g., Adam optimizer [71], with mini-batch $\mathcal{B}$ of size B and learning rate η. Each client $i \in \mathcal{S}_t$ performs E local epochs prior to exchanging the local model with the PS, which is responsible for updating the global model. In the vanilla FL using PS, i.e., FedAvg, the *aggregation* step is conducted using a weighted average based on the number of samples D_i from each client:

$$\mathbf{w}_{\text{PS},t} = \sum_{i \in \mathcal{S}_t} \alpha_{i,\mathcal{S}_t} \mathbf{w}_{i,t}, \tag{1}$$

where $\mathbf{w}_{\text{PS},t}$ is the PS global model, $\mathbf{w}_{i,t}$ is the local model of client i and $\alpha_{i,\mathcal{S}_t} = \frac{D_i}{\sum_{i' \in \mathcal{S}_t} D_{i'}}$ are the mixing weights.

Decentralized FL architectures do not employ the PS but rather share their local model(s) repeatedly over Device-to-Device (D2D) links so as to reach a consensus on a global model (consensus-based FL): the clients form a graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, where each node $i \in \mathcal{V}_t$ corresponds to a client/learner, while the edge $(i, j) \in \mathcal{E}_t$, with $i \neq j$, signifies the presence of a communication link from client i to client j. The consensus-based algorithm, referred to as CFA, operates as follows. At round t, each client $i \in \mathcal{S}_t$ performs a *local model optimization* step and then exchanges the model parameters with its neighbors $\mathcal{N}_{i,t}$. Subsequently, an *aggregation* step is executed, similar to the PS [52]:

$$\boldsymbol{\psi}_{i,t} = \mathbf{w}_{i,t} + \epsilon_t \sum_{k \in \mathcal{N}_{i,t}} \alpha_{k,\mathcal{N}_{i,t}} \left(\mathbf{w}_{k,t} - \mathbf{w}_{i,t}\right), \tag{2}$$

where $\boldsymbol{\psi}_{i,t}$ represents the aggregated model, ϵ_t is the consensus step-size which modulates the memory of previous models and $\alpha_{k,\mathcal{N}_{i,t}} = \frac{D_k}{\sum_{k' \in \mathcal{N}_{i,t}} D_{k'}}$ are the mixing weights related to client k, based on the number of samples retained in each client. While the weights $\alpha_{k,\mathcal{N}_{i,t}}$ enable CFA to partially cope with sample unbalances, they do not fully capture the information gain of each local model in case of different data qualities or other types of non-IID data imbalances. Note that the aggregated model $\boldsymbol{\psi}_{i,t}$ represents an estimate of the global model as seen by client i and its neighborhood $\mathcal{N}_{i,t}$. However, as opposed to (1), here the aggregated model is obtained by taking into account the error between the local model and the neighbor ones. It is worth noting that the algorithm operates in the same manner if clients exchange gradients of the local model update instead of model parameters.

B. FL for non-IID

The importance of IID sampling in training data lies in the fact that it ensures the stochastic gradient is an unbiased estimate of the full gradient. FedAvg and CFA are known to be effective when data distribution across different nodes is the same as for the centrally collected data. However, in practice, data distribution across local nodes is typically non-IID, resulting in local losses $\mathcal{L}_i(\mathbf{w}_{\text{PS},t})$ to be closely related to data distribution $\mathcal{D}_i$ and local updates to gravitate towards the optima of its local loss $\mathcal{L}_i(\mathbf{w}_{\text{PS},t})$ rather than the global loss $\mathcal{L}(\mathbf{w}_{\text{PS},t})$. The inconsistency between local models $\mathbf{w}_{i,t}$ and the global model $\mathbf{w}_{\text{PS},t}$ accumulates during local training, necessitating more communication rounds for convergence. Consequently, multiple local updates during local training can potentially harm convergence and even cause divergence in the presence of non-IID data [3].

When it comes to designing PS-based FL algorithms for non-IID data, two major categories can be distinguished: either based on the local model optimization (e.g., adding a penalizing loss term) or either based on the global model aggregation step (e.g., weighting the local model contributions). Examples of the first category include FedProx [47], Distributed Approximate NEwton (DANE) [72], and Federated Curvature (FedCurv) [73]. FedProx uses parameter stiffness, i.e., an isotropic penalty term in the local loss, to avoid diverging from the global model.

DANE builds upon FedProx by adding a gradient correction term to accelerate convergence, while FedCurv exploits the Fisher information matrix to protect parameters that are important to each task.

The second category of algorithms, on which we focus on, leaves unaltered the local model optimization step, while targeting the optimal aggregation weights that the PS should employ. FedAdp [49] is one of the most relevant works in this direction which aims at designing the aggregation weights $\widetilde{\alpha}_{i,\mathcal{S}_t}$, such that $\sum_{i\in\mathcal{S}_t}\widetilde{\alpha}_{i,\mathcal{S}_t}=1$, in order to increase the convergence rate especially in non-IID settings. The main idea is that the greater the difference between the global gradient (i.e., of the global model) and the local gradients of each client, the higher is the weight that should be assigned to the local update. To this aim, the PS, after receiving the local updates, computes the global gradient starting from the gradient descent update:

$$\mathbf{w}_{\text{PS},t}=\mathbf{w}_{\text{PS},t-1}-\eta\nabla\mathcal{L}(\mathbf{w}_{\text{PS},t}) \tag{3}$$

with

$$\nabla\mathcal{L}(\mathbf{w}_{\text{PS},t})=\sum_{i\in\mathcal{S}_t}\alpha_{i,\mathcal{S}_t}\nabla\mathcal{L}_i(\mathbf{w}_{\text{PS},t}), \tag{4}$$

and $\nabla\mathcal{L}_i(\mathbf{w}_{\text{PS},t})=-\frac{\Delta_i}{\eta}$ represents the approximated local gradients of client i, with $\Delta_i=\mathbf{w}_{i,t}-\mathbf{w}_{\text{PS},t-1}$. A measure of the distance between the gradients can be obtained, as proposed in [49], by using the instantaneous angle $\theta_{i,t}$:

$$\theta_{i,t}=\arccos\frac{\nabla\mathcal{L}(\mathbf{w}_{\text{PS},t})^{\text{T}}\cdot\nabla\mathcal{L}_i(\mathbf{w}_{\text{PS},t})}{\|\nabla\mathcal{L}(\mathbf{w}_{\text{PS},t})\|\|\nabla\mathcal{L}_i(\mathbf{w}_{\text{PS},t})\|}, \tag{5}$$

where $\|\cdot\|$ represents the l2 norm. Furthermore, for numerical stability purposes, a smoothed angle $\widetilde{\theta}_{i,t}$ is used:

$$\widetilde{\theta}_{i,t}=\begin{cases}\theta_{i,t} & t=1\\ \frac{t-1}{t}\widetilde{\theta}_{i,t-1}+\frac{1}{t}\theta_{i,t} & t>1.\end{cases} \tag{6}$$

Finally, the aggregation weights are obtained as:

$$\widetilde{\alpha}_{i,\mathcal{S}_t}=\sum_{i\in\mathcal{S}_t}\alpha_{i,\mathcal{S}_t}e^{f(\widetilde{\theta}_{i,t})}, \tag{7}$$

where $f(\widetilde{\theta}_{i,t})=\alpha_{\text{G}}(1-e^{-e^{-\alpha_{\text{G}}(\widetilde{\theta}_{i,t}-1)}})$ is a variant of the Gompertz function [74] and α_{G} is an hyper-parameter. We highlight that α_{G} regulates the sensitivity with respect to the smoothed angle between the gradients. The higher α_{G}, the more sensitive the output is to the smoothed angle, potentially increasing the difference in contributions from the participating clients.

III. Weighted Consensus Algorithms

In this section, we first describe the proposed weighted consensus algorithm, specifically designed to handle heterogeneous clients. Next, based on the analysis of [49], we discuss the convergence properties which exploit the weighted consensus scheme.

In fully decentralized scenarios where the PS is not available, each client member of the federation must depend on the local updates from its neighbors to compute the aggregated model: the goal is to determine the optimal aggregation weights $\widetilde{\alpha}_{i,\mathcal{S}_t}$ to be used in the aggregation step. Following the FedAdp paradigm, the main challenge is to identify the equivalent global model on which the similarity metrics should be calculated. We propose that each client hosts virtual PS functions, where $\boldsymbol{\psi}_{i,t}$ in (2) can be now interpreted as an instance of the PS global model which is observed by the local client i using the available neighbors. In what follows, we refer to this solution as Consensus-driven FedAdp virtual PS (CFAdp-vPS). An alternative approach that will be explored in the next sections, involves selecting one specific neighbor local model as the reference instance (i.e., reference model) of the global model, and computing the various gradient distances concerning it. In the subsequent sections, we provide a detailed description of the two versions of weighted consensus algorithms, emphasizing their key characteristic steps.

A. CFAdp with Virtual PS

The first CFAdp algorithm is the dual version of the vanilla PS-based FedAdp method adapted to implement linear distributed average consensus among peer clients. The method is described in Algorithm 1. Here, each client i, after receiving the neighbors' local models $\mathbf{w}_{k,t}\ \forall k\in\mathcal{N}_{i,t}$, computes the aggregated gradient as:

$$\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})=\sum_{k\in\mathcal{S}_t}\alpha_{k,\mathcal{S}_t}\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t}), \tag{8}$$

where $\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})=-\frac{\Delta_i}{\eta}$ are the approximated local gradients of client k and $\Delta_k=\mathbf{w}_{k,t}-\boldsymbol{\psi}_{i,t-1}$. Note that (8) is the analogous of (4) where the global model $\mathbf{w}_{\text{PS},t}$ is substituted with its *local* representation $\boldsymbol{\psi}_{i,t}$. Here, the aggregated gradients serve as an approximation of the gradients gathered and combined by a PS, connected with the neighbors $\mathcal{N}_{i,t}$ and the node i itself. The instantaneous angles are computed similarly to (5) in lines 13 of Algorithm 1:

$$\theta_{k,\mathcal{S}_t}=\arccos\frac{\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})^{\text{T}}\cdot\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})}{\|\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})\|\|\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})\|}, \tag{9}$$

where the global gradients are substituted with the aggregated gradients, i.e., $\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})$ with respect to the aggregated model $\boldsymbol{\psi}_{i,t}$ in client i. The instantaneous angle directly relates to the amount of information that can be injected from client k into the learning system. Note that client i has no advantage with respect to all the other clients and that its model contribution can be made negligible according to the angles. Consequently, the new aggregated model is estimated with:

$$\boldsymbol{\psi}_{i,t}=\sum_{k\in\mathcal{S}_t}\widetilde{\alpha}_{k,\mathcal{S}_t}\mathbf{w}_{k,t}, \tag{10}$$

where the mixing weights $\widetilde{\alpha}_{k,\mathcal{S}_t}$ are obtained in line 16 as:

$$\widetilde{\alpha}_{k,\mathcal{S}_t}=\frac{D_k e^{f(\widetilde{\theta}_{k,\mathcal{S}_t})}}{\sum_{k'\in\mathcal{S}_t}D_{k'}e^{f(\widetilde{\theta}_{k',\mathcal{S}_t})}}\quad\forall k\in\mathcal{S}_t. \tag{11}$$

The mixing weights in (11) are obtained similarly to a softmax function, but with the usage of the Gompertz function. This function enables a slow and gradual variation of the weights for big angles, ensuring that major differences between clients do not

Algorithm 1: Consensus-driven FedAdp virtual PS.

procedure CFAdp-vPS $\mathcal{N}_{i,t}, \alpha_{k,\mathcal{S}_t}$ ▷ Run on client i
  authentication with network broker
  receive parameters (E, B) ▷RX from broker
  initialize $\mathbf{w}_{i,0} \leftarrow$ device i
  for each round $t = 1, 2, \ldots$ **do** ▷Training loop
    receive $\{\mathbf{w}_{k,t}\}_{k\in\mathcal{N}_{i,t}}$ ▷ RX from broker
    $Dec\{\mathbf{w}_{k,t}\}_{k\in\mathcal{N}_{i,t}}$ ▷Decipher weights
    $\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t}) = -\frac{\Delta_k}{\eta} \quad \forall k \in \mathcal{S}_t$
    $\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t}) = \sum_{k\in\mathcal{S}_t} \alpha_{k,\mathcal{S}_t} \nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})$
    $\theta_{k,\mathcal{S}_t} = \arccos \frac{\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}} \cdot \nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})}{\|\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})\|\|\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})\|} \quad \forall k \in \mathcal{S}_t$
    $\widetilde{\theta}_{k,\mathcal{S}_t} = \begin{cases} \theta_{k,\mathcal{S}_t} & t = 1 \\ \frac{t-1}{t}\widetilde{\theta}_{k,\mathcal{S}_{t-1}} + \frac{1}{t}\theta_{k,\mathcal{S}_t} & t > 1 \end{cases} \quad \forall k \in \mathcal{S}_t$
    $f(\widetilde{\theta}_{k,\mathcal{S}_t}) = \alpha_{\mathrm{G}}(1 - e^{-e^{-\alpha_{\mathrm{G}}(\widetilde{\theta}_{k,\mathcal{S}_t} - 1)}}) \quad \forall k \in \mathcal{S}_t$
    $\widetilde{\alpha}_{k,\mathcal{S}_t} = \frac{D_k e^{f(\widetilde{\theta}_{k,\mathcal{S}_t})}}{\sum_{k'\in\mathcal{S}_t} D_{k'} e^{f(\widetilde{\theta}_{k',\mathcal{S}_t})}} \quad \forall k \in \mathcal{S}_t$
    $\boldsymbol{\psi}_{i,t} = \sum_{k\in\mathcal{S}_t} \widetilde{\alpha}_{k,\mathcal{S}_t} \mathbf{w}_{k,t}$
    $\mathbf{w}_{i,t+1} = \boldsymbol{\psi}_{i,t} - \eta\nabla\mathcal{L}_i(\boldsymbol{\psi}_{i,t})$ ▷Model update
    send $Enc(\mathbf{w}_{i,t+1})$ ▷Encrypt and TX to broker
  end for
end procedure

lead to abrupt changes in the aggregation weights. As the angles decrease, the function allows for a more significant adjustment in the weights, enabling clients with similar, i.e., smaller angles, to have a more substantial impact on the consensus. Finally, the local model optimization step is performed starting from $\boldsymbol{\psi}_{i,t}$ as in CFA.

We want to point out that the CFAdp algorithm considers both the volume of data and the contribution of each node in the neighborhood (i.e., the correlation between the local and aggregated gradients obtained from neighbors) when determining the appropriate weights for model aggregation. As described in Section III-C, this allows to infer an upperbound to the rate at which the overall FL loss may decline, under specific assumptions on loss function.

B. CFAdp with Client Selection

As clarified in the analysis of Section IV, in some cases, computing the aggregated gradients as in (8) may not be the best way to combine the local gradients observed in the neighborhood. For example, this might happen when a neighbor's local model gives a limited or negative contribution to the FL process. The proposed algorithm, referred to as CFAdp with Client Selection (CFAdp-CS), allows every client to proactively select the best neighbor model which is used to represent the new aggregated model, namely, a new reference $\boldsymbol{\psi}_{i,t}$ for the next round t.

The comprehensive pseudo-code is outlined in Algorithm 2, and it works in the following way. After receiving the neighbor models, each client i computes the instantaneous and smoothed angles, and Gompertz function for each couple

Algorithm 2: Consensus-driven FedAdp with Client Selection.

procedure CFAdp-CS $\mathcal{N}_{i,t}, \epsilon_t$ ▷ Run on client i
  authentication with network broker
  receive parameters (E, B) ▷RX from broker
  initialize $\mathbf{w}_{i,0} \leftarrow$ device i
  for each round $t = 1, 2, \ldots$ **do** ▷Training loop
    receive $\{\mathbf{w}_{k,t}\}_{k\in\mathcal{N}_{i,t}}$ ▷ RX from broker
    $Dec\{\mathbf{w}_{k,t}\}_{k\in\mathcal{N}_{i,t}}$ ▷Decipher weights
    $\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t}) = -\frac{\Delta_k}{\eta} \quad \forall k \in \mathcal{S}_t$
    $\theta_{k_1,k_2,t} = \arccos \frac{\nabla\mathcal{L}_{k_1}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}} \cdot \nabla\mathcal{L}_{k_2}(\boldsymbol{\psi}_{i,t})}{\|\nabla\mathcal{L}_{k_1}(\boldsymbol{\psi}_{i,t})\|\|\nabla\mathcal{L}_{k_2}(\boldsymbol{\psi}_{i,t})\|} \quad \forall k_1, k_2 \in \mathcal{S}_t, k_1 \neq k_2$
    $\widetilde{\theta}_{k_1,k_1,t} = \begin{cases} \theta_{k_1,k_2,t} & t = 1 \\ \frac{t-1}{t}\widetilde{\theta}_{k_1,k_2,t-1} + \frac{1}{t}\theta_{k_1,k_2,t} & t > 1 \end{cases} \quad \forall k_1, k_2 \in \mathcal{S}_t, k_1 \neq k_2$
    $f(\widetilde{\theta}_{k_1,k_2,t}) = \alpha_{\mathrm{G}}(1 - e^{-e^{-\alpha_{\mathrm{G}}(\widetilde{\theta}_{k_1,k_2,t^{-1}})}}) \quad \forall k_1, k_2 \in \mathcal{S}_t, k_1 \neq k_2$
    $k_t^* = \mathrm{argmax}_k \sum_{k'\in\mathcal{S}_t, k'\neq k} f(\widetilde{\theta}_{k,k',t})$
    $\widetilde{\alpha}_{k,k_t^*} = \frac{D_k e^{f(\widetilde{\theta}_{k,k_t^*})}}{\sum_{k'\in\mathcal{N}_{k_t^*}} D_{k'} e^{f(\widetilde{\theta}_{k',k_t^*})}} \quad \forall k \in \mathcal{N}_{k_t^*}$
    $\boldsymbol{\psi}_{i,t} = \mathbf{w}_{k_t^*} + \epsilon_t \sum_{k\in\mathcal{N}_{k_t^*}} \widetilde{\alpha}_{k,k_t^*}(\mathbf{w}_{k,t} - \mathbf{w}_{k_t^*})$
    $\mathbf{w}_{i,t+1} = \boldsymbol{\psi}_{i,t} - \eta\nabla\mathcal{L}_i(\boldsymbol{\psi}_{i,t})$ ▷Model update
    send $Enc(\mathbf{w}_{i,t+1})$ ▷Encrypt and TX to broker
  end for
end procedure

of neighbors, obtaining the matrices $\theta_{k_1,k_2,t}$, $\widetilde{\theta}_{k_1,k_1,t}$ and $f(\widetilde{\theta}_{k_1,k_2,t})\,\forall k_1, k_2 \in \mathcal{S}_t, k_1 \neq k_2$, respectively. In particular, the instantaneous angles are estimated as:

$$\theta_{k_1,k_2,t} = \arccos \frac{\nabla\mathcal{L}_{k_1}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}} \cdot \nabla\mathcal{L}_{k_2}(\boldsymbol{\psi}_{i,t})}{\|\nabla\mathcal{L}_{k_1}(\boldsymbol{\psi}_{i,t})\|\|\nabla\mathcal{L}_{k_2}(\boldsymbol{\psi}_{i,t})\|}. \tag{12}$$

This permits the evaluation of the similarities between clients' updates, holding a method to select the best reference clients with respect to the current update. Then, the reference neighbor model is selected as:

$$k_t^* = \mathrm{argmax}_k \sum_{k'\in\mathcal{S}_t, k'\neq k} f(\widetilde{\theta}_{k,k',t}). \tag{13}$$

The reference neighbor in (13) is chosen by taking among the summations of the Gompertz functions since this helps in identifying the client whose local model is most similar to the majority of the neighboring models, thus maximizing the potential contribution to the consensus. The aggregation weights are computed excluding the reference client as:

$$\widetilde{\alpha}_{k,k_t^*} = \frac{D_k e^{f(\widetilde{\theta}_{k,k_t^*})}}{\sum_{k'\in\mathcal{N}_{k_t^*}} D_{k'} e^{f(\widetilde{\theta}_{k',k_t^*})}} \quad \forall k \in \mathcal{N}_{k_t^*}. \tag{14}$$

As opposed to (10), the aggregated model is now obtained using the selected client k_t^* as reference and implementing an

exponential moving average:

$$\boldsymbol{\psi}_{i,t} = \mathbf{w}_{k_t^*} + \epsilon_t \sum_{k \in \mathcal{N}_{k_t^*}} \widetilde{\alpha}_{k,k_t^*} \left(\mathbf{w}_{k,t} - \mathbf{w}_{k_t^*} \right), \tag{15}$$

where ϵ_t modulates the memory of previous models. The benefit of this approach is that, in a fully-connected network, all clients will select the same reference client during each federated round, leading to a more stable and seamless convergence. This is especially advantageous when one client possesses a highly representative model, as it can serve as a reference to accelerate the convergence process. Notice that (15) can be rewritten as (10) as shown in Appendix A1. Therefore, CFAdp-CS can be interpreted as a special case of CFAdp-vPS for which the same convergence properties as described in Section III-C hold.

As a last remark, note that through Algorithm 2 it is always possible to force each client i to choose its own local model i as the reference, so that $k_t^* = i, \forall t$, and regardless of the reliability of the local data and update. This particular *degenerate* case, referred to as CFAdp Egocentric (CFAdp-Ego), will further analyzed in Section IV. In this case, the model aggregation (15) reduces to:

$$\boldsymbol{\psi}_{i,t} = \mathbf{w}_{i,t} + \epsilon_t \sum_{k \in \mathcal{N}_{i,t}} \widetilde{\alpha}_{k,i} \left(\mathbf{w}_{k,t} - \mathbf{w}_{i,t} \right). \tag{16}$$

In the egocentric approach, each client uses its own local model as the reference. In other words, the instantaneous angles in (12) are computed between the local gradients $\nabla \mathcal{L}_i$ and the neighbors' gradients $\nabla \mathcal{L}_k$. If the local model is highly biased, such as in the scenario depicted in Fig. 6 with 1 IID client and 9 non-IID clients, relying on a biased local reference can hinder optimal aggregation.

The three proposed strategies, namely CFAdp-vPS, CFAdp-CS and CFAdp-Ego, are visually represented in Fig. 2 to highlight their mutual differences. In Fig. 2(a), representing CFAdp-vPS, the reference clients (highlighted in red) are all the available ones in the subset $\mathcal{S}_t$, as described in (8). Therefore, each client acts as a virtual PS, which collects and updates the model independently from its index.

In Fig. 2(b), illustrating CFAdp-CS, the reference client changes at each round according to the model with the highest contribution, as determined by (13). Note that the selected client is the same for all clients in the network during that round, leading to a more stable and seamless convergence.

Finally, in Fig. 2(c), we show the CFAdp-Ego, where each client adopts its local model as the reference. This means that the aggregation is centered around each client's own model, and the similarities are computed between the local gradients and those of the neighbors. As previously discussed, if the local model is highly biased, its dependence can lead to suboptimal aggregation and affect the convergence of the algorithm.

C. Convergence Analysis

We now analyze the theoretical convergence of the proposed CFAdp algorithm by adopting typical FL assumptions [35], [40], [47], [49], [51].

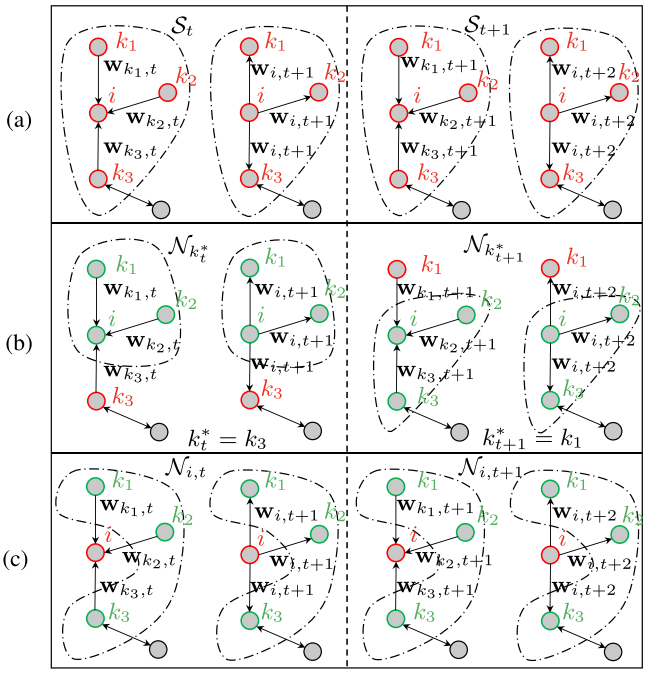


Fig. 2. Three Consensus-driven Federated Adaptive Weighting (CFAdp) strategies, where reference models and neighbors are highlighted in red and green, respectively. From top to bottom: CFAdp with virtual PS (CFAdp-vPS), with client selection (CFAdp-CS) and egocentric (CFAdp-Ego).

Assumption 1. γ-Lipschitz smoothness: Considering a generic client i member of the federation, we let $\mathcal{L}_k(\boldsymbol{\psi}_{i,t})\ \forall k \in \mathcal{S}_t$ be γ-Lipschitz smooth i.e., $\|\nabla \mathcal{L}_k(\boldsymbol{\psi}_{i,t}) - \nabla \mathcal{L}_k(\boldsymbol{\psi}_{i,t+q})\| \leq \gamma \|\boldsymbol{\psi}_{i,t} - \boldsymbol{\psi}_{i,t+q}\|$ for any two parameter vectors $\boldsymbol{\psi}_{i,t}, \boldsymbol{\psi}_{i,t+q}$.

Based on Assumption 1, the local representation of the global objective, defined as $\nabla \mathcal{L}(\boldsymbol{\psi}_{i,t}) = \sum_{k \in \mathcal{S}_t} \alpha_{k,\mathcal{S}_t} \nabla \mathcal{L}_k(\boldsymbol{\psi}_{i,t})$, can also be assumed as γ-Lipschitz smooth since $\sum_{k \in \mathcal{S}_t} \alpha_{k,\mathcal{S}_t} = 1$ for each subset $\mathcal{S}_t$.

Assumption 2. Bounded Local Dissimilarity: For any client i, the dissimilarity between local loss of client k and the local representation of the global objective at $\boldsymbol{\psi}_{i,t}$ is bounded by A and B, i.e., $A\|\nabla \mathcal{L}(\boldsymbol{\psi}_{i,t})\| \leq \|\nabla \mathcal{L}_k(\boldsymbol{\psi}_{i,t})\| \leq B\|\nabla \mathcal{L}(\boldsymbol{\psi}_{i,t})\|$.

Notice that when all the local data samples are the same, it is $A = B = 1$, therefore the local dissimilarity $|A - B|$ in Assumption 2 might be an indicator of the data heterogeneity among clients, under the assumption of the same training configuration.

Assumption 3. Stationarity: For any client i, we assume that the subsets $\mathcal{S}_t$ and $\mathcal{N}_{i,t}\ \forall i \in \mathcal{S}_t$ are stationary over time.

In the context of PS-based FL, the same assumptions have been made in several works [16], [35], [47], [51], [75], [76], [77], [78], [79] to ensure stability and successful convergence of the learning process across distributed datasets. Smoothness [75], [76], [77] is a safeguard to ensure that the learning process is stable, meaning small changes in the model parameters don't result in large variations in learning loss, while the bounded dissimilarity [16], [78] ensures that learning can be effectively coordinated across different clients to converge towards a useful model that generalizes across all participating nodes.

Finally, by assuming stationary neighbors (as done in [79]), the client selection policy remains consistent over time, which is crucial for establishing the convergence guarantees of the WAC strategy. This practical assumption acknowledges that not all nodes participate in every round due to factors like network connectivity issues or device availability, but maintains that the selection of participating clients does not change drastically over time, thus providing a stable and predictable learning environment. However, the algorithm remains operational even in non-stationary environments. In scenarios where the network topology changes, the algorithm can still work effectively if the client selection policy adheres to a consistent random and uniform mechanism. This ensures ergodicity, meaning every client has a fair chance of being selected over time, regardless of dynamic network changes.

It is important to note that the consensus-based FL inherently complements the mesh network structure, where each node not only captures and disseminates its own data, but also serves as a relay for other nodes. In the case of a mesh network, even if the mobility of the clients is partially present, the consensus-based FL model would still work effectively, assuming that the set of neighboring clients remains stationary. This essentially implies that even if a subset of clients changes its position or connection, as long as the overall structure of the neighboring set of clients remains consistent over time, the learning process can proceed uninterrupted.

Theorem 1: With loss function $\mathcal{L}_k(\boldsymbol{\psi}_{i,t})$ satisfying Assumptions 1, 2, 3 and supposing $\boldsymbol{\psi}_{i,t}$ is not a stationary solution, the expected decrease in the global loss function on client i-th and between two consecutive consensus rounds satisfies,

$$\mathcal{L}(\boldsymbol{\psi}_{i,t+1}) \leq \mathcal{L}(\boldsymbol{\psi}_{i,t}) - \eta \mathbb{E}_{k\in\mathcal{S}_t}\left[\left(\frac{\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}}\cdot\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})}{\|\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})\|\|\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})\|} - \frac{B\gamma\eta}{2}\right)\frac{A^2}{2}\|\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})\|^2\right], \tag{17}$$

where the expectation $\mathbb{E}_{k|t}$ refers to the weighting strategy of the client $k \in \mathcal{S}_t$ for global model aggregation.

The proof of Theorem 1 builds upon [49] while to guarantee paper self-consistency, it is discussed in Appendix A2. Theorem 1 provides a bound on how rapid the decrease of the FL loss can be expected on the generic client i. It is straightforward to verify that the convergence upper bound of decentralized FL tool after N_{FL} consensus rounds is given by

$$\mathcal{L}(\boldsymbol{\psi}_{i,N_{\mathrm{FL}}}) \leq \mathcal{L}(\boldsymbol{\psi}_{i,1}) - \eta\sum_{t=1}^{N_{\mathrm{FL}}}\mathbb{E}_{k\in\mathcal{S}_t}\left[\left(\frac{\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}}\cdot\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})}{\|\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})\|\|\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})\|} - \frac{B\gamma\eta}{2}\right)\frac{A^2}{B}\|\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})\|^2\right]. \tag{18}$$

Based on Theorem 1, we have the following remarks.

Remark 1: The decrease of FL loss on the client i and between two consecutive learning rounds shows the same dependencies as in the FedAdp algorithm [49] including the bound gap $[A, B]$. The correlation $\frac{\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}}\cdot\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})}{\|\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})\|\|\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})\|}$ between the local gradient and the representation of the global gradient, obtained from the received neighbor models, is a local metric to measure their alignment level.

Remark 2: Similarly as for PS-based FedAdp, increasing $\mathbb{E}_{k\in\mathcal{S}_t}[\cdot]$ in each global round improves the convergence of decentralized FL. Contributions of each individual neighbor can be measured quantitatively through the correlation $\frac{\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}}\cdot\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})}{\|\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})\|\|\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})\|}$ between the local gradient $\nabla\mathcal{L}_k(\boldsymbol{\psi}_{i,t})$ and the local representation of the global gradient $\nabla\mathcal{L}(\boldsymbol{\psi}_{i,t})$ obtained from the neighbors, and assign larger weights to the nodes with higher contribution to enlarge the expected decrease of FL loss in each global round.

IV. Simulation Experiments

In this section, we first describe the real-world FL network platform, and then clarify the main networking characteristics and tools that underpin the CFAdp-CS, CFAdp-vPS and CFAdp-Ego consensus processes. Finally, we present the results on convergence properties and performances with highly skewed non-IID data distributions.

A. FL Networking Characteristics and Platform

In order to validate the three proposed CFAdp strategies, we adopted both simulated and real fully-distributed clients connected via Wireless Local Area Network (WLAN) and communicating, i.e., exchanging neural network model parameters, through the broker-based Message Queuing Telemetry Transport (MQTT) protocol [80]. The simulated network of clients was implemented in a workstation featuring an Intel(R) Xeon(R) Silver 4210R CPU operating at 2.40GHz, 96GB of RAM, and a Quadro RTX 6000 24GB GPU. This allowed us to have more flexibility in defining the computational capabilities and the number of clients. On the contrary, the real-FL platform prototype was composed of 6 *Jetson Nano* devices [81] equipped with CPU ARM-Cortex-A57 and GPU 128-core Maxwell. The laboratory comprising the IoT devices and the workstation is shown in Fig. 3.

Communications among the clients in the consensus scheme are managed by an MQTT broker, which receives and forwards model updates. Specifically, each client subscribes to the topics related to the model parameters of its neighbors. Upon completing a local training round, a client pushes its updated local model to its designated topic, e.g., `/fl_session_ID/client_ID`, and pulls the updated models from its neighbors. This pulling operation can be performed asynchronously and automatically, allowing the clients to continuously listen for new updates on their neighbors' topics. The MQTT protocol ensures reliable communication by handling possible packet losses and retransmissions, guaranteeing exactly-once packet delivery through Quality-of-Service (QoS) level 2. Notice that, as observed during the experimental tests, a high QoS level might introduce transmission delays.

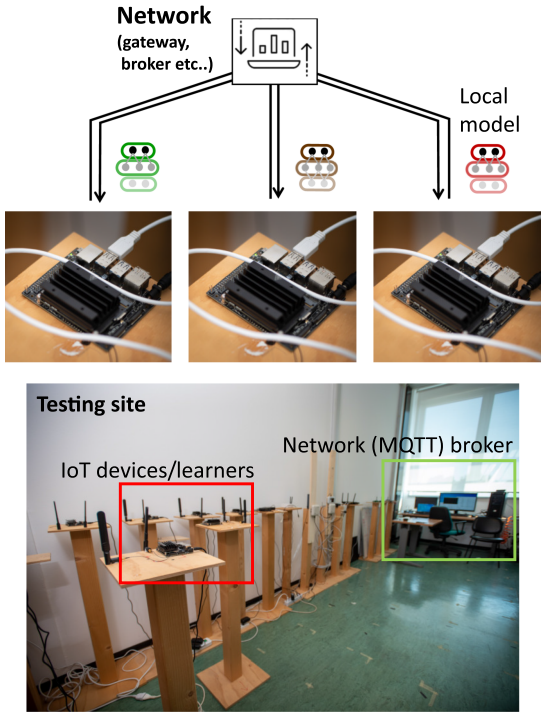


Fig. 3. Real FL platform composed of IoT devices, i.e., *Jetson Nano*, connected via WLAN to a workstation.

The development of strategies to mitigate them has not been considered in this paper, as our primary objective is to design a WAC learning strategy tailored for non-IID local data. Despite the observed delays in the tests, the use of MQTT transport combined with the proposed WAC strategies demonstrated robustness and resilience. For more insights into the potential effects of transmission delays on FL optimization procedures in the presence of a PS, we refer to our previous work [39].

B. FL Dataset and Implementation

Regarding the datasets, we employed a simple dataset widely used for classification tasks, i.e., the Modified National Institute of Standards and Technology (MNIST) [82] dataset using the full validation data, and a more complex scenario with Canadian Institute For Advanced Research (CIFAR)-100 dataset [83]. The corresponding adopted DL models are a Convolutional Neural Network (CNN) (LeNet architecture [84]) and a Convolutional Vision Transformers (CVT) model [85]. For training procedures, we considered the Adam optimization algorithm [71] with an initial learning rate of 0.0001 and momentum values of $m_1 = 0.9$ and $m_2 = 0.999$. To choose the hyper-parameter α_{G}, we tested the values in the range $[1, 10]$ and we reported the results in Fig. 4. Note that increasing α_{G} improved the accuracy up to a certain point, beyond which the performance gains saturated. We found that $\alpha_{\mathrm{G}} = 4$ provided a good trade-off between sensitivity to the smoothed angles and preserving the distinguishability of contributions from nodes with smaller angle differences. Therefore, for the experiments, we assigned α_{G} equal to 4.

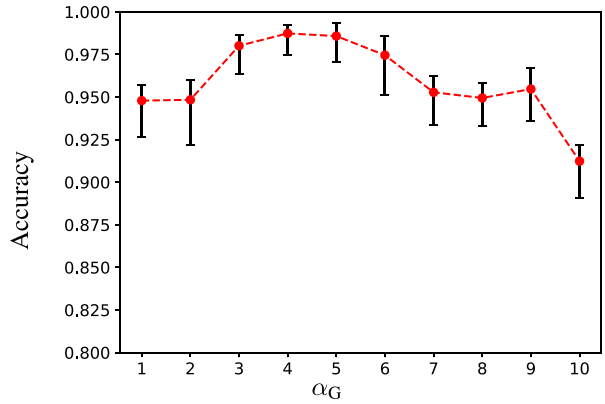


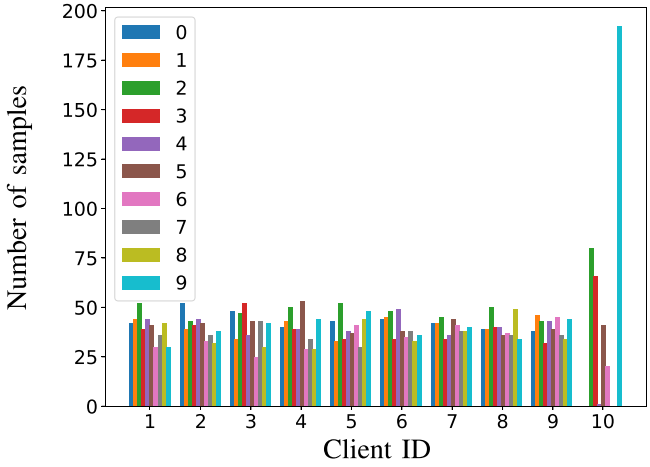
Fig. 4. Tuning of the Gompertz function hyper-parameter α_{G}.

The consensus step-size ϵ_t was set to 0.3. Finally, we adopted a number of local epochs $E = 1$ and a maximum number of federated rounds $N_{\mathrm{FL}} = 100$.

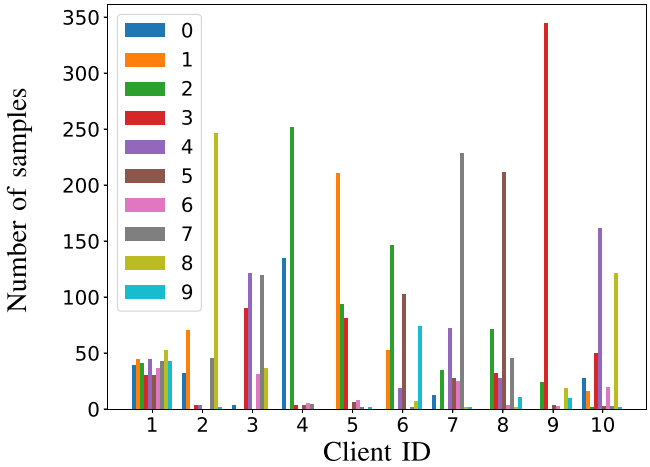
In order to accurately regulate the degree of non-IIDness between clients, we considered two cases. First case adopts a Dirichlet distribution to assign the number of local samples, namely the *quantity skew*. For the second case, the Dirichlet density is used to regulate the label distributions, or *label skew* [40], [86], [87]. In particular, for non-IID sample distributions, the same percentage of labels is retained on each client, while the number of samples in client i is $D_i = p_i^{(\mathrm{S})} D$, where D is the total number of training samples and $\boldsymbol{p}^{(\mathrm{S})} = [p_i^{(\mathrm{S})}]_{i=1}^{K} \sim Dir(\boldsymbol{\beta}^{(\mathrm{S})})$ are the random samples of a Dirichlet distribution with concentration parameters $\boldsymbol{\beta}^{(\mathrm{S})}$. Here, for simplicity, we consider $\boldsymbol{\beta}^{(\mathrm{S})} = [\beta_i^{(\mathrm{S})}]_{i=1}^{K} = [\beta^{(\mathrm{S})}]_{i=1}^{K}$. On the contrary, for non-IID label distributions, the same quantity of samples is kept on each client, while the distribution of labels across clients follows the Dirichlet distribution. In particular, for a client i, the proportion of label ℓ in the local dataset is $p_{i\ell}^{(\mathrm{L})}$, where $\boldsymbol{p}_i^{(\mathrm{L})} = [p_{i\ell}^{(\mathrm{L})}]_{\ell=0}^{9} \sim Dir(\boldsymbol{\beta}^{(\mathrm{L})})$ and $\boldsymbol{\beta}^{(\mathrm{L})} = [\beta_\ell^{(\mathrm{L})}]_{\ell=0}^{9} = [\beta^{(\mathrm{L})}]_{\ell=0}^{9}$.

C. Convergence Analysis

In this first assessment, we aim at verifying the convergence capabilities of the proposed algorithms within a specific client, e.g., $i = 10$, whose local data distribution is much different from those of its neighbors. This is done in order to compare the different methods (i.e., CFAdp methods and non-adaptive baseline strategy CFA) in the worst-case scenario where non-IID clients usually struggle to converge. To this aim, we simulated a network of $K = 10$ clients in two scenarios: (a) 9 clients out of 10 hold a uniform distribution of labels, while a single client holds a non-IID label distributions with $\beta^{(\mathrm{L})} = 0.2$, and (b) only 1 client out of 10 holds a IID label distribution. For an example showing how labels are assigned, we refer to Fig. 5, where we represent the histogram of the labels for each client. Note that the client $i = 10$ under consideration has a very imbalanced distribution of labels, e.g., in Fig. 5(a) digits 7 and 8 are missing.



(a) 9 IID - 1 non-IID scenario



(b) 1 IID - 9 non-IID scenario

Fig. 5. Examples of label distributions in a federation of 10 clients. (a) Clients 1-9 retain a uniform distribution of labels (i.e., colors 0-9), while client 10 holds an imbalanced distribution. (b) Only client 1 has IID local distribution.

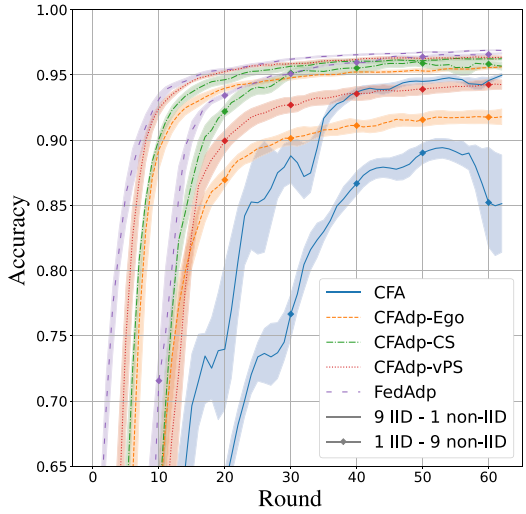


Fig. 6. Validation accuracy in a non-IID client varying the number of training FL rounds, for different consensus algorithms (i.e., baseline CFA and proposed CFAdp-Ego, CFAdp-CS and CFAdp-vPS). Two scenarios are represented: (a) 9 IID - 1 non-IID, (b) 1 IID - 9 non-IID clients.

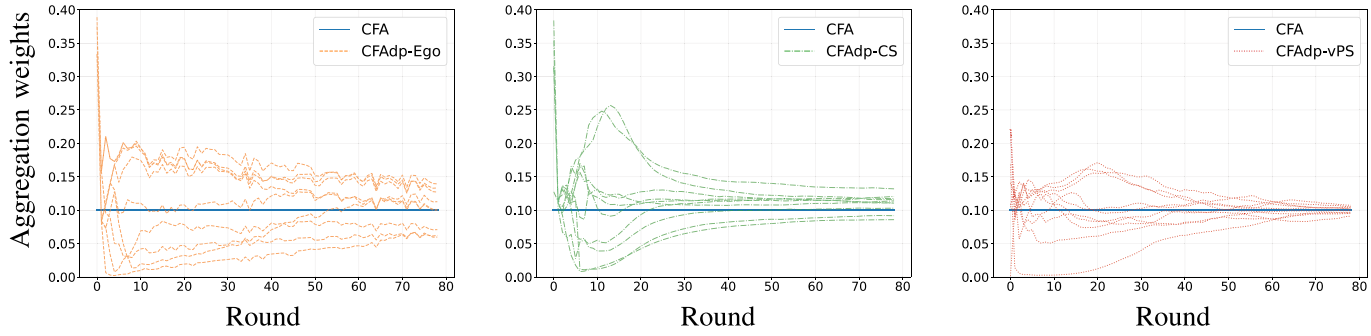
In Fig. 6, we show the average validation accuracy computed by $i = 10$ for each federated round and varying the consensus algorithm, including both scenarios (a) and (b). This is done to establish lower and upper bounds for each consensus algorithm's performance in the presence of non-IID local data. Further intermediate non-IID cases are tested in Section IV-D. The confidence bounds are obtained using the standard deviation as uncertainties. Additionally, we have included the centralized (i.e., PS-based) FedAdp algorithm in our comparisons as an upper bound to the consensus CFAdp versions. This inclusion allows us to benchmark the performance of our proposed algorithms against the best possible scenario in a centralized setting. A common behaviour in the two scenarios is that with CFA, due to the highly imbalanced distribution, the client struggles to converge and presents drops of performances caused by local overfitting. On the contrary, adopting a FL WAC strategy as CFAdp-Ego, we notice a much higher speed of convergence and stability on the training. However, given the fact that the reference local model is highly biased towards the skewed distribution, the performances are inferior to the CFAdp-CS which automatically selects the best neighbors' model as reference. Focusing on scenario (b), with CFAdp-CS we also avoid using the unbalanced models as references and outperform even the CFAdp-vPS. On the contrary, whenever the vast majority of clients have IID distributions as in scenario (a), it is more convenient to adopt the CFAdp-vPS since it represents the equivalent PS-based version.

To further analyze the convergence of the proposed algorithms, in Fig. 7, we represent the aggregation weights in client $i = 10$ and scenario (a) at different training epochs for every neighbor k, i.e., $\alpha_{k,\mathcal{S}_t}$, $\widetilde{\alpha}_{k,i}$, $\widetilde{\alpha}_{k,k_t^*}$ and $\widetilde{\alpha}_{k,\mathcal{S}_t}$ for CFA, CFAdp-Ego, CFAdp-CS and CFAdp-vPS, respectively. Notice that the baseline strategy maintains all the weights $\alpha_{k,\mathcal{S}_t} = \frac{D_k}{\sum_{k' \in \mathcal{S}_t} D_{k'}} = 1/K = 0.1$ since all clients retain the same number of samples. On the contrary, the WAC strategies modulate the weights in order to compensate the unbalance between label distributions. Here the consensus step-size ϵ_t is fixed to 0.3 for all epochs in order not to alter the convergence of the aggregation weights. Between the WAC strategies, we can see that the CFAdp-Ego presents periodical changes of weights values due to tendency on overfitting. Moreover, the convergence times are much slower if compared with CFAdp-CS and CFAdp-vPS. Indeed, selecting the best neighbor dramatically smooths the convergence process and ultimately leads to a faster approximation toward the CFAdp-vPS solution.

D. Quantity and Label Skewness

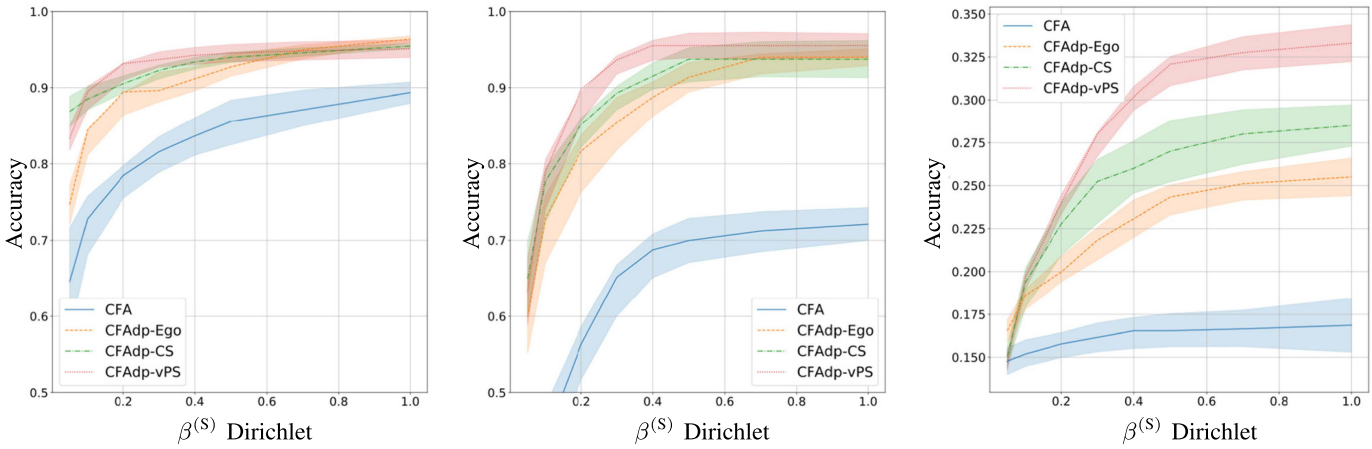
In this section we evaluate two different datasets, i.e., MNIST and CIFAR100, using two DL models, i.e., CNN and CVT, respectively, tested on both simulated and real devices.

In this first experiment, we assess the performances of the proposed methods when the number of samples in each client varies significantly according to a Dirichlet distribution described in Section IV-B. This is important in order to measure the



(a) CFA vs CFAdp-Ego (b) CFA vs CFAdp-CS (c) CFA vs CFAdp-vPS

Fig. 7. Average aggregation weights after 5 different runs of CFA, CFAdp-Ego (a), CFAdp-CS (b) and CFAdp-vPS (c), at varying training federated rounds. We refer to (16), (14) and (11) for the aggregation weights $\widetilde{\alpha}_{k,i}$, $\widetilde{\alpha}_{k,k_t^*}$ and $\widetilde{\alpha}_{k,\mathcal{S}_t}$ of CFAdp-Ego, CFAdp-CS and CFAdp-vPS, respectively. The baseline non-adaptive CFA has all the weights $\alpha_{k,\mathcal{S}_t} = 1/K = 0.1$.



(a) $K = 3$ clients scenario (b) $K = 10$ clients scenario (c) $K = 30$ clients scenario

Fig. 8. Mean reached validation accuracy after 20 rounds of training for varying concentration parameters $\beta^{(S)}$ on the number of samples. The confidence bounds are obtained using the standard deviation as uncertainties.

capabilities of coping with important or negligible local updates, as well as with generalized or overfitted models. To this aim, we simulated three different FL scenarios with $K = \{3, 10, 30\}$ clients and we plot in Fig. 8 the mean and standard deviations of the reached validation accuracy after 20 rounds, varying $\beta^{(S)} \in [0.01, 1]$.

From the results, we note that the observed tendency in the previous experiments is found again among all $\beta^{(S)}$ values, i.e., the simple weight aggregation strategy of CFA, based solely on the number of samples, struggles with very imbalanced distributions. Starting with the three-client scenario, we notice that if the distributions are the same among all clients, i.e., high $\beta^{(S)}$, the proposed CFAdp strategies reach the same level of accuracy. This is intuitive since whenever all models bring the same contributions, there is not advantage in choosing a specific one. Still, the WAC, with adaptive aggregation weights, outperforms the conventional CFA from 7 to 56%. With a higher number of clients, the convergence time increases and the differences between the proposed WAC strategies become more distinct.

In the last assessment, we employ the real FL-prototype composed of 6 IoT devices where each client experiences non-IID label distribution with $\beta^{(L)}$ varying in $[0.05, 100]$. Different from before, here we employ the much more complex CVT model and CIFAR100 dataset. In Fig. 9, we report the validation accuracy reached after $N_{FL} = 100$ federated round, for each consensus algorithm. Note that with different $\beta^{(L)}$ and bigger models, the CFAdp-CS and CFAdp-vPS have almost the same performances. This is due to the fact that an increase in the number of model parameters, coupled with the use of a larger number of classes (100), worsens the overfitting of local models. Ultimately, it results in an equivalent choice between using all clients or the best-one as a reference. On the contrary, CFAdp-Ego, reduces the maximum achievable performance since it always relies on the local model which used as a reference for all the rounds. Finally, the CFA struggles to achieve 80% of accuracy.

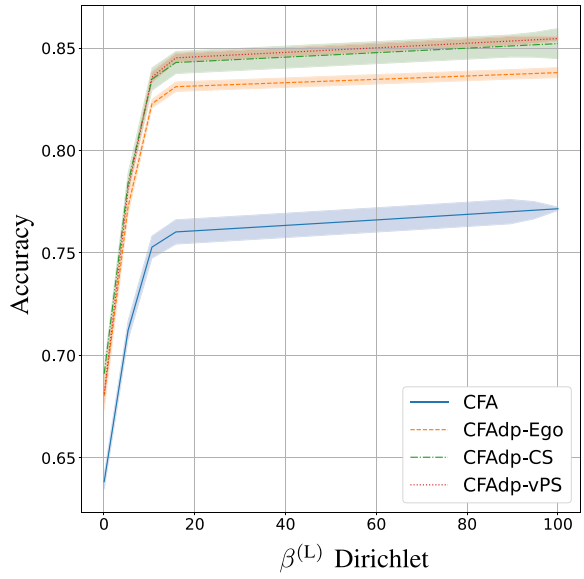


Fig. 9. Mean validation accuracy reached after 100 rounds of training for varying concentration parameters $\beta^{(L)}$ on the label skewness. The confidence bounds are obtained using the standard deviation as uncertainties.

V. Conclusion

This paper addressed the challenges associated with non-IID data distribution in fully-distributed, server-less networked learning systems by introducing a new family of algorithms with roots in WAC tools and adapted for FL processes, namely Consensus-driven FedAdp (CFAdp). Evolved from WAC schemes, the proposed tools have been optimized and adapted specifically for FL, each one employing a unique strategy for calculating the global model. Specifically, we developed three CFAdp algorithms, named CFAdp-Ego, CFAdp-CS, and CFAdp-vPS, where the reference local model is within the client itself, within the best selected neighbor and across all clients, respectively. They are then evaluated in terms of their convergence properties and resilience against non-IID data distribution. The evaluation included both simulated and real experiments using a FL platform implemented over a WLAN network, testing with varying model complexities, i.e., CNN and CVT, and datasets, i.e., MNIST and CIFAR100. Specifically, the tests aimed to simulate complexity and data heterogeneity typically encountered in IoT deployments, including various degrees of sample and label skewness modelled with Dirichlet distributions.

The derived key takeaways are the following. The weighted consensus schemes, i.e., CFAdp, outperform the vanilla tools, such as CFA, up to 56% thanks to the dynamic adjustment of the aggregation weights, disregarding negative client contributions. Whenever the federation of clients has IID or non-IID local distributions, meaning that each client has the same level of non-IID (quantity or label skewness), the CFAdp-vPS, which represents the equivalent PS-based version, achieves the best performances in terms of convergence time and asymptotic accuracy. Conversely, whenever one or few clients retain very high-quality and evenly distributed data, the CFAdp-CS permits to take advantage of the good local model contribution by taking the best client as a global reference.

Appendix

A CFAdp-CS vs CFAdp-VPS: Client Selection Process

In this section we discuss Algorithm 2 which implements CFAdp by selecting a client as opposed to Algorithm 1 that implements weighting average. Algorithm 2 averaging operation in (15) can be rewritten as follows:

$$\boldsymbol{\psi}_{i,t} = \left(1 - \epsilon_t \sum_{k \in \mathcal{N}_{k_t^*}} \widetilde{\alpha}_{k,k_t^*}\right) \mathbf{w}_{k_t^*} + \sum_{k \in \mathcal{N}_{k_t^*}} \epsilon_t \, \widetilde{\alpha}_{k,k_t^*} \mathbf{w}_{k,t}, \quad \text{(A.1)}$$

Compared with Algorithm 2 the consensus weights $\widetilde{\alpha}_{k,\mathcal{S}_t}$ are thus modified as follows:

$$\widetilde{\alpha}_{k,\mathcal{N}_{k_t^*}} = \begin{cases} \sum_{k \in \mathcal{N}_{k_t^*}} (1 - \epsilon_t \, \widetilde{\alpha}_{k,k_t^*}) & k = k_t^* \\ \epsilon_t \, \widetilde{\alpha}_{k,k_t^*} & k \neq k_t^*. \end{cases} \quad \text{(A.2)}$$

In Section IV we provide a numerical comparison between CFAdp weights $\widetilde{\alpha}_{k,\mathcal{S}_t}$ and CFAdp-CS ones $\widetilde{\alpha}_{k,k_t^*}$ as obtained during FL training.

B CFAdp-VPS: Proof of Theorem 1

From the γ-Lipschitz smoothness of $\mathcal{L}(\boldsymbol{\psi}_{i,t})$ in Assumption 1 and Taylor expansion, we have:

$$\mathcal{L}(\boldsymbol{\psi}_{i,t+1}) \leq \mathcal{L}(\boldsymbol{\psi}_{i,t}) + \nabla \mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}} \cdot (\boldsymbol{\psi}_{i,t+1} - \boldsymbol{\psi}_{i,t}) + \frac{\gamma}{2} \|\boldsymbol{\psi}_{i,t+1} - \boldsymbol{\psi}_{i,t}\|^2. \quad \text{(A.3)}$$

The last two terms on the right-hand side of the above inequality are bounded respectively as:

- *Bounding* $\|\boldsymbol{\psi}_{i,t+1} - \boldsymbol{\psi}_{i,t}\|^2$: By the definition of the global aggregation for $\boldsymbol{\psi}_{i,t+1}$, we have:

$$\|\boldsymbol{\psi}_{i,t+1} - \boldsymbol{\psi}_{i,t}\| = \mathbb{E}_{k \in \mathcal{S}_t}[\|\mathbf{w}_{i,t+1} - \boldsymbol{\psi}_{i,t}\|]. \quad \text{(A.4)}$$

By following SGD optimization, for each term within the expectation in the right hand side of (A.4), we have:

$$\mathbf{w}_{i,t+1} = \boldsymbol{\psi}_{i,t} - \eta \nabla \mathcal{L}_k(\boldsymbol{\psi}_{i,t}). \quad \text{(A.5)}$$

Therefore,

$$\begin{aligned} \|\boldsymbol{\psi}_{i,t+1} - \boldsymbol{\psi}_{i,t}\|^2 &= (\mathbb{E}_{k \in \mathcal{S}_t}[\|\mathbf{w}_{i,t+1} - \boldsymbol{\psi}_{i,t}\|])^2 \\ &= \eta^2 (\mathbb{E}_{k \in \mathcal{S}_t}[\|\nabla \mathcal{L}_k(\boldsymbol{\psi}_{i,t})\|])^2 \\ &\leq \eta^2 \mathbb{E}_{k \in \mathcal{S}_t}[\|\nabla \mathcal{L}_k(\boldsymbol{\psi}_{i,t})\|^2], \end{aligned} \quad \text{(A.6)}$$

where inequality holds because of Cauchy-Schwarz inequality.

- *Bounding* $\nabla \mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}} \cdot (\boldsymbol{\psi}_{i,t+1} - \boldsymbol{\psi}_{i,t})$: Again, by the definition of the global aggregation for $\boldsymbol{\psi}_{i,t+1}$ and (A.5) we have:

$$\begin{aligned} &\nabla \mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}} \cdot (\boldsymbol{\psi}_{i,t+1} - \boldsymbol{\psi}_{i,t}) \\ &= -\eta \mathbb{E}_{k \in \mathcal{S}_t}[\nabla \mathcal{L}(\boldsymbol{\psi}_{i,t})^{\mathrm{T}} \cdot \nabla \mathcal{L}_k(\boldsymbol{\psi}_{i,t})]. \end{aligned} \quad \text{(A.7)}$$

The expectation term in (A.7) can be further rewritten as shown in [49] with the following substitutions: $\mathbf{w}_{\mathrm{PS},t}$ becomes $\boldsymbol{\psi}_{i,t}$ and the expectation is defined over all clients $k \in \mathcal{S}_t$.

REFERENCES

[1] P. Vepakomma et al., "No peek: A survey of private distributed deep learning," Dec. 2018, *arXiv:1812.03288*.

[2] J. Konečný, et al., "Federated optimization: Distributed optimization beyond the datacenter," Nov. 2015, *arXiv:1511.03575*.

[3] H. B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. ArSf. Intell. StaSst.*, 2017, pp. 1273–1282.

[4] J. Konečný,, "Federated optimization: Distributed machine learning for on-device intelligence," Oct. 2016, *arXiv:1610.02527*.

[5] B. Camajori Tedeschini, M. Brambilla, and M. Nicoli, "Split consensus federated learning: An approach for distributed training and inference," *IEEE Access*, vol. 12, pp. 119535–119549, Aug. 2024.

[6] L. Italiano, B. C. Tedeschini, M. Brambilla, H. Huang, M. Nicoli, and H. Wymeersch, "A tutorial on 5 G positioning," *IEEE Commun. Surv. Tuts.*, early access, Aug. 23, 2024, doi: 10.1109/COMST.2024.3449031.

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

[8] B. Camajori Tedeschini, G. Kwon, M. Nicoli, and M. Z. Win, "Real-time bayesian neural networks for 6 G cooperative positioning and tracking," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 9, pp. 2322–2338, Sep. 2024.

[9] L. Pu, X. Yuan, X. Xu, X. Chen, P. Zhou, and J. Xu, "Cost-efficient and skew-aware data scheduling for incremental learning in 5 G networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 2, pp. 578–595, Feb. 2022.

[10] H. Gu et al., "Fedaux: An efficient framework for hybrid federated learning," in *Proc. ICC 2022 - IEEE Int. Conf. Commun. Seoul*, Korea, Republic of, May 2022, pp. 195–200.

[11] B. Camajori Tedeschini et al., "Decentralized federated learning for healthcare networks: A case study on tumor segmentation," *IEEE Access*, vol. 10, pp. 8693–8708, 2022.

[12] U. Milasheuski et al., "On the impact of data heterogeneity in federated learning environments with application to healthcare networks," in *Proc. IEEE Conf., Artif. Intell.*, Jun. 2024, pp. 1017–1023.

[13] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1-2, pp. 1–210, 2021.

[14] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[15] J. Perazzone, S. Wang, M. Ji, and K. S. Chan, "Communication-efficient device scheduling for federated learning using stochastic optimization," in *Proc. IEEE INFOCOM 2022 - IEEE Conf. Comput. Commun*. London, United Kingdom, May 2022, pp. 1449–1458.

[16] H. Wu and P. Wang, "Node selection toward faster convergence for federated learning on non-IID data," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3099–3111, Sep.-Oct. 2022.

[17] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang, "Client selection for federated learning with non-IID data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24462–24474, 2021.

[18] W. Guo, R. Li, C. Huang, X. Qin, K. Shen, and W. Zhang, "Joint device selection and power control for wireless federated learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2395–2410, Aug. 2022.

[19] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. ICC 2019-2019 IEEE Int. Conf. Commun.*, Shanghai, China, May 2019, pp. 1–7.

[20] J. Xu, H. Wang, and L. Chen, "Bandwidth allocation for multiple federated learning services in wireless edge networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2534–2546, Apr. 2022.

[21] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *Proc. IEEE INFOCOM 2021 - IEEE Conf. Comput. Commun*. Vancouver, BC, Canada, May 2021, pp. 1–10.

[22] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning in mobile edge networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3606–3621, Dec. 2021.

[23] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM 2019 - IEEE Con. Comput. Commun.*, Apr. 2019, pp. 1387–1395.

[24] X. Ling, R. Li, T. Ouyang, and X. Chen, "Time is gold: A time-dependent incentive mechanism design for fast federated learning," in *Proc. 2022 IEEE/CIC Int. Conf. Commun.*, China, Aug. 2022, pp. 1038–1043.

[25] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[26] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, Mar. 2020.

[27] M. M. Amiri and D. Gündüz,, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[28] Z. Gao, C. Qiu, C. Zhao, Y. Yang, Z. Mo, and Y. Lin, "Fedim: An anti-attack federated learning based on agent importance aggregation," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Oct. 2021, pp. 1445–1451.

[29] A. Bhowmick et al., "Protection against reconstruction and its applications in private federated learning," Jun. 2019, *arXiv:1812.00984*.

[30] Y. Jiang et al., "Improving federated learning personalization via model agnostic meta learning," Jan. 2023, *arXiv:1909.12488*.

[31] V. C. Gogineni, S. Werner, Y. -F. Huang, and A. Kuh, "Communication-efficient online federated learning framework for nonlinear regression," in *Proc. ICASSP 2022-2022 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Singapore, Singapore, May 2022, pp. 5228–5232.

[32] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-IID data," in *Proc. 2020 IEEE Int. Conf. Big Data*, Dec. 2020, pp. 15–24.

[33] P. Han, S. Wang, and K. K. Leung, "Adaptive gradient sparsification for efficient federated learning: An online learning approach," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, Nov. 2020, pp. 300–310.

[34] C. T. Dinh et al., "DONE: Distributed approximate newton-type method for federated edge learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2648–2660, Nov. 2022.

[35] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. Vincent Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[36] C. Zhou et al., "TEA-fed: Time-efficient asynchronous federated learning for edge computing," in *Proc. 18th ACM Int. Conf. Comput. Frontiers. Virtual Event Italy*, May 2021, pp. 30–37.

[37] S. Chen, X. Wang, P. Zhou, W. Wu, W. Lin, and Z. Wang, "Heterogeneous semi-asynchronous federated learning in Internet of Things: A multi-armed bandit approach," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 5, pp. 1113–1124, Oct. 2022.

[38] Z. Wang et al., "Asynchronous federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6961–6978, Sep. 2022.

[39] B. Camajori Tedeschini, S. Savazzi, and M. Nicoli, "A traffic model based approach to parameter server design in federated learning processes," *IEEE Commun. Lett.*, vol. 27, no. 7, pp. 1774–1778, Jul. 2023.

[40] Y. Zhao et al., "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[41] J. Zhang et al., "Fedada: Fast-convergent adaptive federated learning in heterogeneous mobile edge computing environment," *World Wide Web*, vol. 25, no. 5, pp. 1971–1998, Sep. 2022.

[42] K. Tu, S. Zheng, X. Wang, and X. Hu, "Adaptive federated learning via mean field approach," in *Proc. 2022 IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. & Commun. (GreenCom) IEEE Cyber, Phys. & Social Comput. (CPSCom) IEEE Smart Data (SmartData) IEEE Congr. Cybermat. (Cybermatics)*, Espoo, Finland, Aug. 2022, pp. 168–175.

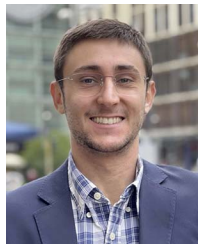
[43] K. Mo, C. Chen, J. Li, H. Xu, and C. J. Xue, "Two-dimensional learning rate decay: Towards accurate federated learning with non-IID data," in *Proc. 2021 Int. Joint Conf. Neural Netw. (IJCNN)*. Shenzhen, China, Jul. 2021, pp. 1–7.

[44] N. Mhaisen, A. A. Abdellatif, A. Mohamed, A. Erbad, and M. Guizani, "Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 55–66, Jan. 2022.

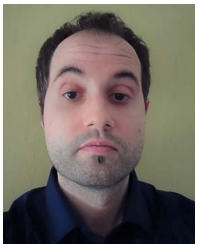
[45] P. Tian, W. Liao, W. Yu, and E. Blasch, "WSCC: A weight-similarity-based client clustering approach for non-IID federated learning," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20243–20256, Oct. 2022.

[46] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL for wireless networks: Cooperative learning mechanism using non-IID data," in *Proc. ICC 2020-2020 IEEE Int. Conf. Commun*, Dublin, Ireland, Jun. 2020, pp. 1–7.

[47] T. Li et al., "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, vol. 2, pp. 429–450.
[48] R. Pathak et al., "Fedsplit: An algorithmic framework for fast federated optimization," in *Adv. Neural Inform. Process. Syst.*, H. Larochelle, Eds., vol. 33. Red Hook, NY, USA: Curran Associates, Mar. 2020, pp. 7057–7066.
[49] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1078–1088, Dec. 2021.
[50] S. Wang et al., "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE INFOCOM 2018 - IEEE Conf. Comput. Commun.*, Honolulu, HI, Apr. 2018, pp. 63–71.
[51] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
[52] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, May 2020.
[53] F. P. -C. Lin, S. Hosseinalipour, S. S. Azam, C. G. Brinton, and N. Michelusi, "Semi-decentralized federated learning with cooperative D2D local model aggregations," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3851–3869, Dec. 2021.
[54] S. Savazzi, M. Nicoli, M. Bennis, S. Kianoush, and L. Barbieri, "Opportunities of federated learning in connected, cooperative, and automated industrial systems," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 16–21, Feb. 2021.
[55] G. Soatti et al., "Distributed signal processing for dense 5 G iot platforms: Networking, synchronization, interference detection and radio sensing," *Ad Hoc Netw.*, vol. 89, pp. 9–21, Jun. 2019.
[56] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
[57] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Trans. Ind. Inform.*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.
[58] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.
[59] S. Savazzi et al., "An energy and carbon footprint analysis of distributed and federated learning," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 248–264, Mar. 2023.
[60] I. Hegeds et al., "Gossip learning as a decentralized alternative to federated learning," in *Distrib. Appl. Interoperable Syst.*, J. Pereira, Eds. Cham: Springer International Publishing, Jun. 2019, vol. 11534, pp. 74–90.
[61] S. Savazzi, M. Nicoli, V. Rampa, and S. Kianoush, "Federated learning with mutually cooperating devices: A consensus approach towards server-less model optimization," in *Proc. ICASSP 2020-2020 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 3937–3941.
[62] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020.
[63] L. Barbieri, S. Savazzi, and M. Nicoli, "A layer selection optimizer for communication-efficient decentralized federated deep learning," *IEEE Access*, vol. 11, pp. 22155–22173, 2023.
[64] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.
[65] G. Soatti, M. Nicoli, S. Savazzi, and U. Spagnolini, "Consensus-based algorithms for distributed network-state estimation and localization," *IEEE Trans. Signal Inform. Process. Over Netw.*, vol. 3, no. 2, pp. 430–444, Jun. 2017.
[66] D. P. Spanos et al., "Distributed sensor fusion using dynamic consensus," in *Proc. IFAC World Congr. Citeseer*, 2005, pp. 1–6.
[67] A. Giuseppi et al., "A weighted average consensus approach for decentralized federated learning," *Mach. Intell. Res.*, vol. 19, no. 4, pp. 319–330, Jul. 2022.
[68] Z. Chen et al., "DACFL: Dynamic average consensus-based federated learning in decentralized sensors network," *Sensors*, vol. 22, no. 9, Apr. 2022, Art. no. 3317.
[69] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.
[70] S. Savazzi, S. Kianoush, V. Rampa, and M. Bennis, "A joint decentralized federated learning and communications framework for industrial networks," in *Proc. IEEE 25th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw. (CAMAD)*, Pisa, Italy, Sep. 2020, pp. 1–7.
[71] D. P. Kingma et al., "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representation*, Dec. 2015, pp. 1–15.
[72] O. Shamir et al., "Communication efficient distributed optimization using an approximate newton-type method," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1000–1008.
[73] N. Shoham et al., "Overcoming forgetting in federated learning on non-IID data," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2019.
[74] D. Mackay and D. J. C. Mackay, "Variational gaussian process classifiers," *IEEE Trans. Neural Netw.*, vol. 11, no. 6, pp. 1458–1464, Nov. 2000.
[75] C. Chen et al., "GIFT: Toward accurate and efficient federated learning with gradient-instructed frequency tuning," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 902–914, Apr. 2023.
[76] S. Park and W. Choi, "Regulated subspace projection based local model update compression for communication-efficient federated learning," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 964–976, Apr. 2023.
[77] C.-H. Hu, Z. Chen, and E. G. Larsson, "Scheduling and aggregation design for asynchronous federated learning over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 874–886, Apr. 2023.
[78] J. Jin et al., "Accelerated federated learning with decoupled adaptive optimization," in *Proc. 39th Int. Conf. Mach. Learn*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, Eds., vol. 162. PMLR, 17-23 Jul. 2022, pp. 10298–10322.
[79] T. Jahani-Nezhad, M. A. Maddah-Ali, S. Li, and G. Caire, "SwiftAgg : Achieving asymptotically optimal communication loads in secure aggregation for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 977–989, Apr. 2023.
[80] "Mqtt v3.1 protocol specification," (n.d.) [Online]. Available: https://mqtt.org
[81] "Jetson nano developer kit," Accessed: Jan. 03, 2021. [Online]. Available: https://tinyurl.com/52mdbjzh
[82] Y. LeCun et al., "MNIST handwritten digit database," ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist, vol. 2, 2010.
[83] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Master's thesis, Department of Computer Science, University of Toronto, 2009.
[84] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
[85] H. Wu et al., "Cvt: Introducing convolutions to vision transformers," in *Proc. 2021 IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*. Montreal, QC, Canada, Oct. 2021, pp. 22–31.
[86] M. Luo et al., "No fear of heterogeneity: Classifier calibration for federated learning with non-IID data," in *Adv. Neural Inform. Process. Syst*, M. Ranzato, Eds., Curran Associates, Inc., 2021, vol. 34, pp. 5972–5984.
[87] T. Lin et al., "Ensemble distillation for robust model fusion in federated learning," in *Adv. Neural Inform. Process Syst.*, H. Larochelle, Eds., Curran Associates, Inc., 2020, vol. 33, pp. 2351–2363.



Bernardo Camajori Tedeschini (Member, IEEE) received the B.Sc. (Hons.) degree in computer science engineering, the M.Sc. (Hons.) degree in telecommunications engineering, and the Ph.D. (Hons.) degree in information technology from the Politecnico di Milano, Milan, Italy, in 2019, 2021, and 2024, respectively. He was a Visiting Ph.D. Researcher with the Wireless Information and Network Sciences Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2023 and 2024. In 2021, he was a Visiting Research Scientist with CERN, Geneva, Switzerland, where he worked on the CAFEIN Project, focusing on the development and deployment of a federated network platform. His research interests include cooperative machine learning in distributed systems and localization methods. Dr. Camajori Tedeschini was the recipient of a Ph.D. grant from the Italian Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) and of the Roberto Rocca Doctoral Fellowship, jointly awarded by MIT and Politecnico di Milano. He was honored with the Best Freshmen Prize from Politecnico di Milano in 2017.



Stefano Savazzi (Member, IEEE) received the M.Sc. and Ph.D. (Hons.) degrees in ICT from the Politecnico di Milano, Milan, Italy, in 2004 and 2008, respectively. In 2012, he joined the Institute of Electronics, Computer and Telecommunication Engineering (IEIIT), Consiglio Nazionale delle Ricerche (CNR), as a Researcher, and Senior Researcher since 2023. He was a Visiting Researcher with Uppsala University, Uppsala, Sweden, in 2005 and University of California at San Diego, San Diego, CA, USA, in 2007. He has coauthored more than 110 scientific publications (Scopus). His research interests include distributed signal processing, machine learning and networking aspects for the Internet of Things, radio localization and vision technologies. Dr. Savazzi won the Dimitris N. Chorafas Foundation Award in 2008. He is Principal investigator for CNR in Horizon EU Projects Holden and TRUSTroke. He is also an Associate Editor for *Frontiers in Communications and Networks, Wireless Communications* and *Mobile Computing,* and Lead Guest Editor of the Special Issue on Radio Sensing and Sensor Networks in Sensors (MDPI).

Monica Nicoli (Senior Member, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees in communication engineering from Politecnico di Milano, Milan, Italy, in 1998 and 2002, respectively. She was a Visiting Researcher with ENI Agip, from 1998 to 1999, and Uppsala University, Uppsala, Sweden, in 2001. In 2002, she joined Politecnico di Milano as a Faculty Member. She is currently an Associate Professor of telecommunications with the Department of Management, Economics and Industrial Engineering. Her research interests include signal processing, machine learning, and wireless communications, with emphasis on smart mobility and Internet of Things (IoT) applications. She was the recipient of the Marisa Bellisario Award, in 1999, and co-recipient of the best paper awards of EuMA Mediterranean Microwave Symposium in 2022, IEEE Symposium on Joint Communications and Sensing, in 2021, IEEE Statistical Signal Processing Workshop, in 2018, and IET Intelligent Transport Systems journal, in 2014. She served as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS from 2020 to 2024, for *EURASIP Journal on Wireless Communications and Networking*, from 2010 to 2017, and Lead Guest Editor of the Special Issue on Localization in Mobile Wireless and Sensor Networks, in 2011.