

# User-Centric Federated Learning: Trading off Wireless Resources for Personalization

Mohamad Mestoukirdi<sup>†</sup>, *Student Fellow, IEEE*, Matteo Zecchin<sup>†</sup>, *Student Fellow, IEEE*,  
David Gesbert, *Fellow, IEEE*, and Qianrui Li, *Member, IEEE*

**Abstract**—Statistical heterogeneity across clients in a Federated Learning (FL) system increases the algorithm convergence time and reduces the generalization performance, resulting in a large communication overhead in return for a poor model. To tackle the above problems 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. In this work, 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 FL client. The proposed aggregation rules are inspired by an upper bound of the weighted aggregate empirical risk minimizer. Secondly, we derive a communication-efficient variant based on user clustering which greatly enhances its applicability to communication-constrained systems. Our algorithm outperforms popular personalized FL baselines in terms of average accuracy, worst node performance, and training communication overhead.

**Index Terms**—Distributed Optimization, Personalized Federated Learning, Statistical learning theory

## I. INTRODUCTION

IN recent years, the evolution of energy and computation-efficient hardware, together with the wide adoption of data-driven solutions have led to an increased interest in pushing intelligence closer to edge devices where data is generated. This contributed to the emergence of autonomous and intelligent systems where decisions are made locally. However, the reliable operation of intelligent edge devices requires periodic ML training and tuning, which mainly relies on pooling data from a multitude of devices toward a central entity. Consequently, privacy concerns arise in such settings, as data owners may be reluctant to share sensible and personal pieces of information [1]. Additionally, the soaring model complexity of modern ML solutions requires vast amounts of data to be harvested to achieve satisfactory inference accuracy. This introduces a large communication overhead and long training delays.

Federated Learning (FL) [2] was introduced to deal with these problems. It offers clients the possibility of collaboratively training models under the orchestration of a parameter server (PS), by iteratively aggregating locally optimized models without the need to offload any raw data centrally. Such

an approach tackles both the privacy and communications challenges mentioned above. 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 [2]) algorithms can perform and generalize well yielding a single model, fitting the common data distribution. However, this assumption is hardly met in practice [3], 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 [4], 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 is 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.

In [5], we proposed 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. [2]. In the case of a set of  $m$  collaborating devices, the original objective in [2] produces a common model at each communication round  $t$  according to

$$\theta^t \leftarrow \sum_{i=1}^m w_i \theta_i^{t-1/2}, \quad (1)$$

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

$$\theta_i^t \leftarrow \sum_{j=1}^m w_{i,j} \theta_j^{t-1/2}, \quad \text{for } i = 1, 2, \dots, m. \quad (2)$$

The key motivation underpinning the use of distinct user-centric personalization rules is that a single model often

<sup>†</sup> Equal contribution.

M. Mestoukirdi, M. Zecchin, and D. Gesbert are with the Communication Systems Department, EURECOM, Sophia-Antipolis, France. Emails: {Mestouki, Zecchin, Gesbert}@eurecom.fr. M. Mestoukirdi and Q. Li are with Mitsubishi Electric R&D Centre Europe. Emails: {M.Mestoukirdi, Q.Li}@fr.merce.mee.com.

The work of M. Zecchin is funded by the Marie Skłodowska Curie action WINDMILL (grant No. 813999).

fails in heterogeneous settings [2]. At the same time, hard clustering strategies [3], [6] are limited to restrictive intra-cluster collaboration and they cannot exploit similarities among different clusters. The authors in [7] proposed FedFomo, a personalization scheme that uses a similar aggregation policy as ours [5]. 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.

#### A. Contributions:

In this work, we build upon the findings of our previous work [5] and extend the results 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.

## II. 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 [8] propose a novel framework for enabling the implementation of FL algorithms over wireless networks by jointly taking into account FL and wireless factors. [3], [6] 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 [9] 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 [10], 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 [7], exploiting the full personalization potential of the solutions in [9], [10] induces a huge overhead over the communication resources in the federated system, which renders their approaches unpractical. Similar to Fedprox [11], the authors in [12] 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 [3], 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 [13] propose Ditto, where users collaborate to train a separate global model akin to [2], 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 [14], 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.

## III. 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  $m$  nodes, each endowed with a data set  $\mathcal{D}_i$  of  $n_i$  IID samples from a local distribution  $P_i$ , the goal is to find a predictor  $f : \mathcal{X} \rightarrow \hat{\mathcal{Y}}$  from a hypothesis class  $\mathcal{F}$  that minimizes

$$L(f, \vec{w}) = \sum_{i=1}^m \frac{w_i}{n_i} \sum_{(x,y) \in \mathcal{D}_i} \ell(f(x), y), \quad (3)$$

where  $\ell : \hat{\mathcal{Y}} \times \mathcal{Y} \rightarrow \mathbb{R}^+$  is a loss function and  $\vec{w} = (w_1, \dots, w_m)$  is a weighting scheme. In the case of identically distributed local data sets, the typical weighting vector is  $\vec{w} = \frac{1}{\sum_i n_i} (n_1, \dots, n_m)$ , 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  $P_{\vec{w}} = \sum_{i=1}^m w_i P_i$ , the mixture of distributions which the final global model is trained to generalize over. Furthermore, it 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. In order 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  $\vec{w}_i = (w_{i,1}, \dots, w_{i,m})$  (where  $w_{i,j}$  defines how much node  $i$  relies on data from user  $j$ ) we define the personalized objective for user  $i$

$$L(f, \vec{w}_i) = \sum_{j=1}^m \frac{w_{i,j}}{n_j} \sum_{(x,y) \in \mathcal{D}_j} \ell(f(x), y), \quad (4)$$

and the resulting personalized model

$$\hat{f}_{\vec{w}_i} = \arg \min_{f \in \mathcal{F}} L(f, \vec{w}_i). \quad (5)$$

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 in order 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 [15], 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.

In order 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 [16] provides a measure of similarity between learning tasks that can be used to this end. For a hypothesis set of functions  $\mathcal{F} : \mathcal{X} \rightarrow \hat{\mathcal{Y}}$  and two distributions  $P, Q$  on  $\mathcal{X}$ , the discrepancy distance is defined as

$$d_{\mathcal{F}}(P, Q) = \sup_{f, f' \in \mathcal{F}} |\mathbb{E}_{x \sim P} [\ell(f, f')] - \mathbb{E}_{x \sim Q} [\ell(f, f')]|, \quad (6)$$

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

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

where  $\gamma = \inf_{f \in \mathcal{F}} (\mathbb{E}_{(x,y) \sim P} [\ell(f, y)] + \mathbb{E}_{(x,y) \sim Q} [\ell(f, y)])$ . 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  $\ell$   $B$ -bounded range, symmetric and satisfying the triangular inequality, with probability  $1 - \delta$  the function  $f_{\vec{w}_i}$  satisfies

$$\begin{aligned} & \mathbb{E}_{z \sim P_i} [\ell(\hat{f}_{\vec{w}_i}, z)] - \mathbb{E}_{z \sim P_i} [\ell(f^*, z)] \leq \\ & B \sqrt{\sum_{j=1}^m \frac{w_{i,j}^2}{n_j}} \left( \sqrt{\frac{2d}{\sum_i n_i} \log \left( \frac{e \sum_i n_i}{d} \right)} + \sqrt{\log \left( \frac{2}{\delta} \right)} \right) + \\ & 2 \sum_{j=1}^m w_{i,j} d_{\mathcal{F}}(P_i, P_j) + 2\gamma, \end{aligned}$$

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

Recently, an alternative bound based on an information-theoretic notion of dissimilarity, the Jensen-Shannon divergence, has been proposed [17]. It is based on less restrictive constraints, as it only requires the loss function  $\ell(f, Z)$  to be sub-Gaussian of some parameter  $\sigma$  for all  $f \in \mathcal{F}$ , and therefore whenever  $\ell(\cdot)$  is bounded, the requirement is automatically satisfied. Measuring similarity by the Jensens-Shannon divergence the following inequality is available

$$\mathbb{E}_{X \sim P} [X] \leq \mathbb{E}_{X \sim Q} [X] + \beta \sigma^2 + \frac{D_{JS}(P||Q)}{\beta}, \quad \text{for } \beta > 0, \quad (7)$$

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

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

$$\begin{aligned} & \mathbb{E}_{z \sim P_i} [\ell(\hat{f}_{\vec{w}_i}, z)] - \mathbb{E}_{z \sim P_i} [\ell(f^*, z)] \leq \\ & B \sqrt{\sum_{j=1}^m \frac{w_{i,j}^2}{n_j}} \left( \sqrt{\frac{2d}{\sum_i n_i} \log \left( \frac{e \sum_i n_i}{d} \right)} + \sqrt{\log \left( \frac{2}{\delta} \right)} \right) + \\ & B \sqrt{2 \sum_{j=1}^m w_{i,j} D_{JS}(P_i||P_j)}, \end{aligned}$$

**Proof of Theorems 1 and 2:** In the Appendix VIII.

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.

#### IV. 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 (3) with  $\vec{w} = \frac{1}{\sum_i n_i} (n_1, \dots, n_m)$ . At each round  $t$ , the PS broadcasts the parameter vector  $\theta^{t-1}$  and then combines the locally optimized models by the clients  $\{\theta_i^{t-1}\}_{i=1}^m$  according to the following aggregation rule

$$\theta^t \leftarrow \sum_{i=1}^m \frac{n_i}{\sum_{j=1}^m n_j} \theta_i^{t-1}.$$

As mentioned in Sec. III, 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

$$\theta_i^t \leftarrow \sum_{j=1}^m w_{i,j} \theta_j^{t-1/2}, \quad \text{for } i = 1, \dots, m, \quad (8)$$

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

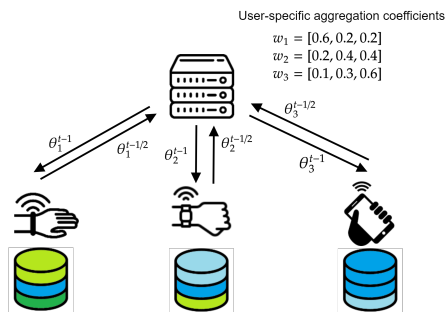


Fig. 1: Personalized Federated Learning with user-centric aggregates at round  $t$ .

next, the mixing coefficients are heuristically defined based on a distribution similarity metric and the data set size ratios. These coefficients are calculated before the start of federated training. The similarity 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 (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 [7] capitalizes on the same intuition.

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 IV-A, 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 at the device side is identical to that of standard federated learning. Each device is simply required to locally optimize the received model. At 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 IV-B 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 V.

##### A. Computing the Collaboration Coefficients

Computing the discrepancy distance (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 similarity metric given by

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

where the quality of the approximation depends on the number of samples  $n_i$  and  $n_j$ . The mixing coefficients for user  $i$  are then set to the following normalized Gaussian kernel function

$$w_{i,j} = \frac{\frac{n_j}{n_i} e^{-\frac{1}{2\sigma_i\sigma_j} \Delta_{i,j}(\hat{\theta})}}{\sum_{j'=1}^m \frac{n_{j'}}{n_i} e^{-\frac{1}{2\sigma_i\sigma_{j'}} \Delta_{i,j'}(\hat{\theta})}}, \quad \text{for } j = 1, \dots, m. \quad (9)$$

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  $\sigma_i^2$  partitioning the local data randomly in  $K$  batches  $\{\mathcal{D}_i^k\}_{k=1}^K$  of size  $n_k$  and computing

$$\sigma_i^2 = \frac{1}{K} \sum_{k=1}^K \left\| \frac{1}{n_k} \sum_{(x,y) \in \mathcal{D}_i^k} \nabla \ell(f_{\hat{\theta}}, y) - \frac{1}{n_i} \sum_{(x,y) \in \mathcal{D}_i} \nabla \ell(f_{\hat{\theta}}, y) \right\|^2, \quad (10)$$

where  $\sigma_i^2$  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-V-F. 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 (8).

The proposed heuristic (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.

### B. Reducing the Communication Load

A full-fledged personalization employing the user-centric aggregation rule (8) would introduce an  $m$ -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

---

### Algorithm 1: User-centric Federated Learning

---

**Input** : number of clients  $m$ , local mini-batch size  $B$ , number of epochs  $E$  and learning rate  $\eta$

PS broadcasts  $\theta^0$  to the users

**foreach** user  $k$  **do**

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

    Compute  $\sigma_k^2$  as in (10)

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

**end**

PS computes  $w_{i,j}$  as in (9)

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

    PS unicasts  $\theta_k^t$  to each node  $k$

**foreach** node  $k$  **do**

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

        return  $\theta_k^{t+1}$  to PS

**end**

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

**end**

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

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

$\theta_k \leftarrow \theta_k^t$

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

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

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

**end**

**end**

**return**  $\theta_k$

---

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  $m_t$ , we run a  $k$ -means clustering scheme over the set of collaboration vectors  $\{\vec{w}_i\}_{i=1}^m$  and we select the centroids  $\{\vec{c}_i\}_{i=1}^{m_t}$  to implement the  $m_t$  personalized streams. Formally, given  $m_t$  and the user-specific weights  $\{\vec{w}_i\}_{i=1}^m$ , the objective is to find  $m_t < m$  clusters  $\mathcal{C}_1, \dots, \mathcal{C}_{m_t}$  such that

$$\sum_{n=1}^{m_t} \sum_{\vec{w}_i \in \mathcal{C}_n} \|\vec{w}_i - \vec{c}_n\| \quad (11)$$

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 save 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  $m_t$ .

---

**Algorithm 2:** Silhouette based scoring
 

---

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

---

### C. Choosing the Number of Personalized Streams

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 a clustering  $\mathcal{C}_1, \dots, \mathcal{C}_{m_t}$ , 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

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

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  $\{\vec{w}_i\}_{i=1}^m$  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 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  $m_t$  based on a communication-personalization trade-off function  $c(k, s_k) : \mathbb{N} \times [-1, 1] \rightarrow \mathbb{R}$  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

$$c(k, s_k) = s_k - \lambda k. \quad (12)$$

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 cost. 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.

## V. EXPERIMENTS

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

### A. Set-up

In our simulation we consider a handwritten character/digit recognition task using the EMNIST data set [18], an image classification task using the CIFAR-10 data set [19], and a text sentiment classification on Stack Overflow questions dataset extracted from the much larger public Stack-Overflow dataset on BigQuery [20]. Data heterogeneity is induced by splitting and transforming the data set differently across the group of devices. In particular, we analyze three 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 [9], [21].
- **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  $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$  respectively. In particular, heterogeneity is imposed such that  $p_i(x|y) \neq p_j(x|y), \forall i \in \mathcal{G}_k, j \in \mathcal{G}_{k'}, k \neq k', \forall (x, y) \in \mathcal{X} \times \mathcal{Y}$ .
- **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  $f_i, f_j : \mathcal{X} \rightarrow \mathcal{Y}$ , 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.

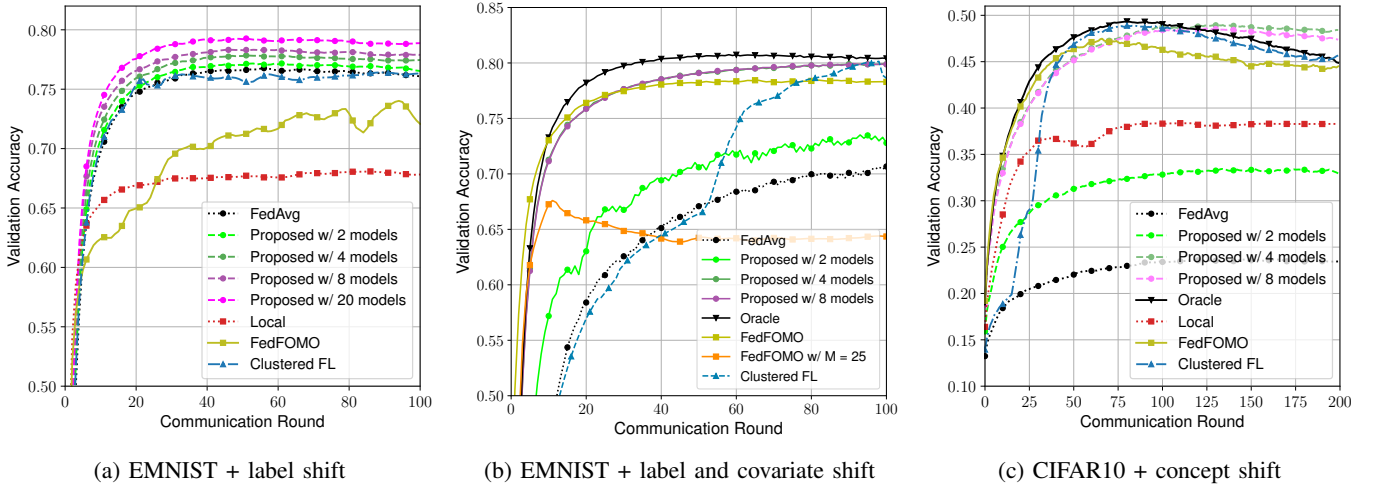


Fig. 2: Average Validation Accuracy across the three different experiments

TABLE I: Average test accuracy of the different algorithms across the three proposed scenarios.

Algorithm	Scenario		
	EMNIST ( $m = 20$ ) label shift	EMNIST ( $m = 100$ ) covariate & label shift	CIFAR10 ( $m = 20$ ) concept shift
Proposed $k = m$	<b>79.4</b> ( $\pm 4.2$ )	<b>77.9</b> ( $\pm 2.7$ )	<b>47.7</b> ( $\pm 2.2$ )
Proposed $k = 4$	77.8 ( $\pm 3.9$ )	<b>79.7</b> ( $\pm 2.5$ )	<b>49.1</b> ( $\pm 1.4$ )
SCAFFOLD [12]	77.2 ( $\pm 4.0$ )	72.5 ( $\pm 2.2$ )	17.5 ( $\pm 1.8$ )
Ditto [13]	78.3 ( $\pm 3.9$ )	74.1 ( $\pm 2.3$ )	44.1 ( $\pm 1.4$ )
pFedMe [14]	77.6 ( $\pm 4.1$ )	75.2 ( $\pm 4.4$ )	46.6 ( $\pm 1.5$ )
Fedprox [11]	<b>79.6</b> ( $\pm 4.8$ )	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 [2]	76.7 ( $\pm 4.0$ )	70.5 ( $\pm 2.2$ )	24.2 ( $\pm 2.6$ )
Oracle ( <i>Upper bound</i> )	-	<b>80.7</b> ( $\pm 1.8$ )	<b>49.5</b> ( $\pm 1.2$ )

TABLE II: Worst user performance averaged over 5 experiments in the three simulation scenarios

Scenario	Algorithm						
	Ditto [13]	FedAvg [2]	Oracle	CFL [3]	FedFOMO [7]	pFedMe [14]	Proposed
EMNIST ( $m = 20$ ) label shift	72.2	68.9	-	70.3	70.0	71.5	<b>73.2</b> ( $k = 20$ )
EMNIST ( $m = 100$ ) covariate & label shift	70.7	67.5	77.4	76.1	73.6	70.9	<b>76.4</b> ( $k = 4$ )
CIFAR10 ( $m = 20$ ) concept shift	43.2	19.6	49.1	48.6	45.5	45.3	<b>48.8</b> ( $k = 4$ )

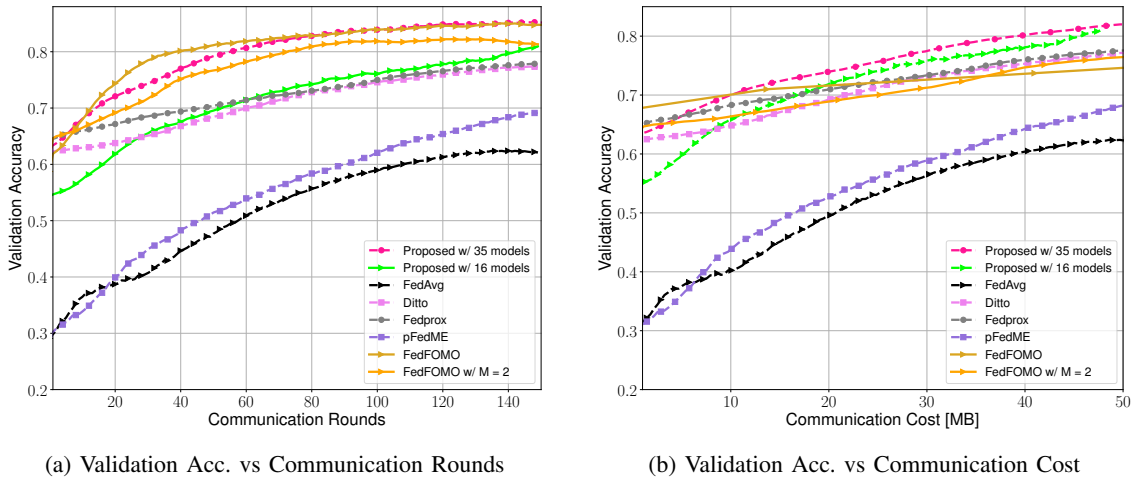


Fig. 3: Average Validation Accuracy across the different algorithms over the Stack-Overflow Sentiment dataset



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 [3], FedFomo [7], pFedMe [14] and Ditto [13]. The second set of baselines include algorithms that yield a single Federated model such as Fedprox<sup>1</sup> [11], SCAFFOLD [12]. FedAvg [2], and Local training algorithms are also included for reference. The image classification tasks are trained using the LeNet-5 [22] 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  $\eta = 0.01$ , momentum  $\beta = 0.9$ , and the number of epochs  $E = 1$  and mini-batch size = 64 for image classification tasks<sup>2</sup>. 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 [23]. 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  $K$ , by choosing the value that maximizes the silhouette score. Nevertheless, we also conducted experiments using different values of  $K$  to illustrate the trade-off between learning accuracy and communication efficiency.

## B. 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.

1) *Multi-Model Baseline Algorithms*: In Fig.2 and Table I, 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.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.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 personalization 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.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 the experiment we run the proposed scheme we consider the proposed scheme with full-fledged personalization in which each user has its own personalized model, and the communication efficient version with a number of personalized streams equal to 16.

In Figure 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, 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 3b. In this case the proposed algorithm converges to a larger validation accuracy compared to all other

<sup>1</sup>The penalization hyperparameters  $\mu$  and  $\lambda = \{0.1, 0.5, 1\}$  were used in the simulations of Fedprox and Ditto, then, the best results were reported.

<sup>2</sup>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



algorithms.

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 I), 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.

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 II 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.

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. 4. Interestingly, we notice that in the EMNIST covariate shift experiment (Fig. 4a), our clustering algorithm can detect similarities among the different groups of users, leveraging inter-cluster collaboration among them, unlike hard clustering algorithms [3]. 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  $\{0, 1, 8\}$ ).

### C. Silhouette Score

In Fig. 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. IV-B. In the labels shift scenario, for which we have seen that a universal model performs almost as well as the personalized ones, the silhouette scores monotonically decrease with  $k$ . 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  $k = 4$ . In Sec. V-B 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.

### D. 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  $\rho = \frac{T_{ul}}{T_{dl}}$  the ratio between model transmission time in uplink (UL) and downlink (DL). Typical values of  $\rho$  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  $m$  devices, we then have

$$T_{\text{comp}} = \mathbb{E}[\max\{T_1, \dots, T_m\}] = T_{\min} + \frac{H_m}{\mu},$$

where  $H_m$  is the  $m$ -th harmonic number. Accordingly, the communication round duration is

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

where  $N$  denotes the number of clients in the federated system, and  $m$  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  $T_{\text{comp}}$  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 asymmetry 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. 6 we report the time evolution of the validation accuracy in 3 different systems: wireless systems with slow UL  $\rho = 4$  and unreliable nodes  $T_{\min} = T_{dl} = \frac{1}{\mu}$ , a wireless system with fast uplink  $\rho = 2$  and reliable nodes  $T_{\min} = T_{dl}, \frac{1}{\mu} = 0$  and a wired system  $\rho = 1$  (symmetric UL and DL) with reliable nodes  $T_{\min} = T_{dl}, \frac{1}{\mu} = 0$ . In all plots, we normalize the time axis by  $T_{dl}$  in order 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  $k = 4$ . Note that FedFOMO incurs a large communication cost as personalized aggregation is performed on the client side.

### E. 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 following sense: In the typical FL framework, at each communication round  $t$ , the PS aggregates the models that

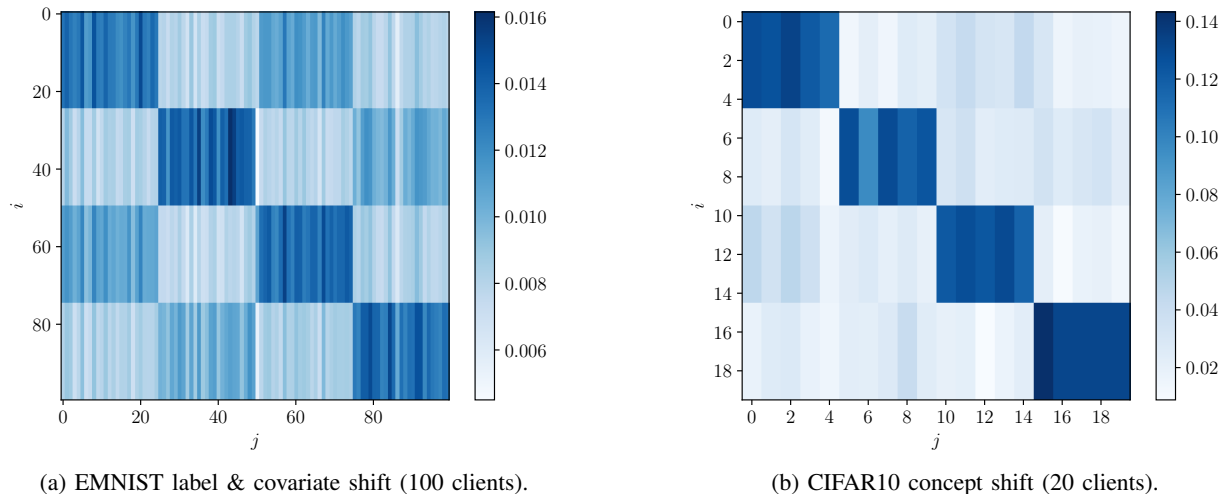


Fig. 4: Clusters formed by our proposed algorithm in the EMNIST label and covariate shift, and CIFAR10 concept shift scenarios. Each 2D point denotes  $w_{i,j}$ : A dark blue point  $w_{i,j}$  conveys a relatively large collaboration between user  $i$  and  $j$ .

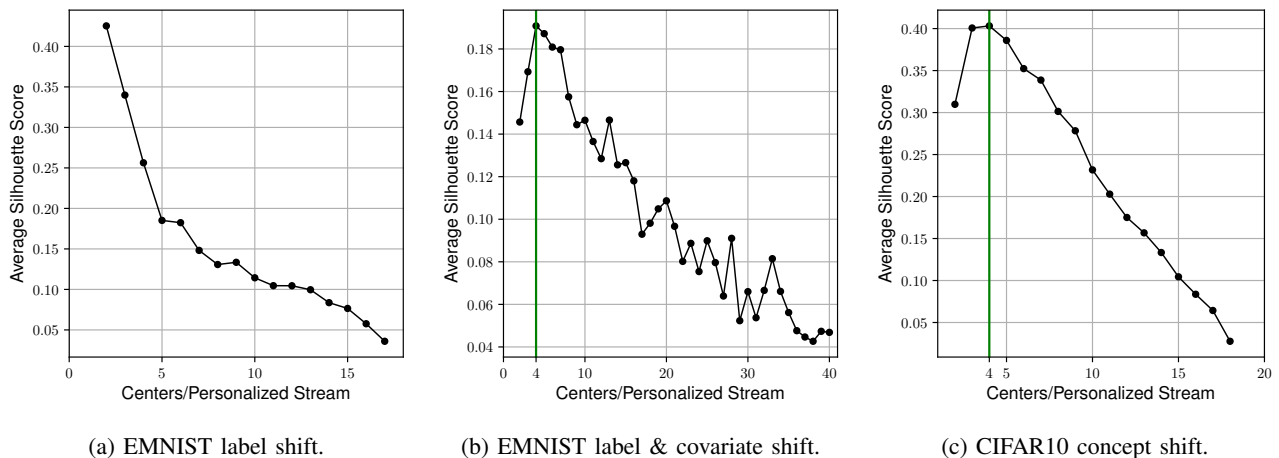


Fig. 5: Average silhouette scores of the  $k$ -means clustering in the three scenarios. In the last two scenarios, in which user inherently belongs to 4 different cluster, the scores indicates the necessity of at least 4 personalized streams.

were locally trained, at each participating device, starting from the same launch model  $\theta^{t-1}$ . 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 to. 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 [7]. 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 (4), we would have to run in parallel as many FL instances as the number of personalized streams  $m_t$  and incur a  $m_t$ -fold computation and uplink communication load.

To assess the quality of our assumption, we consider running in parallel  $m_t$  collaborative FL instances employing the proposed user-centric weights and solving exactly (4) for

each different aggregation rule. At each communication round, each user also optimizes the user-centric models of the other  $m_t - 1$  personalized streams which are then used at the PS server to apply the user-centric aggregation rules

$$\theta_i^t \leftarrow \sum_{j=1}^m w_{i,j} \theta_{i,j}^{t-1/2}, \quad \text{for } i = 1, 2 \dots, m. \quad (13)$$

Note that the aggregation rule in (13) is different from the one in (2), as  $\theta_{i,j}^{t-1/2}$  denotes the update of user  $j$  to the model of user  $i$  obtained by locally optimizing  $\theta_j^{t-1}$ .

We experiment using the EMNIST data set with label shift and the CIFAR10 data set with covariate and label shift. We set  $m_t = m = 20$  and use the same neural network model and settings indicated in Sec. V. In Fig. 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.

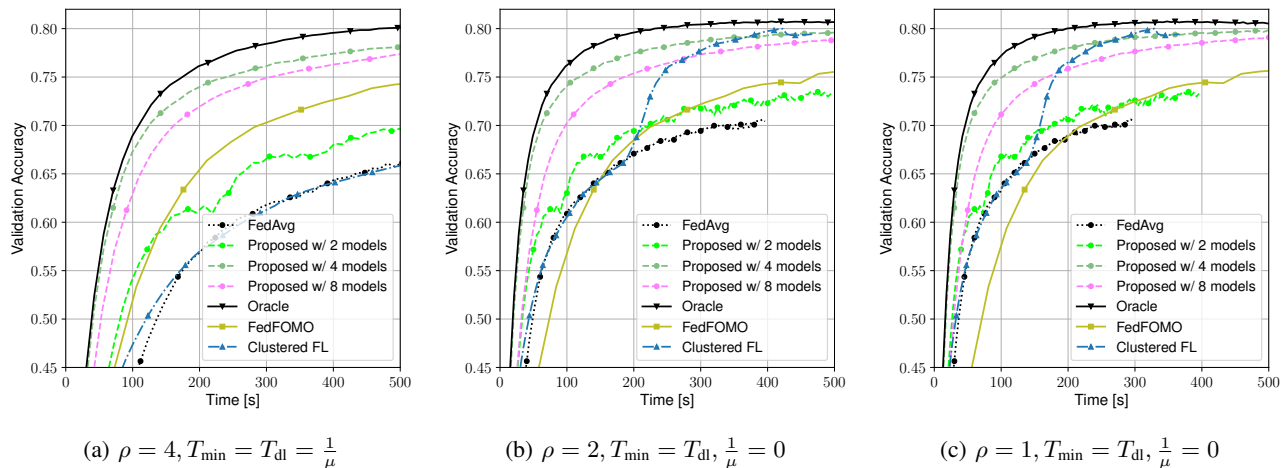


Fig. 6: Evolution of the average validation accuracy against time normalized w.r.t.  $T_{dl}$  for the three different systems.

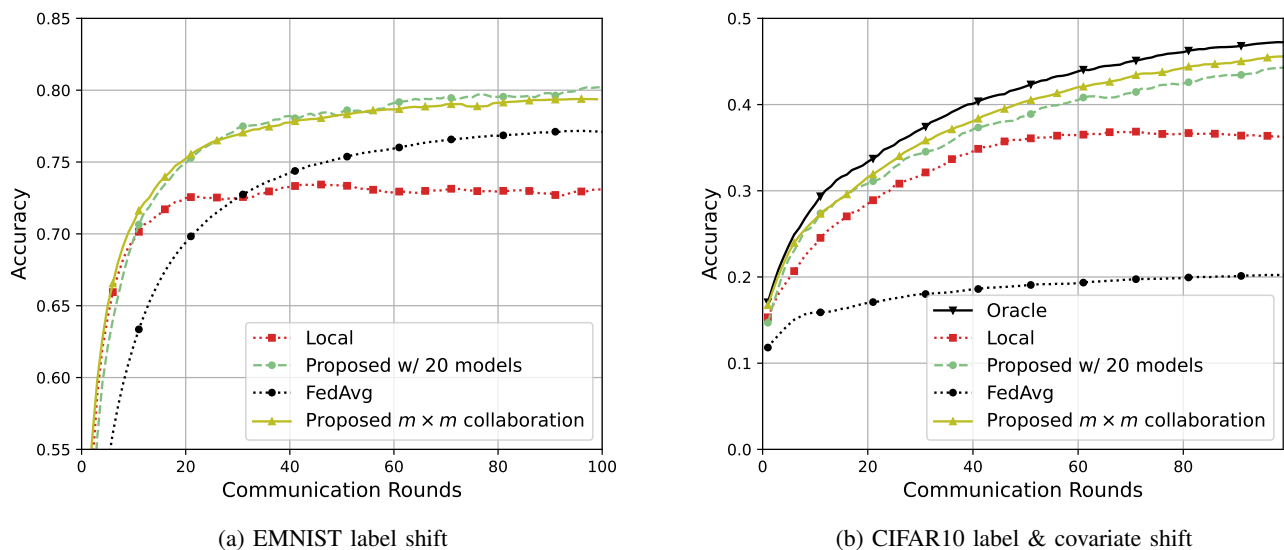


Fig. 7: Comparison between the proposed algorithm and the parallel user centric federated learning approach. The validation accuracy is averaged over 5 experiment runs.

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  $m_t$  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.

#### F. Variance Computation: Mini-batch Size

As mentioned in section IV-A, 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. 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 V-A, respectively. We vary the mini-batch sizes used to calculate the variances from 100  $\rightarrow$  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 (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. 8, achieving a high validation accuracy in the label shift setting while suffering in the co-variate shift experiment with a performance comparable to that of FedAvg attained in Fig. 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, we set the mini-batch size  $n_k = 100$  for the label shift experiment, and  $n_k = N/3$  for the other two EMNIST co-variate and CIFAR10 concept shift experiments, where  $N$  denotes the local data set size of each user.

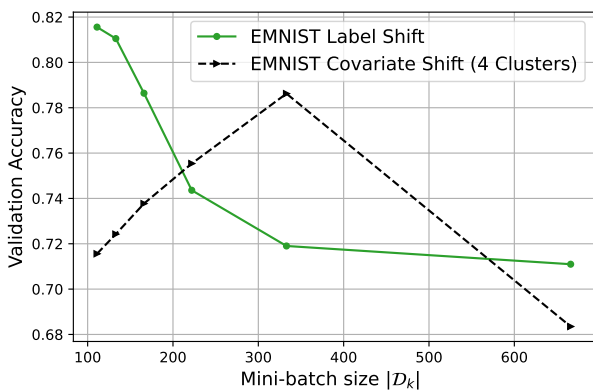


Fig. 8: Effect of the mini-batch sizes on the maximum validation accuracy attained: A proxy to the quality of the calculated collaboration coefficients

## VI. CONCLUSION

In this work, 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 prior to federated training and are designed based on an excess risk upper bound of the weighted aggregated loss minimizer. Additionally, in order 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. In order 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.

## VII. REPRODUCTION STATEMENT

The simulation code for this work is open-sourced and available at <https://github.com/Mestou/User-Centric-Federated-Learning-Trading-off-Wireless-Resources-for-Personalization>. All the hyperparameters necessary to reproduce the results can be found in the paper. We used publicly available standard datasets and included links to them in the manuscript.

## VIII. APPENDIX

### 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  $\hat{f}_{\bar{w}_i}$  as

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

where  $\lambda = \arg \min_{f \in \mathcal{F}} (E_{z \sim P_i} [\ell(f, z)] + E_{z \sim P_{\bar{w}_i}} [\ell(f, z)])$ .

We recognize the estimation error of  $\hat{f}_{\bar{w}_i}$  w.r.t to the measure  $P_{\bar{w}_i}$  that can be bounded following fairly standard approaches. In particular,

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

where

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

is the uniform deviation term and

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

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  $\frac{w_{i,j}}{n_i} B$  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

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

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

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

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

$$\begin{aligned} \text{Rad}(\mathcal{G}) &\leq \sqrt{\sum_{j=1}^m \frac{w_{i,j}^2}{n_j}} \\ &\times \sqrt{\frac{2\text{VCdim}(\mathcal{G}) \left( \log(e \sum_j n_j) + \log(\text{VCdim}(\mathcal{G})) \right)}{\sum_j n_j}}. \end{aligned}$$

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  $\sigma$  is also sub-Gaussian with parameter  $2\sigma$ , we can decompose the excess risk as

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

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

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

Plugging it back into the previous expression and minimizing with respect to  $\beta$  we obtain

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

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,  $\sigma = B/2$ .

### REFERENCES

- [1] J. H. Abawajy and M. M. Hassan, "Federated internet of things and cloud computing pervasive patient health monitoring system," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 48–53, 2017.
- [2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.
- [3] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [4] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.
- [5] M. Mestoukirdi, M. Zecchin, D. Gesbert, Q. Li, and N. 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), Dec. 2021.
- [6] C. Briggs, Z. Fan, and P. 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)*, pp. 1–9, IEEE, 2020.
- [7] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," *arXiv preprint arXiv:2012.08565*, 2020.
- [8] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2021.
- [9] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal, "Federated multi-task learning under a mixture of distributions," *International Workshop on Federated Learning for User Privacy and Data Confidentiality in conjunction with ICML 2021 (FL-ICML'21)*, 2021.
- [10] M. Reisser, C. Louizos, E. Gavves, and M. Welling, "Federated mixture of experts," *arXiv preprint arXiv:2107.06724*, 2021.
- [11] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous networks," *CoRR*, vol. abs/1812.06127, 2018.
- [12] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: stochastic controlled averaging for on-device federated learning," *CoRR*, vol. abs/1910.06378, 2019.
- [13] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proceedings of the 38th International Conference on Machine Learning (M. Meila and T. Zhang, eds.)*, vol. 139 of *Proceedings of Machine Learning Research*, pp. 6357–6368, PMLR, 18–24 Jul 2021.
- [14] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," *CoRR*, vol. abs/2006.08848, 2020.
- [15] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.
- [16] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint arXiv:0902.3430*, 2009.
- [17] C. Shui, Q. Chen, J. Wen, F. Zhou, C. Gagné, and B. Wang, "Beyond h-divergence: Domain adaptation theory with jensen-shannon divergence," 2020.
- [18] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, IEEE, 2017.
- [19] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [20] G. Cloud, "Stack-overflow dataset," 2017. Accessed on May 30, 2023.
- [21] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *arXiv preprint arXiv:2007.07481*, 2020.
- [22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [23] Tensorflow, "Basic text classification," 2023. Accessed on August 30, 2023.