Beyond Random Selection: A Perspective from Model Inversion in Personalized Federated Learning

Zichen Ma (✉)[1,2][0000−0002−8235−4720], Yu Lu[1,2][0000−0001−6971−3680], Wenye Li[1,3][0000−0002−5679−9670], and Shuguang Cui[1,3][0000−0003−2608−775X]

[1] The Chinese University of Hong Kong, Shenzhen, China
{zichenma1,yulu1}@link.cuhk.edu.cn;{wyli,shuguangcui}@cuhk.edu.cn
[2] JD AI Research, Beijing, China
[3] Shenzhen Research Institute of Big Data, Shenzhen, China

Abstract. With increasing concern for privacy issues in data, federated learning has emerged as one of the most prevalent approaches to collaboratively train statistical models without disclosing raw data. However, heterogeneity among clients in federated learning hinders optimization convergence and generalization performance. For example, clients usually differ in data distributions, network conditions, input/output dimensions, and model architectures, leading to the misalignment of clients' participation in training and degrading the model performance. In this work, we propose PFedRe, a personalized approach that introduces individual *relevance*, measured by Wasserstein distances among dummy datasets, into client selection in federated learning. The server generates dummy datasets from the inversion of local model updates, identifies clients with large distribution divergences, and aggregates updates from high relevant clients. Theoretically, we perform a convergence analysis of PFedRe and quantify how selection affects the convergence rate. We empirically demonstrate the efficacy of our framework on a variety of non-IID datasets. The results show that PFedRe outperforms other client selection baselines in the context of heterogeneous settings.

Keywords: Federated learning · Client selection · Personalization.

1 Introduction

The ever-growing attention to data privacy has propelled the rise of federated learning (FL), a privacy-preserving distributed machine learning paradigm on decentralized data [24]. A typical FL system consists of a central server and multiple decentralized clients (e.g., devices or data silos). The training of an FL system is typically an iterative process, which has two steps: (i) each local client is synchronized by the global model and trained using its local data; (ii) the server updates the global model by aggregating the local models.

However, as the number of clients and the complexity of the models grow, new challenges emerge concerning heterogeneity among clients [16]. For example, statistical heterogeneity in that data are not independent and identically

distributed (IID) hinders the convergence of the model and is detrimental to its performance. Thus, methods to overcome the adverse effects of heterogeneity are proposed, including regularization [20, 17], clustering [2, 10], and personalization [6, 30]. Despite these advances, client selection is a critical yet under-investigated topic.

In a cross-device FL training phase, it is plausible that not all of the client contributes to the learning objective [33]. Aggregating local updates from irrelevant clients to update the global model might degrade the system's performance. Moreover, McMahan et al. [24] show that only a fraction of clients should be selected by the server in each round, as adding more clients would diminish returns beyond a certain point. Hence, effective client selection schemes for heterogeneous FL are highly desired to achieve satisfactory model performances.

Thus far, some efforts have been devoted to selecting clients to alleviate heterogeneous issues and improve model performances, roughly grouped into two categories: (i) naive approaches to client selection identify and exclude irrelevant local model updates under the assumption that they are geometrically far from relevant ones [36, 13]; (ii) another line of work assumes the server maintains a public validation dataset and evaluates local model updates using this dataset. Underperforming clients are identified as irrelevant and excluded from aggregation. [34, 35].

Nevertheless, most of the existing client selection schemes have some limitations: (i) keeping a public validation dataset in the server and evaluating local updates on it disobeys the privacy principle of FL to some degree and might be impractical in real-world applications; (ii) current approaches are limited to the empirical demonstration without a rigorous analysis of how selection affects convergence speed.

Against this background, we propose a simple yet efficient personalized technique with client selection in heterogeneous settings. Clients with high relevance, measured by Wasserstein distances among dummy datasets, will be involved in the aggregation on the server, which boosts the system's efficiency.

Contributions of the paper are summarized as follows. First, we provide unique insights into client selection strategies to identify irrelevant clients. Specifically, the server derives dummy datasets from the inversion of local updates, excludes clients with large Wasserstein distances (large distribution divergences) among dummy datasets, and aggregates updates from high relevant clients. The proposed scheme has a crucial advantage: it uses dummy datasets from the inversion of local updates. Thus, there is no need for the server to keep a pubic validation dataset and the algorithm ensures the aggregation only involves highly relevant clients.

Second, we introduce a notion of individual relevance into FL, measured by Wasserstein distance among dummy datasets. As a motivating example, we examine two algorithms' (FedAvg [24] and FedProx [20]) performances with/without irrelevant clients on the MNIST dataset [19] in Figure 1. The objective is to classify odd labeled digits, i.e., {1, 3, 5, 7, 9}. For the case with irrelevant clients, odd labeled data are distributed to six clients, even labeled data are assigned

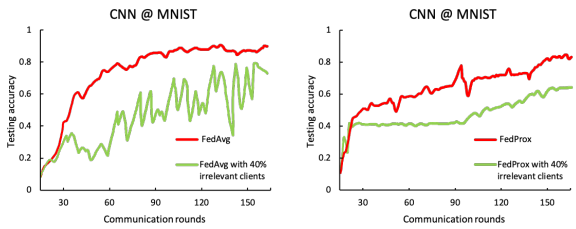

Fig. 1. Impacts of irrelevant clients in FL

to four clients, and even labels on four clients are randomly flipped to one of the odd labels. It is evident from the figure that irrelevant clients annihilate the stability of the training and incur lower accuracy, demonstrating the need to identify and exclude them from the system.

Finally, we explore the influences of client selection on the convergence of PFedRe. Theoretically, we show that, under some mild conditions, PFedRe will converge to an optimal solution for strongly convex function in non-IID settings. We illustrate that PFedRe can promote efficacy through extensive empirical evaluations while achieving superior prediction accuracy relative to recent state-of-the-art client selection algorithms.

2 Related Work

Client Selection in FL Existing work in client selection focuses on (i) detecting and excluding irrelevant clients that are geometrically far from relevant ones. Blanchard et al. [1] explore the problem by choosing the local updates with the smallest distance from others and aggregating them to update the global model. Later, Trimmed Mean and Median [36] removes local updates with the largest and smallest F, and take the remaining mean and median as the aggregated model. In [4, 32], authors alleviate the client selection issue while preserving efficient communication and boosting the convergence rate. However, some recently proposed work shows that irrelevant clients may be geometrically close to relevant ones [9, 28]; (ii) another line of research needs to centralize a public validation dataset on the server and use it to evaluate local model updates in terms of test accuracy or loss. The error rate-based method [9] rejects local model updates that significantly negatively impact the global model's accuracy. Zeno [34, 35] uses the loss decrease on the validation dataset to rank the model's relevance. Nevertheless, these schemes may violate the privacy-preserving principle of FL and may be challenging to implement in practice.

Recently, some work combines the two schemes and proposes hybrid client selection mechanisms. FLTrust [3] adopts a bootstrap on the server's validation dataset and uses the cosine similarity between the local and trained bootstrap models to rank the relevance. Later, DiverseFL [27] introduces a bootstrap method for each client using partial local data and compares this model with its

updates to determine the selection. However, these approaches may also inherit limitations of two ways.

Personalized FL Given the variability of data in FL, personalization is an approach used to improve accuracy, and numerous work has been proposed along this line. Particularly, Smith et al. [29] explore personalized FL via a primal-dual multi-task learning framework. As summarized in [6, 31, 22], the subsequent work has explored personalized FL through local customization [8, 15, 23], where models are built by customizing a well-trained global model. There are several ways to achieve personalization: (i) mixture of the global model and local models combines the global model with the clients' latent local models [14, 6, 23]; (ii) meta-learning approaches build an initial meta-model that can be updated effectively using Hessian or approximations of it, and the personalized models are learned on local data samples [8, 7]; (iii) local fine-tuning methods customize the global model using local datasets to learn personalized models on each client [23, 21].

3 Personalized Federated Learning with Relevance (PFedRe)

To explore client selection in personalized FL, we first formally define the personalized FL objective and introduce the system's workflow (Section 3.1). We then present PFedRe, a personalized algorithm that selects highly relevant clients to participate in training, and our proposed notion of individual relevance (Section 3.2). Finally, in Section 3.3, we analyze the influences of selection behaviors on training convergence.

3.1 Preliminaries and Problem Formulation

Notations Suppose there are M clients and a server in the system and denote by $X_k = \{x_{k,1}, x_{k,2}, ..., x_{k,m_k}\}$ the local data samples in the k-th client, where $x_{k,l}$ is the l-th sample and $l = 1, 2, ..., m_k$. Let $X = \cup_k X_k$ be the set of data among all clients, ω correspond to the global model, $\beta = (\beta_1, \beta_2, ..., \beta_M)$ with β_k being the personalized local models on the k-th client, F_k be the local objective function on the k-th client, and E be the local epochs on clients, respectively. We denote $m = \sum_{k=1}^{M} m_k$ as the total number of samples.

In personalized FL, clients communicate with the server to solve the following problem:

$$\min_{\omega,\beta} F(\omega, \beta) = \frac{1}{m} \sum_{k=1}^{M} \sum_{l=1}^{m_k} f(\omega, \beta_k; x_{k,l}) = \sum_{k=1}^{M} \frac{m_k}{m} F_k(\omega, \beta_k) \tag{1}$$

to find the global model ω and personalized model β. $f(\omega, \beta_k; x_{k,l})$ is the composite loss function for sample $x_{k,l}$ and model ω, β_k. Generally, in clients, Equation (1) is optimized w.r.t. ω and β by stochastic gradient descent (SGD).

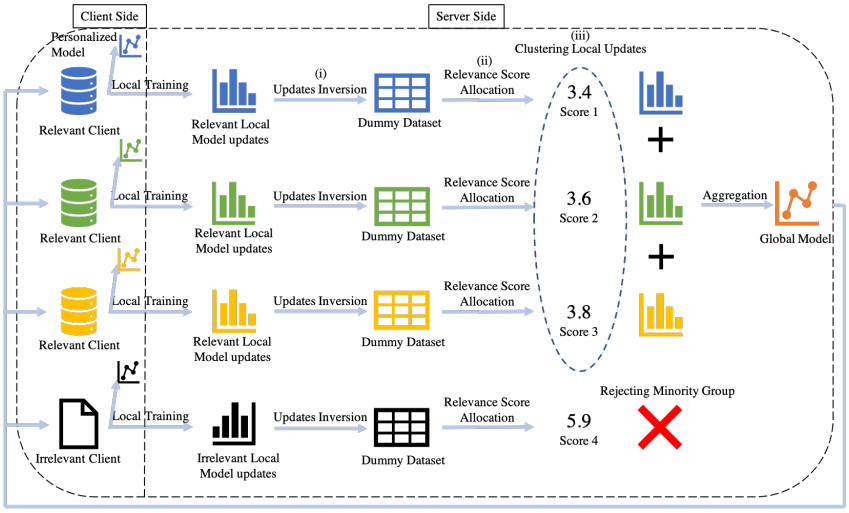

Fig. 2. Workflow of PFedRe

The communications in personalized FL only involve ω, while the personalized models β are stored locally and optimized without being sent to the server. Suppose the k-th client optimizes $F_k(\cdot)$ at most T iterations. After a client receives the global model at the beginning of the τ-th round ($0 \leq \tau < T$), it updates the received model using $\omega_{\tau+1}^k = \omega_\tau^k - \eta_\tau \nabla_{\omega_k} F_k(\omega_\tau^k, \beta_\tau^k)$, and the personalized model is updated by $\beta_{\tau+1}^k = \beta_\tau^k - \delta_\tau \nabla_{\beta_k} F_k(\omega_\tau^k, \beta_\tau^k)$, where η_τ and δ_τ are the learning rates.

After optimizing the personalized model β_k, each available client uploads $\omega_{\tau+1}^k$ every E epochs. The server aggregates the received models by

$$\omega_{\tau+1}^G = \sum_{k=1}^{M} \frac{m_k}{m} \omega_{\tau+1}^k. \tag{2}$$

Personalized FL updates the global model with Equation (2).

However, the server can only randomly select clients to participate in training due to the inaccessibility of clients' local training data and the uninspectable local training processes. As shown in Figure 1, aggregating irrelevant clients' updates, in this case, hampers the stability and performance of the system. Hence, we introduce a client selection mechanism that facilitates the server to prune irrelevant clients.

Figure 2 elucidates the workflow of the proposed framework. It takes a different approach with three key differences compared with traditional methods. (i) local updates inversion: the received local model updates are inverted to generate corresponding dummy datasets on the server; (ii) relevance score allocation: Wasserstein distances among dummy datasets are calculated and recorded as the relevance score; (iii) relevant client clustering: local updates are clustered into

two groups based on their relevance scores, and updates in the majority group are aggregated on the server.

3.2 PFedRe: Algorithm

In PFedRe, the k-th client performs E epochs of the local model updates via mini-batch SGD with a size of B. Then, it submits local model update $\omega_{\tau+1}^k$ in the τ-th round. The server works with the updates it receives from the clients. It first inverts local updates to generate dummy datasets using

$$x'^*_k = \arg\min_{x'_k} ||\frac{\partial F_k((x'_k, y'_k); \omega_\tau^k)}{\partial \omega_\tau^k} - \frac{a_\tau(\omega_{\tau+1}^k - \omega_\tau^k)}{Em_k/B}||_2^2, \tag{3}$$

where (x'_k, y'_k) are the dummy data to be optimized, and $\omega_\tau^k = \omega_\tau^G$ is the current global model. PFedRe performs inversion by matching the dummy gradient with the equivalent gradient $\frac{(\omega_{\tau+1}^k - \omega_\tau^k)}{Em_k/B}$, where the term starts to cancel out, i.e., $\frac{(\omega_{\tau+1}^k - \omega_\tau^k)}{Em_k/B} \to 0$, as the global model converges. To signify the differences among gradients such that the server can identify the differences among clients' data distributions, PFedRe adds a scale factor a_τ, i.e., a to the power of τ, where a is a hyperparameter. The optimum, x'^*_k, substracts the initialization of the dummy data, respectively.

After generating dummy datasets, the server employs the Wasserstein distance metric to derive the relevance scores and the distribution divergences among dummy datasets. The divergence $\mathcal{D}_W$ between (x'_k, y'_k) and (x'_l, y'_l) is given by

$$\mathcal{D}_W(x'_k, x'_l) = \sum_{i=1}^{p}\sum_{j=1}^{q} \text{Wasserstein}[x'^{i,j}_k, x'^{i,j}_l], \tag{4}$$

where $x'^{i,j}_k$ is the vector composed of the j-th features of samples with label i, p and q are the numbers of labels and features of dummy datasets.

It is natural that the dummy datasets derived from irrelevant clients' updates have more considerable distribution divergences than relevant ones. However, this may not hold when statistical heterogeneity exists, i.e., data among clients are non-IID. Thus, instead of simply removing local updates with more significant distribution divergences, PFedRe collects those updates that have moderate distribution divergences.

Denote by $\mathcal{H} = \{\mathcal{D}_W(x'_k, x'_l) | l = 1, ..., M\}$ the set of k-th client's Wasserstein distances with all clients. More formally, individual relevance can be defined as

Definition 1 *Let $r_{\tau+1}^k$ be the relevance score of model $\omega_{\tau+1}^k$. For two models $\omega_{\tau+1}^k$, and $\omega_{\tau+1}^n$ in an FL system, we say $\omega_{\tau+1}^k$ is more relevant than $\omega_{\tau+1}^n$ if $r_{\tau+1}^k < r_{\tau+1}^n$, where*

$$r_{\tau+1}^k = \sum_{\mathcal{H}} |\mathcal{D}_W(x'_k, x'_l)|. \tag{5}$$

Algorithm 1 PFedRe. M clients are indexed by k; a_τ represents the scaling factor; η and δ denote the learning rates; γ is the decay factor; T is the maximal number of communication rounds, and B denotes mini-batch size.

Server executes:

initialize ω^0, β_k^0, and dummy datasets x'. $\mathcal{H} \leftarrow \emptyset$, $\mathcal{R} \leftarrow \emptyset$.
for $\tau = 0, 1, ..., T-1$ **do**
  for each client $k \in \{1, 2, ..., M\}$ **in parallel do**
    $\omega_{\tau+1}^k \leftarrow$ ClientUpdate(k, ω_τ^G)
    $x'^*_k = \arg\min_{x'_k} ||\frac{\partial F_k((x'_k, y'_k); \omega_\tau^G)}{\partial \omega_\tau^G} - \frac{a_\tau(\omega_{\tau+1}^k - \omega_\tau^G)}{Em_k/B}||_2^2$
    $x'_k = x'^*_k - x'$
  end for
  for each client $k \in \{1, 2, ..., M\}$ **do**
    $\mathcal{H} \leftarrow \emptyset$
    for $l \in \{1, 2, ..., M\} \backslash \{k\}$ **do**
      $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{D}_W(x'_k, x'_l)\}$
    end for
    $r_{\tau+1}^k = \sum_{\mathcal{H}} |\mathcal{D}(x'_k, x'_l)|$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{r_{\tau+1}^k\}$
  end for
  $\mathcal{R} \leftarrow$ **2-Median**$(\mathcal{R})$
  $\Lambda \leftarrow \{k | r_{\tau+1}^k \in \mathcal{R}\}$
  $\omega_{\tau+1}^G = \sum_{k \in \Lambda} \frac{m_k}{\sum_{k \in \Lambda} m_k} \omega_{\tau+1}^k$
  return $\omega_{\tau+1}^G$ to participants.
end for

ClientUpdate(k, ω_τ^G)**:**

$B \leftarrow$ split local data into batches.
for $i = 0, ..., E-1$ **do**
  for batch $\xi \in B$ **do**
    $\omega_{\tau+i+1}^k = \omega_{\tau+i}^k - \frac{\eta}{1+\gamma\tau} \frac{\partial F_k(\xi; \omega_{\tau+i}^k)}{\partial \omega_{\tau+i}^k}$
    $\beta_{\tau+i+1}^k = \beta_{\tau+i}^k - \frac{\delta}{1+\gamma\tau} \frac{\partial F_k(\xi; \beta_{\tau+i}^k)}{\partial \beta_{\tau+i}^k}$
  end for
end for
return $\omega_{\tau+E}^k$ to the server.

Let $\mathcal{R} = \{r_{\tau+1}^k | k = 1, ..., M\}$. PFedRe selects the majority group of $\mathcal{R}$ using the 2-Median clustering. Updates from the majority group are aggregated in the server. The details of the algorithm are summarized in Algorithm 1.

The inherent benefits of the proposed selection scheme are that (i) the individual relevance of a client is not the same across communication rounds, i.e., the value of a client's relevance score changes according to the state of the system that varies across rounds. Further, as the global model converges to the

optimum, value of individual relevance also converges. Thus, the selection mechanism adapts to the dynamics of the heterogeneous settings that change over time; (ii) the framework is highly modular and flexible, i.e., we can readily use prior art developed for FL along with the client selection add-on, where the new methods still inherit the convergence benefits, if any.

3.3 PFedRe: Theoretical Analysis

In this section, we analyze the convergence behavior of PFedRe as described in Algorithm 1. We show that the proposed client selection scheme benefits to the convergence rate, albeit at the risk of incorporating a non-vanishing gap between the global optimum $\omega^{G*} = \arg\min_{\omega} F_k(\omega, \beta)$ and personalized optimum $\beta_k^* = \arg\min_{\omega,\beta} F_k(\omega, \beta)$.

Assumption 1 *(L-smoothness) F_k is L-smooth with constant $L > 0$ for $k = 1, 2, ..., M$, i.e. for all v, w,*

$$||\nabla F_k(v) - \nabla F_k(w)|| \leq L||v - w||.$$

Assumption 2 *(μ-strongly convexity) F_k is μ-strongly convex with constant $\mu > 0$ for $k = 1, 2, ..., M$, i.e. for all v, w,*

$$F_k(w) - F_k(v) - \nabla F_k(v)||w - v|| \geq \frac{\mu}{2}||w - v||^2.$$

Assumption 3 *(Unbiased gradient and bounded gradient discrepancy) For the mini-batch ξ uniformly sampled at random from B, the resulting stochastic gradient is unbiased, i.e.,*

$$\mathbb{E}[g_k(\omega_\tau^k, \xi)] = \nabla_{\omega_\tau^k} F_k(\omega_\tau^k). \tag{6}$$

Also, the discrepancy of model gradients is bounded by

$$\mathbb{E}||g_k(\omega_\tau^k, \xi) - \nabla_{\omega_\tau^k} F_k(\omega_\tau^k)||^2 \leq \chi^2, \tag{7}$$

where χ is a scalar.

Assumption 4 *(Bounded model discrepancy) Denote by $\beta_k^* = \arg\min_{\omega,\beta} F(\omega, \beta)$ the optimal model in the k-th client, and ω^0 the initialization of the global model. For a given ratio $q \gg 1$, the discrepancy between ω^0 and ω^{G*} is sufficiently larger than the discrepancy between β_k^* and ω^{G*}, i.e. $||\omega^0 - \omega^{G*}|| > q||\beta_k^* - \omega^{G*}||$.*

Two metrics are introduced, i.e., the personalized-global objective gap, and the selection skew, to help the convergence analysis.

Definition 2 *(Personalized-global objective gap) For the global optimum $\omega^{G*} = \arg\min_{\omega} F(\omega, \beta)$ and personalized optimum $\beta_k^* = \arg\min_{\omega,\beta} F(\omega, \beta)$, we define the personalized-global objective gap as*

$$\Gamma = F^* - \sum_{k=1}^{M} \frac{m_k}{m} F_k^* = \sum_{k=1}^{M} \frac{m_k}{m} (F_k(\omega^{G*}) - F_k(\beta_k^*)) \geq 0. \tag{8}$$

Γ is an inherent gap between the personalized and global objective functions and is independent of the selection strategy. A more significant Γ indicates higher data heterogeneity in the system. When $\Gamma = 0$, the personalized and global optimal values are the same, and no solution bias results from the selection.

The selection skew that captures the effect of the client selection strategy on the personalized-global objective gap can be defined as

Definition 3 *(Selection skew) Let a client selection strategy π be a function that maps the local updates to a selected set of clients $S(\pi, \omega_k)$, we define*

$$\rho(S(\pi,\omega_k),\beta_k) = \frac{\mathbb{E}_{S(\pi,\omega_k)}[\sum_{k\in S(\pi,\omega_k)} \frac{m_k}{m}(F_k(\omega_k) - F_k(\beta_k))]}{F_k(\omega_k) - \sum_{k=1}^{M} \frac{m_k}{m} F_k(\beta_k)} \geq 0, \tag{9}$$

where $\mathbb{E}_{S(\pi,\omega_k)}$ is the expectation over the randomness from the selection strategy π.

We further define two related metrics independent of the global updates and personalized model to obtain a conservative error bound, where

$$\overline{\rho} = \min_{\omega,\beta_k} \rho(S(\pi,\omega_k),\beta_k) \tag{10}$$

and

$$\widetilde{\rho} = \max_{\omega} \rho(S(\pi,\omega_k),\beta_k^*). \tag{11}$$

Equation (9) formulates the skew of a selection π. $\rho(S(\pi,\omega_k),\beta_k)$ is a function of versions of the global model's updates ω_k and personalized model β_k. According to Equations (10) and (11), $\overline{\rho} \leq \widetilde{\rho}$ for a client selection strategy π.

For the client selection strategy π_{random}, we have $\rho(S(\pi_{random},\omega_k),\beta_k) = 1$ for all ω_k and β_k since the numerator and denominator of Equation (9) become equal, and $\overline{\rho} = \widetilde{\rho} = 1$. For the proposed client selection strategy, π chooses clients' updates within the majority group of individual relevance, where $\overline{\rho}$ and $\widetilde{\rho}$ will be more significant. The following analysis shows that a more substantial $\overline{\rho}$ leads to a faster convergence with a potential error gap proportional to $(\frac{\widetilde{\rho}}{\overline{\rho}-1})$.

The convergence results for a selection strategy π with personalized-global objective gap Γ and selection skew $\widetilde{\rho}, \overline{\rho}$ is presented in Theorem 1.

Theorem 1. *Given Assumptions 1 to 4, for learning rate $\eta_\tau = \frac{1}{\mu(\tau+\frac{4L}{\mu})}$, and any client selection strategy π, the error after T rounds satisfies*

$$\mathbb{E}[F_k(\beta_T^k)] - F_k(\beta_k^*) \leq \frac{\mu}{\mu T + 4L}\left[\frac{4L(32E^2q^2 + \frac{\chi^2}{|S|})}{3\mu^2\overline{\rho}} + \frac{8L^2\Gamma}{\mu^2} + \frac{2L^2||\beta_0^k - \beta_k^*||^2}{\mu}\right] + Q(\overline{\rho},\widetilde{\rho}), \tag{12}$$

where $Q(\overline{\rho},\widetilde{\rho}) = \frac{8L\Gamma}{3\mu}(\frac{\widetilde{\rho}}{\overline{\rho}} - 1)$.

Theorem 1 provides the first convergence analysis of personalized FL with a biased client selection strategy π. It shows that a more significant selection skew $\overline{\rho}$ leads to faster convergence rate $O(\frac{1}{T\overline{\rho}})$. Since $\overline{\rho}$ is obtained by taking a minimum of the selection skew $\rho(S(\pi,\omega_k),\beta_k)$ over ω_k and β_k, the conservative bound on

Table 1. Statistics of datasets. The number of devices, samples, the mean and the standard deviation of data samples on each device are summarized.

Dataset	# Devices	# Samples	Mean	SD
CIFAR100	100	59,137	591	32
Shakespeare	132	359,016	2,719	204
Sentiment140	1,503	90,110	60	41
EMNIST	500	131,600	263	93

the actual convergence rate is obtained. If the selection skew $\rho(S(\pi, \omega_k), \beta_k)$ changes in training, the convergence rate can be improved by a more significant or at least a factor equal to $\overline{\rho}$.

The second term $Q(\overline{\rho}, \widetilde{\rho})$ in Equation (12) represents the solution bias, depending on the selection strategy, and $Q(\overline{\rho}, \widetilde{\rho}) \geq 0$ according to the definitions of $\overline{\rho}$ and $\widetilde{\rho}$, if the selection strategy is unbiased, e.g., random selection, $\overline{\rho} = \widetilde{\rho} = 1$, and $Q(\overline{\rho}, \widetilde{\rho}) = 0$. If $\overline{\rho} > 1$, the method has faster convergence by $\overline{\rho}$ and $Q(\overline{\rho}, \widetilde{\rho}) \neq 0$. The proof of Theorem 1 is presented in Appendix.

4 Experiments

We evaluate the efficacy of our approach PFedRe on multiple datasets by considering various heterogeneous settings.

4.1 Setup

Both convex and non-convex models are evaluated on several benchmark datasets. Specifically, we adopt the EMNIST [5] dataset with Resnet50 , CIFAR100 dataset [18] with VGG11 , Shakespeare dataset with an LSTM [24] to predict the next character, and Sentiment140 dataset [11] with an LSTM to classify sentiment. Statistics of datasets are summarized in Table 1.

To demonstrate the effectiveness of PFedRe, we experiment with both vanilla and irrelevant clients, where two ways are adopted to simulate irrelevant clients. The first method flips data samples' labels to other classes on clients, and the second assigns out-of-distribution samples to clients and labels them randomly. Furthermore, three baselines are compared with PFedRe: (i) standard federated averaging (FedAvg) algorithm [24]; (ii) Selecting clients using the Shapely-based valuation (S-FedAvg) method [25]; (iii) Dynamic filtering of clients according to their cumulative losses (AFL) [12].

All experiments are implemented using PyTorch [26] and run on a cluster where each node is equipped with 4 Tesla P40 GPUs and 64 Intel(R) Xeon(R) CPU E5-2683 v4 cores @ 2.10GHz. For reference, details of datasets partition and implementation settings are summarized in Appendix.

Fig. 3. Normalized relevance scores of clients on EMNIST dataset obtained by PFedRe after 1000 communication rounds. The irrelevant client percentage is 10%, 20%, 30%, and 40%. PFedRe identifies irrelevant clients (in blue) in training and excludes them from aggregation.

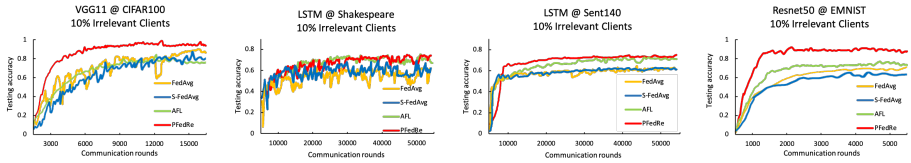

Fig. 4. The evolution of the testing accuracy is presented, where irrelevant clients have out-of-distribution samples. PFedRe outperforms other baselines in this case.

4.2 Detection of Irrelevant Clients

In this experiment, data samples of the EMNIST dataset are partitioned among 500 clients. To introduce the irrelevant clients, we flip data samples' labels to other classes on clients. The irrelevant client percentage in the system is 10%, 20%, 30%, and 40%. Figure 3 shows the normalized relevance score of clients learned using PFedRe after 1000 communication rounds. The yellow bars correspond to relevant clients, whereas the blue bars correspond to irrelevant clients. It is evident from the figure that the relevance scores of relevant clients are lower than that of irrelevant clients. Hence, using PFedRe, the server can differentiate between relevant and irrelevant clients. We further note that due to the dynamic nature of the generated dummy datasets in training, the magnitude of relevance keeps changing across communication rounds. However, the trend between relevant and relevant clients remains consistent.

4.3 Performance Comparison: Assigning Out-of-distribution Data Samples

This experiment shows the impact of irrelevant clients with out-of-distribution data samples on the system's performance. We use four datasets where 10% of clients are irrelevant. Out of-distribution data samples with random labels are assigned to irrelevant clients. Figure 4 shows that even in the presence of out-of-distribution samples at irrelevant clients, the performance of PFedRe is significantly better than that of other baselines, indicating the efficacy of the

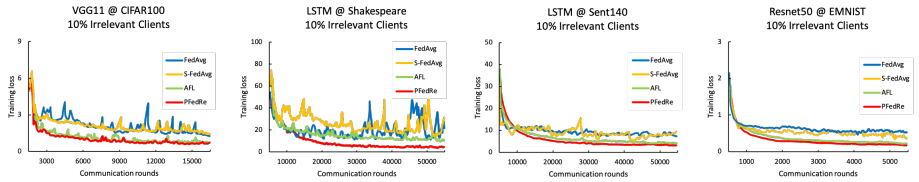

Fig. 5. The evolution of the training loss is presented, where the labels of samples are flipped to other classes on irrelevant clients. PFedRe exhibits better efficacy compared with baselines in this case.

proposed method. A close competitor to PFedRe is AFL, underlining the need for dynamic client filtering in heterogeneous settings.

4.4 Performance Comparison: Flipping Labels to Other Classes

In this experiment, we demonstrate the impact of the proposed client selection strategy on the performance (w.r.t. training loss) of algorithms. We implement PFedRe and the other three baselines independently on four datasets. For irrelevant clients, labels of samples are flipped to different classes on clients, and we set 10% of clients to be irrelevant in the system. Ideally, if PFedRe detects irrelevant clients correctly, the server would aggregate updates derived only from relevant clients, leading to better performance (lower training loss). Figure 5 shows that the system trained using PFedRe outperforms the models trained by other baselines. These results signify that identifying relevant clients and then aggregating updates from them is essential for building an efficient FL system.

4.5 Impact of Removing Clients with High/Low Relevance Score

This experiment shows that removing clients with high relevance scores deteriorates the system's performance, whereas removing clients who usually have low relevance scores helps improve it. We partition datasets to all clients and randomly flip 20% of samples' labels on 10% of clients. Then we run PFedRe for τ_0 rounds ($\tau_0 \ll T$) on datasets. After τ_0, the evolution of three testing accuracy is recorded, where PFedRe (i) keeps all clients in the system; (ii) removes clients determined as relevant more than 50% of rounds before τ_0 and keeps others in training; (iii) removes clients who are judged as irrelevant more than 50% of rounds before τ_0 and keeps others.

As shown in Figure 6, it is evident that removing clients with high relevance scores indeed affects the system's performance adversely. On the contrary, eliminating clients with low relevance scores improves its performance. We can consistently observe that removing as many as 10% of clients with low relevance scores will enhance the system's efficacy. Whereas removing clients with high relevance scores has a noticeable negative impact.

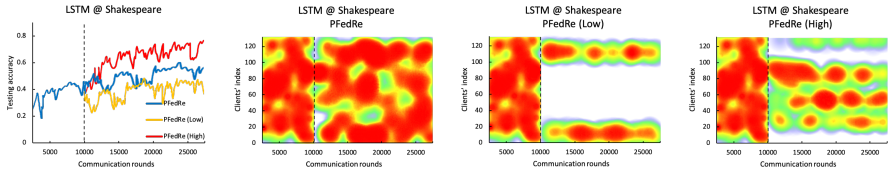

Fig. 6. The evolution of the testing accuracy is presented in the left figure. PFedRe trains models for $\tau_0 = 10000$ rounds. After τ_0, PFedRe (i) keeps all clients and continues training (blue curve); (ii) removes clients determined as relevant for more than 5000 rounds and keeps others (yellow curve); (iii) removes clients judged as irrelevant for more than 5000 rounds and keeps the remaining clients (red curve). The heatmaps record the clients' participation in three methods.

5 Conclusion and Future Work

This paper presents PFedRe, a novel personalized FL framework with client selection to mitigate heterogeneous issues in the system. By introducing the individual relevance into the algorithm, we extend the server to identify and exclude irrelevant clients via local updates' inversion, showing that dynamic client selection is instrumental in improving the system's performance. Both the analysis and empirical evaluations show the ability of PFedRe to achieve better performances in heterogeneous settings. In future work, we will explore potential competing constraints of client selection such as privacy and robustness to attacks and consider the applicability of PFedRe to other notions of the distributed system.

Acknowledgements The work was supported in part by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by the National Key R&D Program of China with grant No. 2018YFB1800800, by Shenzhen Outstanding Talents Training Fund, and by Guangdong Research Project No. 2017ZT07X152 and 2021A1515011825.

References

1. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. Advances in Neural Information Processing Systems **30** (2017)
2. Briggs, C., Fan, Z., Andras, P.: 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)
3. Cao, X., Fang, M., Liu, J., Gong, N.Z.: Fltrust: Byzantine-robust federated learning via trust bootstrapping. In: ISOC Network and Distributed System Security Symposium (NDSS) (2021)

4. Cho, Y.J., Gupta, S., Joshi, G., Yağan, O.: Bandit-based communication-efficient client selection strategies for federated learning. In: 2020 54th Asilomar Conference on Signals, Systems, and Computers. pp. 1066–1069. IEEE (2020)
5. Cohen, G., Afshar, S., Tapson, J., Van Schaik, A.: EMNIST: Extending MNIST to handwritten letters. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 2921–2926. IEEE (2017)
6. Deng, Y., Kamani, M.M., Mahdavi, M.: Adaptive personalized federated learning. arXiv preprint arXiv:2003.13461 (2020)
7. Fallah, A., Mokhtari, A., Ozdaglar, A.: On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In: International Conference on Artificial Intelligence and Statistics. pp. 1082–1092. PMLR (2020)
8. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. Advances in Neural Information Processing Systems **33**, 3557–3568 (2020)
9. Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to byzantine-robust federated learning. In: 29th USENIX Security Symposium (USENIX Security 20). pp. 1605–1622 (2020)
10. Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. Advances in Neural Information Processing Systems **33**, 19586–19597 (2020)
11. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N project report, Stanford **1**(12), 2009 (2009)
12. Goetz, J., Malik, K., Bui, D., Moon, S., Liu, H., Kumar, A.: Active federated learning. arXiv preprint arXiv:1909.12641 (2019)
13. Guerraoui, R., Rouault, S., et al.: The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning. pp. 3521–3530. PMLR (2018)
14. Hanzely, F., Richtárik, P.: Federated learning of a mixture of global and local models. arXiv preprint arXiv:2002.05516 (2020)
15. Jiang, Y., Konečnỳ, J., Rush, K., Kannan, S.: Improving federated learning personalization via model agnostic meta learning. arXiv preprint arXiv:1909.12488 (2019)
16. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)
17. Karimireddy, P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning. pp. 5132–5143. PMLR (2020)
18. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
20. Li, T., Sahu, A., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems **2**, 429–450 (2020)
21. Liang, P.P., Liu, T., Ziyin, L., Allen, N., Auerbach, R., Brent, D., Salakhutdinov, R., Morency, L.P.: Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523 (2020)

22. Ma, Z., Lu, Y., Li, W., Yi, J., Cui, S.: Pfedatt: Attention-based personalized federated learning on heterogeneous clients. In: Asian Conference on Machine Learning. pp. 1253–1268. PMLR (2021)
23. Mansour, Y., Mohri, M., Ro, J., Suresh, A.T.: Three approaches for personalization with applications to federated learning. arXiv preprint arXiv:2002.10619 (2020)
24. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
25. Nagalapatti, L., Narayanam, R.: Game of gradients: Mitigating irrelevant clients in federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 9046–9054 (2021)
26. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
27. Prakash, S., Avestimehr, A.S.: Mitigating byzantine attacks in federated learning. arXiv preprint arXiv:2010.07541 (2020)
28. Shejwalkar, V., Houmansadr, A.: Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In: NDSS (2021)
29. Smith, V., Chiang, C.K., Sanjabi, M., Talwalkar, A.: Federated multi-task learning. arXiv preprint arXiv:1705.10467 (2017)
30. T Dinh, C., Tran, N., Nguyen, T.: Personalized federated learning with moreau envelopes. Advances in Neural Information Processing Systems **33** (2020)
31. Tan, A.Z., Yu, H., Cui, L., Yang, Q.: Towards personalized federated learning. arXiv preprint arXiv:2103.00710 (2021)
32. Tang, M., Ning, X., Wang, Y., Wang, Y., Chen, Y.: Fedgp: Correlation-based active client selection for heterogeneous federated learning. arXiv preprint arXiv:2103.13822 (2021)
33. Wang, H., Kaplan, Z., Niu, D., Li, B.: Optimizing federated learning on non-iid data with reinforcement learning. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications. pp. 1698–1707. IEEE (2020)
34. Xie, C., Koyejo, S., Gupta, I.: Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In: International Conference on Machine Learning. pp. 6893–6901. PMLR (2019)
35. Xie, C., Koyejo, S., Gupta, I.: Zeno++: Robust fully asynchronous SGD. In: International Conference on Machine Learning. pp. 10495–10503. PMLR (2020)
36. Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: International Conference on Machine Learning. pp. 5650–5659. PMLR (2018)