FEDPROF: SELECTIVE FEDERATED LEARNING WITH REPRESENTATION PROFILING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Learning (FL) has shown great potential as a privacy-preserving solution to learning from decentralized data that are only accessible to end devices (i.e., clients). In many scenarios however, a large proportion of the clients are probably in possession of low-quality data that are biased, noisy or even irrelevant. As a result, they could significantly slow down the convergence of the global model we aim to build and also compromise its quality. In light of this, we propose FEDPROF, a novel algorithm for optimizing FL under such circumstances without breaching data privacy. The key of our approach is a data representation profiling and matching scheme that uses the global model to dynamically profile data representations and allows for low-cost, lightweight representation matching. Based on the scheme we adaptively score each client and adjust its participation probability so as to mitigate the impact of low-value clients on the training process. We have conducted extensive experiments on public datasets using various FL settings. The results show that FEDPROF effectively reduces the number of communication rounds and overall time (up to 4.5x speedup) for the global model to converge and provides accuracy gain.

1 INTRODUCTION

With the advances in Artificial Intelligence (AI), we are seeing a rapid growth in the number of AI-driven applications as well as the volume of data required to train them. However, a large proportion of data used for machine learning are often generated outside the data centers by distributed resources such as mobile phones and IoT (Internet of Things) devices. It is predicted that the data generated by IoT devices will account for 75% of the total in 2025 (Meulen, 2018). Under this circumstance, it will be very costly to gather all the data for centralized training. More importantly, moving the data out of their local devices (e.g., mobile phones) is now restricted by law in many countries, such as the General Data Protection Regulation (GDPR)[1] enforced in EU.

We face three main difficulties to learn from decentralized data: i) massive scale of end devices; ii) limited communication bandwidth at the network edge; and iii) uncertain data distribution and data quality. As an promising solution, Federated Learning (FL) (McMahan et al., 2017) is a framework for efficient distributed machine learning with privacy protection (i.e., no data exchange). A typical process of FL is organized in rounds where the devices (clients) download the global model from the server, perform local training on their data and then upload their updated local models to the server for aggregation. Compared to traditional distributed learning methods, FL is naturally more communication-efficient at scale (Konečnỳ et al., 2016; Wang et al., 2019). Nonetheless, several issues stand out.

1.1 MOTIVATION

1) FL is susceptible to biased and low-quality local data. Only a fraction of clients are selected for a round of FL (involving too many clients leads to diminishing gains (Li et al., 2019)). The standard FL algorithm (McMahan et al., 2017) selects clients randomly, which implies that every client (and its local data) is considered equally important. This makes the training process susceptible to local data with strong heterogeneity and of low quality (e.g., user-generated texts (Hard et al.,

[1] https://gdpr.eu/what-is-gdpr/

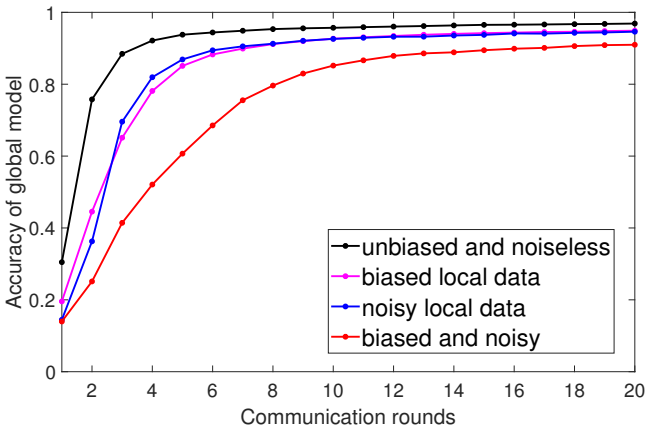


Figure 1: (Preliminary experiment) The global model's convergence under different data conditions. We ran the FL process with 100 clients to learn a CNN model on the MNIST dataset, which is partitioned and allocated to clients in four different ways where the data are 1) *original* (black line): noiseless and evenly distributed across the clients, 2) *biased* (magenta line): locally class-imbalanced, 3) *noisy* (blue line): blended with noise, or 4) *biased and noisy* (red line). The noise (if applied) covers 65% of the clients; the dominant class accounts for >50% of the samples for biased local data. The fraction of selected clients is 0.3 for each round.

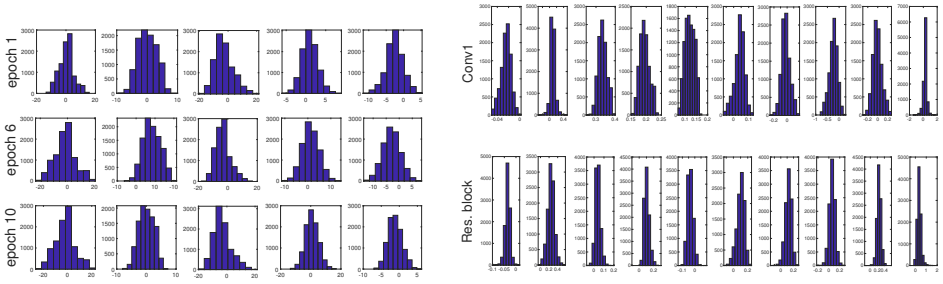
2018) and noisy photos). In some scenarios, local data may contain irrelevant or even adversarial samples (Bhagoji et al., 2019; Bagdasaryan et al., 2020) from malicious clients (Fang et al., 2020; Bagdasaryan et al., 2020; Tolpegin et al., 2020). Traditional solutions such as data augmentation (Yoo et al., 2020) and re-sampling (Lin et al., 2017) prove useful for centralised training but applying them to local datasets may introduce extra noise (Cui et al., 2019) and increase the risk of information leakage (Yu et al., 2021). Another naive solution is to directly exclude those low-value clients with low-quality data, which, however, is often impractical because i) the quality of the data depends on the learning task and is difficult to gauge; ii) some noisy or biased data could be useful to the training at early stages (Feelders, 1996); and iii) sometimes low-quality data are very common across the clients.

In Fig. 1 we demonstrate the impact of involving "low-value" clients by running FL over 100 clients to learn a CNN model on MNIST using the standard FEDAVG algorithm. From the traces we can see that training over clients with problematic or strongly biased data can compromise the efficiency and efficacy of FL, resulting in an inferior global model that takes more rounds to converge.

2) Learned representations can reflect data distribution and quality. Representation learning is vital to the performance of deep models because learned representations can capture the intrinsic structure of data and provide useful information for the downstream machine learning tasks (Bengio et al., 2013). In ML research, The value of representations lies in the fact that they characterize the domain and learning task and provide task-specific knowledge (Morcos et al., 2018; Kornblith et al., 2019). In the context of FL, the similarity of representations are used for refining the model update rules (Li et al., 2021; Feng & Yu, 2020), but the distributional difference between representations of heterogeneous data is not yet explored.

Our study is also motivated by a key observation that representations from neural networks tend to have Gaussian patterns. As a demonstration we trained two different models (LeNet-5 and ResNet-18) on two different datasets (MNIST and CIFAR-100) separately. Fig. 2a shows the neuron-wise distribution of representations extracted from the first dense layer (FC-1) of LeNet-5. Fig. 2b shows the distribution of fused representations (in a channel-wise manner) extracted from a plain convolution layer and a residual block of ResNet-18.

These observations motivate us to study the distributional property of data representations and use it as a means to differentiate clients' value.



(a) Representations from FC-1 of a LeNet-5 model after being trained for 1, 6 and 10 epochs on MNIST.

(b) Fused representations from a standard convolution layer (1st row) and a residual block (2nd row) of a ResNet-18 model trained for 100 epochs on CIFAR-100.

Figure 2: (Preliminary experiment) Demonstration of learned representations from a distributional perspective. The representations are generated by forward propagation in model evaluation. Each box corresponds to a randomly sampled element in the representation vector.

1.2 Contributions

Our contributions are summarized as the following:

- We first provide theoretical proof for the observation that data representations from neural networks tend to follow Gaussian distribution, based on which we propose a representation profiling and matching scheme for fast, low-cost comparison between different representation profiles.
- We present a novel FL algorithm FedProf that adaptively adjusts clients' participation probability based on representation profile dissimilarity.
- Results of extensive experiments show that FedProf reduces the number of communication rounds by up to 77%, shortens the overall training time (up to 4.5x speedup) while increasing the accuracy of the global model by up to 2.5%.

2 Related Work

Different from traditional distributed training methods (e.g., Alistarh et al. (2017); Wu et al. (2018); Zheng et al. (2017)), Federated Learning assumes strict constraints of data locality and limited communication capacity (Konečnỳ et al., 2016). Much effort has been made in optimizing FL and covers a variety of perspectives including communication (Konečnỳ et al., 2016; Niknam et al., 2020; Cui et al., 2021), update rules (Li et al., 2020; Wu et al., 2021a; Leroy et al., 2019; Luping et al., 2019), flexible aggregation (Wang et al., 2019; Wu et al., 2021b) and personalization (Fallah et al., 2020; Tan et al., 2021; Deng et al., 2020).

The control of device participation is imperative in cross-device FL scenarios (Kairouz et al., 2019; Yang et al., 2020) where the quality of local data is uncontrollable and the clients show varied value for the training task (Tuor et al., 2020). To this end, the selection of clients is pivotal to the convergence of FL over heterogeneous data and devices (Nishio & Yonetani, 2019; Wang et al., 2020b; Chai et al., 2019; Acar et al., 2020). Non-uniform client selection is widely adopted in existing studies (Li et al., 2020; Goetz et al., 2019; Cho et al., 2020; Li et al., 2019; Chen et al., 2020b; Wang et al., 2020a) and has been theoretically proven with convergence guarantees (Chen et al., 2020b; Li et al., 2019). Many approaches sample clients based on their performance (Nishio & Yonetani, 2019; Chai et al., 2020) or aim to jointly optimize the model accuracy and training time (Shi et al., 2020; Chen et al., 2020a; 2021). A popular strategy is to use loss as the information to guide client selection (Goetz et al., 2019; Lai et al., 2021; Sarkar et al., 2020). For example, AFL (Goetz et al., 2019) prioritizes the clients with high loss feedback on local data, but it is potentially susceptible to noisy and unexpected data that yield illusive loss values. Data representations are

useful in the context of FL for information exchange (Feng & Yu, 2020) or objective adaptation (Li et al., 2021), but the study on the distribution of data representations is lacking whilst its connection to clients' training value is hardly explored either.

3 Data Representation Profiling and Matching

In this paper, we consider a typical cross-device FL setting (Kairouz et al., 2019), in which multiple end devices collaboratively perform local training on their own datasets $D_i, i = 1, 2, ..., n$. The server owns a validation dataset D^* for model evaluation. Every dataset is only accessible to its owner.

Considering the distributional pattern of data representations (Fig. 2) and the role of the global model in FL, we propose to profile the representations of local data using the global model. In this section, we first provide theoretical proof to support our observation that representations from neural network models tend to follow Gaussian distribution. Then we present a novel scheme to profile data representations and define profile dissimilarity for fast and secure representation comparison.

3.1 Gaussian Distribution of Representations

We first make the following definition to facilitate our analysis.

Definition 1 (The Lyapunov's condition)**.** *A set of random variables $\{Z_1, Z_2, \ldots, Z_v\}$ satisfy the Lyapunov's condition if there exists a δ such that*

$$\lim_{v\to\infty} \frac{1}{s^{2+\delta}} \sum_{k=1}^{v} \mathrm{E}\left[|Z_k - \mu_k|^{2+\delta}\right] = 0, \tag{1}$$

where $\mu_k = \mathrm{E}[Z_k]$, $\sigma_k^2 = \mathrm{E}[(Z_k - \mu_k)^2]$ and $s = \sqrt{\sum_{k=1}^{v} \sigma_k^2}$.

The Lyapunov's condition can be intuitively explained as a limit on the overall variation (with $|Z_k - \mu_k|^{2+\delta}$ being the $(2+\delta)$-th moment of Z_k) of a set of random variables.

Now we present Proposition 1 and Proposition 2.

Proposition 1. *The representations from linear operators (e.g., a pre-activation dense layer or a plain convolutional layer) in a neural network tend to follow the Gaussian distribution if the layer's weighted inputs satisfy the Lyapunov's condition.*

Proposition 2. *The fused representations[2] from non-linear operators (e.g., a hidden layer of LSTM or a residual block of ResNet) in a neural network tend to follow the Gaussian distribution if the layer's output elements satisfy the Lyapunov's condition.*

The proofs of Propositions 1 and 2 are provided in Appendices A.1 and A.2, respectively. In practice, the Lyapunov's condition is typically met when the model is properly initialized and batch normalization is applied. Next, we discuss the proposed representation profiling and matching scheme.

3.2 Distributional Profiling and Matching

Based on the Gaussian pattern of representations, we compress the data representations statistically into a compact form called *representation profiles*. The profile produced by the global model w on a dataset D, denoted by $RP(w, D)$, has the following format:

$$RP(w, D) = \{\mathcal{N}(\mu_i, \sigma_i^2)\}_{i=1}^{q}, \tag{2}$$

where q is the profile length determined by the dimensionality of the representations. For example, q is equal to the number of kernels for channel-wise fused representations from a convolutional layer. The tuple (μ_i, σ_i^2) contains the mean and the variance of the i-th representation element.

Local representation profiles are generated by clients and sent to the server for comparison (the cost of transmission is negligible considering each profile is only $q \times 8$ bytes). Let RP_k denote the local

[2] *Fused* representations refer to the sum of elements in the original representations produced by a single layer (channel-wise for a residual block).

profile from client k, and RP^* denote the baseline profile (generated in model evaluation) on the server. The dissimilarity between RP_k and RP^*, denoted by $div(RP_k, RP^*)$, is defined as:

$$div(RP_k, RP^*) = \frac{1}{q}\sum_{i=1}^{q} \mathrm{KL}(\mathcal{N}_i^{(k)} || \mathcal{N}_i^*), \tag{3}$$

where $\mathrm{KL}(\cdot)$ denotes the Kullback–Leibler (KL) divergence. An advantage of our profiling scheme is that a much simplified KL divergence formula can be adopted because of the Gaussian distribution property (see Roberts & Penny (2002, Appendix B) for details), which yields:

$$\mathrm{KL}(\mathcal{N}_i^{(k)} || \mathcal{N}_i^*) = \log \frac{\sigma_i^*}{\sigma_i^{(k)}} + \frac{(\sigma_i^{(k)})^2 + (\mu_i^{(k)} - \mu_i^*)^2}{2(\sigma_i^*)^2}, \tag{4}$$

Eq. (4) computes the KL divergence without calculating any integral, which is computationally cost-efficient. Besides, the computation of profile dissimilarity can be performed under the Homomorphic Encryption for minimum knowledge disclosure (see Appendix D for details).

4 The Training Algorithm FedProf

Our research aims to optimize the global model over a large group of clients (datasets) of disparate training value. Given the client set $U(|U| = N)$, let D_k denote the local dataset on client k and D^* the validation set on the server, We formulate the optimization problem in (5) where the coefficient ρ_k differentiates the importance of the local objective functions $F_k(w)$ and depends on the data in D_k. Our global objective is in a sense similar to the agnostic learning scenario (Mohri et al., 2019) where a non-uniform mixture of local data distributions is implied.

$$\arg\min_{w} F(w) = \sum_{k=1}^{N} \rho_k F_k(w), \tag{5}$$

where w is the parameter set of the global model. The coefficients $\{\rho_k\}_{k=1}^N$ add up to 1. $F_k(w)$ is client k's local objective function of training based on the loss function $\ell(\cdot)$:

$$F_k(w) = \frac{1}{|D_k|} \sum_{(x_i, y_i) \in D_k} \ell(x_i, y_i; w), \tag{6}$$

Involving the "right" clients facilitates the convergence. With this motivation we score each client with λ_k each round based on the representation profile dissimilarity:

$$\lambda_k = \exp\big(- \alpha \cdot div(RP_k(v_k), RP^*(v_k)\big), \tag{7}$$

where v_k is the version of RP_k (i.e., the version of the global model that client k receives) and α is a preference factor deciding how biased the selection strategy needs to be towards the clients with small profile dissimilarity (i.e., higher training value). With $\alpha = 0$, our strategy is equivalent to random selection. The scores connect the representation profiling and matching scheme to the design of the selective client participation strategy adopted in our FL training algorithm FedProf, which is outlined in Algorithm 1 (see Appendix B for a detailed version with both client and server processes). The key steps of our algorithm are local representation profiling (line 5), baseline representation profiling (line 9) and client scoring (line 3).

The convergence rate of FL algorithms with opportunistic client selection (sampling) has been extensively studied in the literature (Li et al., 2019; Chen et al., 2020b; Wang et al., 2019). Inspired by these studies, we present Theorem 1 to guarantee the global model's convergence for our algorithm. Similar to Stich et al. (2018); Zhang et al. (2013); Li et al. (2019), we assume that $\{F_k\}_{k=1}^N$ are L-smooth and μ-strongly convex and that in expectation, the variance of local stochastic gradients are bounded by ϵ^2 and their squared norms are bounded by G^2. For each local update step t we define $S(t)$ as the set of selected clients in the associated round.

Theorem 1. *Using partial aggregation and a strategy π that selects clients by the probability distribution $\{q_k\}_{k=1}^N$, the global model $w(t)$ converges in expectation by having $q_k = \rho_k$, an aggregation interval $E \geq 1$ and a decreasing step size (learning rate) $\eta_t = \frac{2}{\mu(t+\gamma)}$.*

$$\mathrm{E}\big[F(w(t))\big] - F^* \leq \frac{L}{(\gamma + t)}\Big(\frac{2(\mathcal{B} + \mathcal{C})}{\mu^2} + \frac{\gamma + 1}{2}\Delta_1\Big), \tag{8}$$

Algorithm 1: the FEDPROF algorithm

Input: maximum number of rounds T_{max}, iterations per round E, fraction C

Initialize global model w and generate baseline profile RP^*
Collect initial representation profiles $\{RP_k\}_{k \in U}$ from all clients
for round $T \leftarrow 1$ to T_{max} **do**
  Update client scores $\{\lambda_k\}_{k \in U}$ and compute $\Lambda = \sum_{k \in U} \lambda_k$
  $S \leftarrow$ Choose $K = N \cdot C$ clients by probability distribution $\{\frac{\lambda_k}{\Lambda}\}_{k \in U}$
  for client k in S **in parallel do**
    $RP_k \leftarrow$ *updateProfile*$(k, w, T-1)$
    $w_k \leftarrow$ *localTraining*(k, w, E)
  end
  Collect local profiles from the clients in S
  Update w via model aggregation
  Evaluate w and update RP^*
end
return w

where $t \in T_A = \{nE | n = 1, 2, \ldots\}$, $\gamma = \max\{\frac{8L}{\mu}, E\} - 1$, $\mathcal{B} = \sum_{k=1}^{N} \rho_k^2 \epsilon_k^2 + 6L\Gamma + 8(E-1)^2 G^2$, $\mathcal{C} = \frac{4}{K} E^2 G^2$, $\Gamma = F^* - \sum_{k=1}^{N} \rho_k F_k^*$, $\Delta_1 = \mathrm{E}\|\bar{w}(1) - w^*\|^2$, $K = |S(t)| = N \cdot C$.

The proof of Theorem 1 is provided in Appendix C.

5 EXPERIMENTS

We conducted extensive experiments to evaluate FEDPROF under various FL settings. Apart from FEDAVG (McMahan et al., 2017), we also reproduced several state-of-the-art FL algorithms for comparison. For fair comparison, the algorithms are grouped by the aggregation method (i.e., full aggregation and partial aggregation) and configured following the hyper-parameter settings in their papers (if any). Table 1 summarizes these FL algorithms. Note that our algorithm can adapt to both aggregation methods.

Table 1: The implemented FL algorithms for comparison

Algorithm	Aggregation method	Rule of selection
FEDAVG (McMahan et al., 2017)	full aggregation	random selection
CFCFM (Wu et al., 2021b)	full aggregation	submission order
FEDAVG-RP (Li et al., 2019)	partial (Scheme II)	random selection
FEDPROX (Li et al., 2020)	partial aggregation	weighted random by data ratio
FEDADAM (Leroy et al., 2019)	partial with momentum	random selection
AFL (Goetz et al., 2019)	partial with momentum	local loss valuation
FEDPROF (ours)	full/partial aggregation	weighted random by score

5.1 EXPERIMENT SETUP

We built our simulated FL system and implemented the algorithms based on the Pytorch framework (Build 1.7.0). We first set up a Small-scale Task (S-Task) to learn a multi-layer feed-forward network model from decentralized sensor data for predicting carbon monoxide (CO) and nitrogen oxides (NOx) emissions using the *GasTurbine*[3] dataset. Next, we set up a Large-scale Task (L-Task) to train a CNN over a large population of user devices for image classification (*EMNIST*[4]). In both tasks, data sharing is not allowed between any parties. The data are non-IID across the end devices.[5]

[3]https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set

[4]https://www.nist.gov/itl/products-and-services/emnist-dataset. We use the digits subset of EMNIST.

[5]In the S-Task, local datasets are of different sizes that follow a Gaussian distribution. In the L-Task, local data are largely imbalanced where around 60% of the samples on each client have the same class label.

We introduce a diversity of noise into the local datasets on end devices to simulate the discrepancy in data quality. The S-Task is performed over 50 sensor clients and 50% of them produce noisy data (including 10% invalid). The population for L-Task contains 1000 end devices across which the data spread with strong class imbalance – roughly 60% of the samples on each device fall into the same class. 15% of the local datasets in the L-Task are irrelevant images whereas another 45% of them are low-quality images (blurred or affected by salt-and-pepper noise). Considering the population of the clients, the setting of the selection fraction C is based on the scale of the training participants suggested by Kairouz et al. (2019). For both tasks, the clients are heterogeneous in terms of both performance and communication bandwidth. More experimental settings are listed in Table 4 in Appendix E where the environment setup is also given in details.

5.2 Evaluation Results

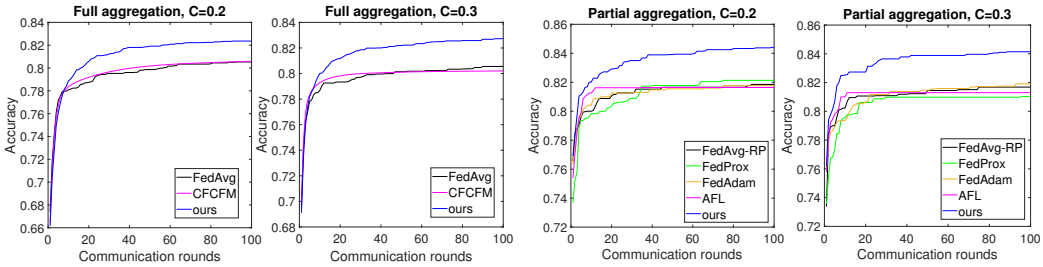
We evaluate the performance of our FedProf algorithm in terms of the efficacy (best accuracy achieved) and efficiency (costs for convergence) in establishing a global model for the two tasks. Tables 2 and 3 summarize the results. Figs. 3 and 4 plot the accuracy traces from the round-wise evaluations of the global model.

Table 2: The results of running the S-Task. The best accuracy is achieved by running for long enough. Other metrics are recorded upon reaching the target accuracy (80% for the S-Task).

Full aggregation								
	C=0.2				**C=0.3**			
	Best acc	For accuracy@0.8			Best acc	For accuracy@0.8		
		Rounds	Time(s)	E(Wh)		Rounds	Time(s)	E(Wh)
FedAvg	0.805	56	2869.66	2.87	0.806	52	3160.01	4.12
CFCFM	0.806	39	1230.81	1.61	0.802	42	1495.34	2.91
Ours	**0.824**	**16**	**803.74**	**0.80**	**0.827**	**12**	**701.18**	**0.94**
Partial aggregation								
	C=0.2				**C=0.3**			
	Best acc	For accuracy@0.8			Best acc	For accuracy@0.8		
		Rounds	Time(s)	E(Wh)		Rounds	Time(s)	E(Wh)
FedAvg-RP	0.819	13	735.48	0.71	0.817	8	466.90	0.64
FedProx	0.821	16	899.93	0.79	0.810	16	841.65	1.08
FedAdam	0.818	8	438.20	0.42	0.819	12	667.00	0.94
AFL	0.816	6	313.81	0.30	0.813	6	298.81	0.42
Ours	**0.844**	**5**	**283.76**	**0.27**	**0.841**	**4**	**235.22**	**0.35**

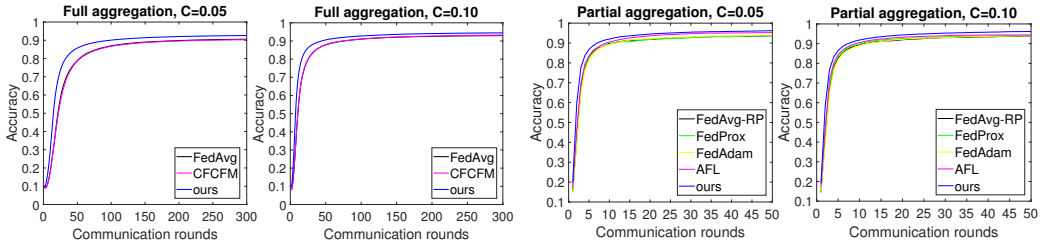
Table 3: The results of running the L-Task. The best accuracy is achieved by running for long enough. Other metrics are recorded upon reaching the target accuracy (90% for the L-Task).

Full aggregation								
	C=0.05				**C=0.1**			
	Best acc	For accuracy@0.9			Best acc	For accuracy@0.9		
		Rounds	Time(s)	E(Wh)		Rounds	Time(s)	E(Wh)
FedAvg	0.906	213	10407.30	60.01	0.929	76	3894.26	43.16
CFCFM	0.923	251	8645.17	61.37	0.932	75	2675.56	37.62
Ours	**0.926**	**98**	**4846.15**	**28.22**	**0.945**	**45**	**2295.03**	**25.98**
Partial aggregation								
	C=0.05				**C=0.1**			
	Best acc	For accuracy@0.9			Best acc	For accuracy@0.9		
		Rounds	Time(s)	E(Wh)		Rounds	Time(s)	E(Wh)
FedAvg-RP	0.937	12	572.90	3.29	0.938	12	603.94	6.57
FedProx	0.936	13	640.01	3.58	0.942	11	559.16	5.91
FedAdam	0.940	12	599.96	3.47	0.939	12	608.85	6.76
AFL	0.952	10	479.73	2.73	0.944	9	476.62	5.26
Ours	**0.962**	**8**	**383.94**	**2.26**	**0.962**	**8**	**413.16**	**4.64**



(a) (b)

Figure 3: The traces of evaluation accuracy of the global model through 100 rounds in the S-Task using (a) the full-aggregation method, and (b) the partial-aggregation method.



(a) (b)

Figure 4: The traces of evaluation accuracy of the global model in the L-Task using (a) the full-aggregation method (through 300 rounds), and (b) the partial aggregation method (through 50 rounds).

1) Convergence in different aggregation modes: Our results show a great difference in convergence rate under different aggregation modes. From Figs. 3 and 4, we observe that partial aggregation facilitates faster convergence of the global model than full aggregation, which is consistent with the observations made by Li et al. (2019). The advantage is especially obvious in the L-Task where partial aggregation requires much fewer communication rounds to reach the 90% accuracy. Our FEDPROF algorithm yields the fastest convergence in both groups of comparison for both tasks because selective participation benefits both aggregation methods.

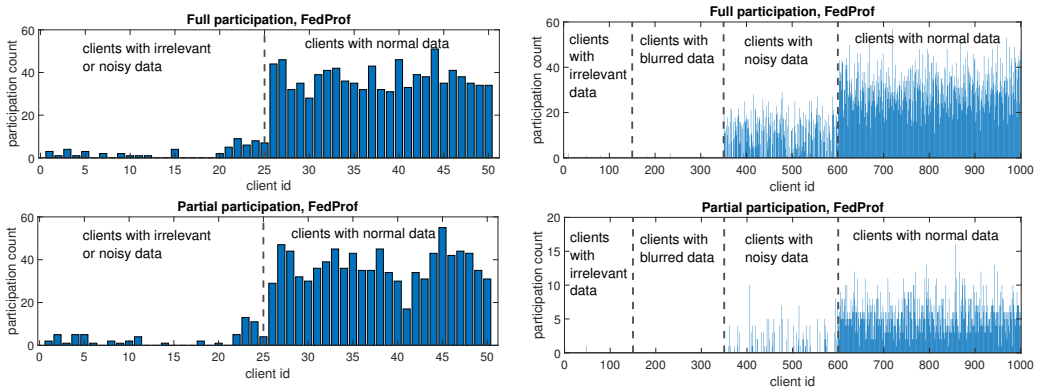
2) Best accuracy of the global model: Through the training process of FL, the global model is evaluated each round on the server. The best global model obtained thus far is stored on the server. As shown in the 2nd column of Tables 2 and 3, our FEDPROF algorithm improves the accuracy achieved by 2.5% when compared against the baselines (FEDAVG and FEDAVG-RP). The AFL algorithm uses a loss-oriented client selection strategy, which shows the closest performance to our algorithm in the L-Task but the worst accuracy in the S-Task.

3) Total communication rounds for convergence: The number of communication rounds required for reaching convergence is a key indicator to the efficiency of FL. In the S-Task, our algorithm takes less than half the communication rounds required by other algorithms in most cases. In the L-Task with C=0.05, our algorithm reaches 90% accuracy within 100 rounds whilst FEDAVG and CFCFM need more than 200. In this case, FEDPROF also achieves approximately 7% higher accuracy in 50 rounds. Partial aggregation turns out to be much more communication-efficient: FEDAVG-RP needs 8 rounds to reach the accuracy target for the S-Task and 12 rounds for the L-Task, whilst our protocol reduces the numbers to 5 and 8, respectively.

4) Total time needed for convergence: The overall time consumption is closely related to total communication rounds needed for convergence and the time cost for each round. Algorithms requiring more rounds to converge typically take longer to reach the accuracy target except the case of CFCFM, which priorities the clients that work faster. Using FEDAVG as the baseline, CFCFM accelerates the training process by 2.1x whilst our algorithm provides a 4.5x speedup for achieving the same target accuracy (S-Task, C=0.3). FEDPROF also has a clear advantage over FEDAVG-

RP, FEDPROX and FEDADAM in the partial aggregation group where it shows a 2.6x speedup over FEDAVG-RP in the S-Task with C=0.2.

5) Energy consumption of end devices: A main concern for the end devices, as the participants of FL, is their power usage (Watt hours). Full aggregation methods experience slower convergence and thus endure higher energy cost on the devices. For example, with a small selection fraction C=0.05 in the L-Task, FEDAVG and CFCFM consume over 60Wh to reach the target accuracy. In this case, our algorithm reduces the cost by more than a half (28.22Wh). With the partial aggregation mode, the reduction by our algorithm is up to 62% (S-Task, $C = 0.2$). Considering all the cases, FEDPROF achieves the target accuracy with the least energy cost, providing an reduction of 29% $\sim$ 53%.



(a) (b)

Figure 5: Total counts by client of participation (i.e., being selected) in (a) the S-Task and (b) the L-Task. For clarity, clients are indexed according to their local data quality.

6) Differentiated participation with FEDPROF: Fig. 5 reflects the preference of our selection strategy. In the S-Task we can observe that the clients with useless samples or noisy data get significantly less involved (<10 on average) in training. In the L-Task our algorithm also effectively limits (basically excludes) the clients who owns the image data of poor quality (i.e., irrelevant or severely blurred), whereas the clients with moderately noisy images are selected with reduced frequency as compared to those with normal data. A potential issue of having the preference towards some of the devices is about fairness. Nonetheless, one can apply our algorithm together with an incentive mechanism (e.g., Yu et al. (2020)) to address it.

6 CONCLUSION

Federated learning provides a privacy-preserving approach to decentralized training but is vulnerable to the heterogeneity and uncertain quality of on-device data. In this paper, we use a novel approach to address the issue without violating the data locality restriction. We first provide key insights for the distribution of data representations and then develop a dynamic data representation profiling and matching scheme. Based on the scheme we propose a selective FL training algorithm FEDPROF that adaptively adjusts clients' participation chance based on their profile dissimilarity. We have conducted extensive experiments on public datasets under various environment settings. Evaluation results show that our algorithm significantly improves the efficiency of FL and reduces the time and energy costs for the global model to converge.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 30:1709–1720, 2017.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948. PMLR, 2020.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pp. 634–643. PMLR, 2019.

Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

Aaron Carroll, Gernot Heiser, et al. An analysis of power consumption in a smartphone. In *USENIX annual technical conference*, volume 14, pp. 21–21. Boston, MA, 2010.

Zheng Chai, Hannan Fayyaz, Zeshan Fayyaz, Ali Anwar, Yi Zhou, Nathalie Baracaldo, Heiko Ludwig, and Yue Cheng. Towards taming the resource and data heterogeneity in federated learning. In *2019 {USENIX} Conference on Operational Machine Learning (OpML 19)*, pp. 19–21, 2019.

Zheng Chai, Ahsan Ali, Syed Zawad, Stacey Truex, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Heiko Ludwig, Feng Yan, and Yue Cheng. Tifl: A tier-based federated learning system. In *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, pp. 125–136, 2020.

Mingzhe Chen, H Vincent Poor, Walid Saad, and Shuguang Cui. Convergence time optimization for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 20 (4):2457–2471, 2020a.

Mingzhe Chen, Nir Shlezinger, H Vincent Poor, Yonina C Eldar, and Shuguang Cui. Communication-efficient federated learning. *Proceedings of the National Academy of Sciences*, 118(17), 2021.

Wenlin Chen, Samuel Horvath, and Peter Richtarik. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020b.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020.

Laizhong Cui, Xiaoxin Su, Yipeng Zhou, and Yi Pan. Slashing communication traffic in federated learning by transmitting clustered model updates. *IEEE Journal on Selected Areas in Communications*, 2021.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to byzantine-robust federated learning. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pp. 1605–1622, 2020.

AJ Feelders. Learning from biased data using mixture models. In *KDD*, pp. 102–107, 1996.

Siwei Feng and Han Yu. Multi-participant multi-class vertical federated learning. *arXiv preprint arXiv:2001.11154*, 2020.

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169–178, 2009.

Jack Goetz, Kshitiz Malik, Duc Bui, Seungwhan Moon, Honglei Liu, and Anuj Kumar. Active federated learning. *arXiv preprint arXiv:1909.12641*, 2019.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019.

Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Oort: Efficient federated learning via guided participant selection. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, pp. 19–35, 2021.

Don S Lemons. *An Introduction to Stochastic Processes in Physics*. Johns Hopkins University Press, 2003.

David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6341–6345. IEEE, 2019.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10713–10722, 2021.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *The 3rd MLSys Conference*, 2020.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. Clustering-based undersampling in class-imbalanced data. *Information Sciences*, 409:17–26, 2017.

WANG Luping, WANG Wei, and LI Bo. Cmfl: Mitigating communication overhead for federated learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 954–964. IEEE, 2019.

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

Robert van der Meulen. What edge computing means for infrastructure and operations leaders. https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders/, 2018. Accessed: 2021-07-10.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Ari S Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 5732–5741, 2018.

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

Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE, 2019.

Stephen J Roberts and Will D Penny. Variational bayes for generalized autoregressive models. *IEEE Transactions on Signal Processing*, 50(9):2245–2257, 2002.

Dipankar Sarkar, Ankur Narang, and Sumit Rai. Fed-focal loss for imbalanced data classification in federated learning. *arXiv preprint arXiv:2011.06283*, 2020.

Wenqi Shi, Sheng Zhou, and Zhisheng Niu. Device scheduling with fast convergence for wireless federated learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6. IEEE, 2020.

Jie Song, Tiantian Li, Zhi Wang, and Zhiliang Zhu. Study on energy-consumption regularities of cloud computing systems by a novel evaluation model. *Computing*, 95(4):269–287, 2013.

Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. *Advances in Neural Information Processing Systems*, 31:4447–4458, 2018.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *arXiv preprint arXiv:2103.00710*, 2021.

Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*, pp. 480–501. Springer, 2020.

Nguyen H Tran, Wei Bao, Albert Zomaya, Minh NH Nguyen, and Choong Seon Hong. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1387–1395. IEEE, 2019.

Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K. Leung. Overcoming noisy and irrelevant data in federated learning. *arXiv preprint arXiv:2001.08300*, 2020.

Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1698–1707. IEEE, 2020a.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020b.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

Jiaxiang Wu, Weidong Huang, Junzhou Huang, and Tong Zhang. Error compensated quantized sgd and its applications to large-scale distributed optimization. In *International Conference on Machine Learning*, pp. 5325–5333. PMLR, 2018.

Wentai Wu, Ligang He, Weiwei Lin, and Rui Mao. Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1539–1551, 2021a.

Wentai Wu, Ligang He, Weiwei Lin, Rui Mao, Carsten Maple, and Stephen A Jarvis. Safa: a semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Transactions on Computers*, 70(5):655–668, 2021b.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. In *International Conference on Learning Representations*, 2020.

Jaejun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn. Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8375–8384, 2020.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. How does data augmentation affect privacy in machine learning? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10746–10753, 2021.

Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 393–399, 2020.

Yuchen Zhang, John C Duchi, and Martin J Wainwright. Communication-efficient algorithms for statistical optimization. *The Journal of Machine Learning Research*, 14(1):3321–3363, 2013.

Shuxin Zheng, Qi Meng, Taifeng Wang, Wei Chen, Nenghai Yu, Zhi-Ming Ma, and Tie-Yan Liu. Asynchronous stochastic gradient descent with delay compensation. In *International Conference on Machine Learning*, pp. 4120–4129. PMLR, 2017.