Dissertation

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy by

Sorbonne University

Author:

Mohamad Mestoukirdi

Defended on the 4th of December, 2023

Reliable and Communication-Efficient Federated Learning for Future Intelligent Edge Networks

Thesis Directors:
David GESBERT (Eurecom, France)
Nicolas GRESSET (Mitsubishi Electric R&D Centre Europe, France)

Jury President:
M. Jérôme HÄRRI

Jury

M. Deniz GUNDUZ, Professor, Imperial College London, United Kingdom	Reviewer
Mrs. Zheng CHEN, Associate Professor HDR, Linköping University, Sweden	Reviewer
M. Giovanni NEGLIA, Research Director, Inria Center of Université Côte d'Azur, France	Examiner
M. Jérôme HÄRRI, Professor, Eurecom, France	Examiner
M. José Mairton BARROS DA SILVA Jr., Assistant Professor, Uppsala University, Sweden	Examiner
M. Qianrui Li, Doctor, CICT Mobile Communications Technology, China	Invited Guest

T H È S E

Soumise pour l'obtention du grade de Docteur par

Sorbonne Université

Auteur:

Mohamad Mestoukirdi

Soutenue le 4 Décembre, 2023

Apprentissage Fédéré Fiable et Efficace en Termes de Communication Dans les Futurs Réseaux Intelligents

Directeurs de Thèse:
David GESBERT (Eurecom, France)
Nicolas GRESSET (Mitsubishi Electric R&D Centre Europe, France)

Président du Jury:
M. Jérôme HÄRRI

Jury

M. Deniz GUNDUZ, Professeur, Imperial College London, Royaume-Uni	Rapporteur
Mme. Zheng CHEN, Maître de conférences HDR, Linköping University, Suède	Rapporteuse
M. Giovanni NEGLIA, Directeur de Recherche, Centre Inria de l'Université Côte d'Azur, France	Examinateur
M. Jérôme HÄRRI, Professeur, Eurecom, France	Examinateur
M. José Mairton BARROS DA SILVA Jr., Professeure Adjointe, Université d'Uppsala, Suède	Examinateur
M. Qianrui LI, Docteur, CICT Mobile Communications Technology, Chine	Invité

To my dear parents,
for their endless love and support.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, David Gesbert, Nicolas Gresset, and Qianrui Li, for their guidance and support throughout the completion of this dissertation. Their crystalline view on challenges and expertise in the field of research have been invaluable to me, and their patience and encouragement have helped me to overcome the various challenges that I encountered during the thesis.

In addition, I would like to extend my gratitude to my colleagues at EURECOM and Mitsubishi Electric. During my journey, I had the chance to meet and work with remarkable minds. Particularly, the past members of the M3 group and the CSE group. I am grateful for the opportunity to have worked with such a talented people.

Furthermore, I would like to thank Mitsubishi Electric for the financial support, which has enabled me to pursue my research interests and complete this dissertation.

Finally, I would like to convey my sincere gratitude to my loved ones, my family, friends, and Viktoria, some of whom are far away in Lebanon, for their unwavering support and encouragement throughout my academic journey. They have been a constant source of love and motivation especially during the most challenging times.

Abstract

In the realm of future 6G wireless networks, integrating the intelligent edge through the advent of AI signifies a momentous leap forward, promising revolutionary advancements in wireless communication. This integration fosters a harmonious synergy, capitalizing on the collective potential of these transformative technologies. Central to this integration is the role of federated learning (FL), a nascent decentralized learning paradigm. Particularly, federated learning, aided by communication networks, facilitates collaborative training of machine learning models across networks of edge devices by leveraging their distributed datasets through an iterative process of decentralized on-device model optimization and centralized aggregation of model updates. This circumvents the need to migrate sensitive user raw data to centralized servers for training or inference, overcoming critical privacy barriers. Additionally, federated learning allows the development of high-accuracy models that harness subtle patterns across vastly more total data than any single device could provide alone. By embracing this paradigm, 6G networks can unlock a myriad of benefits for both wireless networks and edge devices. On one hand, wireless networks stand to benefit from federated learning ability to develop data-driven solutions surpassing the limitations of traditional model-driven approaches, by facilitating cooperative training spanning different cellular networks or other wireless technology domains. This empowers future networks through embedded inference to adapt, optimize performance, and enhance network efficiency dynamically. On the other hand, edge devices benefit from personalized experiences and tailored solutions, catered to their specific requirements. Specifically, edge devices will experience improved performance and reduced latency through localized decision-making, real-time processing, and reduced reliance on centralized infrastructure.

While federated learning has the potential to revolutionize future networks by offering limitless opportunities for distributed model training, its widespread adoption is contingent upon addressing several significant challenges that impede its full utilization. The two most prominent hurdles in federated learning are the communication overhead associated with exchanging model updates over communication channels, which can be prohibitive when dealing with a large number of devices, and ensuring the reliability of trained models in heterogeneous settings, where data distribution and computational resources vary greatly.

In the first part of the thesis, we tackle the predicament of statistical heterogeneity in federated learning stemming from divergent data distributions among devices datasets. Rather than training a conventional one-model-fits-all, which often performs poorly in non-IID settings, we propose user-centric set of rules that produce personalized models tailored to each user objectives. To mitigate the extra communication overhead associated with training distinct personalized model for each user, users are partitioned into clusters based

on their objectives similarity. This enables collective training of cohort-specific personalized models. As a result, the total number of personalized models trained is reduced. This reduction lessens the consumption of wireless resources required to transmit model updates across bandwidth-limited wireless channels.

In the second part, our focus shifts towards integrating the Internet of Things (IoT) remote devices into the intelligent edge by leveraging Unmanned Aerial Vehicles (UAVs) as a federated learning orchestrator. While previous studies have extensively explored the potential of UAVs as flying base stations or relays in wireless networks, their utilization in facilitating model training is still a relatively new area of research. In this context, we leverage the UAV mobility to bypass the unfavorable channel conditions in rural areas and establish learning grounds to remote IoT devices. However, UAV deployments poses challenges in terms of scheduling and trajectory design. To this end, a joint optimization of UAV trajectory, device scheduling, and the learning performance is formulated and solved using convex optimization techniques and graph theory.

In the third and final part of this thesis, we take a critical look at the communication overhead imposed by federated learning on wireless networks. While compression techniques such as quantization and sparsification of model updates are widely used, they often achieve communication efficiency at the cost of reduced model performance. Accordingly, we employ over-parameterized random networks to approximate target networks, through parameter pruning rather than direct optimization to overcome this limitation. This approach has been demonstrated to require transmitting no more than a single bit of information per model parameter under a satisfactory learning performance level. We show that state of the art (SoTA) methods fail to capitalize on the full attainable advantages in terms of communication efficiency by relying on consistent loss objectives. Therefore, we propose a regularized loss function which considers the entropy of transmitted updates, resulting in notable improvements to communication and memory efficiency during federated training on resource-constrained edge devices with slight generalization performance loss in some cases.

Résumé

Dans le domaine des futurs réseaux sans fil 6G, l'intégration de la périphérie intelligente grâce à l'avènement de l'IA représente un bond en avant considérable, promettant des avancées révolutionnaires en matière de communication sans fil. Cette intégration favorise une synergie harmonieuse, capitalisant sur le potentiel collectif de ces technologies transformatrices. Au cœur de cette intégration se trouve le rôle de l'apprentissage fédéré, un paradigme d'apprentissage décentralisé qui préserve la confidentialité des données tout en exploitant l'intelligence collective des appareils interconnectés. En adoptant l'apprentissage fédéré, les réseaux 6G peuvent débloquer une myriade d'avantages à la fois pour les réseaux sans fil et pour les appareils périphériques. D'une part, les réseaux sans fil acquièrent la capacité de fournir des solutions basées sur les données, dépassant les limites des approches traditionnelles basées sur des modèles. En particulier, l'exploitation des données en temps réel permettra aux réseaux 6G de s'adapter, d'optimiser les performances et d'améliorer l'efficacité du réseau de manière dynamique. D'autre part, les appareils périphériques bénéficient d'expériences personnalisées et de solutions sur mesure, adaptées à leurs besoins spécifiques. Plus précisément, les appareils périphériques bénéficieront de meilleures performances, d'une latence réduite et d'une efficacité énergétique accrue, ce qui renforcera leurs capacités. Simultanément, la périphérie intelligente permet aux dispositifs de périphérie de prendre des décisions localisées, d'effectuer des traitements en temps réel et de réduire la dépendance à l'égard de l'infrastructure centralisée. Cette convergence des futurs réseaux 6G et de l'IA révolutionne les réseaux sans fil et renforce l'intelligence périphérique, les propulsant dans une ère de connectivité, d'intelligence et d'innovation sans précédent.

L'apprentissage fédéré a le potentiel de révolutionner les réseaux du futur en offrant des possibilités illimitées pour la formation de modèles distribués. possibilités illimitées pour la formation de modèles distribués, son adoption à grande échelle dépend de la résolution de plusieurs défis importants qui empêchent sa pleine utilisation. son adoption à grande échelle dépend de la résolution de plusieurs problèmes importants qui empêchent sa pleine utilisation. Les deux obstacles les plus Les deux obstacles les plus importants de l'apprentissage fédéré sont le surcoût de communication associé à l'échange de mises à jour de modèles sur les canaux de communication, et le coût de la formation. l'échange de mises à jour de modèles sur les canaux de communication, qui peut être prohibitif lorsqu'il s'agit d'un grand nombre d'appareils de communication, qui peut être prohibitif lorsqu'il s'agit d'un grand nombre d'appareils, et la garantie de la fiabilité des dans des environnements hétérogènes, où la distribution des données et les ressources informatiques varient considérablement.

Dans la première partie de la thèse, nous nous attaquons au problème de l'hétérogénéité statistique dans l'apprentissage fédéré, qui découle des distributions de données divergentes

entre les ensembles de données des dispositifs. Plutôt que d'entraîner un modèle unique conventionnel, qui donne souvent de mauvais résultats avec des données non identifiées, nous proposons un ensemble de règles centrées sur l'utilisateur qui produisent des modèles personnalisés adaptés aux objectifs de chaque utilisateur. Pour atténuer la surcharge de communication prohibitive associée à l'apprentissage d'un modèle personnalisé distinct pour chaque utilisateur, les utilisateurs sont répartis en groupes sur la base de la similarité de leurs objectifs. Cela permet l'apprentissage collectif de modèles personnalisés spécifiques à la cohorte. En conséquence, le nombre total de modèles personnalisés formés est réduit. est réduit. Cette réduction diminue la consommation de ressources sans fil nécessaires à la transmission des mises à jour de modèles sur des canaux sans fil à bande passante limitée.

Dans la deuxième partie, nous nous concentrons sur l'intégration des dispositifs à distance de l'IdO dans la périphérie intelligente en exploitant les véhicules aériens sans pilote en tant qu'orchestrateur d'apprentissage fédéré. Alors que des études antérieures ont largement exploré le potentiel des drones en tant que stations de base volantes ou relais dans les réseaux sans fil, leur utilisation pour faciliter l'apprentissage de modèles est encore un domaine de recherche relativement nouveau. Dans ce contexte, nous tirons parti de la mobilité des drones pour contourner les conditions de canal défavorables dans les zones rurales et établir des terrains d'apprentissage pour les dispositifs IoT distants. Cependant, les déploiements de drones posent des défis en termes de planification et de conception de trajectoires. À cette fin, une optimisation conjointe de la trajectoire du drone, de l'ordonnancement du dispositif et de la performance d'apprentissage est formulée et résolue à l'aide de techniques d'optimisation convexe et de la théorie des graphes.

Dans la troisième et dernière partie de cette thèse, nous jetons un regard critique sur la surcharge de communication imposée par l'apprentissage fédéré sur les réseaux sans fil. Bien que les techniques de compression telles que la quantification et la sparsification des mises à jour de modèles soient largement utilisées, elles permettent souvent d'obtenir une efficacité de communication au prix d'une réduction de la performance du modèle. Pour surmonter cette limitation, nous utilisons des réseaux aléatoires sur-paramétrés pour approximer les réseaux cibles par l'élagage des paramètres plutôt que par l'optimisation directe. Il a été démontré que cette approche ne nécessite pas la transmission de plus d'un seul bit d'information par paramètre du modèle. Nous montrons que les méthodes SoTA ne parviennent pas à tirer parti de tous les avantages possibles en termes d'efficacité de la communication en utilisant cette approche. En conséquence, nous proposons une fonction de perte régularisée qui prend en compte l'entropie des mises à jour transmises, ce qui se traduit par des améliorations notables de l'efficacité de la communication et de la mémoire lors de l'apprentissage fédéré sur des dispositifs périphériques à ressources limitées, sans sacrifier la précision.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

5G	Fifth Generation
6G	Sixth Generation
ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
AP	Access Point
BS	Base Station
CoV	Coefficient of Variation
CFL	Clustered Federated Learning
CNN	Convolutional Neural Network
DL	Downlink
FL	Federated Learning
FedAVG	Federated Averaging
IoT	Internet of Things
IID	Independent and Identically Distributed
LoS	Line-of-Sight
ML	Machine Learning
NLoS	Non-Line-of-Sight
PER	Packet Error Rate
PS	Parameter Server
SGD	Stochastic Gradient Descent
SNR	Signal-to-Noise-Ratio

TDMA Time Division Multiple Access

UAV Unmanned Aerial Vehicle

UL Uplink

Chapter 1
Introduction

The ubiquitous evolution of 5G (Fifth Generation) and upcoming 6G (Sixth Generation) networks are expected to enable a new paradigm of intelligent edge computing. As network speeds increase dramatically and latency decreases, more processing and intelligence can be pushed to the edge rather than relying solely on cloud data centers [1–3]. This allows for real-time data processing and decision-making closer to the end-user or device, adhering to future network requirements. In newly deployed 5G networks, multi-access edge computing cloud allows compute and storage resources to be deployed at the network's edge [4]. Looking ahead to 6G, some of the proposed visions promote for highly distributed intelligent networks with AI enabled (Artificial Intelligence-enabled) processing directly embedded into end-users edge devices. The integration of AI capabilities and edge devices encompasses tasks such as training and inferring machine learning models locally, enabling innovative applications such as industrial automation [5], autonomous vehicles [6], augmented reality [7], and other services requiring ultra-low latency. For instance, 3GPP (The 3rd Generation Partnership Project) has been pushing towards integrating AI in wireless networks and the edge. In Release 18, various techniques have been studied to enhance the performance and efficiency of wireless networks, including beam management, channel state information feedback, and positioning accuracy [8].

The rapidly growing number of connected devices further motivates this shift. As shown in Fig. 1.1, the estimated number of worldwide IoT devices has increased over 140% from 2018 to 2023, reaching nearly 15 billion [9,10]. Accommodating and leveraging this massive influx of such sensory and computationally capable edge devices is only feasible with the help of distributed edge intelligence.

In recent years, a notable paradigm shift has occurred in the integration of machine learning models within edge devices. The conventional approach of transmitting data from devices to centralized servers for model training or inference, and subsequently deploying the trained models or inference decisions to the edge, has witnessed a decline in favor [11, 12]. This change can be primarily attributed to two pivotal factors: privacy and communication overhead.

The preservation of privacy has emerged as a paramount concern in the era of data-driven technologies. Traditional model training and data inference approaches has raised substantial apprehensions regarding the protection of sensitive personal information [13].

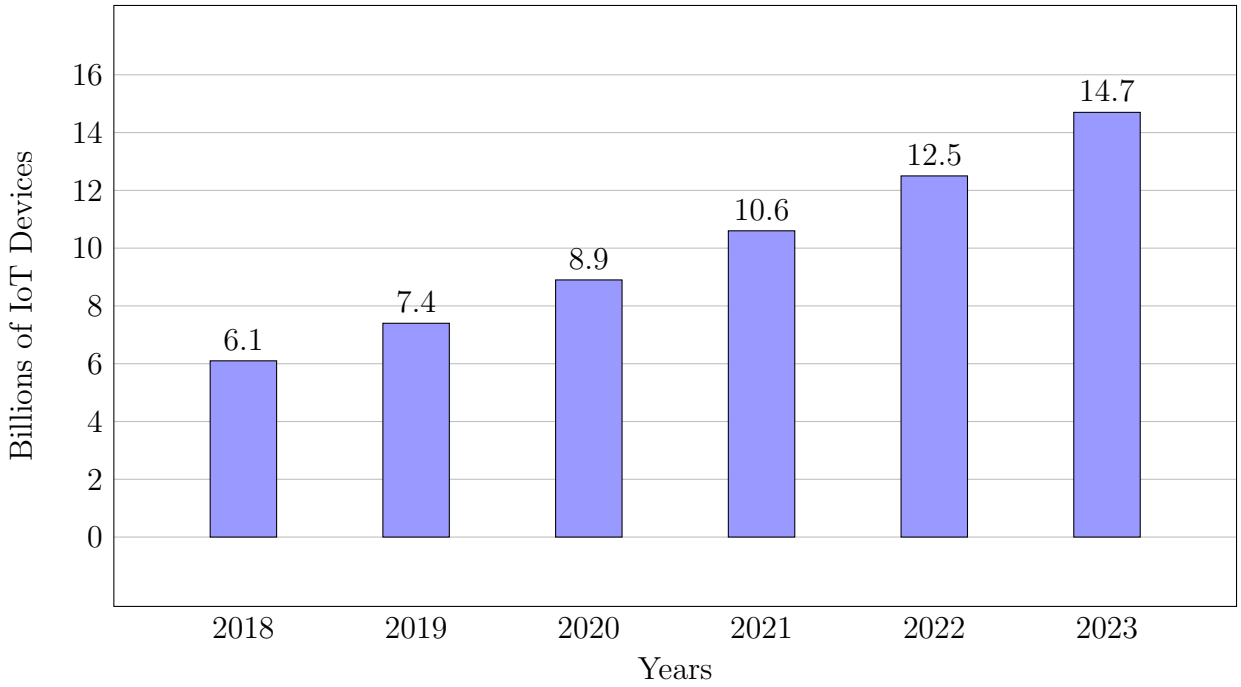


Figure 1.1: The Rise of Connected IoT

This concern becomes particularly pronounced in domains that handle confidential data, such as healthcare applications, where wearable devices collect patient-specific information. Consequently, safeguarding data privacy has become a pivotal consideration in the design and deployment of machine learning methodologies.

Additionally, the communication overhead associated with transferring data between devices and central servers has gained attention as a crucial challenge. The transmission of substantial amounts of data for centralized training or inference over networks not only introduces latency issues but also imposes a burden on network resources. In scenarios where real-time data collection and decision-making are paramount, such as industrial environments relying on IoT devices for sensor data acquisition, the inefficiencies introduced by the excessive communication overhead can impede the overall system performance.

Furthermore, this shift in paradigm gains further momentum due to the widespread availability of powerful edge devices equipped with significant computational capabilities and sensory functionalities. These advancements have made it feasible to perform data acquisition, processing, and model training directly on the device.

However, a challenge arises when conducting on-device local training. Edge devices such as smartphones typically grapple with limited datasets sampled from their immediate environments. The inherent limitations of the datasets can be interpreted through the lens of their representational quality and size. Representational quality, which is synonymous with

the expressiveness of the datasets, measures their effectiveness in training a model specifically designed for each device task. This effectiveness relies on factors such as the sampling resolution and the sensory capabilities of the devices. These data limitations impede the potential of local training to generate models that exhibit robust generalization performance over the devices' tasks.

1.1 Federated Learning

Due to the limitations mentioned earlier, Federated Learning (FL) surfaced as a potential solution to overcome these challenges [14]. FL is a nascent sub-field of machine learning that provides devices (also known as *clients* or *users*) with the opportunity for collaborative training of models, supervised by a central *orchestrator*, without the need to share their raw training data. Instead, FL allows edge devices to collectively train models that exhibit improved generalization and performance, while promoting data privacy by design. The diversity of data across devices enables the trained models to capture a broader range of patterns and variations, enhancing their ability to handle diverse user preferences and generalize better compared to locally learned models. This distributed learning paradigm also reduces the dependence on centralized servers, enabling real-time processing and context-aware decision-making directly on the edge. This facilitates faster response times, reduced latency, and improved user experience in various applications. In its prototypical configuration [15], FL involves distributed training executed iteratively over several *communication rounds*. A communication round refers to a single iteration or cycle of the training process between the orchestrator and the participating devices. During a communication round, the following steps typically occur:

1. **Model Distribution**: The central server sends a global ML model to a selected subset of devices over a downlink (DL) channel.

2. **Local Model Training**: Each device performs local training using its own local dataset and the received global model. The device optimizes its local model over its local data.

3. **Uplink Model Transmission**: The locally trained models from the participating devices are sent back to the central server over an uplink (UL) communication channel.

4. **Model Aggregation**: The central server aggregates the received models from the edge devices to produce an updated global model. The updated global model is then distributed to a new set of devices for the next communication round.

The training typically involves multiple communication rounds to iteratively refine and optimize the global model till convergence. An illustration of the FL process is given in Fig. 1.2.

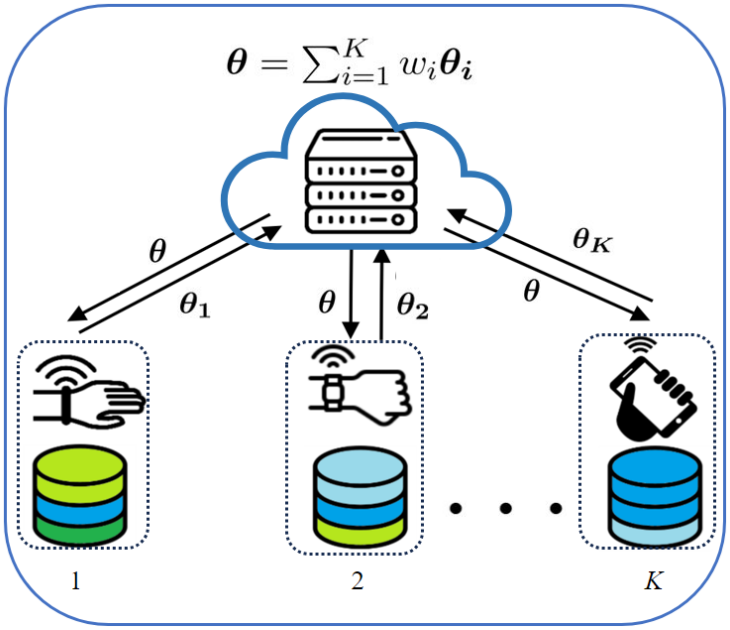


Figure 1.2: Federated Learning: decentralized model optimization and centralized aggregation.

FL training can be split into two main settings: cross-silo and cross-device training. Cross-silo training in FL refers to the process of training a machine-learning model on data from multiple silos or domains. Each silo represents a separate entity or organization (e.g. hospitals or banks) that has its data and wants to collaborate with other silos to train a shared model. All participating data silos are reliable and are almost always available during training. Training allows these silos to work together to train a model that is more accurate and robust than any individual silo could achieve on its own. On the other hand, cross-device training in FL involves training a machine-learning model using data from multiple devices, such as smartphones, smart home devices, or IoT devices. In this scenario, each device represents a separate data source with a relatively small volume of data that contributes to the training of a shared model. These devices are naturally less reliable due to factors like availability, poor network connectivity, and hardware failures.

By virtue of its inherent privacy guarantees, FL has been adopted by several major industrial companies. For instance, Nvidia applied FL across various domains such as medical imaging, and genetics research [16]. Additionally, Apple employs FL in the development of biometric identification systems like Face ID and voice commands for digital assistants

like Siri [17]. One prominent example is Google's keyboard app, Gboard [18], which utilizes FL to improve its language model without compromising user data privacy. By leveraging collaborating among multiple devices, Gboard can train a shared model that adapts to users' typing habits and preferences, enhancing the accuracy of its predictions. Another notable application is found in wireless communication, where there is a growing interest to complement the traditional model-driven design approaches with data-driven solutions [19]. The traditional methods often rely on idealized models that are insufficient for capturing the intricate complexities of real-world scenarios. These models are frequently based on simplifying assumptions that do not accurately reflect the practical realities, which can limit their effectiveness in addressing the challenges of wireless communication [20–23].

Realizing the full potential of federated learning necessitates addressing several critical challenges that arise mainly in practical cross-device deployment.

For instance, communication-efficient aggregation of model updates from participating devices over bandwidth-limited wireless networks is imperative to ensure feasible and scalable FL. Furthermore, heterogeneous hardware, and data distributions across devices must also be reconciled to enable effective federated training. In section 1.4, we delve deeper into each of these key issues, while highlighting the main challenges that this thesis targets.

1.2 Federated Learning System Modelling

The standard objective of FL [15, 24] is to find a global model $\theta \in \mathbb{R}^d$ that minimizes the weighted loss of the K devices in the system, over their local data distribution $\{P_k\}_{k=1}^K$:

$$\min_{\theta \in \mathbb{R}^d} \left[L(\theta) \triangleq \sum_{k=1}^{K} w_k \ell_k(\theta) \right], \tag{1.1}$$

where $\{\ell_k\}$ are the devices' loss functions and $\{w_k\}$ denote their corresponding weights, such that $\sum_k w_k = 1$. The local losses can be defined as:

$$\ell_k(\theta) \triangleq \mathbb{E}_{x \sim P_k} \left[\ell_k(\theta, x) \right], \tag{1.2}$$

where $\mathbb{E}[\cdot]$ denotes the mathematical expectation. In the case where the devices are endowed with finite datasets denoted $\{\mathcal{D}_k\}$ sampled from the local data distributions with a cardinality $\{|\mathcal{D}_k| < \infty\}$, then the global objective in (1.1) is termed the empirical risk minimization (ERM). Accordingly, the local losses can be written as :

$$\ell_k(\theta) = \frac{1}{|\mathcal{D}_k|} \sum_{j=1}^{|\mathcal{D}_k|} \ell_k(\theta, x_{k,j}), \tag{1.3}$$

where $|\mathcal{D}_k|$ denotes the dataset cardinality of device k, and $x_{k,j}$ denotes the j^{th} sample of $\mathcal{D}_k$. Commonly [24–26], the weights found in (1.1) are based on factors like the devices' dataset sizes, and are given by :

$$w_i = \frac{|\mathcal{D}_i|}{\sum_k |\mathcal{D}_k|}. \tag{1.4}$$

In this case, the objective is to train a model parameterized by θ to minimize the weighted losses, across the union of the datasets of the entire system denoted as $\mathcal{D} = \bigcup_k \mathcal{D}_k$. This dataset is sampled from the mixture of distributions denoted by $P = \sum_k w_k P_k$. In this scenario, the assumption is that all devices will encounter data sampled from the target distribution P. Consequently, a key concern is that the model discovered should exhibit effective generalization over P.

In certain scenarios, users may require a personalized experience [27]. Prototypical federated optimization (1.1) aggregates updates from diverse devices to train a single model. However, severe distributional divergences across the devices' target distributions P_k can render a one-model-fits-all approach ineffective. Instead, tailored models catering to specific user preferences or even individual users may be necessary. Nonetheless, accommodating personalized experiences presents an additional challenge: how can statistical divergences be inferred without direct data access, while preserving privacy? This question remains pertinent today, given the various forms of heterogeneity that can exist across devices' data. Ultimately, it is the responsibility of the training service provider to ensure that the chosen training algorithm can effectively handle diverse data distributions while maintaining data privacy.

1.3 Federated Optimization

In order to address problem (1.1), it is imperative to recognize that the global gradient can be expressed as the weighted average of the local gradients:

$$\nabla L(\theta) = \sum_k w_k \nabla \ell_k(\theta). \tag{1.5}$$

Once the global gradient has been computed, it can be utilized to optimize the objective in (1.1), by applying Gradient Descent (`GD`) during each communication round t, according to the following update rule:

$$\theta(t+1) = \theta(t) - \eta_t \nabla L(\theta(t)), \tag{1.6}$$

where $\eta^t > 0$ represents the learning step size that may vary across different communication rounds.

Within a FL setting, wherein devices have access to their local data, and receive the global model $\theta(t)$ in the DL from the orchestrator, local GD can be employed to compute the local gradients. Considering the star configuration prevalent in federated learning, devices transmit their computed local gradients in the UL to the orchestrator. These gradients are subsequently weighted and aggregated to yield the global update described in (1.5). This global update is then applied to the global model according to (1.6). The updated global model is subsequently sent to all devices to initiate the subsequent communication round. The training process concludes when convergence is achieved, an indication that the global gradient norm approaches zero within a small margin.

Despite its theoretical applicability and being extensively researched, GD is not commonly used in practical FL settings. This is primarily attributed to the computational overhead it imposes, as it requires the computation of full local gradients [28]. This computational requirement can pose challenges, particularly for devices with limited resources. Moreover, GD involves a single local update step per communication round, which leads to a relatively slow convergence rate. Consequently, numerous communication rounds between devices and the central server are necessary until convergence is achieved. This significantly adds to the communication overhead, particularly in networks with limited resources. Instead, stochastic and adaptive variants are used, such as SGD and ADAM [29].

1.4 Challenges

Given the applicability of FL in cross-device settings, supported by the devices' rising capabilities and pervasiveness, the use of FL brings about several critical issues that are necessary to address to realize its full potential. Specifically, the communication efficiency during the federated process over wireless networks requires optimization. Also, devices exhibiting hardware, software, and data heterogeneity must be reconciled. Additionally, even with FL privacy provisions, residual privacy concerns persist that warrant further technical amelioration. In the following subsections, each of these key challenges is explored in more detail, while elucidating our contributions towards tackling them.

1.4.1 Communication Bottleneck

The communication bottleneck presents a substantial challenge within the context of FL, mainly when applied to resource-constrained edge networks. In a cross-device setting, FL necessitates the frequent aggregation of model updates from numerous participating devices. However, transmitting complete model updates across wireless networks with limited bandwidth can often prove to be unfeasible. The sizes of raw model updates, particularly for large deep learning models, can reach hundreds of megabytes, leading to network congestion.

This can impede user participation in a federated system, prompting users to withdraw from training due to wireless resource limitations [30].

Several approaches have been proposed in the literature to mitigate the communication bottleneck of FL in both the UL and DL directions. In the UL, dedicated point-to-point connections are established between individual users and the orchestrator to transmit the model updates. However, in the DL, updates are transmitted to multiple users via a multicast or broadcast channel when they are all served by a single cell, a common scenario in industrial environments utilizing private cellular networks. Alternatively, point-to-point communication links are employed in settings where devices are served by local Wi-Fi connectivity. Notably, the DL overhead can be comparable to that of the UL in the latter scenario, and therefore, should be addressed with equal importance. One set of methods involves compressing updates through sparsification [31–33] and quantization [34, 35] to reduce their size. Nonetheless, this often results in a trade-off with decreased model accuracy. Another set of algorithms focuses on minimizing communication overhead by limiting the number of devices involved in each communication round [36–38]. This is accomplished by selecting reliable devices based on specific criteria, such as their wireless channel condition, battery status, and time zone. This thesis examines communication-efficient federated learning in Chapters 2 and 4, proposing techniques to mitigate communication bottlenecks without sacrificing model performance.

While communication-efficient techniques help mitigate bottlenecks for bandwidth limited devices, more fundamental orchestration innovations may be necessary for extremely remote endpoints like IoT devices in rural areas. IoT devices role as data generators positions them as highly suitable candidates for the participation in training models, contributing to the development of an intelligent edge [39]. However, the intermittent connectivity from poor coverage and unreliable channels creates barriers to their participation [40, 41]. Therefore, integrating these devices requires optimizations beyond efficient transmission protocols. Fully decentralized learning may help in this setting, leveraging peer-to-peer communication between the devices without the need for a centralized orchestrator [42, 43]. However, managing coordination in these settings remains an open problem. Modern approaches propose the utilization of UAVs as dynamic relays capable of supervising FL training [44–46]. This strategy is particularly well-suited to situations characterized by challenging connectivity, and on-demand training requirements. Chapter 3 of this thesis thoroughly examines this problem and explores the use of UAVs as orchestrators for FL in remote areas.

1.4.2 Statistical Data Heterogeneity

Statistical heterogeneity poses a significant challenge in FL stemming from the non-IID (non-identically or independently distributed) local datasets of the participating devices [47].

When referencing non-IID data in FL, this typically refers to the underlying differences between the local data distributions P_i and P_j for different devices i and j. This heterogeneity in data is manifested in individual preferences, geographic-specific features capturing localized traits, and time-specific transient dynamics [48]. For instance, IoT devices may differ in their sampling rates or data collection frequencies; some devices might collect data every minute, while others collect data every hour. These variations can affect the temporal resolution of the data and therefore introduce heterogeneity.

There are numerous ways in which data tend to diverge from being identically distributed. If we consider the local distributions supported by $(\mathcal{X}, \mathcal{Y})$, as in supervised learning settings, where $\mathcal{X}$ denotes the input feature space and $\mathcal{Y}$ denotes the ground truth label space, then the local data distributions of the devices are defined as $\{P_k(x, y)\}_{k=1}^{K}$. The most prominent forms of data heterogeneity are [14]:

- Covariate Shift: The conditional probability distributions of the input variables $P_i(x|y)$, may differ across different client populations. For instance, in a collaborative health monitoring FL system, some clients might use high-end medical sensors, while others use simpler wearable devices. The noise among the measurements captured by the different devices alters the input distributions.

- Label Skew: The distribution of labels, represented by $P_i(y)$, varies among different devices. For instance, in a federated sentiment analysis scenario, one client with a substantial dataset may contribute mostly positive reviews, while another client may have more negative reviews. Consequently, the global model could exhibit bias towards the dominant labels present in the device with the larger dataset, considering the weighting scheme mentioned in equation (1.4). As a result, the performance of the global model might be subpar for labels that are under-represented in the datasets.

- Concept Shift: The conditional distribution $P_i(y|x)$ varies across devices datasets. In this case, devices may assign different labels to the same input feature vectors. For instance, labels associated with predicting the next word at different devices (e.g. in Gboard), given a starting phrase (i.e. the input feature), can exhibit variations based on personal choices and regional differences.

In practical scenarios, datasets often comprise a mixture of those effects, and the typical FL objective (1.1) results in suboptimal models when confronted with these effects. While a few algorithms in the literature have successfully dealt with the combined manifestation of these effects across users' data, many approaches have focused on addressing these effects individually while disregarding the interplay among them [14]. Chapter 2 of the thesis investigates the interplay among the mentioned effects that promote heterogeneity and offers a personalized modeling approach as a potential solution.

1.4.3 System Heterogeneity

System heterogeneity in FL leads to divergences in the capabilities and properties of client devices. This heterogeneity spans hardware, connectivity, and availability [47,49]. Hardware differences across mobile, embedded, and server devices induce variable compute parallelism, affecting local training speeds. Connectivity heterogeneity produces differences in communication channel conditions and reliability. Availability heterogeneity arises when stragglers, defined as slower nodes that delay overall execution, emerge due to issues like unreliable participation, power limitations, and mobility. The multitude of hardware, connectivity, and availability divergences together pose significant systems challenges in FL settings. Chapter 3 of the thesis presents an in-depth exploration of the impact of channel heterogeneity among low-powered remote devices in a UAV-orchestrated federated learning setting.

1.4.4 Privacy

Federated Learning was initially developed as a means to address privacy concerns arising from the sharing of users' data with cloud servers for ML model training. By exchanging model updates instead of raw data, FL aimed to provide privacy guarantees to users during the training process. However, recent research has unveiled potential vulnerabilities wherein adversaries could exploit the model updates to infer the content of user datasets [50, 51]. To address this issue, ongoing research focuses on enhancing the privacy of FL through techniques such as secure aggregation, differential privacy, and encryption. Secure aggregation methods aim to ensure that model updates from individual devices are combined in a way that prevents adversaries from extracting information about individual data samples [52, 53]. Differential privacy techniques introduce random noise to the model updates to protect against privacy leakage. Encryption methods can be employed to secure model updates during transmission, preventing unauthorized access and tampering.

While privacy is a vital consideration in FL, this thesis does not explore the difficulties associated with preserving privacy during FL training.

1.5 Thesis Considerations:

We now list some assumptions made across the different chapters of the thesis.

- **Device Participation:** In the following chapters, our analysis is based on the assumption that all devices are available for training, unless stated otherwise. This indicates that the total number of devices available for training is fixed. However, whether all devices are selected for training or whether the updates are successfully transmitted

through the communication channels will depend on the specific assumptions made in each chapter.

- **Local Training:** Local training is a widely used technique in FL to minimize the communication overhead. Accordingly, in all chapters, we assume that devices use gradient stochastic approximation methods, such as stochastic mini-batch gradient descent, to compute the local updates before sending them to the orchestrator in the uplink. Such stochastic approximations help reduce the local computational costs at the devices. The number of local steps is determined by the number of epochs predefined by the orchestrator.

1.6 Contributions and Thesis Outline

This thesis is divided into three distinct parts, each addressing specific challenges that arise from the integration of federated learning in wireless edge networks. The proposed solutions are algorithmic in nature, substantiated by either theoretical or empirical results using information theoretic tools and Monte-Carlo methods respectively. The thesis is structured as follows:

- In the first part of this thesis, we address the problem of data heterogeneity across the devices datasets in federated learning. We propose a user-centric approach that deviates from the traditional one-model-fits-all approach that often performs poorly in these settings [14, 24], and instead offers personalized, fine-tuned models tailored to each user's unique objectives. To mitigate the high communication overhead associated with training the personalized models, we propose a clustering method that groups users with similar objectives, allowing them to collaborate to produce a shared personalized model. Our proposed algorithm demonstrates superior convergence rates compared to several state-of-the-art personalization algorithms. This part is associated with Chapter 2, and is based on the following two published works:

  – [54] Mohamad Mestoukirdi, Matteo Zecchin, David Gesbert, and Qianrui Li. "**User-centric federated learning, Trading off Wireless Resources for Personalization**." *IEEE Transactions on Machine Learning in Communications and Networking, 1:346–359, 2023.*

  – [55] Mohamad Mestoukirdi, Matteo Zecchin, David Gesbert, Qianrui Li and Nicolas Gresset, "**User-Centric Federated Learning**," *IEEE Globecom Workshop, Madrid, Spain, 2021, pp. 1-6, doi: 10.1109/GCWkshps.*

- In the second part of the thesis, we focus on incorporating IoT remote devices in the intelligent edge by leveraging UAVs as federated learning orchestrator. While UAVs have been thoroughly investigated for their potential to act as flying base stations or relays in wireless networks [56,57], the application of UAVs in facilitating model training remains a nascent field. The UAVs deployment offers several advantages, including cost-effective and on-demand deployment which aligns with periodic model training and refinement requirements. Moreover, UAVs mobility enables the establishment of LoS (Line-of-Sight) communication links with devices in difficult areas, circumventing unfavorable channel conditions. However, UAV deployment in such settings poses challenges in terms of scheduling and trajectory design. To optimize UAV trajectory and device scheduling, we propose a heuristic metric that serves as a proxy for training performance. Based on this metric, we define a surrogate objective that enables joint optimization of UAV trajectory and device scheduling using convex optimization techniques and graph theory. This segment summarizes Chapter 3, which is based on the published work:

  - [45] Mohamad Mestoukirdi, Omid Esrafilian, David Gesbert, and Qianrui Li, "**UAV-Aided Multi-Community Federated Learning**," *IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022 , pp. 1314-1319.*

- In the final part of the thesis, we shift our focus towards addressing the challenge of the communication burden associated with exchanging model updates. In archetypical FL algorithms, during each round of communication, model updates are often quantized or sparsified before being sent on the UL or DL channels, leading to improved communication efficiency. However, this compression often comes at the cost of reduced model accuracy. To overcome this trade-off, extensive research has been conducted to explore alternative algorithms that can decouple model accuracy from communication efficiency in FL. One recent promising approach involves pruning a random network to approximate a target network, according to the subset-sum approximation problem [58,59], which has demonstrated significant gains in communication efficiency and model generalization. However, we show that existing state-of-the-art algorithms that adopt such schemes in federated settings fail to fully harness the potential for communication efficiency improvements. As a result, we propose a novel algorithm that achieves substantially higher communication gains. Our approach promotes additional pruning of the random networks, resulting in sparser model updates. Importantly, the proposed solution is demonstrated to have a negligible impact on the generalization performance of the produced model compared to the communication-efficiency gains achieved.

This part is covered by Chapter 4 and is based on :

– [60] Mohamad Mestoukirdi, Omid Esrafilian, David Gesbert, Qianrui Li, and Nicolas Gresset, 2023. **Sparser Random Networks Exist: Enforcing Communication-Efficient Federated Learning via Regularization.** arXiv preprint arXiv:2309.10834. (Submitted to IEEE Communication letters)

Part I

On Statistical Heterogeneity in Federated Learning

Chapter 2

User-Centric Federated Learning

Federated learning aims to enable collaborative training between devices that individually may not have sufficient local data to train sufficiently good models. By aggregating updates from many devices' local datasets, the goal is to leverage supplementary training data existing across devices. However, a fundamental trade-off arises when the data distribution on each device is not well aligned. For instance, when the labeling or annotation functions used to generate training data differ across devices, then for the same input data points, contradictory or inconsistent labels may be applied. When models are trained on aggregated data from those devices, it creates conflicting optimization objectives. Herein, the model has to compromise and learn weights that perform adequately but not optimally on any localized labeling distribution. In such cases, attempting to learn a single global model yields poor generalization performance compared to training customized models per device that can adapt to each one's unique labeling behavior. The personalization of models, while providing significant in heterogeneous settings, also demands a greater utilization of communication resources, particularly in the downlink channels to transmit the unique personalized models.

Accordingly, in this chapter, we tackle the predicament of statistical heterogeneity in federated learning stemming from the divergent data distributions among devices datasets. To address this problem without violating the privacy constraints that FL imposes, personalized FL methods have to couple statistically similar clients without directly accessing their data in order to guarantee a privacy-preserving transfer. We design user-centric aggregation rules at the parameter server (PS) that are based on readily available gradient information and are capable of producing personalized models for each device. Secondly, we derive a communication-efficient variant based on user clustering which greatly enhances its applicability to communication-constrained systems.

Early FL algorithms were devised under the assumption that the data distribution of clients' data sets is common. In this case, clients are said to share the same learning task, and traditional FL (e.g FedAvg [24]) algorithms can perform and generalize well yielding a single model, fitting the common data distribution. However, this assumption is hardly met in practice [61], as data distribution heterogeneity often arises in distributedly generated data sets. In such cases, traditional FL (e.g. FedAvg) approaches exhibit slow convergence and often fail to generalize well [62], especially when conflicting objectives among users

exist. This is a direct consequence of the fact that in heterogeneous settings, a convex combination of locally trained models may not be fit for any particular client data distribution. Hence, heterogeneous distributions bring about an interesting trade-off: On the one hand the advantage of exploiting training data at other clients when the local training data are insufficient, and on the other hand the problem of having the trained model steered towards improper directions due to differences in data distributions among clients. This trade-off motivates the search for new FL strategies that can navigate the compromise between model aggregation benefits and the threat of model mismatch.

We propose a novel user-centric aggregation rule to tackle the underlying heterogeneity among clients and overcome the shortcomings of the traditional FL schemes. The proposed strategy leverages user-centric aggregation rules at the PS to produce models at each device that are tailored to their local data distribution. This is achieved by generalizing the aggregation rule introduced by McMahan et al. [24]. Particularly, in the case of a set of K collaborating devices, the original objective in [24] produces a common model at each communication round t according to

$$\theta^t \leftarrow \sum_{i=1}^{K} w_i \theta_i^{t-1/2}, \tag{2.1}$$

where each w_i weights the contribution of the locally optimized model $\theta_i^{t-\frac{1}{2}}$ of user i, to the update global model θ^t. On the other hand, the proposed aggregation rule replaces each device i weighting coefficient w_i, by user-specific weighting vectors $\vec{w}_i = (w_{i,1}, \ldots, w_{i,K})$ and it produces a personalized model update for each FL client

$$\theta_i^t \leftarrow \sum_{j=1}^{K} w_{i,j} \theta_j^{t-1/2}, \qquad \text{for } i = 1, 2, \cdots, K. \tag{2.2}$$

The key motivation underpinning the use of distinct user-centric personalization rules is that a single model often fails in heterogeneous settings [24]. At the same time, hard clustering strategies [61,63] are limited to restrictive intra-cluster collaboration and they cannot exploit similarities among different clusters. The authors in [64] proposed FedFomo, a personalization scheme that uses a similar aggregation policy as ours [65]. However, FedFomo's weighting scheme is repeatedly refined during training and it relies on sharing local models among clients at each communication round. This strategy can violate the FL privacy-preserving nature, and introduces a large communication burden to the training procedure. In contrast, our personalization policy is shown experimentally to enjoy faster convergence, being able to capture the data heterogeneity at the start of training without the need for further refinements at later stages.

The main contributions of this chapter can be summarized as follows :

- We establish an upper bound on the expected risk of the minimizer of the user-centric learning objective.
- Based on the developed bound, we motivate the use of heuristically determined collaboration coefficients as an alternative to theoretically optimal ones, thereby presenting a practical approach.
- To reduce the communication costs associated with the transmission of multiple personalized models, we introduce a k-means clustering algorithm that operates on the user-centric weighing vectors $\vec{w}_i$. This approach effectively limits the number of personalized streams while considering the heterogeneity of local data and learning tasks. It effectively balances learning accuracy and communication load.
- We show that by evaluating the quality of the k-means solution using the silhouette score, we can identify the underlying heterogeneity a priori. This technique provides a principled way of determining the number of user-centric rules.
- Through extensive numerical experiments conducted on popular FL benchmarks, we validate the performance of our proposed strategy and compare it with state-of-the-art solutions. Specifically, we evaluate the inference accuracy and communication costs, thereby highlighting the effectiveness of our approach and its relevance in scenarios characterized by scarce communication resources.

2.1 Related Work

Several recent studies investigate the challenges that arise due to the underlying task heterogeneity and communication efficiency present across learners in Federated Learning settings. For instance, the authors in [36] propose a novel framework for enabling the implementation of FL algorithms over wireless networks by jointly taking into account FL and wireless factors. [61, 63] devised a hierarchical clustering scheme to group users that share the same learning task and enable collaboration among them only. However, their strategy is based on the assumption that heterogeneous tasks are either tangential or parallel, which is not necessarily true, as tasks are defined by the users' target data distributions which are often different for each of them. In this sense, hard-clustering strategies – despite maintaining communication efficiency – limit the degree of collaboration across learners and may not always be able to capture the differences across users' tasks. In [66] a distributed

Expectation-Maximization (EM) algorithm has been proposed, that concurrently converges to a set of shared hypotheses and a personalized linear combination of them at each device. Similarly in [67], a Mixture of Experts' formulation has been devised to learn a personalized mixture of the outputs of a jointly trained set of models. Similar to [64], exploiting the full personalization potential of the solutions in [66, 67] induces a huge overhead over the communication resources in the federated system, which renders their approaches unpractical. Similar to Fedprox [26], the authors in [68] propose SCAFFOLD to tackle the "*client drifts*" that emerge as a result of the heterogeneity of the clients' data sets during the global model training. However, in some heterogeneous settings, "*client drifts*" can act as an indication of the existence of opposing target tasks among the learners. Therefore, intelligently employing the drifts can highlight similarity patterns among the clients' tasks [61], which in turn can aid in training multiple refined models to fit each of the available tasks, yielding better-personalized models in contrast to a single global model trained by SCAFFOLD. More recently, the authors in [25] propose Ditto, where users collaborate to train a separate global model akin to [24], which is then used to steer the training of the local personalized model at each user via local model adaptation. Their approach embodies the intuition of pFedMe [69], which decouples personalized model optimization from the global model learning by introducing a penalizing term to regularize the clients' local adaptation step. Despite resulting in a per-user personalized model, collaboration among users in Ditto and pFedMe is limited to updating the global model, while relying solely on the local data sets to train their personalized models, rather than leveraging collaboration among statistically similar learners to refine those models. Consequently, the resulting personalized models may generalize poorly, especially in settings where local data sets are small in size.

2.2 Learning with heterogeneous data sources

In this section, we provide theoretical guarantees for learners that combine data from heterogeneous data distributions. The setup mirrors the one of personalized federated learning and the results are instrumental to derive our user-centric aggregation rule. In the following, we limit our analysis to the discrepancy distance, but it can be readily extended to other divergences as we show later.

In the federated learning setting, the weighted combination of the empirical loss terms of the collaborating devices represents the customary training objective. Namely, in a distributed system with K nodes, each endowed with a data set $\mathcal{D}_i$ of $|\mathcal{D}_i|$ IID samples from a local distribution P_i, the goal is to find a predictor $f : \mathcal{X} \to \hat{\mathcal{Y}}$ from a hypothesis class $\mathcal{F}$

that minimizes

$$L(f, \vec{w}) = \sum_{i=1}^{K} \frac{w_i}{|\mathcal{D}_i|} \sum_{(x,y)\in\mathcal{D}_i} \ell(f(x), y), \tag{2.3}$$

where $\ell : \hat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}^+$ is a loss function and $\vec{w} = (w_1, \ldots, w_K)$ is a weighting scheme. In the case of identically distributed local data sets, the typical weighting vector is $\vec{w} = \frac{1}{\sum_i |\mathcal{D}_i|}(|\mathcal{D}_1|, \ldots, |\mathcal{D}_K|)$, the relative fraction of data points stored at each device. This particular choice minimizes the variance of the aggregated empirical risk, which is also an unbiased estimate of the local risk at each node in this scenario. However, in the case of heterogeneous local distributions, the minimizer of $\vec{w}$-weighted risk may transfer poorly to certain devices whose target distribution differs from $P_{\vec{w}} = \sum_{i=1}^{K} w_i P_i$, the mixture of distributions which the final global model is trained to generalize over. Furthermore, there may not exist a single weighting strategy that yields a universal predictor with satisfactory performance for all participating devices. To address the above limitation of a universal model, personalized federated learning allows adapting the learned solution at each device. To better understand the potential benefits and drawbacks coming from the collaboration with statistically similar but not identical devices, let us consider the point of view of a generic node i that has the freedom of choosing the degree of collaboration with the other devices in the distributed system. Namely, identifying the degree of collaboration between node i and the rest of users by the weighting vector $\vec{w}_i = (w_{i,1}, \ldots, w_{i,K})$ (where $w_{i,j}$ defines how much node i relies on data from user j) we define the personalized objective for user i

$$L(f, \vec{w}_i) = \sum_{j=1}^{K} \frac{w_{i,j}}{|\mathcal{D}_j|} \sum_{(x,y)\in\mathcal{D}_j} \ell(f(x), y), \tag{2.4}$$

and the resulting personalized model

$$\hat{f}_{\vec{w}_i} = \arg\min_{f\in\mathcal{F}} L(f, \vec{w}_i). \tag{2.5}$$

We now seek an answer to: *"What's the proper choice of $\vec{w}_i$ in order to obtain a personalized model $\hat{f}_{\vec{w}_i}$ that performs well on the target distribution P_i?"*. This question is deeply tied to the problem of domain adaptation, in which the goal is to successfully aggregate multiple data sources to produce a model that transfers positively to a different and possibly unknown target domain. In our context, the data set $\mathcal{D}_i$ is made of data points drawn from the target distribution P_i, and the other devices' data sets provide samples from the sources $\{P_j\}_{j\neq i}$. Leveraging results from domain adaptation theory [70], we provide learning guarantees on the performance of the personalized model $\hat{f}_{\vec{w}_i}$ to gauge the effect of collaboration that we later use to devise the weights for the user-centric aggregation rules.

To avoid negative transfer, it is crucial to upper bound the performance of the predictor w.r.t. to the target task. The discrepancy distance introduced in [71] provides a measure of similarity between learning tasks that can be used to this end. For a hypothesis set of functions $\mathcal{F} : \mathcal{X} \to \hat{\mathcal{Y}}$ and two distributions P, Q on $\mathcal{X}$, the discrepancy distance is defined as

$$d_{\mathcal{F}}(P, Q) = \sup_{f, f' \in \mathcal{F}} \left| \mathbb{E}_{x \sim P} \left[\ell(f, f') \right] - \mathbb{E}_{x \sim Q} \left[\ell(f, f') \right] \right|, \tag{2.6}$$

where we streamlined notation denoting $f(x)$ by f. For bounded and symmetric loss functions that satisfy the triangular inequality, the previous quantity allows to obtain the following inequality

$$\mathbb{E}_{(x,y)\sim P}[\ell(f, y)] \leq \mathbb{E}_{(x,y)\sim Q}[\ell(f, y)] + 2d_{\mathcal{F}}(P, Q) + 2\gamma,$$

where $\gamma = \inf_{f \in \mathcal{F}} \left(\mathbb{E}_{(x,y)\sim P}[\ell(f, y)] + \mathbb{E}_{(x,y)\sim Q}[\ell(f, y)] \right)$. We can exploit the inequality to obtain the following risk guarantee for $\hat{f}_{\vec{w}_i}$ w.r.t the true minimizer f^* of the risk for the distribution P_i.

Theorem 1. *For a loss function ℓ with B-bounded range, symmetric and satisfying the triangular inequality, with probability $1 - \delta$ the function $f_{\vec{w}_i}$ satisfies*

$$\begin{aligned} &E_{z\sim P_i}[\ell(f_{\vec{w}_i}, z)] - E_{z\sim P_i}[\ell(f^*, z)] \leq \\ &B\sqrt{\sum_{j=1}^{K} \frac{w_{i,j}^2}{|\mathcal{D}_j|}} \left(\sqrt{\frac{2d}{\sum_i |\mathcal{D}_i|} \log\left(\frac{e \sum_i |\mathcal{D}_i|}{d} \right)} + \sqrt{\log\left(\frac{2}{\delta} \right)} \right) + \\ &2 \sum_{j=1}^{K} w_{i,j} d_{\mathcal{F}}(P_i, P_j) + 2\gamma, \end{aligned}$$

where $\gamma = \min_{f \in \mathcal{F}} \left(E_{z\sim P_i}[\ell(f, z)] + E_{z\sim P_{\vec{w}_i}}[\ell(f, z)] \right)$ and d is the VC-dimension of the function space resulting from the composition of $\mathcal{F}$ and ℓ.

In a scenario with similar underlying data distributions and sufficiently large datasets, the selection of weight vector $\vec{w}_i$ has a negligible effect. As the cardinality of the datasets approaches infinity, the functions $f_{\vec{w}_i}$ converges to the true minimizer f^*, irrespective of choice of the user-centric weights. Recently, an alternative bound based on an information-theoretic notion of dissimilarity, the Jensen-Shannon divergence, has been proposed [72]. It is based on less restrictive constraints, as it only requires the loss function $\ell(f, Z)$ to be sub-Gaussian of some parameter σ for all $f \in \mathcal{F}$, and therefore whenever $\ell(\cdot)$ is bounded, the requirement is automatically satisfied. Measuring similarity by the Jensen-Shannon

divergence the following inequality is available

$$E_{X\sim P}[X] \leq E_{X\sim Q}[X] + \beta\sigma^2 + \frac{D_{JS}(P||Q)}{\beta}, \quad \text{for } \beta > 0, \tag{2.7}$$

where $D_{JS}(P\|Q) = \mathrm{KL}\left(P\middle\|\frac{P+Q}{2}\right) + \mathrm{KL}\left(Q\middle\|\frac{P+Q}{2}\right)$. Exploiting the above inequality we obtain the following estimation error bound.

Theorem 2. *For a loss function ℓ B-bounded range, the function $f_{\vec{w}_i}$ satisfies*

$$E_{z\sim P_i}[\ell(f_{\vec{w}_i}, z)] - E_{z\sim P_i}[\ell(f^*, z)] \leq$$
$$B\sqrt{\sum_{j=1}^{K}\frac{w_{i,j}^2}{|\mathcal{D}_j|}}\left(\sqrt{\frac{2d}{\sum_i|\mathcal{D}_i|}\log\left(\frac{e\sum_i|\mathcal{D}_i|}{d}\right)} + \sqrt{\log\left(\frac{2}{\delta}\right)}\right) +$$
$$B\sqrt{2\sum_{j=1}^{K} w_{i,j} D_{JS}(P_i||P_j)},$$

Proof of Theorems 1 and 2: In the Appendix.
The theorems highlight that a fruitful collaboration should strike a balance between the bias term due to the dissimilarity between the local distributions and the risk estimation gains provided by the data points of other nodes. Minimizing the upper bound in Theorems 1 and 2 with respect to the user-specific weights, and using the optimal weights in our aggregation rule seems an appealing solution to tackle the data heterogeneity during training; however, the distance terms ($d_{\mathcal{F}}(P_i, P_k)$ and $D_{JS}(P_i||P_j)$) are difficult to compute, especially under the privacy constraints that federated learning imposes. For this reason, in the following, we consider a heuristic method based on the similarity of the readily available users' model updates to estimate the collaboration coefficients.

2.3 User-centric aggregation

For a suitable hypothesis class parametrized by $\theta \in \mathbb{R}^d$, federated learning approaches use an iterative procedure to minimize the aggregate loss (2.3) with $\vec{w} = \frac{1}{\sum_i|\mathcal{D}_i|}(|\mathcal{D}_1|, \ldots, |\mathcal{D}_K|)$. At each round t, the PS broadcasts the parameter vector θ^{t-1} and then combines the locally optimized models by the clients $\{\theta_i^{t-1}\}_{i=1}^K$ according to the following aggregation rule

$$\theta^t \leftarrow \sum_{i=1}^{K}\frac{|\mathcal{D}_i|}{\sum_{j=1}^{K}|\mathcal{D}_j|}\theta_i^{t-1}.$$

As mentioned in Sec. 2.2, this aggregation rule has two shortcomings: it does not take into account the data heterogeneity across users, and it is bounded to produce a single solution. For this reason, we propose a user-centric model aggregation scheme that takes into account the data heterogeneity across the different nodes participating in training and aims at neutralizing the bias induced by a universal model. Our proposal generalizes the naïve aggregation of FedAvg, by assigning a unique set of mixing coefficients $\vec{w}_i$ to each user i and, consequently, a user-specific model aggregation at the PS side. Namely, on the PS side, the following set of user-centric aggregation steps are performed

$$\theta_i^t \leftarrow \sum_{j=1}^{K} w_{i,j}\theta_j^{t-1/2}, \quad \text{for } i = 1, \ldots, K, \tag{2.8}$$

where now, $\theta_j^{t-1/2}$ is the locally optimized model at node j starting from θ_j^{t-1}, and θ_i^t is the user-centric aggregated model for user i at communication round t. As we elaborate next,

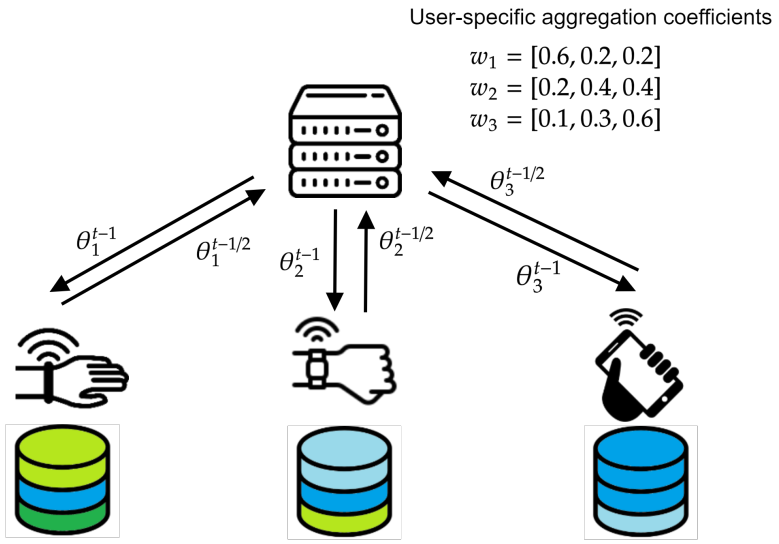


Figure 2.1: Personalized Federated Learning with user-centric aggregates at round t.

the mixing coefficients are heuristically defined based on a distribution dissimilarity metric and the data set size ratios. These coefficients are calculated before the start of federated training. The dissimilarity score we propose is designed to favour collaboration among similar users and takes into account the relative data set sizes, as more intelligence can be harvested from clients with larger data availability. Using these user-centric aggregation rules, each node ends up with its personalized model that yields better generalization for the local data distribution. It is worth noting that the user-centric aggregation rule does not produce a minimizer of the user-centric aggregate loss given by (2.4). At each round, the PS aggregates model updates are computed starting from a different set of parameters.

Nonetheless, we find it to be a good approximation of the true update since personalized models for similar data sources tend to propagate in a close neighborhood. The aggregation in [64] capitalizes on the same intuition.

2.3.1 Computing the Collaboration Coefficients

Computing the discrepancy distance (2.6) can be challenging in high-dimension, especially under the communication and privacy constraints imposed by federated learning. For this reason, we propose to compute the mixing coefficient based on the relative data set sizes and the distribution dissimilarity metric given by

$$\begin{aligned}\Delta_{i,j}(\hat{\theta}) &= \left\| \frac{1}{|\mathcal{D}_i|} \sum_{(x,y)\in\mathcal{D}_i} \nabla \ell(f_{\hat{\theta}}, y) - \frac{1}{|\mathcal{D}_j|} \sum_{(x,y)\in\mathcal{D}_j} \nabla \ell(f_{\hat{\theta}}, y) \right\|^2 \\ &\approx \left\| \mathbb{E}_{z\sim P_i} \nabla \ell(f_{\hat{\theta}}, y) - \mathbb{E}_{z\sim P_j} \nabla \ell(f_{\hat{\theta}}, y) \right\|^2,\end{aligned}$$

where the quality of the approximation depends on the number of samples $|\mathcal{D}_i|$ and $|\mathcal{D}_j|$. The mixing coefficients for user i are then set to the following normalized Gaussian kernel function

$$w_{i,j} = \frac{\frac{|\mathcal{D}_j|}{|\mathcal{D}_i|} e^{-\frac{1}{2\sigma_i\sigma_j}\Delta_{i,j}(\hat{\theta})}}{\sum_{j'=1}^{K} \frac{|\mathcal{D}_{j'}|}{|\mathcal{D}_i|} e^{-\frac{1}{2\sigma_i\sigma_j}\Delta_{i,j'}(\hat{\theta})}}, \qquad \text{for } j = 1, \dots, K. \tag{2.9}$$

The mixture coefficients are calculated at the PS during a special round before federated training. During this round, the PS broadcasts an initialized model denoted ($\hat{\theta} = \theta^0$) to the users, which computes the full gradient on their local data sets. At the same time, each node i locally estimates the value σ_i^2 partitioning the local data randomly in N batches $\{\mathcal{D}_i^k\}_{k=1}^N$ of size $n_{i,k}$ and computing

$$\sigma_i^2 = \frac{1}{N} \sum_{k=1}^{N} \left\| \frac{1}{n_{i,k}} \sum_{(x,y)\in\mathcal{D}_i^k} \nabla \ell(f_{\hat{\theta}}, y) - \frac{1}{|\mathcal{D}_i|} \sum_{(x,y)\in\mathcal{D}_i} \nabla \ell(f_{\hat{\theta}}, y) \right\|^2, \tag{2.10}$$

where σ_i^2 is an estimate of the gradient variance (i.e. noise) computed over local data sets $\mathcal{D}_i^k$ sampled from the same target distribution P_i. The variances are computed as a function of the partitioned mini-batch sizes. Consequently, the size of the mini-batches shall be chosen carefully to successfully capture clients of similar data distributions during training. We discuss the suitable choice of mini-batch sizes to compute the variances in Sec. V-2.4.6. Once all the necessary quantities are computed, they are uploaded to the PS, which proceeds to calculate the mixture coefficients and initiates the federated training using the custom

aggregation scheme given by (2.8).

The proposed heuristic (2.8) embodies the intuition provided by Theorems 1 and 2. Specifically, a device should collaborate with peers that have similar data distributions and large data sets instead of devices that have small data sets or dissimilar data. Moreover, the proposed weighting scheme recovers as special case scenarios for which is possible to explicitly characterize the value of the collaboration coefficients. For example, in the case of homogeneous users, it falls back to the standard FedAvg aggregation rule, while if node i has an infinite amount of data, it degenerates to the local learning rule that is optimal in this case.

Algorithm 1: User-centric Federated Learning

Input : number of clients K, local mini-batch size B, number of epochs E and learning rate η

PS broadcasts θ^0 to the users

foreach user k **do**

  Compute $\nabla \ell(\theta^0, \mathcal{D}_k)$

  Compute σ_k^2 as in (2.10)

  Transmit $\{\nabla \ell(\theta^0, \mathcal{D}_k), \sigma_k^2\}$ to PS

end

PS computes $w_{i,j}$ as in (2.9)

for $t = 0, \dots, T$ **do**

  PS unicasts θ_k^t to each node k

  foreach *node* k **do**

    $\theta_k^{t+1} \leftarrow$ **ClientUpdate**$(\theta_k^t, \mathcal{D}_k)$

    return θ_k^{t+1} to *PS*

  end

  PS computes $\theta_k^{t+1} \leftarrow \sum_{j=1}^{K} w_{k,j} \theta_j^{t+1}$

end

PROCEDURE: ClientUpdate$(\theta_k^t, \mathcal{D}_k)$:

$\mathcal{B} \leftarrow$ Split $\mathcal{D}_k$ into mini-batches of size B

$\theta_k \leftarrow \theta_k^t$

for $t = 0, \dots, E$ **do**

  foreach mini-batch $\mathbf{b} \in \mathcal{B}$ **do**

    $\theta_k \leftarrow \theta_k - \eta \nabla \ell(\theta_k, \mathbf{b})$

  end

end

return θ_k

2.3.2 Algorithm Complexity

We will now analyze the overhead, per-round computation, and communication complexity of the proposed user-centric aggregation strategy, which can be found in Algorithm 1. In user-centric federated learning, the overhead is primarily introduced during the computation of collaboration coefficients before training. As discussed in Section 2.3.1, the computation and communication complexity of this procedure are equivalent to that of one round of standard federated learning. Specifically, the collaboration coefficients are computed by the parameter server (PS) broadcasting a single model to the devices, each device computing a model update and sending it back to the PS server. Consequently, the overhead resulting from the computation of collaboration coefficients becomes negligible as the overall number of federated learning rounds increases.

The computation complexity of user-centric federated learning on the device side is identical to that of standard federated learning. Each device is simply required to locally optimize the received model. On the PS side, the use of multiple aggregation rules increases the computation complexity compared to standard federated learning, where the aggregation rule is singular. However, since each aggregation rule involves a simple linear combination of model parameter vectors, its computational cost remains limited and negligible.

Regarding communication complexity, the uplink communication of user-centric federated learning is the same as that of standard federated learning, with each device sending a single model to the PS. On the other hand, in the downlink, the communication cost grows linearly with the number of personalized models. While this cost is significant, Section 2.3.3 provides a procedure to efficiently reduce the number of personalized streams, thereby mitigating the communication cost in the downlink.

In conclusion, although the per-round communication cost of user-centric federated learning is higher than that of standard federated learning, the use of personalized aggregation rules greatly reduces the number of training rounds required to achieve convergence under heterogeneous settings. Consequently, the aggregate communication and computation cost is lower in these settings, as demonstrated in the experiments in Section 2.4.

2.3.3 Reducing the Communication Load

A full-fledged personalization employing the user-centric aggregation rule (2.8) would introduce an K-fold increase in communication load during the downlink phase as the original broadcast transmission is replaced by unicast ones. Although from a learning perspective, the user-centric learning scheme is beneficial, it is also possible to consider overall system performance from a learning-communication trade-off point of view. The intuition is that, for small discrepancies between the user data distributions, the same model transfers positively

to statistically similar devices. To strike a suitable trade-off between learning accuracy and communication overhead we hereby propose to adaptively limit the number of personalized downlink streams. In particular, for a number of personalized models m, we run a k-means clustering scheme over the set of collaboration vectors $\{\vec{w}_i\}_{i=1}^K$ and we select the centroids $\{\vec{c}_i\}_{i=1}^m$ to implement the m personalized streams. Formally, given m and the user-specific weights $\{\vec{w}_i\}_{i=1}^K$, the objective is to find $m < K$ clusters $\mathcal{C}_1, \ldots, \mathcal{C}_m$ such that

$$\sum_{n=1}^{m} \sum_{\vec{w}_i \in \mathcal{C}_n} \|\vec{w}_i - \vec{c}_n\| \tag{2.11}$$

is minimized, where $\vec{c}_n$ is the centroid of cluster $\mathcal{C}_n$. We then proceed to replace the unicast transmission with group broadcast ones, in which all users belonging to the same cluster i receive the same personalized model associated with the centroid $\vec{c}_i$. Choosing the right value for the number of personalized streams is critical to saving communication bandwidth but at the same time obtain satisfactory personalization capabilities. In the following, we experimentally show that clustering quality indicators such as the Silhouette score can be used to guide the search for a suitable number of clusters m.

2.3.4 Choosing the Number of Personalized Streams

Algorithm 2: Silhouette based scoring

Input : Collaboration vectors $\{\vec{w}_i\}_{i=1}^K$ from Algorithm 1 and a trade-off function $c(k, s_k)$.
Output: Number of clusters m
for $k = 1, 2, \ldots, K$ **do**
  $\mathcal{C}_k \leftarrow k$-means clustering of $\{\vec{w}_i\}_{i=1}^K$
  $s_k \leftarrow$ the silhouette score of $s(\mathcal{C}_k)$
end
return $m = \arg\max_{k=1,\ldots,K} c(k, s_k)$

Choosing an insufficient number of personalized streams can yield unsatisfactory performance, while concurrently learning many models can prohibitively increase the communication load of personalized federated learning. Therefore, properly tuning this free parameter is essential to obtain a well-performing but still practical algorithm. Being agnostic w.r.t. the underlying data generating distributions at the devices, there does not exist a universal number of personalized streams that fits all problems. However, we now illustrate that the silhouette coefficient, a quality measure of the clustering, provides a rule of thumb for choosing the number of personalized streams. In order to compute the silhouette score of the

clusters $\mathcal{C}_1, \ldots, \mathcal{C}_m$, we define the intra-cluster similarity of the collaboration vector $\vec{w}_i \in \mathcal{C}_k$ as

$$a(\vec{w}_i) = \frac{1}{|\mathcal{C}_j| - 1} \sum_{\vec{w}_j \in \mathcal{C}_k, \vec{w}_j \neq \vec{w}_i} \|\vec{w}_j - \vec{w}_i\|,$$

and the smallest mean distance between the collaboration vector $\vec{w}_i \in \mathcal{C}_k$ and the closest cluster

$$b(\vec{w}_i) = \min_{\mathcal{C}_j \neq \mathcal{C}_k} \frac{1}{|\mathcal{C}_j|} \sum_{\vec{w}_j \in \mathcal{C}_j} \|\vec{w}_j - \vec{w}_i\|.$$

The average silhouette score s is then defined as

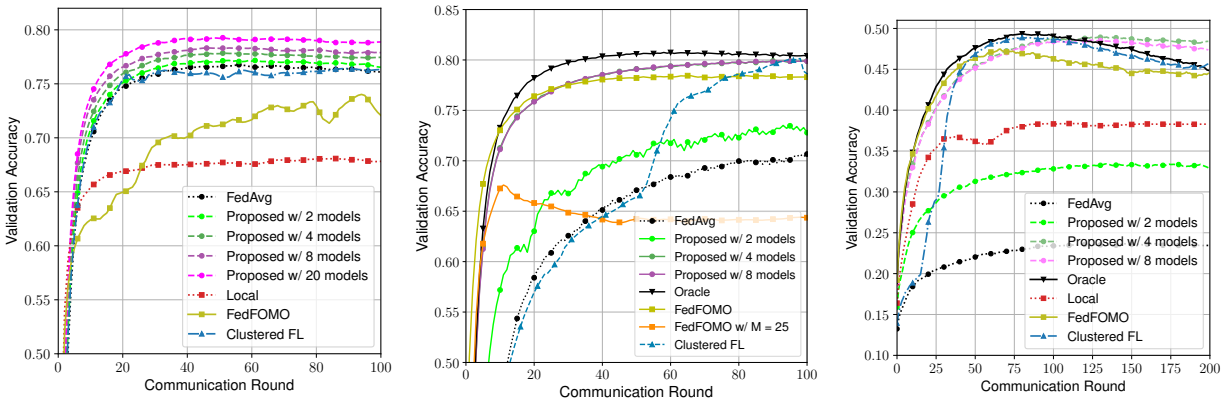
$$s(\mathcal{C}) = \frac{1}{K} \sum_{i=1}^{K} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

and it is a number in the range $[-1, 1]$, directly proportional to the quality of the clustering. In turn, a good clustering of the collaboration vectors $\{\vec{w}_i\}_{i=1}^{K}$ implies that users belonging to the same clusters are similar and that the centroid $\vec{c}_j$ is a good approximation of the collaboration coefficient of users in $\mathcal{C}_j$. Consequently, whenever the silhouette score is large, the loss in terms of personalization performance resulting from the reduced number of aggregation rules compared to the full-fledged personalization system is modest. For this reason, the silhouette score provides a proxy to the inference performance and at the same time, it allows for a trade-off between communication load and personalization capabilities in a principled way. In Algorithm 2 we provide the pseudocode of the procedure that autonomously chooses the optimal number of personalized streams m based on a communication-personalization trade-off function $c(k, s_k) : \mathbb{N} \times [-1, 1] \to \mathbb{R}$ scoring the utility of pairs of the systems based on the number of user-centric rules and the resulting silhouette scores.

Possible communication-personalization trade-off functions are of the form

$$c(k, s_k) = s_k - \lambda k. \tag{2.12}$$

This particular function allows us to measure the trade-off between two factors. In a system with limited bandwidth, having a high value of k would result in increased communication costs. Conversely, a relatively large value of s_k reflects the quality of clustering among the associated clients, given k clusters. The parameter $\lambda > 0$, serves as a design parameter balancing the trade-off between the clustering quality s_k and the communication cost induced by the presence of k clusters.



(a) EMNIST + label shift

(b) EMNIST + label and covariate shift

(c) CIFAR10 + concept shift

Figure 2.2: Average Validation Accuracy across the three different experiments

2.4 Experiments

We now provide a series of experiments to showcase the personalization capabilities and communication efficiency of the proposed algorithm.

2.4.1 Set-up

In our simulation we consider a handwritten character/digit recognition task using the EMNIST data set [73], an image classification task using the CIFAR-10 data set [74], and a text sentiment classification on Stack Overflow questions dataset extracted from the much larger public Stack-Overflow dataset on BigQuery [75]. Data heterogeneity is induced by splitting and transforming the data set differently across the group of devices. In particular, we analyze four different scenarios:

- **Character/digit recognition with user-dependent label shift** in which 10k EMNIST data points are split across 20 users according to their labels. The label distribution follows a Dirichlet distribution with parameter $\alpha = 0.4$, as in [66, 76].

- **Character/digit recognition with user-dependent label shift and covariate shift** in which 100k samples from the EMNIST data set are partitioned across 100 users each with a different label distribution ($\alpha = 8$), as in the previous scenario. Additionally, users are clustered in 4 groups $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4\}$, and at each group images are rotated by $\{0°, 90°, 180°, 270°\}$ respectively. In particular, heterogeneity is imposed such that $P_i(x|y) \neq P_j(x|y)$, $\forall\, i \in \mathcal{G}_k, j \in \mathcal{G}_{k'}, k \neq k'$, $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$.

Table 2.1: Average test accuracy of the different algorithms across the three proposed scenarios.

Algorithm	Scenario		
	EMNIST ($K = 20$) label shift	EMNIST ($K = 100$) covariate & label shift	CIFAR10 ($K = 20$) concept shift
Proposed $m = K$	**79.4** ($\pm$ 4.2)	**77.9** ($\pm$ 2.7)	**47.7** ($\pm$ 2.2)
Proposed $m = 4$	77.8 ($\pm$ 3.9)	**79.7** ($\pm$ 2.5)	**49.1** ($\pm$ 1.4)
SCAFFOLD [68]	77.2 ($\pm$ 4.0)	72.5 ($\pm$ 2.2)	17.5 ($\pm$ 1.8)
Ditto [25]	78.3 ($\pm$ 3.9)	74.1 ($\pm$ 2.3)	44.1 ($\pm$ 1.4)
pFedMe [69]	77.6 ($\pm$ 4.1)	75.2($\pm$ 4.4)	46.6 ($\pm$ 1.5)
Fedprox [26]	**79.6** ($\pm$ 4.8)	72.4 ($\pm$ 2.4)	22.3 ($\pm$ 2.2)
Local	68.2 ($\pm$ 5.3)	62.8 ($\pm$ 3.3)	38.3 ($\pm$ 1.2)
FedAvg [24]	76.7 ($\pm$ 4.0)	70.5 ($\pm$ 2.2)	24.2 ($\pm$ 2.6)
Oracle *(Upper bound)*	-	80.7 ($\pm$ 1.8)	49.5 ($\pm$ 1.2)

- **Image classification with group dependent concept shift** in which the CIFAR-10 data set is distributed across 20 users which are grouped in 4 clusters, for each group we apply a different random label permutation. More specifically, given an image $x \in \mathcal{X}$ and the labelling functions $f_i, f_j : \mathcal{X} \to \mathcal{Y}$, then $f_i(x) \neq f_j(x), \forall i \in \mathcal{G}_k, j \in \mathcal{G}_{k'}, k \neq k'$.

- **Text sentiment classification with user-dependent label shift** in which 16k samples from the Stack-Overflow dataset are distributed across 35 users. A sample tuple (Q, t) is composed of a question Q and its corresponding tag t (either Python, CSharp, JavaScript, or Java). Heterogeneity is imposed across the tag/label distributions, which follow a Dirichlet distribution with $\alpha = 0.4$ akin to the first experiment.

For each scenario, we aim to solve the task at hand by leveraging the distributed and heterogeneous data sets. We compare our algorithm against two sets of baseline algorithms. The first set includes algorithms that achieve personalization by resulting in multiple personalized models. Those include CFL [61], FedFomo [64], pFedMe [69] and Ditto [25]. The second set of baselines include algorithms that yield a single Federated model such as Fedprox [1] [26], SCAFFOLD [68]. FedAvg [24], and Local training algorithms are also included for reference. The image classification tasks are trained using the LeNet-5 [77] convolutional neural network architecture, while for the text sentiment classification task, we train a feedforward network consisting of an embedding layer, a dense layer, and a softmax output layer. We use stochastic gradient descent optimizer with fixed learning rate $\eta = 0.01$,

[1]The penalization hyperparameters μ and $\lambda = \{0.1, 0.5, 1\}$ were used in the simulations of Fedprox and Ditto, then, the best results were reported.

Table 2.2: Worst user performance averaged over 5 experiments in the three simulation scenarios

Scenario	Algorithm					
	Ditto [25]	FedAvg [24]	Oracle	CFL [61]	FedFOMO [64]	Proposed
EMNIST ($K = 20$) label shift	72.2	68.9	-	70.3	70.0	**73.2** ($m = 20$)
EMNIST ($K = 100$) covariate & label shift	70.7	67.5	77.4	76.1	73.6	**76.4** ($m = 4$)
CIFAR10 ($K = 20$) concept shift	43.2	19.6	49.1	48.6	45.5	**48.8** ($m = 4$)

momentum $\beta = 0.9$, and the number of epochs $E = 1$ and mini-batch size $= 64$ for image classification tasks[2]. For the Stack-Overflow sentiment classification problem, we set the mini-batch size to 32 and $\eta = 0.01$. The feature preprocessing, feature extraction, and word embedding steps were done akin to [78]. For FedFomo, we experiment with different values of the hyper-parameter M. M denotes the number of models shared with each user at each round to compute FedFOMO aggregation weights. In our simulations, we chose the number of clusters, denoted as m, by choosing the value that maximizes the silhouette score. Nevertheless, we also conducted experiments using different values of m to illustrate the trade-off between learning accuracy and communication efficiency.

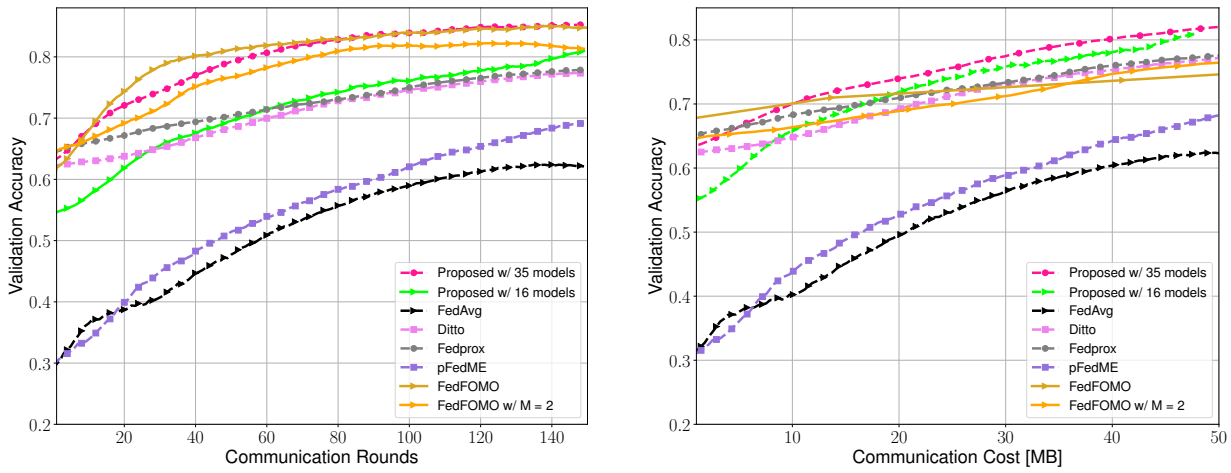
2.4.2 Personalization Performance

We now report the average accuracy over 5 trials attained by the different approaches. We also study the personalization performance of our algorithm when we restrain the overall number of personalized streams, namely the number of personalized models that are concurrently learned.

2.4.2.1 Multi-Model Baseline Algorithms

In Fig. 2.2 and Table 2.1, we report the average validation accuracy of the baseline algorithms that yield multiple personalized models, alongside FedAvg, Fedprox, SCAFFOLD, and local training. In the EMNIST label shift scenario (Fig.2.2a), we first notice that harvesting intelligence from the data sets of other users amounts to a large performance gain compared to the localized learning strategy. This indicates that data heterogeneity is moderate and collaboration is fruitful. Nonetheless, personalization can still provide gains compared to FedAvg. Our solution yields a validation accuracy which is increasing in the number of personalized streams. Allowing maximum personalization, namely a different model for each user, we obtain a 3% gain in the average accuracy compared to FedAvg. CFL is not able to transfer intelligence among different groups of users and attains performance similar to the FedAvg. This behavior showcases the importance of soft clustering compared to the hard one for the task at hand. We find that FedFOMO, despite excelling in case of strong statistical heterogeneity, fails to harvest intelligence in the label shift scenario. In Fig. 2.2b we report the personalization performance for the second scenario. In this case, we also consider the Oracle baseline, which corresponds to running 4 different FedAvg instances, one for each cluster of users, as if the 4 groups of users were known beforehand. Different from the previous scenario, the additional shift in the covariate space renders personaliza-

[2]Exception: the hyperparameters $\eta_{global} = \eta_{local} = 0.01$, batch-size $= 20$, $E = 20$ and mini-batch size $= 20$ were used for pFedMe, and $\eta = 0.01$, $E = 5$ for SCAFFOLD



(a) Validation Acc. vs Communication Rounds (b) Validation Acc. vs Communication Cost

Figure 2.3: Average Validation Accuracy across the different algorithms over the Stack-Overflow Sentiment dataset

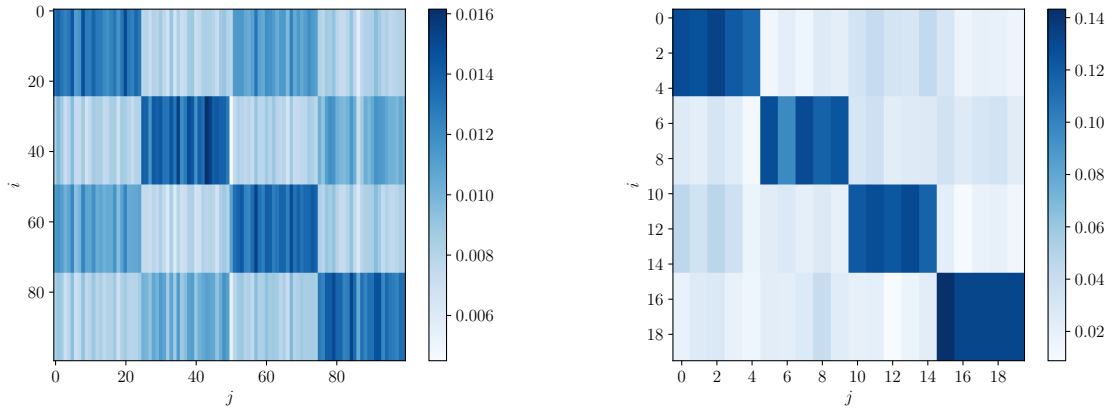
tion necessary to attain satisfactory performance. The Oracle training largely outperforms FedAvg. Furthermore, as expected, our algorithm matches the Oracle final performance when the number of personalized streams is 4 or more. Also, CLF and FedFOMO can correctly identify the 4 clusters. However, the former exhibits slower convergence due to the hierarchical clustering over time while the latter plateaus to a lower average accuracy level. Additionally, we note that FedFOMO with M set to 25 – the number of users in each cluster – fails to recognize the distinct clusters and shows poor generalization performance. We turn now to the more challenging CIFAR-10 image classification task. In Fig. 2.2c we report the average accuracy of the proposed solution for a varying number of personalized streams, the baselines, and the oracle solution. As expected, the label permutation renders collaboration extremely detrimental as the different learning tasks are conflicting. As a result, local learning provides better accuracy than FedAvg. On the other hand, personalization can still leverage data among clusters and provide gains in this case. Our algorithm matches the Oracle performance for a suitable number of personalized streams. This scenario is particularly suitable for hard clustering, which isolates conflicting data distributions. As a result, CFL matches the proposed solution. FedFOMO promptly detects clusters and therefore quickly converges, but it attains lower average accuracy compared to the proposed solution. On the other hand, Ditto and pFedMe perform relatively better than the aforementioned two approaches, given their personalization capabilities. However, they fall short while leveraging collaboration among users towards training the global model only and disregarding the potential generalization gain that could be achieved by enabling collaboration among statistically similar users towards refining their local personalized models.

We now focus on the text sentiment classification task. In particular, we assess the validation performance of different algorithms for a fixed number of communication rounds and communication cost. In this experiment, we consider the proposed scheme with full-fledged personalization in which each user has its personalized model, and the communication efficient version with several personalized streams equal to 16.

In Figure 2.3a we report the validation accuracy against the number of communication rounds. The proposed user-centric algorithm and FedFOMO exhibit faster convergence compared to other algorithms. In particular, the FedFomo algorithm converges faster during the initial stages of training but it plateaus at the same validation accuracy as the proposed algorithm. The situation changes if we consider a fixed communication budget as reported in Figure 2.3b. In this case, the proposed algorithm converges to a larger validation accuracy compared to all other algorithms.

2.4.2.2 Single-Model Baseline Algorithms

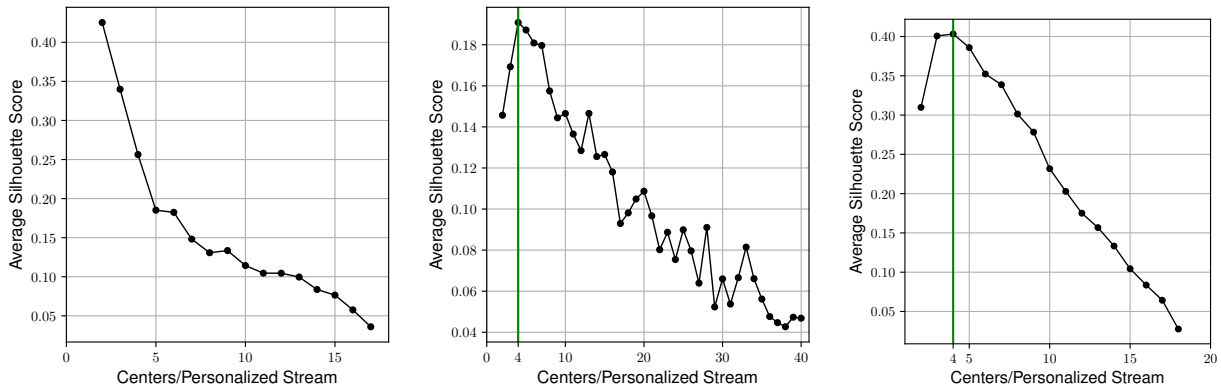
Despite that all algorithms that yield a single model (i.e. Fedprox and SCAFFOLD) excel in the label shift setting (Table 2.1), our proposed algorithm stands out in the two other scenarios. This stems from their inadequacy in addressing the conflicting nature of the available target tasks via a single global model in the other two proposed heterogeneous scenarios.



(a) EMNIST label & covariate shift (100 clients).

(b) CIFAR10 concept shift (20 clients).

Figure 2.4: Clusters formed by our proposed algorithm in the EMNIST label and covariate shift and CIFAR10 concept shift scenarios. Each 2D point denotes $w_{i,j}$: A dark blue point $w_{i,j}$ conveys a relatively large collaboration between user i and j.



(a) EMNIST label shift. (b) EMNIST label & covariate shift. (c) CIFAR10 concept shift.

Figure 2.5: Average silhouette scores of the k-means clustering in the three scenarios. In the last two scenarios, in which users inherently belong to 4 different clusters, the scores indicate the necessity of at least 4 personalized streams.

2.4.2.3 Average Worst Performance

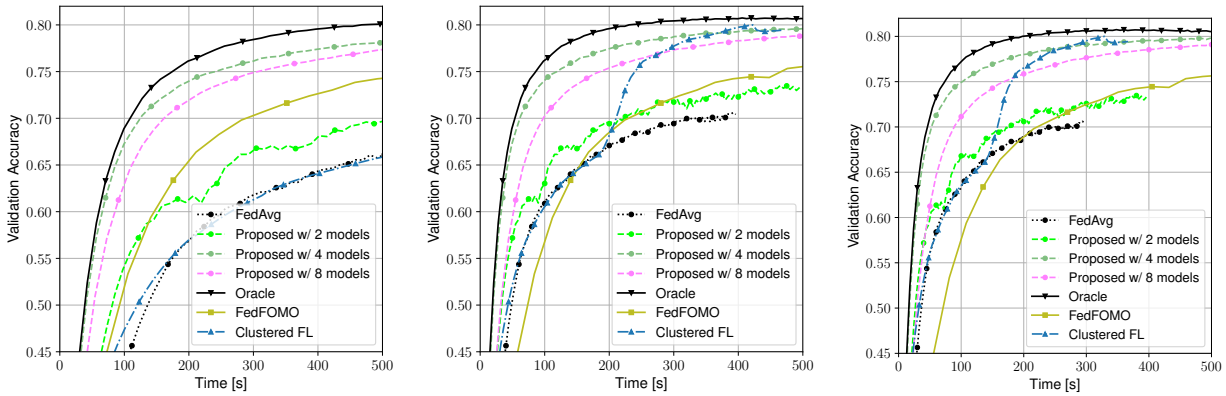
The performance reported so far is averaged over users and therefore fails to capture the existence of outliers performing worse than average. To assess the fairness of the training procedure, in Table 2.2 we report the worst user performance in the federated system across the different algorithms. The proposed approach produces models with the highest worst case in all three scenarios.

2.4.2.4 Inter-Cluster Collaboration

We illustrate the clustering performance of our proposed solution in the EMNIST co-variate shift and the CIFAR10 concept shift scenarios (Experiments two and three) with four clusters each in Fig. 2.4. Interestingly, we notice that in the EMNIST covariate shift experiment (Fig. 2.4a), our clustering algorithm can detect similarities among the different groups of users, leveraging inter-cluster collaboration among them, unlike hard clustering algorithms [61]. This stems from the fact that some digits and letters features are invariant to the 180° rotation applied (e.g. letters $X, Z, O, N,$ etc ... and the digits $\{0, 1, 8\}$).

2.4.3 Silhouette Score

In Fig. 2.5 we plot the average silhouette score obtained by the k-means algorithm when clustering the federated users based on the procedure proposed in Sec. 2.3.3. In the labels shift scenario, for which we have seen that a universal model performs almost as well as



(a) $\rho = 4, T_{\min} = T_{\text{dl}} = \frac{1}{\mu}$ (b) $\rho = 2, T_{\min} = T_{\text{dl}}, \frac{1}{\mu} = 0$ (c) $\rho = 1, T_{\min} = T_{\text{dl}}, \frac{1}{\mu} = 0$

Figure 2.6: Evolution of the average validation accuracy against time normalized w.r.t. T_{dl} for the three different systems.

the personalized ones, the silhouette scores monotonically decrease with m. In fact, in this simulation setting, a natural cluster-like structure among clients' tasks does not exist. On the other hand, in the covariate shift and the concept shift scenarios, the silhouette score peaks around $m = 4$. In Sec. 2.4.2 this has shown to be the minimum number of personalized models necessary to obtain satisfactory personalization performance in the system. This behavior of the silhouette score is expected and desired, in this case, the number of clusters matches exactly the number of underlying different tasks among the participants in FL that was induced by the rotation of the covariates and the permutation of the labels. We then conclude that the silhouette score provides meaningful information to tune the number of user-centric aggregation rules before training.

2.4.4 Communication Efficiency

Personalization comes at the cost of increased communication load in the downlink transmission from the PS to the federated user. To compare the algorithm convergence time, we parametrize the distributed system using two parameters. We define by $\rho = \frac{T_{\text{ul}}}{T_{\text{dl}}}$ the ratio between model transmission time in UL and DL. Typical values of ρ in wireless communication systems are in the $[2, 4]$ range because of the larger transmitting power of the base station compared to the edge devices. Furthermore, to account for unreliable computing devices, we model the random computing time T_i at each user i by a shifted exponential r.v. with a cumulative distribution function

$$P[T_i > t] = 1 - \mathbb{1}(t \geq T_{\min}) \left[1 - e^{-\mu(t - T_{\min})}\right],$$

where $T_{\min}$ represents the minimum possible computing time and $1/\mu$ is the average additional delay due to random computation impairments. Therefore, for a population of K devices, we then have

$$T_{\text{comp}} = \mathbb{E}\left[\max\{T_1, \ldots, T_K\}\right] = T_{\min} + \frac{H_K}{\mu},$$

where H_k is the k-th harmonic number. Accordingly, the communication round duration is

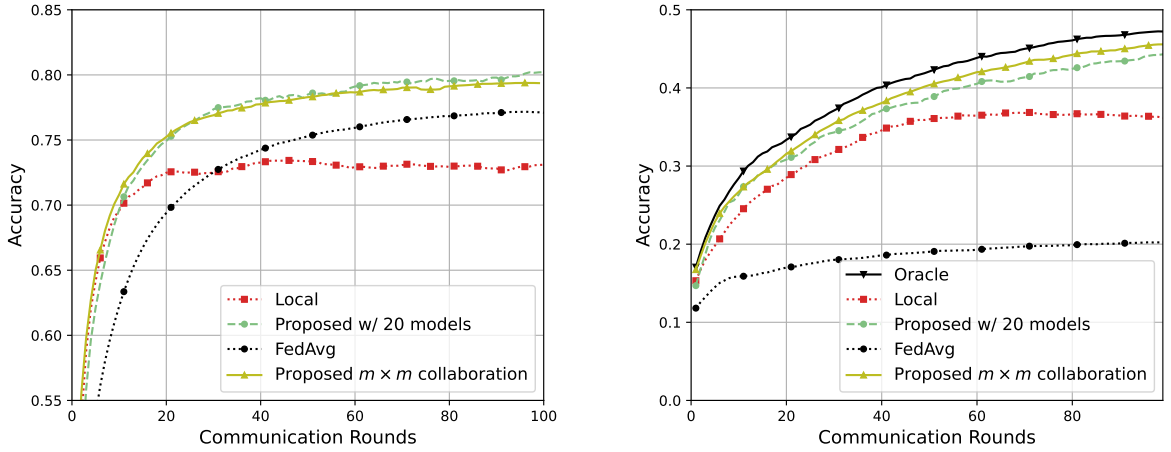
$$\begin{aligned} T_{\text{round}} &= mT_{\text{dl}} + KT_{\text{ul}} + T_{\text{comp}} \\ &= T_{\text{dl}}(m + \rho K) + T_{\text{comp}} \end{aligned}$$

where m denotes the number of personalized streams transmitted in the DL. Note that T_{dl} and T_{ul} implicitly depend on the model size and the available communication resources. On the other hand, the computation time T_{comp} is related to the computational power of the devices and the randomness in the completion time of the local training procedure, and it does not depend on the number of users or the number of personalized streams. This abstraction of the system is simple, yet effective, in capturing different types of distributed learning systems based on the reliability of its workers and the asymmetricity of the UL and DL communication.

To study the communication efficiency we consider the simulation scenario with the EMNIST data set with label and covariate shift. In Fig. 2.6 we report the time evolution of the validation accuracy in 3 different systems: wireless systems with slow UL $\rho = 4$ and unreliable nodes $T_{\min} = T_{\text{dl}} = \frac{1}{\mu}$, a wireless system with fast uplink $\rho = 2$ and reliable nodes $T_{\min} = T_{\text{dl}}$, $\frac{1}{\mu} = 0$ and a wired system $\rho = 1$ (symmetric UL and DL) with reliable nodes $T_{\min} = T_{\text{dl}}$, $\frac{1}{\mu} = 0$. In all plots, we normalize the time axis by T_{dl} to make the plots independent of the scale of this quantity. From the results, we note that the increased DL cost is negligible for wireless systems with strongly asymmetric UL/DL rates and in these cases, the proposed approach largely outperforms the baselines. In the case of more balanced UL and DL transmission times $\rho = [1, 2]$ and reliable nodes, it becomes necessary to properly choose the number of personalized streams to render the solution practical. Nonetheless, the proposed approach remains the best even in this case for $m = 4$. Note that FedFOMO incurs a high communication cost as personalized aggregation is performed on the client side.

2.4.5 Comparison with Parallel User-centric FL

Even if the proposed user-centric aggregation rules outperform state-of-the-art personalized FL approaches, the resulting optimization procedure departs from the standard FL in the



(a) EMNIST label shift

(b) CIFAR10 label & covariate shift

Figure 2.7: Comparison between the proposed algorithm and the parallel user-centric federated learning approach. The validation accuracy is averaged over 5 experiment runs.

following sense: In the typical FL framework, at each communication round t, the PS aggregates the models that were locally trained, at each participating device, starting from the same launch model θ^{t-1}. On the contrary, according to our proposed framework, devices may optimize different models depending on the specific user-centric aggregation rule they have been assigned. This design choice is motivated by the assumption that the models of statistically similar propagate towards the same neighborhood of the parameter space during the optimization [64]. As a result, in the proposed aggregation rule, models that are largely weighted, therefore associated with similar users, were locally optimized starting from similar initial parameters. Furthermore, if we were to adhere to the traditional FL procedure, and produce an exact minimizer of (2.4), we would have to run in parallel as many FL instances as the number of personalized streams K and incur a K-fold computation and uplink communication load.

To assess the quality of our assumption, we consider running in parallel K collaborative FL instances employing the proposed user-centric weights and solving exactly (2.4) for each different aggregation rule. At each communication round, each user also optimizes the user-centric models of the other $K-1$ personalized streams which are then used at the PS server to apply the user-centric aggregation rules

$$\theta_i^t \leftarrow \sum_{j=1}^{K} w_{i,j} \theta_{i,j}^{t-1/2}, \qquad \text{for } i = 1, 2 \cdots, K. \tag{2.13}$$

Note that the aggregation rule in (2.13) is different from the one in (2.2), as $\theta_{i,j}^{t-1/2}$ denotes the update of user j to the model of user i obtained by locally optimizing θ_j^{t-1}.

We experiment using the EMNIST data set with label shift and the CIFAR10 data set with covariate and label shift. We set $K = 20$ and use the same neural network model and settings indicated in Sec. 2.4. In Fig. 2.7, we report the performance of the parallel collaborative FL approach compared to our personalization strategy. For reference, we also report the performance of the FedAvg, local learning, and Oracle baselines. First, we notice that the fully collaborative solution performance serves as an upper bound to our personalization approach and that the oracle slightly outperforms the fully collaborative approach, which highlights the sub-optimality of our heuristic weighting scheme. However, the slight performance gain of the fully collaborative approach compared to our personalization strategy comes at the expense of m times larger uplink communication load and computation cost at each edge device. These empirical results support our assumption: Even if the updated models are trained starting from different points in the parameter space at each communication round, the user-centric weighting scheme can direct statistically similar models in a neighborhood across the loss landscape during training.

2.4.6 Variance Computation: Mini-batch Size

As mentioned in section 2.3.1, the mini-batch sizes chosen to calculate the variances play an essential role in the quality of the derived weights, i.e. their ability to couple statistically similar users in the federated system. In Fig. 2.8, we report the validation accuracy attained in an EMNIST label shift and covariate shift experiments. In both experiments, we randomly split 100k EMNIST data points across 100 users, i.e. 1000 samples per user. Heterogeneity is introduced in both settings akin to the "label shift", and "label and covariate shift" settings in section 2.4.1, respectively. We vary the mini-batch sizes used to calculate the variances from between 100 and 660 samples to explore the effect of this parameter on the validation accuracy of our personalization strategy in both scenarios. First, we note that according to (2.10), decreasing the mini-batch size would yield an increase in the variance value as a result of the noisy gradients obtained compared to the average gradient computed over each user data set. In this case, our proposed aggregation rule renders similar to FedAvg, enabling collaboration among all users in the federated system, while still managing to softly couple statistically similar users under the assumption that $\mathbb{E}_{\mathcal{D}_i,\mathcal{D}_j\sim P_i}[\Delta_{i,j}] \leq \mathbb{E}_{\mathcal{D}_i\sim P_i,\mathcal{D}_k\sim P_k}[\Delta_{i,k}]$ given that $d_{\mathcal{F}}(P_i, P_k) > 0$. This condition is favorable in the label shift setting while being detrimental to the extremely heterogeneous co-variate shift experiment, as it enables collaboration among users with competing tasks. Our claim is verified by the performance attained by our personalization rule in Fig. 2.8, achieving a high validation accuracy in the label shift setting while suffering in the co-variate shift exper-

iment with a performance comparable to that of FedAvg attained in Fig. 2.2b ($\sim 70.5\,\%$). However, as we increase the mini-batch size, the variances converge toward zero and our personalization algorithm degenerates to local training which is detrimental to both settings. Therefore, we conclude that the mini-batch size can be seen as a hyper-parameter for our algorithm, to be tuned according to the local data set size and the type of heterogeneity present across the learners. In our experiments presented in Fig. 2.2, we set the mini-batch size $n = 100$ for the label shift experiment, and $n = |\mathcal{D}|/3$ for the other two EMNIST co-variate and CIFAR10 concept shift experiments, where $|\mathcal{D}$ denotes the local data set size of each user.

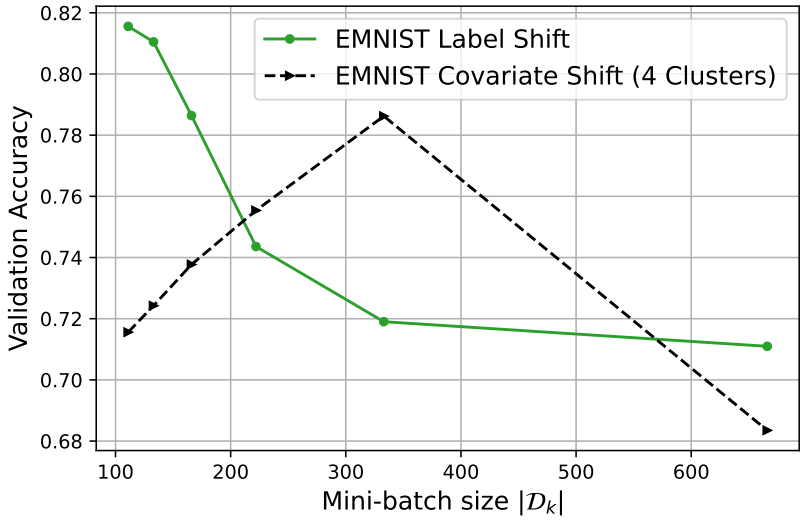


Figure 2.8: Effect of the mini-batch sizes on the maximum validation accuracy attained: A proxy to the quality of the calculated collaboration coefficients

2.5 Conclusion

In this chapter, we have presented a novel FL personalization framework that exploits multiple user-centric aggregation rules to produce personalized models. The aggregation rules are based on user-specific mixture coefficients that can be computed during one communication round before federated training and are designed based on an excess risk upper bound of the weighted aggregated loss minimizer. Additionally, to limit the communication burden of personalization, we have proposed a K-means clustering algorithm to lump together users based on their similarity and serve each group of similar users with a single personalized model. To effectively trade communication resources for personalization capabilities, we have proposed to use the silhouette score to tune the number of user-centric aggregation

rules at the PS before training commences. We have studied the performance of the proposed solution across different tasks. Overall, our solution yields personalized models with higher testing accuracy while at the same time being more communication-efficient compared to other state-of-the-art personalized FL baselines.

In this chapter, a static orchestrator is considered. In the subsequent one, we explore the advantages of using a mobile orchestrator in federated learning, with a particular focus on the gains in terms of the reliability of the wireless communication channels between devices and the orchestrator, the learning performance, and learning fairness among different communities of devices, each defined by a distinct task.

Part II

Federated Learning Using Mobile Orchestrators

Chapter 3

UAV-Aided Multi-Community Federated Learning

In this third chapter, we focus on incorporating remote IoT devices in the intelligent edge by leveraging a mobile orchestrator in lieu of the standard static one. The use of mobile agents has been explored in scenarios ranging from cooperative vehicular networks [79], to 3D-mobile agents such as UAVs supporting emergency and disaster response [80] and expanding the coverage area of ground base stations by acting as a relay to ground devices [81]. However, two main limitations arise in such deployments: first, the algorithmic complexity imposed by dynamic mobility patterns and the wireless channels, and second, the limited energy capacity constraining operational longevity. Nevertheless, mobility grants several advantages that can overcome challenges faced in static deployments. For instance, intelligent 3D mobility patterns enable mobile relays to dynamically shape wireless channel distributions, mitigating obstruction and improving link reliability for users, providing means to counter system heterogeneity in FL with respect to channel heterogeneity, which can otherwise lead devices to drop or not be properly scheduled, resulting in a model biased towards devices with strong channels conditions. Additionally, mobile relays such as UAVs can operate in nomadic modes, moving among and dwelling in optimized locations to reduce energy expenditure and prolong autonomy.

Building on the previous insights, in this chapter, we investigate the problem of an online trajectory design for the UAV in a federated learning setting where several communities exist, each defined by a unique task to be learned. In this setting, spatially distributed devices belonging to each community collaboratively contribute towards training their community model via wireless links provided by the UAV. Accordingly, the UAV acts as a mobile orchestrator coordinating the transmissions and the learning schedule among the devices in each community, intending to accelerate the learning process of all tasks.

The utilization and incorporation of UAVs have augmented significantly thanks to their fast on-demand deployment and their inherent maneuvering capabilities. Their role evolved to complement or even substitute static access points in multiple areas [82]. More recently, the usage of UAVs to facilitate FL model training of ground and airborne units has gained significant attention. In [83], a UAV trajectory path planning problem has been formulated in order to govern the participation of the straggling devices during training. Their solution

optimizes the UAV trajectory to balance the local model updates computation and transmission times at each learner to fit in each communication round time slot and guarantee the widest participation of devices. In [84], a joint power allocation and scheduling design is proposed to optimize the convergence rate of FL training among a swarm of UAVs.

While orchestrating FL training is not a typical use-case for UAVs in urban areas — where wireless connectivity is guaranteed and static access points (APs) can act as robust front-haul orchestrators —, it sounds appealing to deploy UAVs as FL orchestrators for IoT devices in remote areas where a multitude of data are expected to be generated by massive numbers of machines and sensors. The generated data are envisioned to help in the management and optimization of the industrial and agricultural economy by serving as a training data feed for predictive machine learning models. In this setting, training models centrally by pooling the massive amounts of data from edge IoT devices may inflict a hit over their energy budget, especially if data are high dimensional. Additionally, the deployment of static APs in rural unpopulated areas is costly, and inefficient as they are mostly under-utilized. Moreover, the static nature of APs does not guarantee a good wireless channel quality to the IoT edge devices, worsening their transmission times and consequently expanding their energy expenditure. Alternatively, devices can rely on decentralized machine learning (ML) schemes over wireless device-to-device (D2D) networks to train ML models [85]. However, due to the limited communication range of edge devices, their connectivity is not always guaranteed. As a result, distributed D2D ML algorithms may perform poorly.

Unlike the previous works, we investigate an online path planning problem of a UAV missioned to orchestrate the FL training among devices belonging to different communities. Each community consists of statistically heterogeneous devices (i.e with non-IID datasets) that wish to train a model corresponding to their unique community task. The model updates are transmitted by the devices through a lossy channel, therefore, the successful participation of all devices during each training round is not guaranteed. Our goal is to establish learning fairness among the different community tasks/models during training, therefore, guaranteeing a desirable inference performance of all the different tasks at the end of training. This is achieved by employing a heuristically derived metric that is able to capture the training performance and the scheduling requirements of the different tasks throughout the course of training. Capitalizing on this metric, we devise a surrogate optimization problem which is solved by the UAV at each communication round, to dynamically schedule devices and optimize the UAV trajectory to successfully pool their model updates. Our solution aims at steering the scheduling and the UAV control in favor of users belonging to communities that are seen to lag behind in terms of convergence and as a result, establish learning fairness among all the tasks.

3.1 System Model

We consider a scenario where a UAV acts as a flying orchestrator for FL training across different communities of devices in a service area. The considered area is composed of a total of C communities. Communities can be viewed as distinct groups of devices that desire to train personalized models akin to those delineated in the preceding chapter, or perhaps entirely different tasks. Each community c consists of $|\mathcal{K}_c|$ ground devices where $\mathcal{K}_c$ is a set of devices' index in community c and $|.|$ denotes the cardinality function. The total number of devices in the system is $\sum_{c=1}^{C} |\mathcal{K}_c| = K$. The devices within each community wish to collaboratively train a supervised learning model to fit to their corresponding community task in a federated manner. We emphasize that the models that we wish to train at the different communities are unrelated, hence, there is no collaboration among devices that are not in the same community. The k-th ground device, is located at $\mathbf{u}_k = [x_k, y_k]^T \in \mathbb{R}^2$. By no means, the ground-level device assumption is restrictive and the proposed solution can in principle be applied to a scenario where the devices are located in 3D. The UAV's mission consists of M communication rounds. During each communication round $m \in [1, M]$ the UAV collects the locally optimized models from the devices (yet to be optimally scheduled later) in different communities. At the end of each round, the UAV aggregates the collected model updates from the devices of each community, to obtain new community-specific global models. Each global model is then broadcasted back to its corresponding community devices, therefore initiating a new communication round. The UAV is characterized by a battery budget which allows it to maneuver for a distance of $\bar{L}_{total}$ meters with a constant velocity of v m/s. Moreover, the UAV is assumed to fly at an altitude $z(t)$ above the ground, and the horizontal location of the UAV at time t is denoted by $\mathbf{v}(t) = [x(t), y(t)]^T$. We assume that the UAV is equipped with a GPS, hence, its location is known at each time stamp t. We do not consider the optimization over the UAV altitude and assume that the UAV flies at a fixed altitude $z(t) = H$. Since controlling the UAV in continuous time is cumbersome, we discretize each communication round into N time steps. Hence, the UAV trajectory is defined by a set of discrete locations $\{\mathbf{v}[n] = [x[n], y[n]]^T, n \in [1, N]\}$, where each two consecutive UAV locations are connected with a straight line.

3.1.1 Channel Model

We define the wireless channel gain between device k and the UAV at time step n as a log-normal fading channel under Additive White Gaussian Noise (AWGN) , given by :

$$h_{k,s}[n] = \frac{\beta_s}{d_k[n]^{\alpha_s}} \xi_s, \tag{3.1}$$

where ξ_s denotes the shadowing component that is modeled as a Log-normal distribution $\xi_s \sim \text{Lognormal}(0, \sigma_s^2)$. $s \in \{\text{LoS}, \text{NLoS}\}$ emphasizes the strong dependence of the propagation parameters on the Line-of-Sight (LoS) or Non-Line-of-Sight (NLoS) segments. β_s is the average gain at the reference point $d = 1$ meter, and $d_k[n] = \sqrt{\|\mathbf{u_k} - \mathbf{v}[n]\|^2 + H^2}$ is the distance between the ground device k and the UAV at step n. Note that the channel gain model only includes the large-scale fading effects and does not directly account for small-scale fading as we rather average over them. This approximation is valid because the small-scale fading has a negligible impact on the trajectory optimization and user scheduling problems addressed later. The LoS event probability of the link between the UAV at time step n and device k is given by [86] :

$$\rho_k[n] = \frac{1}{1 + \exp(-a_1\theta_k[n] + a_2)}, \tag{3.2}$$

where $\theta_k[n] = \arctan(\frac{H}{\|\mathbf{u_k} - \mathbf{v}[n]\|})$ is the elevation angle, parameters $\{a_1, a_2\}$ denote the model coefficients of the LoS probability which depends on the structure of the city and can be obtained according to [86].

Without loss of generality, we assume that the model updates are transmitted in packets across a lossy channel and that the UAV has enough power to transmit the global models in the downlink for all devices with an average packet success rate equal to one, from every point inside the service area. In the next subsection, we derive the expression of the average Packet Error Rate (PER) experienced by the ground devices while transmitting their updates in the uplink.

3.1.2 Average Packet Error Rate

We define $q(\gamma)$, the instantaneous PER, representing the probability of packet detection error at a given signal-to-noise ratio (SNR) γ. We assume that the packets are erroneously detected with a probability $0 \leq q(\gamma) \leq 1$ if the instantaneous SNR resides below a threshold γ_0, and $q(\gamma) = 0$, otherwise. The instantaneous SNR experienced by the UAV when device k is in $s \in \{\text{LoS, NLoS}\}$ at step n, is given by :

$$\gamma_{k,s}[n] = \frac{P_k\, h_{k,s}[n]}{N_0}, \tag{3.3}$$

where P_k is the transmission power of device k, and N_0 is the noise power level. In accordance with (3.1), $\gamma_{k,s}[n]$ follows a log-normal distribution $\gamma_{k,s}[n] \sim \text{Lognormal}(\mu_{k,s}[n], \sigma_s^2)$, where $\mu_{k,s}[n] = \log(\frac{P_k\, \beta_s}{N_0\, d_k[n]^{\,\alpha_s}})$. We denote by $g_{\gamma_{k,s}}(\gamma)$ the probability density function of $\gamma_{k,s}[n]$.

The average PER experienced by device k during the model transmission in the UL at

UAV location at time step n can be written as :

$$\bar{q}_k[n] = \mathbb{E}_s \left[\mathbb{E}_{\gamma_{k,s}[n]} \left[q(\gamma_{k,s}[n]) \right] \right]. \tag{3.4}$$

The inner expectation is over the instantaneous SNR randomness, while the outer expectation over s is with respect to the channel LoS/NLoS segments probabilities. Hereafter we drop the time step index n for ease of notation. For a given time step n, averaging over the LoS/NLoS probabilities, we can rewrite (3.4) as :

$$\begin{aligned} \bar{q}_k &= \rho_k \, \mathbb{E}_{\gamma_{k,\text{LoS}}} \left[q(\gamma_{k,\text{LoS}}) \right] + (1-\rho_k) \, \mathbb{E}_{\gamma_{k,\text{NLoS}}} \left[q(\gamma_{k,\text{NLoS}}) \right] \\ &\overset{(a)}{\leq} \rho_k \int_0^{\gamma_0} g_{\gamma_{k,\text{LoS}}}(\gamma) d\gamma + \bar{\rho}_k \int_0^{\gamma_0} g_{\gamma_{k,\text{NLoS}}}(\gamma) d\gamma \\ &= \rho_k \, \phi\left(\gamma_0, \gamma_{k,\text{LoS}}\right) + \bar{\rho}_k \, \phi\left(\gamma_0, \gamma_{k,\text{NLoS}}\right), \end{aligned} \tag{3.5}$$

where $\bar{\rho}_k = (1-\rho_k)$. Step (a) holds given that $q(\gamma) \leq 1, \forall \, \gamma \in (0, \gamma_0)$. $\phi\left(\gamma_0, \gamma_{k,s}\right) = \mathbb{P}\left(\gamma_{k,s} < \gamma_0\right)$ is the cumulative density function of $\gamma_{k,s}$, and is written as :

$$\phi\left(\gamma_0, \gamma_{k,s}\right) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{\log(\gamma_0) - \mu_{k,s}}{\sigma_s \sqrt{2}}\right)\right], \tag{3.6}$$

where $\text{erf}(x)$ is the error function.

3.2 Community FL and UAV trajectory Modelling

In this section, we describe the Federated Learning training process across devices belonging to different communities.

3.2.1 Classical Federated Learning

In classical FL settings [24], the goal is to collaboratively train a model across different learners to find a global model parameterized by ϑ_c that minimizes the expected local risk over the learners datasets. Given a set of different communities and their corresponding tasks, this reflects as finding the predictor of each community c that minimizes :

$$L(\vartheta_c) = \sum_{k \in \mathcal{K}_c} p_k \ell_k(\vartheta_c), \quad \forall c, \tag{3.7}$$

where $\vec{p} = (p_1, \ldots, p_{|\mathcal{K}_c|})$ is a weighting scheme such that $\sum_{k \in \mathcal{K}_c} p_k = 1$.

3.2.2 UAV-aided Orchestration

Unlike traditional FL implementations, a mobile UAV is deployed to orchestrate the training of the different community tasks available. In this setting, a communication round starts as the UAV finds an optimized trajectory as well as schedules a set of devices from the different available communities to participate in the training. Then, the global models are broadcasted to all communities during the DL phase with a PER equal to zero, as explained in Sec. 3.1.1. The models are then optimized locally at the scheduled devices. Model updates are then sent back in the UL phase, as the UAV maneuvers following the optimized path found earlier, while governing a favorable channel condition for the scheduled devices, and consequently, a low packet error transmission rate, to successfully gather their updates.

To limit the energy spent by the UAV during the UL phase, we limit the total distance that can be travelled by the UAV during each round to $\bar{L}_{max}$ meters. We assume that the ground devices are served by the UAV in a Time-Division Multiple Access (TD-MA) manner in the UL, and that a maximum of $\overline{K}$ devices can be served by the UAV at each time step. Note that each device can be scheduled at most once during each round, to implicitly preserve their energy, especially when experiencing unfavorable channel conditions that induce high packet error rates.

3.3 Accounting for the learning performance

Accounting for the learning performance is essential to the online path planning optimization problem that we wish to solve. In a single community vanilla FL setting, the for-seen advantage of sampling a device during a communication round is proportional to its dataset sizes. However, this is partially true in a multi-community FL setting, as scheduling (i.e device sampling) and resource allocation should be carried out while accounting to the relative learning performance of the different available communities. Particularly in our case, scheduling and UAV trajectory planning should be considered to insure a low PER for devices with tasks that are seen to fall behind other communities in terms of convergence.
Theoretically, the convergence rate of the FL models can be quantified based on the level of heterogeneity of the datasets available at the learners, their participation rate, and the model architecture. Unfortunately, computing the convergence rates in practice is not trivial, especially in settings where the loss landscape is non-convex and datasets are heterogeneously distributed [26]. Consequently, we choose the Coefficient of Variation (CoV), computed periodically during training, over the average validation accuracy of each community devices, as a metric of choice, to quantify the training performance of each community model.
The motivation behind using the CoV is its ability to capture the current model performance difference among the devices belonging to the same community, compared to their

average performance, ergo convey how well the current model performs over the devices' local datasets. Consequently, the CoV being calculated periodically offers a measure of goodness of the available different community models during training, which we can rely on to compare the training performance of the underlying tasks and quantify their scheduling and resource allocation requirements.

The CoV of each community c, is updated by the UAV every $\bar{t}$ communication rounds, and is given by

$$\psi_c = \frac{\sqrt{\sum_{k\in\mathcal{K}_c}(\varepsilon_k - \bar{\varepsilon}_c)^2}}{\bar{\varepsilon}_c} \quad \forall c, \tag{3.8}$$

where ε_k is the average validation accuracy for device k computed over $\bar{t}$ rounds as follows :

$$\varepsilon_k = \frac{1}{\bar{t}} \sum_{j=m-\bar{t}}^{m-1} \varepsilon_k(\vartheta_j). \tag{3.9}$$

$\varepsilon_k(\vartheta_j)$ is the validation accuracy computed locally at device k over the validation dataset, using the global model parameterized by ϑ_j transmitted in the DL during round j. $\bar{\varepsilon}_c$ denotes the weighted average validation accuracy over all devices in community c which is given by :

$$\bar{\varepsilon}_c = \sum_{k\in\mathcal{K}_c} p_k \varepsilon_k, \text{ such that } p_k = \frac{|\mathcal{D}_k|}{\sum_{i\in\mathcal{K}_c}|\mathcal{D}_i|}, \tag{3.10}$$

where $|\mathcal{D}_k|$ is the training data set size of device k.

We assume that ε_k is transmitted alongside the local models during the UL phase at each round by the scheduled devices. However, if ε_k is not received during the round in which the CoV is updated, the last successfully received value is considered for the update.

3.4 UAV Trajectory Planning

In this section, we seek to find an optimized UAV trajectory during each communication round, in order to improve the overall learning performance within the communities.

We introduce a surrogate optimization problem which enables us to optimize the UAV trajectory for collecting the model updates from a subset of devices of each community to improve the performance of learning. The objective function proposed is equivalent to that proposed in [36] for static AP, while we also aim at providing learning fairness among the different communities in a UAV-aided federated setting. We define the surrogate

optimization problem at each communication round as follows

$$\max_{\mathcal{V},\mathcal{W}} \quad \sum_{n\in[1,N]} \sum_{k\in[1,K]} \omega_k[n]\,(1-\bar{q}_k[n])\,\delta_k \tag{3.11a}$$

$$\text{s.t.} \quad \sum_{n\in[1,N]} \omega_k[n] \leq 1, \forall k, \tag{3.11b}$$

$$\sum_{k\in[1,K]} \omega_k[n] \leq \overline{K}, \forall n, \tag{3.11c}$$

$$\sum_{n=1}^{N-1} \|\mathbf{v}[n+1]-\mathbf{v}[n]\| \leq \bar{L}_{max}, \tag{3.11d}$$

$$\mathbf{v}[1] = \mathbf{v}_{\mathrm{I}}, \tag{3.11e}$$

where $\mathcal{V} = \{\mathbf{v}[n], \forall n\}$ is the UAV trajectory, and $\mathcal{W} = \{\omega_k[n] \in \{0,1\}, \forall n, k\}$ is the set of scheduling binary variables where $\omega_k[n]$ indicates if device k is scheduled at time step n. Constraint (3.11b) implies that a device can only be served once by the UAV at each communication round, and (3.11c) indicates the maximum of $\overline{K}$ devices can be served by the UAV at each time step. Constraint (3.11d) is the maximum length of the UAV trajectory allowed in each round, and $\mathbf{v}_{\mathrm{I}}$ is the staring location at each round (i.e. $\mathbf{v}_{\mathrm{I}}$ can be the location of the UAV at the end of the previous communication round to guarantee a continuous trajectory throughout the entire mission). δ_k captures the importance of participation of device k during the current round. δ_k is a function of device k weight p_k given in (3.10), and the CoV of the community which it belongs to ψ_c. Moreover, in order to guarantee fairness over the participation of devices throughout the course of training, we impose an extra weight λ ($\lambda > 1$), for devices that have failed to transmit their model updates successfully, or have not been scheduled during the previous round. Hence, the importance of scheduling device $k \in \mathcal{K}_c$ at each round is given by

$$\delta_k = \begin{cases} p_k\,\psi_c\,\lambda, & \text{if } \forall\, n,\ w_k[n] = 0 \text{ during the previous round,} \\ p_k\,\psi_c, & \text{Otherwise.} \end{cases} \tag{3.12}$$

Solving problem (3.11) is challenging since the exact close form of $\bar{q}_k[n]$ is not available. To solve this problem we first simplify the objective function by finding an approximate for $\bar{q}_k[n]$. Since $\bar{q}_k[n]$ comprises the erf(.) function, a closed-form approximation can be obtained by using the logistic function. Therefore, an approximate for $\bar{q}_k[n]$ is given by

$$\bar{q}_k[n] \approx \tilde{q_k}[n] \triangleq \frac{1}{1+\exp(b_1\theta_k[n]+b_2)}, \tag{3.13}$$

where $\theta_k[n]$ is the elevation angle between the UAV at time step n and the k-th device. The parameters $\{b_1, b_2\}$ can be found using regression techniques on the samples taken from (3.5) for different UAV and device locations. For further simplification, we also relax the binary scheduling variables $\mathcal{W}$ into continuous variables. Hence, problem (3.11) by substituting $\tilde{q_k}[n]$ and relaxed scheduling variables can be reformulated as follows

$$\max_{\mathcal{V},\mathcal{W}} \sum_{n\in[1,N]} \sum_{k\in[1,K]} \omega_k[n]\,(1 - \frac{1}{1+\exp(b_1\theta_k[n]+b_2)})\,\delta_k \tag{3.14a}$$

$$\text{s.t.} \quad (3.11b), (3.11c), (3.11d), (3.11e), \tag{3.14b}$$

$$0 \le \omega_k[n] \le 1, \forall n, k. \tag{3.14c}$$

However, having simplified the objective function, this problem is still difficult to solve as it is a non-convex optimization problem. To tackle this difficulty, we split the optimization problem (3.14) into two sub-problems of device scheduling and UAV trajectory optimization. In the first phase, the devices are scheduled while fixing the UAV trajectory. Then in the second phase, given the scheduled devices from the first phase the UAV trajectory is optimized. The algorithm iterates between two phases until convergence.

3.4.1 Device Scheduling

For a given UAV trajectory $\mathcal{V}$, the ground device scheduling can be optimized as follows

$$\max_{\mathcal{W}} \quad \sum_{n\in[1,N]} \sum_{k\in[1,K]} \omega_k[n]\,(1 - \tilde{q_k}[n])\,\delta_k \tag{3.15a}$$

$$\text{s.t.} \quad (3.11b), (3.11c), (3.14c). \tag{3.15b}$$

This problem is a standard Linear Program (LP) and can be solved by using any optimization tools such as CVX [87].

3.4.2 Trajectory Optimization

Having optimized the scheduling variables $\mathcal{W}$, the optimal UAV trajectory can be obtained by solving the following optimization

$$\max_{\mathcal{V}} \sum_{n\in[1,N]} \sum_{k\in[1,K]} \omega_k[n]\,(1 - \frac{1}{1+\exp(b_1\theta_k[n]+b_2)})\,\delta_k \tag{3.16a}$$

$$\text{s.t.} \quad (3.11d), (3.11e). \tag{3.16b}$$

This problem is still non-convex. By introducing slack variables $\mathcal{S} = \{S_k[n], \forall n, k\}$, $\mathcal{T} = \{\theta_k[n], \forall n, k\}$, and $\mathcal{R} = \{r_k[n], \forall n, k\}$ problem (3.16) can be rewritten as

$$\max_{\mathcal{V},\mathcal{S},\mathcal{T},\mathcal{R}} \sum_{n\in[1,N]} \sum_{k\in[1,K]} \omega_k[n] \left(1 - \frac{1}{1+S_k[n]}\right) \delta_k \tag{3.17a}$$

$$\text{s.t.} \quad S_k[n] \leq \exp(b_1\theta_k[n] + b_2), \forall n, k, \tag{3.17b}$$

$$\theta_k[n] \leq \arctan(\frac{H}{r_k[n]}), \forall n, k, \tag{3.17c}$$

$$r_k[n] = \|\mathbf{v}[n] - \mathbf{u}_k\|, \forall n, k, \tag{3.17d}$$

$$(3.11d), (3.11e). \tag{3.17e}$$

Without loss of optimality the constraints (3.17b) and (3.17c) can be met with equality. It can be verified that objective function (3.17a) is a concave function for $S_k[n] \geq 0$, however, problem (3.17) is still non-convex. To solve this problem efficiently, we employ the sequential convex programming techniques by using a local first-order Taylor estimation to convert the problem into a convex form. To do so, it can be shown that the right hand side functions in constraints (3.17b), (3.17c), and (3.17d) are convex functions of $\theta_k[n]$, $r_k[n]$, and $\mathbf{v}[n]$, respectively, when $\theta_k[n], r_k[n] \geq 0$. Since every convex function can be lower-bounded by its first-order Taylor approximation, a lower bound of problem (3.17) is given by

$$\max_{\mathcal{V},\mathcal{S},\mathcal{T},\mathcal{R}} \sum_{n\in[1,N]} \sum_{k\in[1,K]} \omega_k[n] \left(1 - \frac{1}{1+S_n[k]}\right) \delta_k \tag{3.18a}$$

$$\text{s.t.} \quad S_k[n] \leq \tilde{S}(\theta_k[n]), \forall n, k, \tag{3.18b}$$

$$\theta_k[n] \leq \tilde{\theta}(r_k[n]), \forall n, k, \tag{3.18c}$$

$$r_k[n] \geq \tilde{r}(\mathbf{v}[n]), \forall n, k, \tag{3.18d}$$

$$S_n[k], \theta_k[n], r_k[n] \geq 0, \forall n, k, \tag{3.18e}$$

$$(3.11d), (3.11e). \tag{3.18f}$$

where $\tilde{S}(\theta_k[n]), \tilde{\theta}(r_k[n])$, and $\tilde{r}(\mathbf{v}[n])$ are the local first-order Taylor approximation of functions in the right hand side of constraints (3.17b), (3.17c), and (3.17d) with respect to $\theta_k[n], r_k[n]$, and $\mathbf{v}[n]$, respectively.

3.4.3 Overall Algorithm and Convergence

According to the preceding analysis, now we propose an iterative algorithm to solve the optimization problem (3.14) by applying the alternating optimization method. We split

the problem into two phases i) user scheduling, and ii) UAV trajectory optimization. In the first phase, the devices are scheduled while keeping the UAV trajectory fixed. Then in the second phase, given the optimized scheduling variables $\mathcal{W}$ from the first phase, the UAV trajectory is optimized. The algorithm iterates between two phases until convergence. Moreover, the obtained solution in each iteration is used as the input for the next iteration. The convergence of the aforementioned algorithm is guaranteed since, at each phase of device scheduling and the UAV trajectory optimization, the objective function is optimized and does not decrease compared to the previous phase which results in convergence to at least a local optima. The details of the proof are omitted for the sake of the limited space.

3.4.4 Trajectory Initialization

Due to the non-convexity of problem (3.14), the iterative solution proposed above will converge to a local minima. Therefore, it is of a crucial importance to suitability initialize the UAV trajectory. To do so, we use a low complexity graph-based algorithm to find a good candidate for the initial UAV trajectory. We define the graph $G(\mathcal{N}, \mathcal{E})$ comprising a set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E}$. The graph nodes includes a set of UAV locations where the UAV can fly to and is defined as $\mathcal{N} = \left\{\boldsymbol{\nu}_k = [x_k, y_k, H]^T, k \in [1, K]\right\}$. This implies that the UAV has to fly to the location at top of the devices at a fixed altitude H. We also add $\mathbf{v}_\mathrm{I}$ as node zero to $\mathcal{N}$. The graph edges consists of all the possible combinations of the segments between the nodes and is defined as $\mathcal{E} = \{e_{i,j} = (\boldsymbol{\nu}_i, \boldsymbol{\nu}_j), \boldsymbol{\nu}_i, \boldsymbol{\nu}_j \in \mathcal{N}, \boldsymbol{\nu}_i \neq \boldsymbol{\nu}_j\}$. We assign a reward $r_{i,j}$ to each edge $e_{i,j}$ of the graph defined

$$
\begin{aligned}
r_{i,j} := \max_{\omega_k[j], k \in [1,K]} \quad & \sum_{k \in [1,K]} \omega_k[j] \left(1 - \tilde{q}_k[j]\right) \delta_k \\
\text{s.t.} \quad & \sum_{k \in [1,K]} \omega_k[j] \leq \overline{K},
\end{aligned}
$$

where the index j indicates when the UAV is at location $\boldsymbol{\nu}_j \in \mathcal{N}$. Then a trajectory is defined as set of connected edges starting from $\mathbf{v}_\mathrm{I}$ in graph G that maximized the sum rewards while satisfying the maximum UAV trajectory length constraint (3.11d) and the constraint (3.11b). To solve this problem, a greedy algorithm is used where an optimized initial trajectory is found within the graph iteratively.

3.5 Experiments

In our simulations, the UAV flies at constant altitude $H = 60\,m$ with a constant velocity $v = 20\,m/s$, a travel budget $\bar{L}_{total} = 40\,km$, and $\bar{L}_{max} = 800\,m$. The true propagation

parameters are chosen similar to [81]. The transmission power for all ground devices is set to -20 dB, with a noise level of -95 dB. We chose $\overline{K} = 3$, $\gamma_0 = 10$, the periodicity of CoV updates $\bar{t} = 4$, $\lambda = 1.5$. We set $\psi_c = 1 \; \forall \, c$ at the first round.

We consider a sub-urban area of size $800 \times 800 \, m^2$, containing two communities ($C = 2$), of 6 devices each, defined by two different tasks to train, namely the CIFAR10 [74] and MNIST [88] image classification tasks. We distribute the devices randomly inside the service area, and as a data partitioning strategy, and to enforce heterogeneity among the datasets, we randomly assign 2 different label IDs to each member across the different communities as in [15]. Then, we randomly and equally divide the samples corresponding to each label across devices which own that label. For both tasks, we use Fed-Prox [47] with parameter $\mu = 0.1$, to tackle the heterogeneity burden induced by the partial participation and the data heterogeneity of the devices. The SGD optimizer is used with a fixed learning rate of 0.01, and momentum = 0.9. The batch size is set to 16, and number of epochs is set to 1. In addition to our proposed solution, we analyze 4 different, handpicked static and mobile deployments :

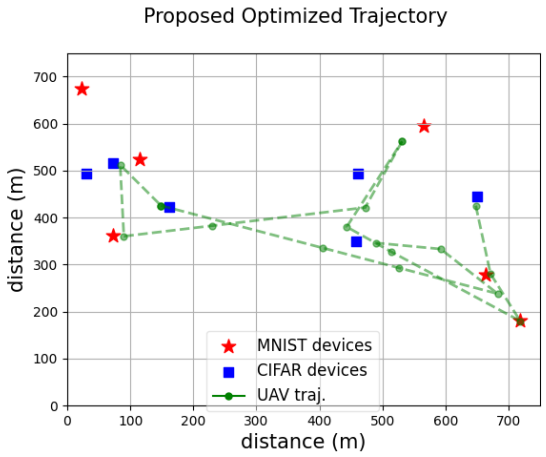
- **A static UAV hovering at the Barycenter** of the devices emulating a BS deployment. Given that the UAV hovers still, we assume that each round lasts for 5 seconds in this particular experiment, which accounts to 100 meters traveled distance per round.

- **A rectangular UAV trajectory (Fig. 3.1b):** where the UAV attempts to cover the whole service area during its mission. Communication rounds are initiated over a set of hovering locations scattered on the predefined rectangular trajectory.

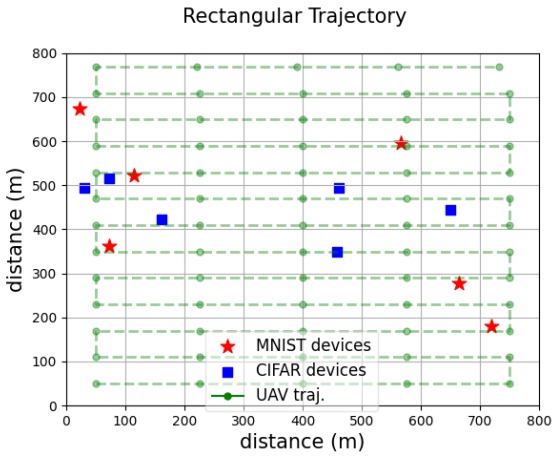
- **Optimal UAV control with naive scheduling (No-CoV):** where the UAV attempts to maximize the objective in (3.11) while setting $\delta_k = p_k$ if device k participated in the previous round, and $\delta_k = p_k \, \lambda$ otherwise.

- **Ideal case:** Representing the maximum achievable performance in the case where all users are scheduled and enjoy no packet loss.

In all experiments, the devices are scheduled akin to Sec. 3.4.1. Moreover, in all deployments (excluding the Barycenter case), we consider that the duration of the DL/UL transmissions and local training is negligible compared to the maneuvering time taken by the UAV at each round. In Fig. 3.2, we report the average validation accuracy attained by both tasks over Monte-Carlo (MC) simulations. At each MC iteration, the devices are distributed randomly. As expected, our solution achieves the highest average validation accuracy, compared to the other benchmarks. In the Rectangular trajectory case, the UAV energy budget is wasted on traversing predefined paths which does not take into account the exact devices' locations.

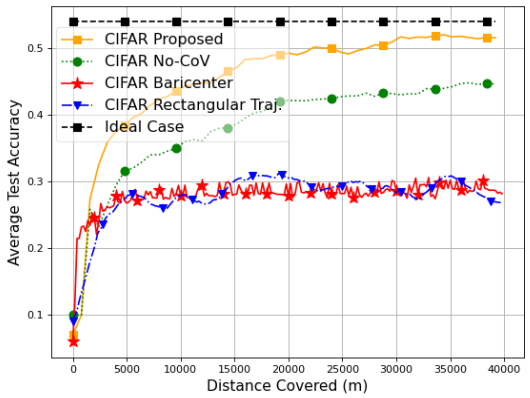


(a) Snapshot: Optimized UAV Trajectory (3 comm. rounds)

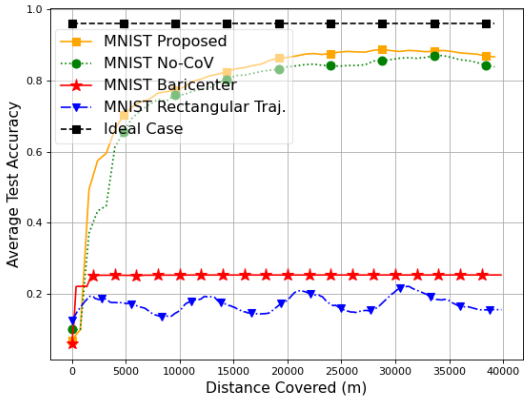


(b) Snapshot: Rectangular Trajectory

Figure 3.1: Optimized UAV Trajectory vs Rectangular trajectory



(a) CIFAR10 Test Accuracy



(b) MNIST Test Accuracy

Figure 3.2: Average Validation Accuracy attained by different strategies

Accordingly, the UL packet transmissions endure high average PER, resulting in low participation count during each round, undermining the convergence rate and hampering the training performance. In the Barycenter case, despite that the UAV hovers still at the mean devices location, the channel yet imposes a strong PER penalty over the devices packets transmission, especially for devices that reside far from the UAV, given their low transmission power. The poor performance of those two benchmarks is mainly related to the wrong UAV placement. Hence, in order to quantify the gain of our scheduling algorithm incorporating the CoV of the different communities, we devise the No-CoV experiment, in which the UAV attempts to maximize the objective in (3.14) while naively assigning the importance of

the devices as a function of their dataset sizes, and ignoring their tasks training performance that is quantified by the CoV. As expected, employing the CoV in our optimization leads to faster convergence and a percentage gain of 14% for the CIFAR10 task compared to the No-CoV experiment, while maintaining a similar performance in the MNIST case. This advantage stems from the inherent ability of the CoV in quantifying the training performance of the two different communities throughout the course of training, ergo enabling the UAV to establish learning fairness among them by prioritizing the CIFAR10 task in terms of device scheduling and trajectory planning, which is well recognized as a more complex task compared to the MNIST task.

3.6 Conclusion

In this chapter, we studied the problem of an online path planning for a UAV missioned to orchestrate the training of different communities' tasks. We proposed a heuristic metric that is able to quantify the training performance and the scheduling requirements of the different tasks. Hinging on this metric, we devise a surrogate optimization problem which we solve iteratively using Convex optimization techniques, to schedule devices and find the optimal trajectory to successfully pool their updates, while aiming at achieving learning fairness among the available tasks. The performance of the proposed algorithm was evaluated via simulations, which highlighted its advantage compared to other benchmarks.

In the subsequent chapter, we explore novel methods that can ensure communication-efficient federated learning without relying on compression or sparsification, which often compromise learning accuracy and communication efficiency.

Part III

On Communication-Efficient Federated Learning

Chapter 4

Communication-Efficient Federated Learning via Sparse Random Networks

In this chapter, our goal shifts to explore novel approaches through which federated learning can be communication efficient. Particularly, we seek to identify methods that can enhance the communication efficiency of FL beyond conventional means such as quantization and sparsification, which often lead to subpar model performance. Accordingly, we present a new method for enhancing communication efficiency in federated learning by leveraging over-parameterized random networks. In this setting, a binary mask is optimized instead of the model weights, which are kept fixed. The mask characterizes a sparse sub-network that is able to approximate a smaller target network. Importantly, sparse binary masks are exchanged rather than the floating point weights in traditional federated learning, reducing communication cost to at most 1 bit per parameter. We show that previous state of the art stochastic methods fail to find the sparse networks that can reduce the communication and storage overhead using consistent loss objectives. To address this, we propose adding a regularization term to local objectives that encourages sparser solutions by eliminating redundant features across sub-networks. Extensive experiments demonstrate significant improvements in communication and memory efficiency of up to five magnitudes compared to the literature, with minimal performance degradation in validation accuracy in some instances.

Recent efforts have focused on reducing communication overhead leveraging compression via quantization and model sparsification techniques [89–91] on the exchanged model weights. Despite these efforts, exchanged compressed models are still represented according to float bit-representations (e.g., 32/16 bits per model weight), leading to significant communication overhead as the size of trained models increases (e.g., LLM).

A recent work [92] has revealed that in over-parameterized random neural networks, it is possible to find smaller sub-networks that perform just as well as a fully trained target network in terms of generalization. These sub-networks are produced by element-wise multiplication of a sparse binary mask with the initial weights of the over-parameterized network (i.e while fixing the weights). In this case, the binary mask is optimized to identify

the initial weights that would constitute a sub-network with similar generalization performance as the target network. Subsequently, the authors in [59] leverage the subset-sum approximation problem [58] to prove the existence of those sub-networks. They show that dense target networks with width d (neuron count in a layer) and depth l (number of layers) can be closely approximated by pruning an over-parameterized dense random network with a width $O(\log(dl))$ times larger and a depth twice as deep. This discovery is particularly interesting for FL training, due to the lower communication overhead associated with exchanging binary masks in the UL and DL instead of float-bit representations of the weight updates. In [93], the authors introduce FedMask, a personalized federated learning algorithm based on pruning over-parameterized random networks. FedMask is a deterministic algorithm that involves pruning a random network by optimizing personalized binary masks using Stochastic Gradient Descent (SGD), aiming to approximate the personalized target networks that fit the heterogeneous datasets found at the devices. Their approach has been shown to ensure a 1-bit-per-parameter (1bpp) communication cost per each round of communication to exchange the updates in FL training. This can be attributed to the nature of their algorithm, which optimizes the binary masks within a constrained search space, wherein the masks demonstrate an equiprobable occurrence of ones and zeros. Recently, a stochastic approach called FedPM [94] was introduced as an alternative to the deterministic FedMask. FedPM requires edge devices to identify a global probability mask, as apposed to the deterministic mask in FedMask. Binary masks then are sampled from the global probability mask, characterizing sub-networks with strong generalization capabilities over the diverse datasets of the edge devices. The results of their approach demonstrate state-of-the-art accuracy and communication efficiency compared to FedMask and other baseline methods. However, our subsequent analysis reveals that their method fails to discover sparse networks, leaving a significant amount of unnecessary redundancy in terms of the size of the found sub-networks.

The proposed solution builds upon the foundation of stochastic masking techniques [92, 94], leveraging their favorable generalization performance and convergence while aiming to enhance communication and memory efficiency. Our main contributions are summarized as follows:

- We introduce a new objective function leading to effectively narrow down the search space to discover a limited set of sub-networks within the over-parameterized random network. These sub-networks offer both communication efficiency and strong generalization performance compared to the literature.
- Through simulations, we demonstrate that our approach, which enforces non-structural sparsity through regularization, results in significantly sparser solutions compared to state-of-the-art algorithms such as FedPM. Importantly, this sparsity gain is achieved

without sacrificing generalization, leading to up to about 5 times more communication and memory efficiency during training.

- Additionally, simulations reveal that our proposed algorithm allows for a flexible trade-off between accuracy, communication overhead and memory efficiency if required. This feature makes it highly suitable for systems with limited bandwidth and memory resources, providing an effective solution to optimize resource utilization in such constrained environments.

4.1 System Model and Problem Formulation

The training procedure commences as the parameter server sends a randomly initialized network to the edge devices. This is accomplished by providing the devices with both the network's structure and an initialization seed, enabling them to construct the network's layers and weights. We denote the initialized weights of the network by $\boldsymbol{w}_{\text{init}} = (w_1, \cdots, w_n) \in \mathbb{R}^n$. The primary objective is to identify a global binary mask $\boldsymbol{m} \in \{0,1\}^n$, yielding a sub-network $y_{\boldsymbol{m}} \in \mathbb{R}$ given according to[1]:

$$y_{\boldsymbol{m}}(\boldsymbol{x}) = (\boldsymbol{m} \otimes \boldsymbol{w}_{\text{init}})^T \cdot \boldsymbol{x}, \tag{4.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes a data point, $\otimes$ denotes the element-wise multiplication operator, and $(\cdot)$ denotes vector multiplication. The produced sub-network minimizes the empirical risk function in accordance to:

$$\min_{\boldsymbol{m}} L(\boldsymbol{m}) = \frac{1}{\sum_i |\mathcal{D}_i|} \sum_{k=1}^{K} |\mathcal{D}_k| \ell_k(y_{\boldsymbol{m}}, \mathcal{D}_k), \tag{4.2}$$

where $f(\boldsymbol{m})$ denotes the empirical risk of $y_{\boldsymbol{m}}$ over the devices datasets.

We denote the target network that we aim at approximating by y_{target}. The number of sub-networks that can be found within the over-parameterized network to approximate y_{target} increases with its size [59,92,95]. Accordingly, constructing a sufficiently over-parameterized random network according to the rules derived in [59] guarantees with a high probability that a sub-network y exists, such that $y \approx y_{\text{target}}$. To this end, we aim at identifying the individual weights of $\mathbf{w}_{\text{init}}$ that play a role in producing sub-networks capable of generalizing as effectively as y_{target}. This is achieved by maximizing the likelihood of these weights while disregarding the weights that do not offer any meaningful contribution towards that objective. Akin to [94], along-side the initialized weights, the users receive a global probability

[1]For ease of representation, we use a linear model in (4.1).

mask vector[2] $\boldsymbol{\theta} \in [0,1]^n$:

$$\boldsymbol{\theta}_i \longleftarrow \boldsymbol{\theta}(t) \tag{4.3}$$

using the probability mask, each user i derives a local score vector $\boldsymbol{s}_i$, according to :

$$\boldsymbol{s}_i = \sigma^{-1}(\boldsymbol{\theta}_i) \tag{4.4}$$

where $\sigma(\boldsymbol{s})$ denotes the sigmoid function applied to each element of the vector $\boldsymbol{s}$. The probability mask represents the likelihood of each particular weight in $\boldsymbol{w}_{\text{init}}$ contributing to the chosen sub-network in (4.1). Once each device i receive the global probability mask $\boldsymbol{\theta(t)}$ from the server, training starts by sampling a binary mask which characterizes the local sub-network $y_{\boldsymbol{m}_i^h}$ to minimize its loss, given by

$$y_{\boldsymbol{m}_i^h}(\boldsymbol{x}) = \left(\boldsymbol{m}_i^h \otimes \boldsymbol{w}_{\text{init}}\right)^T \cdot \boldsymbol{x} \quad , \boldsymbol{m}_i^h \sim \text{Bernoulli}(\boldsymbol{\theta}_i^h), \tag{4.5}$$

Here h denotes the local mini-batch iterations count, where $\boldsymbol{\theta}_i^{h=0} = \boldsymbol{\theta}(t)$. Similar to [92], instead of directly optimizing $\boldsymbol{\theta}_i^h$, the score vector is employed in the optimization process. This ensures smooth and unbiased[3] updates of $\boldsymbol{\theta}$. The scores and probability masks are updated at *each* mini-batch iteration h according to:

$$\boldsymbol{\theta}_i^h = \sigma(\boldsymbol{s}_i^{h-1} - \frac{\eta}{|\mathcal{B}^h|} \nabla_{\boldsymbol{s}_i^{h-1}} \ell_i(y_{\boldsymbol{m}_i^{h-1}}, \mathcal{B}^h)), \tag{4.6}$$

where η is the learning rate, $\mathcal{B}^h \subseteq \mathcal{D}_i$ is a mini-batch, and $|\mathcal{B}^h|$ denotes its cardinality. $\nabla_{\boldsymbol{s}_i^{h-1}} \ell_i(y, \mathcal{B}^h)$ denotes the gradient of the loss function (e.g. the cross entropy loss in classification tasks) of the local sub-network $y_{\boldsymbol{m}_i^{h-1}}$ – sampled during the current iteration h – over the mini-batch $\mathcal{B}^h$ at device i, with respect to the scores vector $\boldsymbol{s}_i^{h-1}$. Accordingly, each element indexed k of the score vector $\boldsymbol{s}_{i,k}^{h-1}$ are optimized locally using the chain rule according to:

$$\boldsymbol{s}_{i,k}^h = \boldsymbol{s}_{i,k}^{h-1} - \eta \left(\frac{\partial \ell_i}{\partial y_{\boldsymbol{m}_i^{h-1}}} \times \frac{\partial y_{\boldsymbol{m}_i^{h-1}}}{\partial m_{i,k}^{h-1}} \times \frac{\partial m_{i,k}^{h-1}}{\partial \theta_{i,k}^{h-1}} \times \frac{\partial \theta_{i,k}^{h-1}}{\partial s_{i,k}^{h-1}} \right). \tag{4.7}$$

$m_{i,k}^{h-1}$ and $\theta_{i,k}^{h-1}$ denote the k^{th} elements of $\boldsymbol{m}_i^{h-1}$ and $\boldsymbol{\theta}_i^{h-1}$ respectively. We omit the local iteration count h in the following expressions for ease of representation. Note that the sampling operation $m_{i,k}^h \sim \text{Bernoulli}(\theta_{i,k}^h)$ is not differentiable. Therefore $\frac{\partial m_{i,k}^h}{\partial \theta_{i,k}^h}$ can be

[2] All mask probabilities are set to 0.5 during the first round

[3] For instance, FedMask relies on optimizing a deterministic mask via SGD, and then thresholding the resultant updated mask. The thresholding operation results in severely biased updates which harms the convergence.

approximated using straight-through estimators [92, 94]. Next, after optimizing the scores for a number of local iterations, let $\hat{\boldsymbol{\theta}}_i(t)$ denote the locally produced probability mask at round t. For each client i, a binary mask $\hat{\boldsymbol{m}}_i$ is sampled according to:

$$\hat{\boldsymbol{m}}_i(t) \sim \text{Bernoulli}(\hat{\boldsymbol{\theta}}_i(t)).$$

These binary masks are then sent to the server. The masks highlight the weights contributing to the best sub-networks. This approach effectively reduces the communication cost (entropy) to a maximum of 1 bit per parameter (1bpp), where the actual entropy depends on the sparsity of the mask. The server then performs averaging to generate a global probability mask according to:

$$\boldsymbol{\theta}(t+1) \longleftarrow \frac{1}{K}\sum_i \hat{\boldsymbol{m}}_i(t). \tag{4.8}$$

The resultant global probability mask $\boldsymbol{\theta}(t+1)$ is re-distributed to the devices in the DL to commence the next communication round. The global mask has been demonstrated in [94] to be asymptotically unbiased estimate of the true global probability mask $\bar{\boldsymbol{\theta}}$, which is given by $\bar{\boldsymbol{\theta}}(t+1) = \frac{1}{K}\sum_i \hat{\boldsymbol{\theta}}_i(t)$.

4.2 Intuition and Proposed Algorithm

4.2.1 Intuition

We first delineate the shortcomings of the current state-of-the-art technique [94] with regards to the sparsity level of the networks identified. Accordingly, we first undertake a thorough analysis of the results outlined in [59]. These outcomes serve as guiding directives stipulating the extent of over-parameterization necessary for a random network to approximate a smaller target network. Subsequently, we conduct a comprehensive evaluation of the original optimization algorithm employed within the framework of FedPM. This evaluation is conducted from the vantage point of each individual learner, under the premise of an absence of regularization in the loss function.

4.2.1.1 Lack of a unique solution to the approximated Subset-Sum Problem

In [59], the authors investigated the estimation of a target weight w_t through the lens of the approximated subset-sum problem [58]. Under this context, they proved that a target weight value w_t can be approximated by a subset-sum of n random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ sampled from a uniform distribution, within a specified margin of error ϵ, with probability

$1-\gamma$. The number of variables n required is in the order of $\mathcal{O}(\log(2/\min(\gamma,\epsilon)))$. Formally, let $n = \mathcal{O}\left(\log\left(2/\min(\gamma,\epsilon)\right)\right)$, then, $\exists\, \mathcal{S} \subseteq \mathcal{X}$ w.p. $1-\gamma$, a feasible solution of:

$$\begin{aligned} &\text{find } \mathcal{S} \subseteq \mathcal{X} \\ &\text{subject to : } |\sum_{X \in \mathcal{S}} X - w_t| < \epsilon \end{aligned} \tag{4.9}$$

Expanding upon these findings, the authors introduce a systematic approach to discern the required size (e.g. width and depth) of an over-parameterized dense random network, in order to effectuate the accurate approximation of a target dense network weights. Note that (4.9) does not necessarily deem a single feasible solution. Accordingly, the objective of finding a sub-network within an over-parameterized random network by optimizing a mask via SGD using consistant loss functions [94] (e.g. cross entropy loss in classification tasks), is not synonymous to solving (4.9), but equivalent to solving :

$$\min_{\mathcal{S}} |\sum_{X \in \mathcal{S}} X - w_t| \tag{4.10}$$

which entails the identification of a solution that aims at reducing the average loss of the sub-network chosen, without factoring in its size and without investigating alternative sparser feasible solutions that can offer a small trade-off of accuracy in response to large sparsity gains. Therefore, we posit the addition of a regularization term over the average number of chosen weights in the global mask, serving to find those sparser sub-networks that can generalize well.

4.2.1.2 FedPM stochasticity results in redundant trained sub-networks

During FedPM training, within every local iteration (e.g. mini-batch update), individual devices sample a distinct instance sub-network based on a received probability mask as outlined in equation (4.1). As a result of the considerable scale of the over-parameterized random network, the sampled sub-networks may be entirely new for the devices at each local iteration. Subsequently, each device calculates the loss specific to the sampled network and then back-propagates the gradients to minimize the loss. This is done by adjusting the scores in directions that activate or deactivate the fixed random weights appropriately. In subsequent local iterations, additional networks are sampled, and their weight scores are tuned to minimize their corresponding loss. From a broader perspective, the local stochastic sub-network sampling step designed in FedPM implicitly promotes the minimization of the weighted average loss of all sub-networks sampled from the probability mask at each device. Due to the substantial number of existing sub-networks that can generalize well,

this sampling step results in redundancy in terms of the number of optimized sub-networks and accordingly the number of activated weights. This considerably increases the size of the sampled sub-networks.

4.2.2 Proposed Loss function

Particularly, we integrate a regularization term alongside the conventional cross-entropy loss between the predicted output and the ground truth value for classification tasks. Therefore, our loss function imposes unstructured sparsity on the sub-networks independently discovered by each individual device, by accounting to the normalized average number of chosen parameters within the original over-parameterized network through a regularization term. Accordingly, the definition of the local loss function at device i over a mini-batch $\mathcal{B} \subseteq \mathcal{D}_i$ is given as follows:

$$\ell_i(y_{\boldsymbol{m}_i}, \mathcal{B}) = \bar{\ell}_i(y_{\boldsymbol{m}_i}, \mathcal{B}) + \frac{\lambda}{n} \sum_{k=1}^{n} \sigma(s_{i,k}), \tag{4.11}$$

where now $\bar{\ell}$ denotes the model average local loss and λ serves as a regularization parameter that governs the level of sparsity exhibited by the resulting sub-networks. Note that the regularization term in (4.11) should be balanced with the values attained by the model loss function, in order to produce sparse models that can generalize well, while avoiding any bias to either objectives. Accordingly the regularization term λ should be carefully chosen, given a bound on the model loss function to reach that goal. The regularization term introduced aims at expediting the deactivation of weights with minimal impact on the current sampled network. This regularization reduces the likelihood of sampling entirely new and distinct sub-networks in subsequent iterations, favoring the sampling of sub-networks sharing substantial features with early stages samples. Upon transmitting the updated mask under regularization to the server, the resulting global probability mask defined in (4.8) introduces redundant sub-networks once again due to the inherent stochasticity in the local sub-network sampling process on each device. However, this global mask also characterizes a more constrained search space for the devices in the subsequent rounds as the training progress, given the limited number of distinct sub-networks optimized by each device during successive iterations. The training proceeds until a probability mask is found that can produce sub-networks with sparsity guided by the parameter λ, thereby ensuring both communication and memory efficiency, alongside achieving good generalization performance.

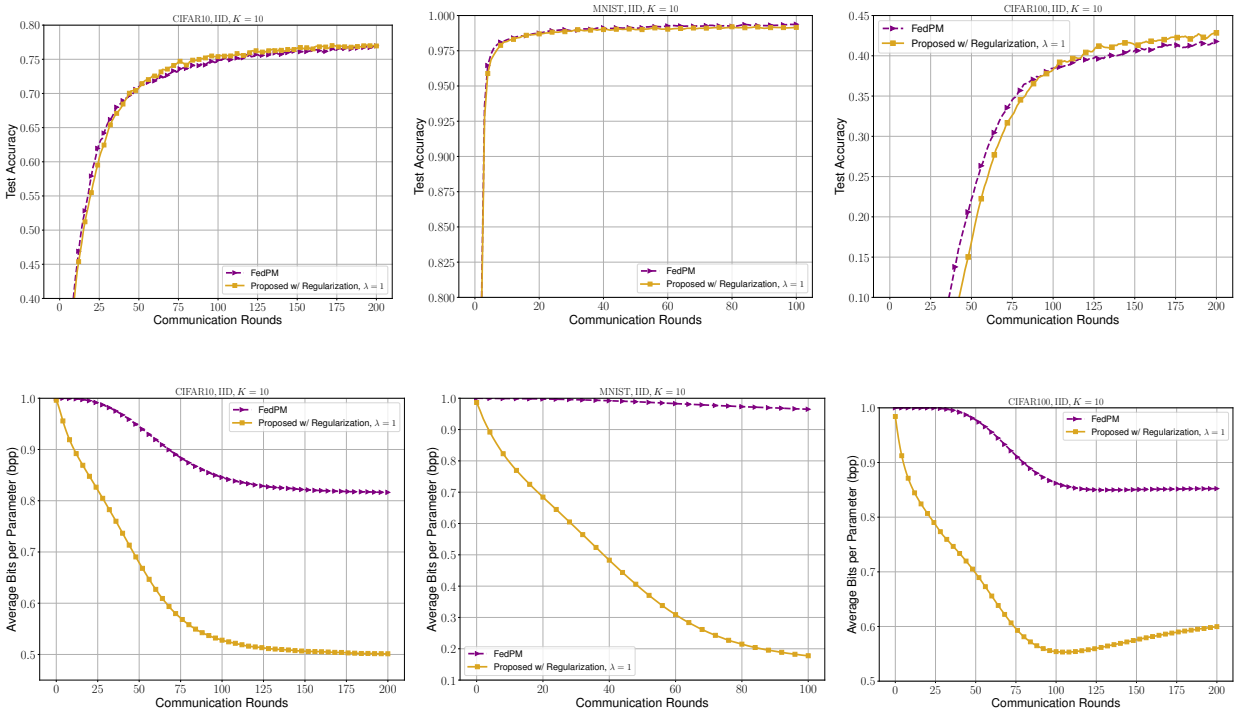
4.3 Experiments

To assess the effectiveness of our proposed approach in comparison to FedPM, we carry out a series of experiments involving image classification tasks. These experiments are conducted under both homogeneous and heterogeneous conditions, as follows:

- In an Independent and Identically Distributed (IID) scenario, we evenly distribute the datasets CIFAR10, CIFAR100 [96], and MNIST [97] across 10 devices.
- We distribute the CIFAR10 dataset across 10 devices while introducing heterogeneity by randomly assigning each device a subset of $c = \{2, 4\}$ classes from the available 10 classes.

For these experiments, we present the average testing accuracy over the population target distribution (top row) and the average bits per parameter required (lower row) as a function of the number of rounds (e.g an average of three simulation runs). The bits per parameter required represents the average entropy of the binary masks transmitted in the UL by the devices. The number of local epochs is set to three with $|\mathcal{B}| = 128$. We utilize three feed-forward convolutional networks (4Conv, 6Conv and 10Conv [95]) to train over MNIST, CIFAR10 and CIFAR100 respectively. The initial score vector is sampled from a standard normal distribution with identity covariance matrix. As in [92], the model random weights are sampled from a uniform distribution over $\{-\varsigma, \varsigma\}$, where ς denotes the standard deviation of the Kaiming normal distribution [98].

Figure 4.1 illustrates the validation accuracy of FedPM with our proposed regularization term ($\lambda = 1$) compared to the original algorithm under IID settings. The validation accuracy of both techniques is similar across all simulations. However, FedPM combined with our proposed regularization term achieves significant improvement in communication efficiency compared to the original algorithm. Specifically, on CIFAR10 experiments, an average efficiency gain of 0.31 bits per parameter (bpp) is achieved using our proposed modification. On MNIST experiment, we achieve 0.8 bpp greater efficiency, while on CIFAR100 experiment, we gain 0.25 bpp higher efficiency relative to original algorithm. Therefore, our proposed recipe provides notable gains in communication efficiency while maintaining the generalization performance of FedPM in the IID settings configuration. We now examine Fig. 4.2, which evaluates the performance of the two algorithms on non-IID CIFAR10 datasets. The regularization term value is varied to highlight the potential trade-off between generalization and communication efficiency in this setting. For $\lambda = 1$, the communication efficiency gain trend persists, where we observe substantial improvements of 0.52 bits per parameter (bpp) when label heterogeneity is present with $c = 2$, and 0.44 bpp when $c = 4$. However, unlike the IID setting, a slight loss in generalization performance is observed (around 3% and 4%



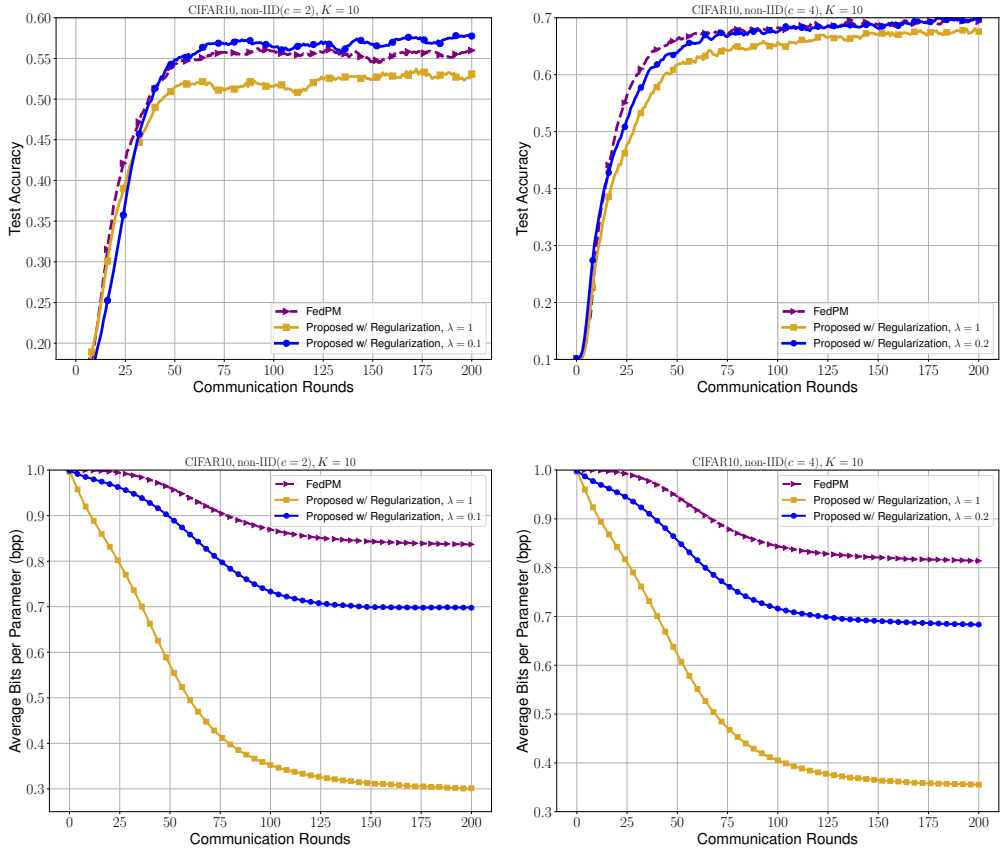
(a) CIFAR10, IID (b) MNIST, IID (c) CIFAR100, IID

Figure 4.1: From left to right: CIFAR10, MNIST, CIFAR100 experiments. First row: Validation Accuracy vs Rounds. Second row: The corresponding Average Bit-per-parameter (bpp) required vs Rounds.

respectively). However, when λ is set to 0.1 and 0.2 for $c = 2$ and $c = 4$ respectively, our algorithm converges to a sub-network with comparable generalization to FedPM while ensuring around 0.12 bpp gain in communication efficiency and final model size. In summary, Fig. 2 shows that our approach can identify sub-networks with generalization performance on par with FedPM in non-IID settings too, while still providing moderate gains in communication and memory efficiency. Moreover, it demonstrates that our algorithm allows for flexible trade-off between accuracy and communication and memory efficiency if required by tuning the regularization hyperparameter λ.

4.4 Conclusion

In this chapter, we demonstrate that state-of-the-art federated learning methods for sparse random networks, which rely on consistent objectives, fail to uncover highly sparse sub-networks within the over-parameterized random models. To address this limitation, we propose and validate the incorporation of a regularization term within the local loss func-



(a) CIFAR10, non-IID, $c = 2$ (b) CIFAR10, non-IID, $c = 4$

Figure 4.2: Trade-off between validation accuracy and communication efficiency (bpp) for different regularization values λ in non-IID CIFAR10 settings. Higher λ prioritizes communication-efficiency over accuracy, while lower value of λ prioritizes accuracy over sparsity reduction.

tions to discover sparser sub-networks. The sparse models obtained through our approach lead to significant improvements in communication and memory efficiency during federated training on resource-constrained edge devices, without sacrificing accuracy. Through extensive experiments, we show that our method outperforms existing state-of-the-art techniques by a large margin in terms of the sparsity and efficiency gains achieved. Additionally, the flexibility of our algorithm enables customizing the trade-off between accuracy and efficiency as per application requirements.

Chapter 5
Conclusion

In this thesis, we have explored three independent contributions that we hope to contribute to the advancement of intelligent edge in future networks. Each chapter presents a novel approach that addresses specific challenges in federated learning.

In the first chapter, we focused on the data heterogeneity challenge that is prevalent among users' datasets in a federated learning system from the lens of personalization. We introduced a novel, user-centric algorithm that produces personalized models for each device, tailored to their unique objectives. To mitigate the communication overhead associated with the training of a distinct model for each device, we utilized a K-means clustering approach that groups users with similar objectives and provides a single personalized model for each group.

The second chapter focused on leveraging UAVs as federated learning orchestrators to integrate remote IoT systems in the intelligent edge. Accordingly, an online path planning problem has been formulated for the UAV tasked with coordinating the federated learning training of different communities of devices, each with their own objectives. By quantifying the training performance, channel conditions, and resource requirements of the different devices, we established and solved a convex optimization problem to facilitate efficient pooling of updates while ensuring learning fairness among the different communities. Our findings are presented in the context of a single drone, and the optimized trajectory is determined accordingly.

In Chapter 3, we focused on addressing the problem of excessive communication overhead associated with transmitting the raw model updates used in federated learning. While compression techniques such as quantization and sparsification exist, the provided communication efficiency often comes at the cost of lower model accuracy. To overcome this limitation, we leverage over-parameterized random networks [92,99] to approximate smaller target networks, through pruning of parameters rather than optimizing the network parameters. This method has been shown to potentially require transmitting at most one bit per model parameter in federated learning settings. We showed that existing state-of-the-art methods fail to leverage the full potential for enhanced communication efficiency attainable through this approach and proposed a regularized loss function that takes into account the entropy of the transmitted updates, leading to significant improvements in communication and memory efficiency during federated training on resource-constrained edge devices with

minimal performance degradation in validation accuracy in some instances.

Future Directions

While the contributions in this thesis addresses key federated learning challenges, ample opportunities remain for further exploration.

- One direction is to explore alternative metrics that can more accurately capture similarities in user objectives while preserving privacy, as discussed in chapter 2. Despite our solution effectiveness, the algorithm's foundation is built upon a heuristic metric that assesses the similarity between users' learning tasks based on their gradients' similarity. In this context, an interesting future direction would be to investigate alternative metrics that can more accurately capture the underlying similarities among user objectives while inherently preserving privacy.

- For the UAV-assisted federated learning framework in chapter 3, investigating scenarios involving multiple drones and developing a jointly optimized trajectory while also accounting to the energy expenditure of the devices are natural potential extensions, building upon the results presented in this chapter. This area of research is still in its nascent stages, primarily due to the prevalence of terrestrial based orchestration approaches, the energy constraints and the computational constraints that prevent on-device training of IoT devices frequently deployed in remote environments. However, exploring the prospective applications of UAV-assisted learning that may emerge in future use-cases warrants further investigation.

- Chapter 4 demonstrated the promise of communication-efficient federated learning via sparse random networks. A natural extension would involve examining the fundamental limits of the potential communication gains in decentralized settings achievable through such methods that approach training models via pruning, rather than parameter tuning, while preserving the model's capacity for effective generalization.

  For instance, the initial work [99] which showed the potential generalization performance of sub-networks pruned from over-parameterized networks, required over-parameterization that grows polynomially as a function of the target network depth and width. This obviously is inefficient in federated settings in terms of the communication overhead when considering large target models.

  Modern attempts managed to reduce the required over-parameterization needed. Particularly, [59] proved that logarithmic over-parameterization in terms of the target network depth and width is sufficient.

Another approach [100] showed that random re-sampling of a portion of the pruned weights during training can provably be effective in reducing the required over parameterization while preserving the same generalization performance. As a result, they show that the required width of the over-parameterized networks could be reduced to twice as wide as the target network when the number of resampling operations is sufficiently large, in contrast to the prior results without resampling assumption [59, 99]. The proposed resampling technique involves periodically resampling a subset of the pruned weights during training. Although this approach has shown promise in centralized settings, its direct application to federated learning poses challenges related to the large number of communication rounds required to execute the necessary number of resampling operations.

Finally, an interesting direction is to explore how the aggregation rules and heterogeneity imposed by user data in FL can affect the approximation performance of sparse random networks. This could involve developing new theoretical frameworks for analyzing the performance of such approaches in decentralized heterogeneous data settings.

Appendix of Chapter 1

Proof of Theorem 1

Denote by f^* the $\arg\min_{f\in\mathcal{F}} E_{z\sim P_i}[\ell(f,z)]$ and bound the estimation error of $\hat{f}_{\vec{w}_i}$ as

$$
\begin{aligned}
&Exc(\hat{f}_{\vec{w}_i}, P_i) = E_{z\sim P_i}[\ell(\hat{f}_{\vec{w}_i}, z)] - E_{z\sim P_i}[\ell(f^*, z)] \\
&\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z)] - E_{z\sim P_{\vec{w}_i}}[\ell(f^*, z)] + 2d_{\mathcal{F}}(P_i, P_{\vec{w}_i}) + 2\gamma \\
&\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z)] - \inf_{f\in\mathcal{F}} E_{z\sim P_{\vec{w}_i}}[\ell(f, z)] \\
&+ 2\sum_{j=1}^{K} w_{i,j} d_{\mathcal{F}}(P_i, P_j) + 2\gamma,
\end{aligned}
$$

where $\gamma = \arg\min_{f\in\mathcal{F}} \Big(E_{z\sim P_i}[\ell(f,z)] + E_{z\sim P_{\vec{w}_i}}[\ell(f,z)]\Big)$. We recognize the estimation error of $\hat{f}_{\vec{w}_i}$ w.r.t to the measure $P_{\vec{w}_i}$ that can be bounded following fairly standard approaches. In particular,

$$
E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z)] - \inf_{f\in\mathcal{F}} E_{z\sim P_{\vec{w}_i}}[\ell(f, z)] \leq 2\Delta(\mathcal{G}, Z),
$$

where

$$
\Delta(\mathcal{G}, Z) = \sup_{g\in\mathcal{G}} \left| E_{P_{\vec{w}_i}}[g(Z)] - \sum_{j=1}^{K} \frac{w_{i,j}}{|\mathcal{D}_i|} \sum_{z\in\mathcal{D}_i} g(z) \right|,
$$

is the uniform deviation term and

$$
\mathcal{G} = \{Z \to \ell(f, Z) : f \in \mathcal{F}\},
$$

is the class resulting from the composition of the loss function $\ell(\cdot)$ and $\mathcal{F}$. The uniform deviation bound can be bounded in different ways, depending on the type of knowledge about the random variable $g(Z)$, in the following we assume that the loss function is bounded with range B and we exploit Azuma's inequality. In particular, the Doob's Martingale associated with the weighted loss will still have increments bounded by $\frac{w_{i,j}}{|\mathcal{D}_i|}B$ depending to which loss term the increment is associated. Recognizing this, we can then directly apply Azuma's concentration bound and state that w.p. $1-\delta$ the following holds

$$
\Delta(\mathcal{G}, Z) \leq E_P[\Delta(\mathcal{G}, Z)] + B\sqrt{\sum_{j=1}^{K} \frac{w_{i,j}^2}{|\mathcal{D}_j|} \log\left(\frac{2}{\delta}\right)}.
$$

Finally, the expected uniform deviation can be bounded by the Rademacher complexity as follows

$$E_P[\Delta(\mathcal{G}, Z)] \leq 2\text{Rad}(\mathcal{G}),$$

where

$$\text{Rad}(\mathcal{G}) = E_{\vec{\sigma},\mathcal{D}_1,\ldots,\mathcal{D}_j}\left[\sup_{g\in\mathcal{G}} \sum_{j=1}^{K} \frac{w_{i,j}}{|\mathcal{D}_i|} \sum_{i=1}^{|\mathcal{D}_i|} \sigma_{i,j} g(Z_{i,j})\right].$$

By a direct application of Massart's and Sauer's Lemma we obtain

$$\text{Rad}(\mathcal{G}) \leq \sqrt{\sum_{j=1}^{K} \frac{w_{i,j}^2}{|\mathcal{D}_j|}} \times \sqrt{\frac{2\text{VCdim}(\mathcal{G})\left(\log(e\sum_j |\mathcal{D}_j|) + \log(\text{VCdim}(\mathcal{G}))\right)}{\sum_j |\mathcal{D}_j|}}.$$

Combining everything together, we get the final result.

Proof of Theorem 2

Thanks to the upper bound on the target domain risk and the fact that the sum of two sub-Gaussian random variables of parameter σ is also sub-Gaussian with parameter 2σ, we can decompose the excess risk as

$$\begin{aligned} Exc(\hat{f}_{\vec{w}_i}, P_i) &= E_{z\sim P_i}[\ell(\hat{f}_{\vec{w}_i}, z)] - \inf_{f\in\mathcal{F}} E_{z\sim P_i}[\ell(f, z)] \\ &= E_{z\sim P_i}[\ell(\hat{f}_{\vec{w}_i}, z) - \ell(f^*, z)] \\ &\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z) - \ell(f^*, z)] + 2\beta\sigma^2 + \frac{D_{JS}(P_i||P_{\vec{w}_i})}{\beta}. \end{aligned}$$

From the convexity of the KL-divergence, we can bound the Jensen-Shannon divergence as follows

$$
\begin{aligned}
D_{JS}(P_i||P_{\vec{w}_i}) &= \frac{1}{2}KL\left(P_i||\frac{P_i+P_{\vec{w}_i}}{2}\right) + \frac{1}{2}KL\left(P_{\vec{w}_i}||\frac{P_i+P_{\vec{w}_i}}{2}\right) \\
&= \frac{1}{2}KL\left(P_i||\frac{\sum_j w_{i,j}(P_i+P_j)}{2}\right) + \frac{1}{2}KL\left(\sum_j w_{i,j}P_j||\frac{\sum_j w_{i,j}(P_i+P_j)}{2}\right) \\
&\leq \frac{1}{2}\sum_j w_{i,j}\left(KL\left(P_i||\frac{(P_i+P_j)}{2}\right) + KL\left(P_j||\frac{(P_i+P_j)}{2}\right)\right) \\
&= \sum_j w_{i,j}D_{JS}(P_i||P_j).
\end{aligned}
$$

Plugging it back into the previous expression and minimizing with respect to β we obtain

$$
\begin{aligned}
Exc(\hat{f}_{\vec{w}_i}, P_i) &\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z)] - \inf_{f\in\mathcal{F}} E_{z\sim P_{\vec{w}_i}}[\ell(f,z)] \\
&\quad + 2\beta\sigma^2 + \frac{\sum_j \vec{w}_{i,j}D_{JS}(P_i||P_j)}{\beta} \\
&\leq E_{z\sim P_{\vec{w}_i}}[\ell(\hat{f}_{\vec{w}_i}, z)] - \inf_{f\in\mathcal{F}} E_{z\sim P_{\vec{w}_i}}[\ell(f,z)] \\
&\quad + 2\sigma\sqrt{2\sum_{j=1}^{K} w_{i,j}D_{JS}(P_i||P_j)}.
\end{aligned}
$$

We identify the estimation error and we bound as previously done for Theorem 1 to obtain the final result. Moreover, for B-bounded random variables, $\sigma = B/2$.

Résumé

L'évolution omniprésente des réseaux 5G et des réseaux 6G à venir permet un nouveau paradigme d'informatique périphérique intelligente. Avec l'augmentation spectaculaire de la vitesse des réseaux et la diminution de la latence, davantage de traitement et d'intelligence peuvent être transférés à la périphérie plutôt que de s'appuyer uniquement sur les centres de données en nuage. Cela permet un traitement des données en temps réel et une prise de décision plus proche de l'utilisateur final ou de l'appareil, en respectant les exigences futures du réseau. Dans les réseaux 5G récemment déployés, le nuage informatique périphérique multi-accès permet de déployer des ressources de calcul et de stockage à la périphérie du réseau. Dans la perspective de la 6G, la vision est celle de réseaux intelligents hautement distribués avec un traitement basé sur l'IA directement intégré dans les dispositifs périphériques des utilisateurs finaux. L'intégration des capacités d'IA et des dispositifs périphériques englobe des tâches telles que la formation et l'inférence de modèles d'apprentissage automatique au niveau local, ce qui permet des applications innovantes telles que l'automatisation industrielle, les véhicules autonomes, la réalité augmentée et d'autres services nécessitant une latence ultra-faible. Par exemple, le 3GPP s'efforce d'intégrer l'IA dans les réseaux sans fil et à la périphérie. Dans la version 18, diverses techniques ont été étudiées pour améliorer les performances et l'efficacité des réseaux sans fil, notamment la gestion des faisceaux, le retour d'informations sur l'état des canaux et la précision du positionnement.

L'augmentation rapide du nombre d'appareils connectés motive encore davantage cette évolution. Comme le montre la figure 1, le nombre estimé de connexions IoT dans le monde a augmenté de plus de 140 % entre 2018 et 2023, pour atteindre près de 15 milliards de connexions [9,10]. La prise en charge et l'exploitation de cet afflux massif d'appareils périphériques divers ne sont possibles que par le biais d'une intelligence périphérique distribuée.

Ces dernières années, un changement de paradigme notable s'est produit dans l'intégration des modèles d'apprentissage automatique dans les appareils périphériques. L'approche conventionnelle consistant à transmettre les données des appareils à des serveurs centralisés pour l'apprentissage ou l'inférence des modèles, puis à déployer les modèles appris ou les décisions d'inférence à la périphérie, a perdu de sa popularité. Ce changement peut être principalement attribué à deux facteurs essentiels : la protection de la vie privée et les frais généraux de communication.

La préservation de la vie privée est devenue une préoccupation majeure à l'ère des technologies basées sur les données. L'approche conventionnelle consistant à transmettre de grands volumes de données brutes des dispositifs périphériques aux serveurs centraux pour l'apprentissage des modèles a suscité des craintes considérables quant à la protection des informations personnelles sensibles. Cette préoccupation devient particulièrement prononcée dans les domaines qui traitent des données confidentielles, comme les applications de soins de santé, où les dispositifs portables collectent des informations spécifiques aux patients. Par conséquent, la protection de la confidentialité des données est devenue une considération essentielle dans la conception et le déploiement des méthodologies d'apprentissage automatique.

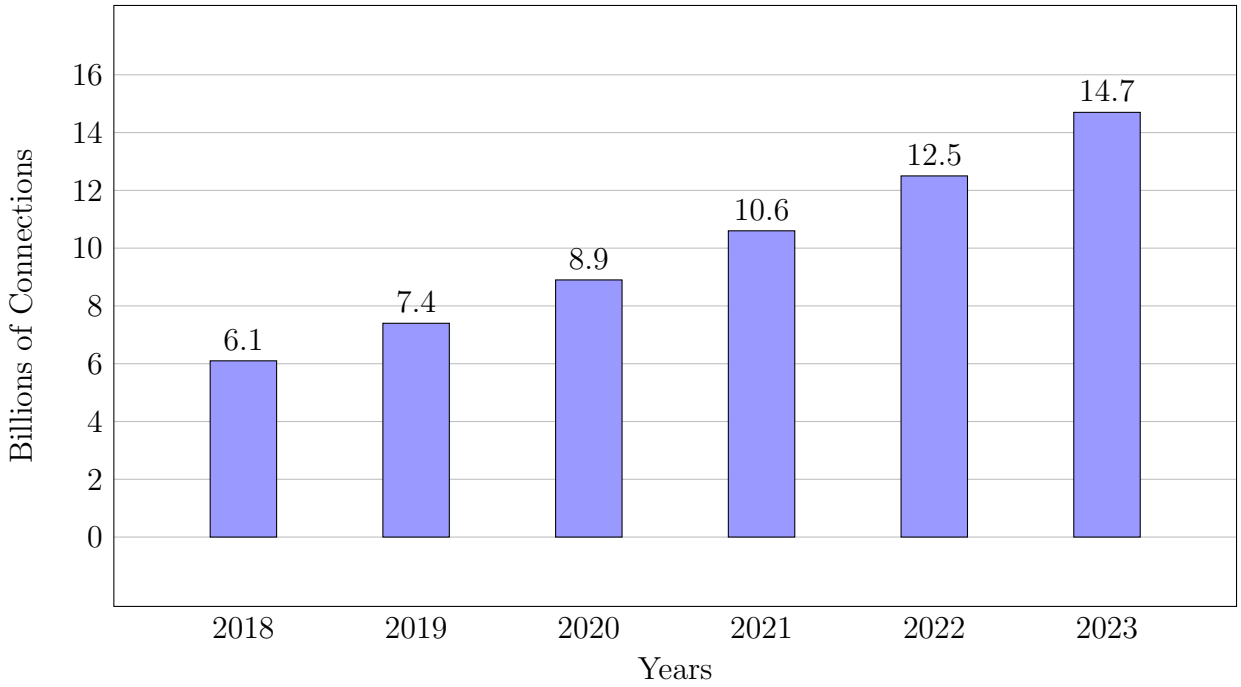


Figure 1: L'essor de l'IdO connectée

En outre, le surcoût de communication associé au transfert de données entre les appareils et les serveurs centraux est devenu un problème crucial. La transmission de grandes quantités de données pour l'apprentissage centralisé ou l'inférence sur les réseaux n'introduit pas seulement des problèmes de latence, mais impose également une charge sur les ressources du réseau. Dans les scénarios où la collecte de données et la prise de décision en temps réel sont primordiales, tels que les environnements industriels qui s'appuient sur des dispositifs IoT (Internet des objets) pour l'acquisition de données de capteurs, les inefficacités introduites par la surcharge de communication excessive peuvent entraver les performances globales du système.

Ce changement de paradigme a pris de l'ampleur grâce à la disponibilité généralisée de puissants appareils périphériques dotés d'importantes capacités de calcul et de fonctionnalités sensorielles. Ces progrès ont rendu possible l'acquisition de données, le traitement et l'apprentissage de modèles directement sur l'appareil.

Cependant, un défi se pose lorsqu'il s'agit d'effectuer un apprentissage local sur l'appareil. Les appareils périphériques tels que les smartphones et les appareils IoT sont généralement confrontés à des ensembles de données limités échantillonnés dans leur environnement immédiat. Les limites inhérentes aux ensembles de données peuvent être interprétées sous l'angle de leur qualité de représentation et de leur taille. La qualité de représentation, qui est synonyme d'expressivité des ensembles de données, mesure leur efficacité dans la formation d'un modèle spécifiquement conçu pour chaque tâche de l'appareil. Cette efficacité dépend de facteurs tels que la résolution d'échantillonnage et les capacités sensorielles des appareils. Ces limitations des données entravent

le potentiel de l'apprentissage local à générer des modèles qui présentent une généralisation robuste sur les tâches des appareils.

Apprentissage fédéré

En raison des limites mentionnées précédemment, l'apprentissage fédéré (AF) est apparu comme une solution potentielle pour surmonter ces défis. L'apprentissage fédéré est un sous-domaine naissant de l'apprentissage automatique qui offre aux appareils (également appelés "clients" ou "utilisateurs") la possibilité de former des modèles en collaboration, sous la supervision d'un "orchestrateur" central, sans qu'il soit nécessaire de partager leurs données d'apprentissage brutes. Au lieu de cela, FL permet aux appareils périphériques de former collectivement des modèles qui présentent une généralisation et des performances améliorées, tout en promouvant la confidentialité des données par conception. La diversité des données entre les appareils permet aux modèles formés de capturer un plus large éventail de modèles et de variations, améliorant ainsi leur capacité à gérer les diverses préférences des utilisateurs et à mieux se généraliser par rapport aux modèles appris localement. Ce paradigme d'apprentissage distribué réduit également la dépendance à l'égard des serveurs centralisés, ce qui permet un traitement en temps réel et une prise de décision contextuelle directement à la périphérie. Cela permet d'accélérer les temps de réponse, de réduire la latence et d'améliorer l'expérience de l'utilisateur dans diverses applications. Dans sa configuration prototypique, le FL implique une formation distribuée exécutée de manière itérative sur plusieurs *circuits de communication*. Un cycle de communication correspond à une itération ou à un cycle unique du processus de formation entre l'orchestrateur et les dispositifs participants. Au cours d'un cycle de communication, les étapes suivantes se déroulent généralement :

1. **Distribution du modèle** : Le serveur central envoie un modèle global de ML à un sous-ensemble sélectionné de dispositifs sur un canal de liaison descendante (DL).

2. **Formation au modèle local** : Chaque appareil effectue un apprentissage local à l'aide de son propre ensemble de données locales et du modèle global reçu. L'appareil optimise son modèle local sur ses données locales.

3. **Modèle de liaison montante Transmission** : Les modèles formés localement par les appareils participants sont renvoyés au serveur central par un canal de communication en liaison montante (UL).

4. **Agrégation de modèles** : Le serveur central agrège les modèles reçus des appareils périphériques pour produire un modèle global actualisé. Le modèle global mis à jour est ensuite distribué à un nouvel ensemble de dispositifs pour le prochain cycle de communication.

La formation implique généralement plusieurs cycles de communication pour affiner et optimiser de manière itérative le modèle global jusqu'à la convergence.

L'apprentissage automatique peut être divisé en deux catégories principales : l'apprentissage inter-silos et l'apprentissage inter-appareils. La formation inter-silos en FL fait référence au processus de formation d'un modèle d'apprentissage automatique sur des données provenant de plusieurs silos ou domaines. Chaque silo représente une entité ou une organisation distincte (par exemple, des hôpitaux ou des banques) qui dispose de ses propres données et souhaite collaborer avec d'autres silos pour former un modèle partagé. Tous les silos de données participants sont fiables et sont presque toujours disponibles pendant la formation. La formation permet à ces silos de travailler ensemble pour former un modèle plus précis et plus robuste que ce qu'un silo individuel pourrait réaliser seul. D'autre part, la formation inter-appareils en FL implique la formation d'un modèle d'apprentissage automatique à l'aide de données provenant de plusieurs appareils, tels que des smartphones, des appareils domestiques intelligents ou des appareils IoT. Dans ce scénario, chaque appareil représente une source de données distincte avec un volume de données relativement faible qui contribue à la formation d'un modèle partagé. Ces appareils sont naturellement moins fiables en raison de facteurs tels que la disponibilité, la mauvaise connectivité réseau et les défaillances matérielles.

En raison de ses garanties inhérentes en matière de respect de la vie privée, FL a été adopté par plusieurs grandes entreprises industrielles. Par exemple, Nvidia a appliqué le FL dans divers domaines tels que l'imagerie médicale et la recherche génétique [16]. En outre, Apple utilise la FL dans le développement de systèmes d'identification biométrique tels que Face ID et de commandes vocales pour des assistants numériques tels que Siri [17]. L'application de clavier de Google, Gboard [18], en est un exemple frappant : elle utilise le FL pour améliorer son modèle linguistique sans compromettre la confidentialité des données de l'utilisateur. En tirant parti de la collaboration entre plusieurs appareils, Gboard peut former un modèle partagé qui s'adapte aux habitudes de frappe et aux préférences des utilisateurs, améliorant ainsi la précision de ses prédictions. Une autre application notable se trouve dans les communications sans fil, où FL a été utilisé pour optimiser les réseaux d'accès radio. En tirant parti des capacités de calcul des appareils périphériques, le FL permet l'apprentissage de modèles susceptibles d'optimiser les performances du réseau, telles que l'allocation des ressources radio et la formation de faisceaux d'utilisateurs, sans nécessiter un partage important de données entre les opérateurs de réseau. Ces applications sont bien adaptées aux techniques collaboratives basées sur les données, telles que l'apprentissage fédéré, plutôt qu'aux approches basées sur des modèles reposant sur des hypothèses simplifiées. Dans ce contexte, les méthodes fondées sur des modèles imposent fréquemment des contraintes irréalistes qui ne parviennent pas à saisir pleinement les subtilités de la complexité du problème et sont souvent encombrées d'hypothèses théoriques idéalistes déconnectées des réalités pratiques [20–23].

Formulation du problème

Définition des normes

L'objectif standard de FL [15, 24] est de trouver un modèle global $\theta \in \mathbb{R}^d$ qui minimise la perte pondérée des K appareils du système, sur leur distribution locale de données $\{P_k\}_{k=1}^K$.:

$$\min_{\theta \in \mathbb{R}^d} \left[L(\theta) \triangleq \sum_{k=1}^{K} w_k \ell_k(\theta) \right], \tag{1}$$

où $\{\ell_k\}$ sont les fonctions de perte des dispositifs et $\{w_k\}$ représentent leurs poids correspondants, de sorte que $\sum_k w_k = 1$. Les pertes locales peuvent être définies comme suit :

$$\ell_k(\theta) \triangleq \mathbb{E}_{x \sim P_k} \left[\ell_k(\theta, x) \right], \tag{2}$$

où $\mathbb{E}[\cdot]$ représente l'espérance mathématique. Dans le cas où les appareils sont dotés d'ensembles de données finis notés $\{\mathcal{D}_k\}$ échantillonnés à partir des distributions de données locales avec une cardinalité $\{|\mathcal{D}_k| < \infty\}$ L'objectif global dans (1) est alors appelé minimisation empirique du risque (ERM). En conséquence, les pertes locales peuvent être écrites comme suit :

$$\ell_k(\theta) = \frac{1}{|\mathcal{D}_k|} \sum_{j=1}^{|\mathcal{D}_k|} \ell_k(\theta, x_{k,j}), \tag{3}$$

où $|\mathcal{D}_i|$ représente la cardinalité de l'ensemble de données de l'appareil i, et $x_{k,j}$ représente le j^{th} échantillon de $\mathcal{D}_k$. Généralement, les poids trouvés dans (1) sont basés sur des facteurs tels que la taille des ensembles de données des appareils, et sont donnés par :

$$w_i = \frac{|\mathcal{D}_i|}{\sum_k |\mathcal{D}_k|}. \tag{4}$$

De manière équivalente dans ce cas, l'objectif est de former un modèle paramétré par θ pour minimiser les pertes pondérées, sur l'union des ensembles de données de l'ensemble du système désigné par $\mathcal{D} = \bigcup_k \mathcal{D}_k$. Cet ensemble de données est échantillonné à partir du mélange de distributions désigné par $P = \sum_k w_k P_k$. Dans ce scénario, l'hypothèse est que tous les dispositifs rencontreront des données échantillonnées à partir de la distribution cible P. Par conséquent, l'une des principales préoccupations est que le modèle découvert présente une généralisation efficace sur P.

Cadre personnalisé

Dans certains scénarios, les utilisateurs peuvent avoir besoin d'une expérience personnalisée. L'optimisation fédérée prototypique (1) agrège les mises à jour de divers dispositifs pour former un modèle unique.

Cependant, de graves divergences de distribution entre les distributions cibles P_k des appareils peuvent rendre inefficace un modèle unique. Au lieu de cela, des modèles sur mesure répondant aux préférences spécifiques des utilisateurs ou même à des utilisateurs individuels peuvent être nécessaires. Par exemple, des approches de regroupement [61, 63] ont été proposées pour identifier les dispositifs ayant des distributions de données suffisamment similaires pour permettre la collaboration. L'objectif global (1) est ensuite appliqué à chaque groupe homogène d'appareils. Cependant, le partitionnement des appareils tout en tenant compte de la personnalisation présente un défi supplémentaire : les divergences statistiques doivent être déduites sans accès direct aux données afin de préserver la vie privée. Cela dit, pour faire progresser la FL personnalisée, il faut mettre au point des mesures d'évaluation localisées afin de déterminer si l'optimisation globale ou locale est appropriée en fonction du degré d'hétérogénéité.

Optimisation de l'apprentissage fédéré

Pour résoudre le problème (1), il est impératif de reconnaître que le gradient global peut être exprimé comme la somme des gradients locaux pond

$$\nabla L(\theta) = \sum_k w_k \nabla \ell_k(\theta). \tag{5}$$

Une fois le gradient global calculé, il peut être utilisé pour optimiser l'objectif dans (1), en appliquant la descente de gradient (`GD`) au cours de chaque cycle de communication t, conformément à la règle de mise à jour suivante :

$$\theta(t+1) = \theta(t) - \eta_t \nabla L(\theta(t)), \tag{6}$$

où $\eta^t > 0$ représente la taille du pas d'apprentissage qui peut varier entre les différents cycles de communication.
Dans le cadre de l'apprentissage fédéré, où les appareils ont accès à leurs données locales et reçoivent le modèle global $\theta(t)$ sur la liaison descendante (DL) de l'orchestrateur, le modèle local `GD` peut être utilisé pour calculer les gradients locaux. Compte tenu de la configuration en étoile qui prévaut dans l'apprentissage fédéré, les appareils transmettent à l'orchestrateur les gradients locaux qu'ils ont calculés sur la liaison montante (UL). Ces gradients sont ensuite pondérés et agrégés pour obtenir la mise à jour globale décrite dans (5). Cette mise à jour globale est ensuite appliquée au modèle global conformément à (6). Le modèle global mis à jour est ensuite envoyé à tous les dispositifs pour lancer le cycle de communication suivant. Le processus de formation s'achève lorsque la convergence est atteinte, ce qui indique que la norme du gradient global s'approche de zéro avec une faible marge. Une illustration du processus FL est donnée dans la Fig. 2.

Malgré son applicabilité théorique et les recherches approfondies dont elle a fait l'objet, la méthode `GD` n'est pas couramment utilisée dans les contextes pratiques de la FL. Cela est principalement dû à la surcharge de calcul qu'elle impose, car elle nécessite le calcul de gradients locaux

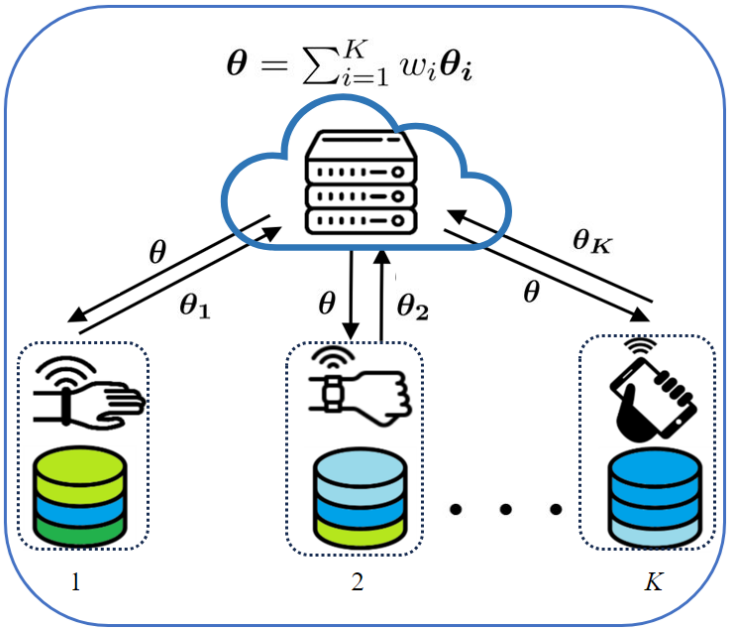


Figure 2: Illustration de l'apprentissage fédéré

complets. Cette exigence de calcul peut poser des problèmes, en particulier pour les appareils disposant de ressources limitées. En outre, `GD` implique une seule étape de mise à jour locale par cycle de communication, ce qui conduit à un taux de convergence relativement lent. Par conséquent, de nombreux cycles de communication entre les appareils et le serveur central sont nécessaires jusqu'à ce que la convergence soit atteinte. Cela augmente considérablement les frais généraux de communication, en particulier dans les réseaux à bande passante limitée. Des variantes stochastiques et adaptatives de `GD` sont utilisées, telles que `SGD` et `ADAM`.

Défis

Étant donné l'applicabilité des FL dans des contextes multi-appareils, soutenue par leurs capacités croissantes et leur omniprésence, elles sont confrontées à plusieurs problèmes critiques qu'il est nécessaire de résoudre pour réaliser leur plein potentiel. En particulier, l'efficacité de la communication au cours du processus fédéré sur les réseaux sans fil doit être optimisée. Les appareils présentent une hétérogénéité matérielle, logicielle et de données qui doit être conciliée. En outre, même avec les dispositions de FL en matière de protection de la vie privée, il subsiste des problèmes résiduels de protection de la vie privée qui justifient d'autres améliorations techniques. Dans les sous-sections suivantes, chacun de ces défis clés est examiné plus en détail.

Coûts de communication

Le goulot d'étranglement de la communication représente un défi important dans le contexte du FL [101], principalement lorsqu'il est appliqué à des réseaux périphériques à ressources limitées. Dans un cadre inter-appareils, FL nécessite l'agrégation fréquente des mises à jour de modèles provenant de nombreux appareils participants. Toutefois, la transmission de mises à jour complètes de modèles sur des réseaux sans fil à largeur de bande limitée peut souvent s'avérer irréalisable. La taille des mises à jour de modèles bruts, en particulier pour les grands modèles d'apprentissage profond, peut atteindre des centaines de mégaoctets, ce qui entraîne une congestion du réseau et un épuisement rapide des batteries des appareils. Cela peut entraver la participation des utilisateurs à un système fédéré, les incitant à se retirer de la formation en raison de la limitation des ressources sans fil ou du faible niveau des batteries [30].

Diverses approches ont été proposées dans la littérature pour atténuer le goulot d'étranglement de la communication. Un ensemble de méthodes implique la compression des modèles par la sparsification [31–33] et la quantification [34,35] afin de réduire la taille des mises à jour échangée . Néanmoins, il en résulte souvent un compromis avec une diminution de la précision du modèle. Un autre ensemble d'algorithmes se concentre sur la minimisation des frais généraux de communication en limitant le nombre d'appareils impliqués dans chaque cycle de communication [36–38]. Pour ce faire, des appareils fiables sont sélectionnés sur la base de critères spécifiques, tels que l'état du canal sans fil, l'état de la batterie et le fuseau horaire. Le chapitre 4 et une partie du chapitre 2 sont consacrés à la prise en compte de la charge de communication de la FL.

Les dispositifs IdO déployés dans des environnements éloignés ou à ressources limitées sont souvent caractérisés par une connectivité réseau restreinte et intermittente [40,41]. Néanmoins, leur rôle en tant que générateurs de données les positionne comme des candidats très appropriés pour la participation à la formation de modèles, contribuant ainsi au développement d'un bord intelligent [39]. Néanmoins, l'accès sporadique au réseau qui prévaut dans les zones rurales constitue un défi pour leur participation à la formation FL, étant donné l'exigence d'une communication cohérente avec les serveurs centraux [42, 43]. L'apprentissage entièrement décentralisé peut être utile dans ce contexte, en tirant parti de la communication de pair à pair entre les dispositifs IoT sans qu'un orchestrateur centralisé ne soit nécessaire. Cependant, la gestion de la coordination dans ces contextes reste un problème ouvert. Les approches modernes proposent l'utilisation de véhicules aériens sans pilote (UAV) comme relais dynamiques capables de superviser FL à la demande [44]. Cette stratégie est particulièrement bien adaptée aux situations caractérisées par une connectivité difficile, telles que celles couramment rencontrées par les dispositifs IoT dans les zones rurales. Le chapitre 3 de cette thèse examine en détail ce problème et explore l'utilisation de drones comme orchestrateurs de FL dans les zones reculées.

Hétérogénéité des données

L'hétérogénéité statistique constitue un défi important dans le domaine du FL, en raison des ensembles de données locales non IID (distribuées de manière non identique ou indépendante) des

dispositifs participants. Lorsque l'on parle de données non IID dans FL, on fait généralement référence aux différences sous-jacentes entre les distributions de données locales P_i et P_j pour les différents appareils i et j. Cette hétérogénéité des données se manifeste par des préférences individuelles, des caractéristiques géographiques spécifiques capturant des traits localisés, et des dynamiques transitoires spécifiques au temps [48]. Par exemple, les appareils IoT peuvent différer dans leurs taux d'échantillonnage ou leurs fréquences de collecte de données ; certains appareils peuvent collecter des données toutes les minutes, tandis que d'autres collectent des données toutes les heures. Ces variations peuvent affecter la résolution temporelle des données et introduire de l'hétérogénéité.

Il existe de nombreux scénarios typiques dans lesquels les données tendent à s'écarter d'une distribution identique. Si nous considérons les distributions locales soutenues par $(\mathcal{X}, \mathcal{Y})$, comme dans les environnements d'apprentissage supervisé, où $\mathcal{X}$ désigne l'espace des caractéristiques d'entrée et $\mathcal{Y}$ désigne l'espace des étiquettes de la vérité de terrain, alors la distribution des données de est définie comme $P_i(x, y)$. Les formes les plus courantes d'hétérogénéité des données sont [14] :

- Déplacement des covariables : Les distributions de probabilité des variables d'entrée $P_i(x)$ peuvent différer d'une population de clients à l'autre. Par exemple, dans un système collaboratif de surveillance de la santé FL, certains clients peuvent utiliser des capteurs médicaux haut de gamme, tandis que d'autres utilisent des dispositifs portables plus simples. Le bruit entre les mesures capturées par les différents appareils modifie les distributions d'entrée.

- Label Skew (biais d'étiquetage) : La distribution des étiquettes, représentée par $P_i(y)$, varie d'un appareil à l'autre. Par exemple, dans un scénario d'analyse fédérée des sentiments, un client disposant d'un ensemble de données important peut fournir des avis majoritairement positifs, tandis qu'un autre client peut avoir davantage d'avis négatifs. Par conséquent, le modèle global pourrait présenter un biais en faveur des étiquettes dominantes présentes dans l'appareil disposant de l'ensemble de données le plus important, compte tenu du système de pondération mentionné dans l'équation (4). Par conséquent, les performances du modèle global peuvent être médiocres pour les étiquettes sous-représentées dans les ensembles de données.

- Changement de concept : La distribution conditionnelle $P_i(y|x)$ varie d'un ensemble de données à l'autre. Les ensembles de données des appareils. Dans ce cas, les appareils peuvent attribuer des étiquettes différentes aux mêmes vecteurs de caractéristiques d'entrée. Par exemple, les étiquettes associées à la prédiction du mot suivant (par exemple, Gboard), compte tenu d'une phrase de départ (c'est-à-dire la caractéristique d'entrée), peuvent varier en fonction des choix personnels et des différences régionales.

Dans les scénarios pratiques, les ensembles de données comprennent souvent un mélange de ces effets, et l'objectif FL typique (1) aboutit à des modèles sous-optimaux lorsqu'il est confronté à ces effets. Alors que quelques algorithmes dans la littérature ont réussi à traiter la manifestation combinée de ces effets à travers les données des utilisateurs [14], de nombreuses approches se sont

concentrées sur le traitement de ces effets individuellement tout en négligeant l'interaction entre eux []. Le chapitre 2 de la thèse étudie l'interaction entre les effets mentionnés qui favorisent l'hétérogénéité et propose une approche de modélisation personnalisée comme solution potentielle.

Hétérogénéité du système

L'hétérogénéité du système FL entraîne des divergences dans les capacités et les propriétés des appareils clients. Cette hétérogénéité s'étend au matériel, à la connectivité et à la disponibilité. Les différences matérielles entre les appareils mobiles, intégrés et les serveurs induisent un parallélisme de calcul variable, qui affecte les vitesses d'apprentissage locales. L'hétérogénéité de la connectivité entraîne des différences dans les conditions et la fiabilité des canaux de communication. L'hétérogénéité de la disponibilité survient lorsque des traînards, définis comme des nœuds plus lents qui retardent l'exécution globale, apparaissent en raison de problèmes tels qu'une participation peu fiable, des limitations d'énergie et la mobilité. La multitude de divergences en matière de matériel, de connectivité et de disponibilité pose d'importants défis systémiques dans les environnements FL. Le chapitre 3 de la thèse présente une exploration approfondie de l'impact de l'hétérogénéité des canaux parmi les dispositifs IoT de faible puissance dans un cadre d'apprentissage fédéré orchestré par des drones.

Confidentialité de l'apprentissage

L'apprentissage fédéré a été initialement développé comme un moyen de répondre aux préoccupations en matière de protection de la vie privée découlant du partage des données des utilisateurs avec des serveurs en nuage pour l'entraînement des modèles de ML. En échangeant des mises à jour de modèles plutôt que des données brutes, l'apprentissage fédéré visait à fournir des garanties de confidentialité aux utilisateurs pendant le processus de formation. Toutefois, des recherches récentes ont mis en évidence des vulnérabilités potentielles dans lesquelles des adversaires pourraient exploiter les mises à jour de modèles pour déduire le contenu des ensembles de données des utilisateurs. Pour résoudre ce problème, les recherches en cours se concentrent sur l'amélioration de la confidentialité du FL grâce à des techniques telles que l'agrégation sécurisée, la confidentialité différentielle et le cryptage. Les méthodes d'agrégation sécurisée visent à garantir que les mises à jour de modèles provenant de dispositifs individuels sont combinées de manière à empêcher les adversaires d'extraire des informations sur des échantillons de données individuels. Les techniques de confidentialité différentielle introduisent un bruit aléatoire dans les mises à jour du modèle afin de protéger contre les fuites de confidentialité. Des méthodes de cryptage peuvent être employées pour sécuriser les mises à jour de modèles pendant la transmission, afin d'empêcher l'accès non autorisé et la falsification.

Bien que la protection de la vie privée soit un élément essentiel de la formation en FL, cette thèse n'explore pas les difficultés liées à la préservation de la vie privée au cours de la formation en FL.

Considérations sur la thèse:

Nous allons maintenant énumérer certaines hypothèses formulées dans les différents chapitres de la thèse.

- **Participation des appareils:** Dans les chapitres suivants, notre analyse repose sur l'hypothèse que tous les appareils sont disponibles pour la formation, sauf indication contraire. Cela signifie que le nombre total d'appareils disponibles pour la formation est fixe. Toutefois, le fait que tous les appareils soient sélectionnés pour la formation ou que les mises à jour soient transmises avec succès par les canaux de communication dépendra des hypothèses spécifiques formulées dans chaque chapitre.

- **Local Training:** L'apprentissage local est une technique largement utilisée en FL pour minimiser le surcoût de communication. Par conséquent, dans tous les chapitres, nous supposons que les dispositifs utilisent des méthodes d'approximation stochastique du gradient, telles que la descente stochastique du gradient par mini-lots, pour calculer les mises à jour locales avant de les envoyer à l'orchestrateur sur la liaison montante. Ces approximations stochastiques permettent de réduire les coûts de calcul locaux au niveau des appareils. Le nombre d'étapes est déterminé par le nombre d'époques prédéfini par l'orchestrateur.

Contributions et plan de thèse

Cette thèse est divisée en trois parties distinctes, chacune abordant des défis spécifiques découlant de l'intégration de l'apprentissage fédéré dans les réseaux périphériques sans fil.

- Dans le deuxième chapitre de cette thèse, nous abordons le problème de l'hétérogénéité des données à travers les ensembles de données de dispositifs dans l'apprentissage fédéré. Nous proposons une approche centrée sur l'utilisateur qui s'écarte du un modèle pour tous, souvent peu performant dans ce contexte, et offre à la place des modèles personnalisés et adaptés aux objectifs uniques de chaque utilisateur. Pour atténuer le surcoût de communication élevé associé à l'entraînement des modèles personnalisés, nous proposons une méthode de clustering qui regroupe les utilisateurs ayant des objectifs similaires, ce qui leur permet de collaborer pour produire un modèle personnalisé partagé. L'algorithme que nous proposons démontre des taux de convergence supérieurs à ceux de plusieurs algorithmes de personnalisation de pointe. Cette partie est associée au chapitre 2, et est basée sur les deux travaux publiés suivants:

  - Mohamad Mestoukirdi, Matteo Zecchin, David Gesbert, and Qianrui Li. "**User-centric federated learning, Trading off wireless resources for personalization**. Minor Review phase, submitted to "*IEEE Transactions on Machine Learning in Communications and Networking, 2023.*

– Mohamad Mestoukirdi, Matteo Zecchin, David Gesbert, Qianrui Li and Nicolas Gresset, "**User-Centric Federated Learning**," *IEEE Globecom Workshop, Madrid, Spain, 2021*, pp. 1-6, doi: 10.1109/GCWkshps52748.2021.9682003.

- Dans la deuxième partie de la thèse, nous nous concentrons sur l'intégration des dispositifs à distance de l'IdO dans la périphérie intelligente en tirant parti des drones en tant qu'orchestrateur d'apprentissage fédéré. Alors que les drones ont fait l'objet d'études approfondies pour leur potentiel à agir comme des stations de base volantes ou des relais dans les réseaux sans fil [56,57], l'application des drones dans la facilitation de la formation de modèles reste un domaine naissant. Le déploiement de drones offre plusieurs avantages, notamment la rentabilité et les capacités de formation à la demande. En outre, la mobilité des drones permet d'établir des liens de communication LoS (Line-of-Sight) avec des appareils situés dans des zones rurales, ce qui permet de contourner les conditions défavorables des canaux. Cependant, le déploiement de drones dans de tels environnements pose des problèmes en termes de programmation et de conception des trajectoires. Pour optimiser la trajectoire des drones et la programmation des appareils, nous proposons une mesure heuristique qui sert d'indicateur de la performance de l'entraînement. Sur la base de cette métrique, nous définissons un objectif de substitution qui permet l'optimisation conjointe de la trajectoire du drone et de l'ordonnancement du dispositif à l'aide de techniques d'optimisation convexe et de la théorie des graphes. Notre solution est plus performante que d'autres déploiements statiques et mobiles sélectionnés, comme le démontrent les résultats de la simulation. Ce segment résume le chapitre 3, qui est basé sur le travail publié :

  – Mohamad Mestoukirdi, Omid Esrafilian, David Gesbert, and Qianrui Li, "**UAV-Aided Multi-Community Federated Learning**," *IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022* , pp. 1314-1319.

- Dans la dernière partie de la thèse, nous nous concentrons sur le défi que représente la charge de communication associée à l'échange de mises à jour de modèles. Dans les algorithmes FL archétypiques, lors de chaque cycle de communication, les mises à jour du modèle sont souvent quantifiées ou réduites avant d'être envoyées sur les canaux UL ou DL, ce qui permet d'améliorer l'efficacité de la communication. Toutefois, cette compression se fait souvent au détriment de la précision du modèle. Pour remédier à ce problème, des recherches approfondies ont été menées afin d'explorer d'autres algorithmes permettant de dissocier la précision du modèle de l'efficacité de la communication en FL. Une approche prometteuse récente consiste à élaguer un réseau aléatoire pour obtenir une approximation d'un réseau cible, conformément au problème d'approximation de la somme des sous-ensembles, ce qui a permis d'obtenir des gains significatifs en termes d'efficacité de la communication et de généralisation du modèle. Cependant, nous montrons que les algorithmes de pointe existants qui adoptent de tels schémas dans des contextes fédérés ne parviennent pas à exploiter pleinement le potentiel d'amélioration de l'efficacité de la communication. En conséquence, nous

proposons un nouvel algorithme qui permet d'obtenir des gains de communication nettement plus importants. Notre approche favorise un élagage supplémentaire des réseaux aléatoires, ce qui se traduit par des mises à jour de modèles plus clairsemées. Il est important de noter que la solution que nous proposons garantit que cet élagage accru n'affecte pas négativement la performance de généralisation du modèle produit.

Cette partie est couverte par le chapitre 4 et est basée sur :

– Mohamad Mestoukirdi, Omid Esrafilian, David Gesbert, Qianrui Li, and Nicolas Gresset, 2023. **Sparser Random Networks Exist: Enforcing Communication-Efficient Federated Learning via Regularization.** arXiv preprint arXiv:2309.10834. (to be submitted to IEEE Communication letters)

Bibliography

[1] Guangxu Zhu, Dongzhu Liu, Yuqing Du, Changsheng You, Jun Zhang, and Kaibin Huang. Toward an intelligent edge: Wireless communication meets machine learning. *IEEE Communications Magazine*, 58(1):19–25, 2020.

[2] Ahmed Imteaj and M. Hadi Amini. Distributed sensing using smart end-user devices: Pathway to federated learning for autonomous iot. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1156–1161, 2019.

[3] Mingzhe Chen, Deniz Gündüz, Kaibin Huang, Walid Saad, Mehdi Bennis, Aneta Vulgarakis Feljan, and H Vincent Poor. Distributed learning in wireless networks: Recent progress and future challenges. *IEEE Journal on Selected Areas in Communications*, 39(12):3579–3605, 2021.

[4] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, 19(3):1657–1681, 2017.

[5] Benjamin Maschler and Michael Weyrich. Deep transfer learning for industrial automation: A review and discussion of new techniques for data-driven machine learning. *IEEE Industrial Electronics Magazine*, 15(2):65–75, 2021.

[6] Ahmet M. Elbir, Burak Soner, Sinem Çöleri, Deniz Gündüz, and Mehdi Bennis. Federated learning in vehicular networks. In *2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 72–77, 2022.

[7] Latif U. Khan, Shashi Raj Pandey, Nguyen H. Tran, Walid Saad, Zhu Han, Minh N. H. Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine*, 58(10):88–93, 2020.

[8] 3GPP release 18 features. Online. Accessed on September 19, 2023.

[9] Cisco. Cisco annual internet report (2018-2023). White paper, 2018.

[10] IoT Analytics. State of iot 2023. Retrieved September 19, 2023, from https://iot-analytics.com/number-connected-iot-devices/, May 24 2023.

[11] Sawsan Abdulrahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 8(7):5476–5497, 2021.

[12] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus H. Maier-Hein, Sébastien Ourselin, Micah J. Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of digital health with federated learning. *CoRR*, abs/2003.08119, 2020.

[13] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *Official Journal of the European Union*, L 119:1–88, 2016.

[14] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, K. A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G.L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. 2019.

[15] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.

[16] Nefi Alarcon. Clara train 3.1 brings secure, enterprise-grade federated learning to developers, 2020. [Online; accessed 20-September-2023].

[17] Karen Hao. How apple personalizes siri without hoovering up your data. MIT Technology Review, December 2019.

[18] Andrew Hard, Chloé M Kiddon, Daniel Ramage, Francoise Beaufays, Hubert Eichner, Kanishka Rao, Rajiv Mathews, and Sean Augenstein. Federated learning for mobile keyboard prediction, 2018.

[19] Solmaz Niknam, Harpreet S. Dhillon, and Jeffrey H. Reed. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, 58(6):46–51, 2020.

[20] Tianyu Wang, Shaowei Wang, and Zhi-Hua Zhou. Machine learning for 5g and beyond: From model-based to data-driven mobile wireless networks. *China Communications*, 16(1):165–175, 2019.

[21] Mingzhe Chen, Ursula Challita, Walid Saad, Changchuan Yin, and Mérouane Debbah. Artificial neural networks-based machine learning for wireless networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 21(4):3039–3071, 2019.

[22] Jingjing Wang, Chunxiao Jiang, Haijun Zhang, Yong Ren, Kwang-Cheng Chen, and Lajos Hanzo. Thirty years of machine learning: The road to pareto-optimal wireless networks. *IEEE Communications Surveys & Tutorials*, 22(3):1472–1514, 2020.

[23] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. Deep learning in mobile and wireless networking: A survey. *IEEE Communications surveys & tutorials*, 21(3):2224–2287, 2019.

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[25] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6357–6368. PMLR, 18–24 Jul 2021.

[26] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *CoRR*, abs/1812.06127, 2018.

[27] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[28] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[30] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[31] Emre Ozfatura, Kerem Ozfatura, and Deniz Gündüz. Time-correlated sparsification for communication-efficient federated learning. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 461–466. IEEE, 2021.

[32] Yuxuan Sun, Sheng Zhou, Zhisheng Niu, and Deniz Gündüz. Time-correlated sparsification for efficient over-the-air model aggregation in wireless federated learning. In *ICC 2022-IEEE International Conference on Communications*, pages 3388–3393. IEEE, 2022.

[33] Pengchao Han, Shiqiang Wang, and Kin K Leung. Adaptive gradient sparsification for efficient federated learning: An online learning approach. In *2020 IEEE 40th international conference on distributed computing systems (ICDCS)*, pages 300–310. IEEE, 2020.

[34] Ahmed M Abdelmoniem and Marco Canini. Towards mitigating device heterogeneity in federated learning via adaptive model quantization. In *Proceedings of the 1st Workshop on Machine Learning and Systems*, pages 96–103, 2021.

[35] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020.

[36] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H Vincent Poor, and Shuguang Cui. A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 20(1):269–283, 2020.

[37] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE international conference on communications (ICC)*, pages 1–7. IEEE, 2019.

[38] Sawsan AbdulRahman, Hanine Tout, Azzam Mourad, and Chamseddine Talhi. Fedmccs: Multicriteria client selection model for optimal iot federated learning. *IEEE Internet of Things Journal*, 8(6):4723–4735, 2020.

[39] Nguyen Cong Luong, Dinh Thai Hoang, Ping Wang, Dusit Niyato, Dong In Kim, and Zhu Han. Data collection and wireless communication in internet of things (iot) using economic analysis and pricing models: A survey. *IEEE Communications Surveys & Tutorials*, 18(4):2546–2590, 2016.

[40] Jiaming Pei, Shike Li, Zhi Yu, Laishan Ho, Wenxuan Liu, and Lukun Wang. Federated learning encounters 6g wireless communication in the scenario of internet of things. *IEEE Communications Standards Magazine*, 7(1):94–100, 2023.

[41] Zubair Md Fadlullah and Nei Kato. On smart iot remote sensing over integrated terrestrial-aerial-space networks: An asynchronous federated learning approach. *IEEE Network*, 35(5):129–135, 2021.

[42] Hong Xing, Osvaldo Simeone, and Suzhi Bi. Decentralized federated learning via sgd over wireless d2d networks. In *2020 IEEE 21st international workshop on signal processing advances in wireless communications (SPAWC)*, pages 1–5. IEEE, 2020.

[43] Hong Xing, Osvaldo Simeone, and Suzhi Bi. Federated learning over wireless device-to-device networks: Algorithms and convergence analysis. *IEEE Journal on Selected Areas in Communications*, 39(12):3723–3741, 2021.

[44] Matteo Zecchin, David Gesbert, and Marios Kountouris. Uav-aided decentralized learning over mesh networks. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 702–706. IEEE, 2022.

[45] Mohamad Mestoukirdi, Omid Esrafilian, David Gesbert, and Qianrui Li. Uav-aided multi-community federated learning. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 1314–1319, 2022.

[46] Ruslan Zhagypar, Nour Kouzayha, Hesham ElSawy, Hayssam Dahrouj, and Tareq Y Al-Naffouri. Characterization of the global bias problem in aerial federated learning. *IEEE Wireless Communications Letters*, 2023.

[47] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks, 2020.

[48] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 794–797. IEEE, 2020.

[49] Bing Luo, Wenli Xiao, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pages 1739–1748. IEEE, 2022.

[50] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[51] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. Can you really backdoor federated learning? *CoRR*, abs/1911.07963, 2019.

[52] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[53] Jialing Liao, Zheng Chen, and Erik G. Larsson. Over-the-air federated learning with privacy protection via correlated additive perturbations. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–8, 2022.

[54] Mohamad Mestoukirdi, Matteo Zecchin, David Gesbert, and Qianrui Li. User-centric federated learning: Trading off wireless resources for personalization. *IEEE Transactions on Machine Learning in Communications and Networking*, 1:346–359, 2023.

[55] Mohamad Mestoukirdi, Matteo Zecchin, David Gesbert, Qianrui Li, and Nicolas Gresset. User-centric federated learning. In *2021 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2021.

[56] Omid Esrafilian, Rajeev Gangula, and David Gesbert. Uav-relay placement with unknown user locations and channel parameters. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 1075–1079, 2018.

[57] Omid Esrafilian and David Gesbert. Simultaneous user association and placement in multi-uav enabled wireless networks. In *WSA 2018; 22nd International ITG Workshop on Smart Antennas*, pages 1–5, 2018.

[58] George S. Lueker. Exponentially small bounds on the expected optimum of the partition and subset sum problems. *Random Structures & Algorithms*, 12(1):51–62, 1998.

[59] Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris S. Papailiopoulos. Optimal lottery tickets via subsetsum: Logarithmic over-parameterization is sufficient. *CoRR*, abs/2006.07990, 2020.

[60] Mohamad Mestoukirdi, Omid Esrafilian, David Gesbert, Qianrui Li, and Nicolas Gresset. Sparser random networks exist: Enforcing communication-efficient federated learning via regularization, 2023.

[61] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[62] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[63] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020.

[64] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021.

[65] Mohamad Mestoukirdi, Matteo Zecchin, David Gesbert, Qianrui Li, and Nicolas Gresset. User-Centric federated learning. In *2021 IEEE Globecom Workshops (GC Wkshps): Workshop on Wireless communications for distributed intelligence (GC 2021 Workshop - WCDI)*, Madrid, Spain, December 2021.

[66] Othmane MARFOQ, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In *Advances in Neural Information Processing Systems*, 2021.

[67] Matthias Reisser, Christos Louizos, Efstratios Gavves, and Max Welling. Federated mixture of experts. *arXiv preprint arXiv:2107.06724*, 2021.

[68] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for on-device federated learning. *CoRR*, abs/1910.06378, 2019.

[69] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *CoRR*, abs/2006.08848, 2020.

[70] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[71] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Proceedings of The 22nd Annual Conference on Learning Theory (COLT 2009)*, Montréal, Canada, 2009.

[72] Changjian Shui, Qi Chen, Jun Wen, Fan Zhou, Christian Gagné, and Boyu Wang. Beyond h-divergence: Domain adaptation theory with jensen-shannon divergence. *ArXiv*, abs/2007.15567, 2020.

[73] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[74] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Report, University of Toronto*, 2009.

[75] Google Cloud. Stack-overflow dataset, 2017. Accessed on May 30, 2023.

[76] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. 2020.

[77] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[78] Tensorflow. Basic text classification, 2023. Accessed on August 30, 2023.

[79] Jason Posner, Lewis Tseng, Moayad Aloqaily, and Yaser Jararweh. Federated learning in vehicular networks: Opportunities and solutions. *IEEE Network*, 35(2):152–159, 2021.

[80] Francesco Malandrino, Carla-Fabiana Chiasserini, Claudio Casetti, Luca Chiaraviglio, and Andrea Senacheribbe. Planning uav activities for efficient user coverage in disaster areas. *Ad Hoc Networks*, 89:177–185, 2019.

[81] Omid Esrafilian, Rajeev Gangula, and David Gesbert. Autonomous uav-aided mesh wireless networks. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 634–640, 2020.

[82] Zdenek Becvar, Michal Vondra, Pavel Mach, Jan Plachy, and David Gesbert. Performance of mobile networks with uavs: Can flying base stations substitute ultra-dense small cells? In *European Wireless 2017; 23th European Wireless Conference*, pages 1–7, 2017.

[83] Igor Donevski, Nithin Babu, Jimmy Jessen Nielsen, Petar Popovski, and Walid Saad. Federated learning with a drone orchestrator: Path planning for minimized staleness. *IEEE Open Journal of the Communications Society*, 2:1000–1014, 2021.

[84] Tengchan Zeng, Omid Semiari, Mohammad Mozaffari, Mingzhe Chen, Walid Saad, and Mehdi Bennis. Federated learning in the sky: Joint power allocation and scheduling with UAV swarms. 2020.

[85] Eunjeong Jeong, Matteo Zecchin, and Marios Kountouris. Asynchronous decentralized learning over unreliable wireless networks. abs/2202.00955, 2022.

[86] Akram Al-Hourani, Sithamparanathan Kandeepan, and Simon Lardner. Optimal LAP altitude for maximum coverage. *IEEE Wireless Communications Letters*, 2014.

[87] Michael Grant, Stephen Boyd, and Yinyu Ye. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx., 2013.

[88] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 2012.

[89] Jun Sun, Tianyi Chen, Georgios Giannakis, and Zaiyue Yang. Communication-efficient distributed learning via lazily aggregated quantized gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[90] Dan Alistarh, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: randomized quantization for communication-optimal stochastic gradient descent. *CoRR*, abs/1610.02132, 2016.

[91] Ran Ben Basat, Shay Vargaftik, Amit Portnoy, Gil Einziger, Yaniv Ben-Itzhak, and Michael Mitzenmacher. Quic-fl: Quick unbiased compression for federated learning, 2023.

[92] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? *CoRR*, abs/1911.13299, 2019.

[93] Ang Li, Jingwei Sun, Xiao Zeng, Mi Zhang, Hai Li, and Yiran Chen. Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys '21, page 42–55, New York, NY, USA, 2021. Association for Computing Machinery.

[94] Berivan Isik, Francesco Pase, Deniz Gunduz, Tsachy Weissman, and Michele Zorzi. Sparse random networks for communication-efficient federated learning, 2023.

[95] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. *CoRR*, abs/1905.01067, 2019.

[96] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research).

[97] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[98] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[99] Eran Malach, Gilad Yehudai, Shai Shalev-Shwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. *CoRR*, abs/2002.00585, 2020.

[100] Daiki Chijiwa, Shin'ya Yamaguchi, Yasutoshi Ida, Kenji Umakoshi, and Tomohiro Inoue. Pruning randomly initialized neural networks with iterative randomization. *CoRR*, abs/2106.09269, 2021.

[101] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.