IEEE Open Journal of the
Computer Society

Received 4 October 2023; accepted 6 November 2023. Date of publication 13 November 2023;
date of current version 28 December 2023. The review of this article was arranged by Associate Editor Peng Li.

Digital Object Identifier 10.1109/OJCS.2023.3332351

Benchmark for Personalized Federated Learning

KOJI MATSUDA, YUYA SASAKI, CHUAN XIAO, AND MAKOTO ONIZUKA

Osaka University, Suita, Osaka 565-0871, Japan

CORRESPONDING AUTHOR: YUYA SASAKI (e-mail: sasaki@ist.osaka-u.ac.jp).

This work was supported by JSPS KAKENHI under Grants JP22H03700, JP20H00584, and JP17H06099, Japan.

ABSTRACT Federated learning is a distributed machine learning approach that allows a single server to collaboratively build machine learning models with multiple clients without sharing datasets. Since data distributions may differ across clients, data heterogeneity is a challenging issue in federated learning. To address this issue, numerous federated learning methods have been proposed to build personalized models for clients, referred to as personalized federated learning. Nevertheless, no studies comprehensively investigate the performance of personalized federated learning methods in various experimental settings such as datasets and client settings. Therefore, in this article, we aim to benchmark the performance of existing personalized federated learning methods in various settings. We first survey the experimental settings in existing studies. We then benchmark the performance of existing methods through comprehensive experiments to reveal their characteristics in computer vision and natural language processing tasks which are the most popular tasks based on our survey. Our experimental study shows that (i) large data heterogeneity often leads to highly accurate predictions and (ii) standard federated learning methods (e.g. FedAvg) with fine-tuning often outperform personalized federated learning methods.

INDEX TERMS Benchmarking, Distributed Computing, Federated Learning.

I. INTRODUCTION

Federated learning has emerged as a promising distributed machine learning approach that enables a single server and multiple clients to collaboratively build machine learning models without sharing their datasets, thereby reducing privacy risks and communication traffic [39]. Due to its effectiveness in distributed scenarios, federated learning has received considerable attention from research communities. A vast array of federated learning methods has been proposed in recent years [11], [14], [21], [26], [30], [32], [51], [52], [55].

The general procedure of federated learning consists of two main steps: client training and model aggregation. In the client training step, clients train their own models on their local data and send their trained models to the server. In the model aggregation step, the server aggregates these models to build a global model and distributes the updated model to the clients. It repeatedly conducts two steps until reaching a given the number of epochs. This procedure can incorporate clients' local data into the global model without sharing data between the server and clients.

One of the key challenges in federated learning is data heterogeneity, where each client has local data with different distributions. This challenge poses difficulties in training a single global model that is optimal for all clients. As reported in previous studies, typical federated learning methods encounter a divergent issue when clients have non-IID local data [32], [33]. To overcome this challenge, recent research has focused on personalized federated learning (PFL), which aims to build personalized models optimized for individual clients [4], [12], [31], [34], [36], [38], [45], [50], [57].

Motivation: The number of PFL methods has significantly increased over the years. As a result, it is essential to understand the characteristics of existing PFL methods to develop new methods and select the optimal method for the user's situation.

To the best of our knowledge, a comprehensive comparison and analysis of state-of-the-art PFL methods in various settings have not been conducted yet. In addition, they did not investigate what factors (e.g., the number of clients) are important to evaluate the performance of existing methods.

Therefore, it is necessary (i) to benchmark the performance of existing PFL methods for a deeper understanding of them and (ii) to design experimental settings for fairly comparing PFL methods.

Contributions: In this article, we provide a comprehensive evaluation to benchmark the performance of the state-of-the-art personalized federated learning (PFL) methods in various experimental settings.

To start with, we examine the experimental settings used in existing studies, as each study uses different settings. We survey commonly used benchmarking datasets and the number of clients in federated learning settings. From our survey, computer vision and natural language processing tasks are the most popular tasks in federated learning, so we used them in our benchmarking. In addition, many studies use standard machine learning datasets (e.g., MNIST) after splitting the whole dataset into sub-datasets by using data-splitting methods that control the characteristics of data distributions. Thus, we also investigate what data-splitting methods are often used in existing studies.

Next, we conduct empirical studies to benchmark the performance of PFL methods in various experimental settings in terms of accuracy, training time, and communication traffic in computer vision and natural language processing tasks. In our setting, we evaluate the impact of the number of clients, the size of datasets, and the degree of data heterogeneity (i.e., the skewness of labels in local data). We evaluate eight state-of-the-art PFL methods, two non-personalized federated learning methods, and two non-federated learning methods. We also investigate the effectiveness of fine-tuning for personalization, which has not been well explored in previous studies.

Our empirical study reveals the pros and cons of existing methods. We report that highly accurate methods often require a large communication traffic and training time. We also find that standard federated learning methods with fine-tuning are capable of building highly accurate personalized models, which have not been evaluated fairly in previous studies. Additionally, we show that the PFL methods perform better when the degree of data heterogeneity is larger because personalized models can easily fit local data. We also demonstrate that the size of datasets has a smaller impact to evaluate their performance than the number of clients and data heterogeneity. Our experimental setting can help to design experimental studies. This article provides a valuable summary of the techniques of existing methods and performance comparison in various settings for researchers to develop new methods and practitioners to select optimal methods.

To facilitate further research, we open `FedMeasure`, a Jupyter notebook-based tool that supports easy experimental studies with various methods, experimental settings, and datasets under the MIT license[1].

II. PRELIMINARIES

A. PROBLEM FORMULATION

We describe the problem formulation of personalized federated learning. Consider a server and a set of clients which collaboratively build personalized models. Let S denote the set of clients. $|S|$ is the number of clients. We use a subscript i for the index of the i-th client. D_i and n_i denote the local data and the number of data samples (e.g., records, images, and texts) of client i, respectively. N denotes the sum of n_i across all the clients. x_i and y_i are the features and the labels of samples contained in the local data of client i, respectively. T and E are the total numbers of global communication rounds and local training rounds, respectively, where global communication refers to the communication between the server and the clients during training and local training refers to the training of each client's model using its local data.

In standard federated learning, a server and clients aim to build a single global model w_g. We define standard federated learning as the following optimization problem:

$$\min_{w_g \in \mathbb{R}^d} \sum_{i=1}^{|S|} \mathcal{T}_i(w_g), \tag{1}$$

where $\mathcal{T}_i$ is the objective for client i and is defined as follows:

$$\mathcal{T}_i(w) = \frac{1}{n_i} \sum_{(x_i, y_i) \in D_i} f_i(x_i, y_i, w), \tag{2}$$

where f_i is a loss function.

In personalized federated learning, a server and clients aim to create a personalized model w_p for each client. We define personalized federated learning as the following optimization problem:

$$\min_{\left\{w_{p_1}, \ldots, w_{p_{|S|}}\right\} \in \mathbb{R}^d} \sum_{i=1}^{|S|} \mathcal{T}_i(w_{p_i}), \tag{3}$$

where w_{p_i} is the personalized model of client i.

B. RELATED WORK

Existing PFL methods: Distinct personalized federated learning (PFL) approaches employ a variety of techniques to address data heterogeneity. We classify PFL methods into five primary categories: (1) clustering, (2) model mixture, (3) model parameter decoupling, (4) knowledge distillation, and (5) meta-learning.

- *Clustering* (e.g. [7], [36], [44]): Methods with clustering divide clients into multiple groups and utilize the groups to build personalized models.
- *Model mixture* (e.g. [31], [36], [48], [57]): Methods with model mixture update multiple model parameters by appropriately averaging weighted personalized and/or global models.
- *Model parameter decoupling* (e.g. [4], [12], [34]): In methods with model parameter decoupling, a part of a

[1] [Online]. Available: https://github.com/OnizukaLab/FedMeasure

TABLE 1. The Characteristics of Each Approach

Approach	Characteristics
Clustering	• No effect to client algorithms • Difficultly to determine the optimal clustering-based approaches
Model mixture	• Early convergences • Requiring substantial computation costs to decide model parameters
Model parameter decoupling	• Communication and training-efficient • Difficulty to determine how to split models
Knowledge distillation	• No model architecture restrictions for each client
Meta-learning	• Effectively building both global and personalized models

model is aggregated in the server, and each client combines the part with other locally updated parts to build their whole personalized models.

- *Knowledge distillation* (e.g. [35], [38], [45]): Knowledge distillation [22] is a technique for transferring the knowledge of a large model (called teacher model) to a small model (called student model) so that the student models mimic the output of the teacher model. In PFL settings, each client builds its own personalized model by using outputs of global or other clients' personalized models.
- *Meta-learning* (e.g. [2], [25], [50]): Meta-learning is a technique to improve learning algorithms by training on multiple tasks. Methods with meta-learning build a meta-model that helps to build personalized models only by re-training using each client's local data.

Each approach exhibits unique characteristics. A comprehensive summary of the characteristics associated with each method can be found in Table 1.

Existing benchmarks and tools on PFL methods: A few studies addressed empirical evaluations of PFL methods. Li et al. [29] empirically evaluated non-personalized federated learning in environments with data heterogeneity. Abdelemoniem et al. [1] evaluated the performance of FedAvg [39], which is the most basic algorithm, in various settings. In particular, they focused on the differences in the devices of clients. Chen et al. [10] conducted an empirical study on personalized federated learning. They focus on the effectiveness of add-on methods to some existing PFL methods such as fine-tuning and FedBN, but they used a small number of PFL methods. Thus, although they showed the effectiveness of combinations of PFL with add-on methods it is not comprehensive to benchmark PFL methods. Wu et al. [54] reviewed existing methods and compared three basic federated learning methods in a single dataset to show a case study, while it did not aim to benchmark the performance of existing methods. Therefore, to the best of our knowledge, there are no studies that benchmark various PFL methods. We focus on the performance of existing PFL methods and them combined with fine-tuning in various client settings.

Libraries and tools for federated learning are also developed such as Flower [6], Leaf [8], and Fedscope [53]. Although these provide some PFL methods and datasets, it is not sufficient to evaluate a variety of PFL methods. For example, there are several benchmarking (e.g., [10]) based on Fedscope, while it only provides four PFL methods. Therefore, our framework is useful to compare PFL methods in various settings. In addition, our framework aims to benchmark various settings, so it can apply new federated settings such as federated class-incremental learning [16], [17].

III. SETTINGS FOR FEDERATED LEARNING

Each previous study uses different (i) benchmarking datasets, (ii) data splitting methods to divide datasets into local data on clients, and (iii) the number of clients. We review benchmarking datasets, splitting methods, and the number of clients used in previous studies. Tables 2 and 3 summarize the datasets/data-splitting methods and the numbers of clients used in existing studies, respectively.

A. DATASETS

First, we show datasets built for federated learning experiments. Each dataset has an attribute that indicates who generates data samples and/or their domains, so we can divide the whole dataset into local data by using the attribute. We tag datasets with their data types: image, text, and numerical data.

- *FEMNIST (image) [8]:* It includes images of handwritten characters with 62 labels and is divided into 3,400 sub-datasets by writers.
- *Shakespeare (text) [32]:* It includes lines in "The Complete Works of William Shakespeare" and is divided into 143 sub-datasets of actors.
- *Sent140 (text) [8]:* It includes the text of tweets with two labels, either positive sentiment or negative sentiment. This dataset is divided into 660,120 sub-datasets of Twitter users.
- *Office-Home (image):* It includes images with four domains: Art, Clipart, Product, and Real world. All domains share the same 65 typical categories in office and home.
- *Human activity recognition (numerical)*[2] It includes mobile phone accelerometer and gyroscope data collected from 30 individuals, with six labels (walking, walking-upstairs, walking-downstairs, sitting, standing, and lying-down).
- *Vehicle sensor networks (numerical) [18]:* It includes sensor data collected from a distributed network of 23 sensors to predict vehicle types (AAV-type or DW-type).

[2] [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

TABLE 2. Benchmarking Data Summary

dataset	Split	Reference
FEMNIST	Writer	[7], [9], [12], [13], [25], [28], [31], [32], [37], [38]
	Random	[28]
	Class	[32]
Shakespeare	Role	[27], [37], [39], [38], [32], [13]
Office-Home	Domain	[49]
Sent140	User	[12], [32]
Vehicle Sensors Network	Sensor	[31], [48], [13]
Human Activity Recognition	Smartphone	[48], [13]
GLEAM	Smart glass	[48]
FLICKR-AES	Worker	[4]
MNIST	Random	[45], [39], [15], [5], [13]
	Class	[46], [24], [50], [39], [9] , [7], [15], [32], [19], [34]
	Dirichlet dist.	[58]
	Similarity	[57]
	Swapping out.	[44], [7]
Permuted MNIST	Random	[13]
Fashion MNIST	Class	[31], [56], [24], [9]
EMNIST	Class	[56], [24], [37], [36]
	Similarity and Class.	[26]
CelabA	Random	[31]
	Class	[58]
Stackoverflow	random	[31], [47]
CIFAR-10	Random	[45], [39], [15], [23], [5]
	Class	[2], [56], [3], [12], [4], [9], [15], [19], [34]
	Dirichlet dist.	[58], [35], [11], [37], [38], [23], [21], [52]
	Swapping out.	[44]
CIFAR-100	Class.	[2], [24], [12], [3] , [4], [11], [37], [28]
	Similarity	[57]
	Dirichlet dist.	[35], [21]
	Random	[11], [45], [28]
	Similarity	[57]
CINIC-10	Class	[3]
	Dirichlet dist.	[21]
ImageNet	Dirichlet dist.	[35]
Tiny-ImageNet	Random	[11]
	Class	[11]
SST2	Dirichlet dist.	[35]
AG News	Dirichlet dist.	[35]
WikiText2	Random	[15]
	Class	[15]
VQA	Class	[34]
madelon	?	[20]
ala	?	[20]
mushrooms	?	[20]
duke	?	[20]
MovieLens	?	[47]
Syntheic		[12], [32], [37], [52]

"?" indicates that splitting methods are not described in the paper.

- *GLEAM (numerical)*[3]*:* It includes two hours of high-resolution sensor data collected from 38 participants wearing Google Glass for activity recognition to predict activities (e.g., walking, talking, drinking).
- *FLICKR-AES (image) [43]:* It includes 40,000 photographs from flickr with aesthetic ratings (between 1 and 5) collected via Amazon Mechanical Turk by 210 annotators.

From Table 2, we can see that FEMNIST and Shakespeare are often used in existing studies. However, there are no standard benchmarking datasets to evaluate PFL methods.

B. DATA-SPLITTING METHODS

Many existing studies also used standard machine learning datasets that are commonly used in general machine

[3] [Online]. Available: http://www.skleinberg.org/data/GLEAM.tar.gz

TABLE 3. **Summary of the Number of Clients That are Used in Existing Studies**

# of clients	# of papers
5	3
10	4
15 or 16	3
20	7
30	4
50	4
100	14
101–499	7
500+	8

learning tasks such as MNIST, CIFAR-10, and CIFAR-100. The datasets are divided into sub-datasets by splitting methods. Since the splitting methods determine the characteristics of data distribution, the performance of existing methods changes depending on how to divide the datasets. Several splitting methods are used in existing studies as follows:

- *Random:* Divides a dataset into sub-datasets in a uniform random distribution.
- *Class:* Divides a dataset into sub-datasets so that the sub-datasets include only a limited number of labels. In a common way, we first sort data samples by labels and sequentially divide them into the same size sub-datasets.
- *Dirichlet-distribution:* Divides a dataset into sub-datasets according to Dirichlet distribution so that the sub-datasets include different biased labels. Given α_{label} (> 0) and the number of clients $|S|$, the Dirichlet distribution generates random numbers based on the standard gamma distribution for every label, taking α_{label} and $|S|$ as parameters, and divides the dataset into sub-datasets based on those random number proportions.
- *Similarity:* Divides a dataset into sub-datasets so that the features of the data samples within the same sub-dataset are similar. It often uses k-means to divide the dataset to sub-datasets.
- *Swapping out:* Randomly distribute data samples to clients and swap out two labels within each sub-dataset (i.e., data samples with similar features may have different labels across sub-datasets).

The class-based splitting method can create a peaky setting where sub-data typically includes one or two types of labels. The Dirichlet-distribution-based splitting method can create a setting where labels in sub-data are biased, but the number of types of labels is not too small. The Dirichlet-distribution-based splitting method is more realistic than the class-based one. The similarity-based splitting method uses features instead of labels. It assumes that local data includes data samples with similar features; for example, some clients have blue images, but others have red images. The swapping-out-based splitting method can create a special setting where labels are different, even if their features are the same. For example, this setting assumes that people add different labels to images. Furthermore, some studies generate synthetic datasets that follow their assumptions.

From Table 2, Random, Class, and Dirichlet-distribution are often used as data-splitting methods. Since Dirichlet-distribution can control the degree of data heterogeneity, it imitates Random and Class. Thus, we adopt Dirichlet-distribution in our experimental studies.

C. NUMBER OF CLIENTS

Even if some existing studies used the same datasets and splitting methods, the number of clients is often different. From Table 3, we can see that 20 and 100 are often used in existing studies. Only six papers changed the number of clients on the same datasets, so most papers do not evaluate the effect of the number of clients and the size of local data on each client. Since the number of clients significantly impacts accuracy, it is necessary to compare the accuracy of each method by varying the number of clients.

IV. EMPIRICAL STUDY

In this section, we introduce experimental settings and report our experimental results. We evaluate the performance of personalized and non-personalized federated learning methods in terms of accuracy, convergence speed, communication traffic, and training time. To validate their robustness for datasets/settings, we evaluate “Average rank” which indicates the sum of ranks for each dataset/setting divided by the number of datasets/settings.

To simplify the experiments, we used Pytorch to implement virtual clients and the server on a single GPU machine. Experiments were performed on a Linux server with NVIDIA Tesla V100 SXM2 GPU (16 GB memory) and Intel Xeon Gold 6148 Processor CPU (384 GB memory).

A. EXPERIMENTAL DESIGN DIMENSIONS

In federated learning, datasets, client, and training settings affect the performance of learning methods. To evaluate the performance of existing methods and understand their characteristics, we consider the following four design dimensions in this study.

Degree of data heterogeneity: As the degree of data heterogeneity increases, the accuracy of non-personalized federated learning decreases, while personalized federated learning rather improves accuracy because it builds a model that fits each client. Previous studies have not comprehensively evaluated this impact on the performance of personalized federated learning methods. In this paper, we compare the accuracy of existing methods by varying the degree of data heterogeneity.

Number of clients: The number of clients may significantly differ, depending on use cases. For example, the number of clients may be around 10 for small institutions, while the number of clients may be 100 or even more for mobile devices. As the number of clients increases, it becomes more difficult to aggregate models on the server, resulting in less accuracy. Therefore, a robust method for varying numbers of clients is desirable.

TABLE 4. Data Statistics

Datasets	Total size	Test size	$\|S\|$	Mean	SD	Max	Min
FEMNIST	749,068	77,483	3,400	220.3	85.20	465	19
Shakespeare	517,106	103,477	143	3,616.1	6,832.37	41,305	3
Sent140	74,589	7,895	927	80.5	40.02	549	50
MNIST	70,000	10,000	20	3,500.0	1,050.17	5,534	1,554
CIFAR-10	60,000	10,000	20	3,000.0	1,233.60	6,043	1,360

Total size and test size indicate the numbers of data samples in the entire dataset and the test data of the dataset, respectively. Mean, Max, and Min indicate average, maximum, and minimum number of data samples in local datasets, and SD indicates the standard deviation.

Total number of data samples: Like the number of clients, the total number of data samples also depends on the use case, and the performances of federated learning methods may differ when we vary the total number of data samples. Even if the server is aware of the number of data samples of the clients, it is challenging to select an optimal method. A robust method for different numbers of data samples is desirable. To this end, it is necessary to evaluate how the performances of existing methods vary with the total number of data samples.

B. EXPERIMENTAL SETUP

Datasets, tasks, and models: We use five datasets: FEMNIST, Shakespeare, Sent140, MNIST, and CIFAR-10 which are often used in existing studies. In FEMNIST and Shakespeare, we use original datasets. In Sent140 [8], we use 927 sub-datasets with more than 50 tweets in the experiment. In MNIST and CIFAR-10, we divide them into sub-datasets by the Dirichlet-distribution-based splitting method. The number of clients, $|S|$, is selected from $\{5, 10, 20, 100\}$. We change the total number of data samples using a ratio D to the entire dataset (i.e., the total number of data samples is $D \cdot N$), whose range is $\{0.25, 0.5, 0.75, 1.0\}$. We use a parameter α_{label} to control the degree of heterogeneity for the labels on the clients. α_{label} is selected from $\{0.1, 0.5, 1.0, 5.0\}$. The default values of $|S|$, D, and α_{label} are 20, 1.0, and 0.5, respectively. We vary these parameters to evaluate their impacts while using the above values as default parameters unless otherwise indicated.

The five datasets are pre-partitioned into training and test data. In FEMNIST, Shakespeare, and Sent140, we randomly select $|S|$ sub-datasets as local data. In MNIST and CIFAR-10, we randomly divide the whole train and test data into $|S|$ local data based on Dirichlet distribution. The distributions of test and train data follow the same distribution. We split the training data into 7 : 3 for FEMNIST, Shakespeare, and Sent140, and into 8 : 2 for MNIST and CIFAR-10. The two splits are used for training and validation, respectively. Table 4 shows the statistics of the above datasets.

In tasks and models, we follow the previous studies [11], [12], [28], [32], [36], [39], [42], [51]. In task settings, we conduct an image classification task for FEMNIST, MNIST, and CIFAR-10. For Shakespeare, we conduct a next-character prediction that infers the next characters after given sentences. For Sent140, we conduct a binary classification that categorizes whether a tweet is a positive or negative sentiment. We use different models for each task following the existing works [12], [42], [51]. For FEMNIST and MNIST we use CNN, and for Shakespeare we use LSTM. For CIFAR-10, we use VGG with the same modification reported in [51]. For Sent140, we use a pre-trained 300-dimensional GloVe embedding [41] and train RNN with an LSTM module.

Methods and hyperparameter tuning: We compare three types of methods: (1) non-PFL methods, (2) PFL methods, and (3) non-federated learning methods. For (1), we use FedAvg [39] and Fedprox [32]. For (2), we select PFL methods based on our survey. We use HypCluster [36] (i.e., with clustering), FML [45] (i.e., with knowledge-distillation), FedMe [38] (i.e., with knowledge-distillation), LG-FedAvg [34] (i.e., with model parameter decoupling), FedPer [4] (i.e., with model parameter decoupling), FedRep [12] (i.e., with model parameter decoupling), Ditto (i.e., with model mixture) [31], and pFedMe [50] (i.e., with meta-learning). For (3), we use Local Data Only, in which clients build their models on their local data, and Centralized, in which a server collects local data from all clients (centralized can be considered as an oracle). We use fine-tuning on each client for FedAvg, Fedprox, HypCluster, FedMe, and Centralized after building their models and denote them as "method + FT". In FML, LG-FedAvg, FedPer, FedRep, Ditto, and pFedMe, we do not use fine-tuning because techniques similar to fine-tuning are included in these methods.

We set the number of global communication rounds to be 300, 200, 500, 100, and 100 for FEMNIST, MNIST, CIFAR-10, Shakespeare, and Sent140, respectively. We set the local epoch E to be 2 for all the settings. All the clients participate in each global communication round following recent studies [4], [45], [51]. We conduct training and test five times and report mean and standard deviation (std) of accuracy over five times of experiments with different clients.

We describe hyperparameter tuning. The learning rate is selected from $\{10^{-3}, 10^{-2.5}, 10^{-2}, \ldots, 10^{0.5}\}$ and optimized for each method on default parameters. The optimal learning rate is selected for default parameters and used the same value for other experiments. The optimization method is SGD (stochastic gradient descent) with momentum 0.9 and weight decay 10^{-4}. The batch sizes of FEMNIST, MNIST, CIFAR-10, Shakespeare, and Sent140 are 20, 20, 40, 10, and 4, respectively. Hyperparameters specific to each method is described in our Github.

C. OVERALL PERFORMANCE COMPARISON

We compare the methods in terms of accuracy, convergence speed, training speed, and communications traffic in the default parameter setting.

Accuracy: Table 5 shows the accuracy and average ranking of each method. We note that the standard deviations of FEMNIST, Shakespeare, and Sent140 are relatively large because the clients differ in each test (we randomly select 20 clients from the set of clients). From Table 5, we can see that the most accurate method is FedMe+FT for FEMNIST, Ditto for Shakespeare, Hypcluster for Sent140, FedAvg+FT for MNIST, and FedMe+FT for CIFAR-10. From this result,

TABLE 5. **Test Accuracy (Mean ± Std Between Runs / Std Between Clients)**

	FEMNIST	Shakespeare	Sent140	MNIST	CIFAR-10	Average rank
FedAvg	75.79±1.65 / 10.64	44.94±1.96 / 9.22	58.83±11.88 / 34.45	98.90±0.10 / 0.53	86.05±0.48 / 3.62	9.2 / 6.0
FedAvg+FT	77.25±3.99 / 10.03	42.53±2.19 / 9.92	74.66± 6.20 / 26.65	**99.23±0.09** / 0.45	89.59±0.94 / 4.14	3.6 / **3.8**
FedProx	76.08±2.12 / 10.90	48.59±3.59 / 9.84	58.83±11.88 / 34.45	98.87±0.06 / 0.54	86.01±0.38 / 3.59	8.6 / 6.6
FedProx+FT	76.96±3.42 / 9.70	45.17±2.83 / 11.61	74.66± 6.20 / 26.65	99.20±0.10 / 0.50	89.76±0.62 / 3.85	3.6 / 4.6
HypCluster	75.99±2.94 / 10.99	41.82±3.33 / 11.64	**77.08±4.69** / 24.37	98.90±0.09 / 0.53	85.21±1.22 / 4.19	7.4 / 6.6
HypCluster+FT	76.29±3.15 / 9.87	41.10±3.29 / 11.35	73.16±9.41 / 27.24	99.15±0.12 / 0.49	88.54±1.42 / 4.29	7.2 / 5.8
FML	67.91±2.53 / 12.25	28.73±1.78 / 13.04	72.49±8.87 / 27.88	98.26±0.16 / 1.11	79.89±1.44 / 7.42	12.0 / 12.6
FedMe	77.64±2.39 / 9.89	46.98±2.30 / 10.10	73.99±8.29 / 26.83	98.92±0.14 / 0.83	88.15±0.52 / 5.09	5.8 / 7.4
FedMe+FT	**78.06±3.00** / 9.92	45.83±2.48 / 10.19	74.41±8.16 / 26.49	99.17±0.07 / 0.53	**90.96±0.84** / 3.59	**2.8** / **3.8**
LG-FedAvg	65.14±3.12 / 12.32	23.17±1.93 / 12.90	73.41±10.07 / 26.83	97.80±0.16 / 1.25	78.53±1.57 / 8.46	13.0 / 12.0
FedPer	65.96±2.81 / 12.69	30.83±3.32 / 12.43	74.16±7.59 / 26.85	99.11±0.08 / 0.59	90.00±0.83 /4.66	8.0 / 10.6
FedRep	66.04±2.20 / 11.86	31.71±2.29 / 11.80	73.91±8.33 / 26.87	99.06±0.07 / 0.61	88.96±0.48 /4.33	8.8 / 10.0
Ditto	75.68±3.63 / 10.47	**49.33±1.85** / 11.53	74.28±8.10 / 26.63	99.22±0.06 / 0.50	90.41±0.67 / 3.56	3.8 / 4.2
pFedMe	72.92±3.54 / 11.38	40.33±2.27 / 10.52	71.20±10.25 / 28.16	98.96±0.05 / 0.64	79.46±2.08 / 6.56	10.6 / 10.2
Local Data Only	64.71±2.90 / 12.67	24.77±1.95 / 12.99	74.33±7.86 / 26.55	97.60±0.28 / 1.34	73.17±1.55 / 9.71	- / -
Centralized	76.08±1.65 / 11.65	47.64±2.63 / 11.15	58.83±11.88 / 34.45	98.89±0.05 / 0.51	85.96±0.54 / 3.74	- / -
Centralized+FT	79.35±2.29 / 9.13	48.43±3.32 / 11.42	67.91±7.41 / 31.68	99.27±0.08 / 0.47	90.80±0.92 / 3.47	- / -

We report average ranking on mean accuracy (the higher the better) and std between clients (the smaller the better).
The bold values indicate the highest accuracy among federated learning methods.

we find that none of the existing state-of-the-art personalized federated learning methods outperform the others in all the datasets.

We can also see that FedMe+FT has the highest average rank. On the other hand, the other personalized federated learning methods have lower average ranks than the standard federated learning methods such as FedAvg and FedProx with fine-tuning. From this result, we can find that only a few state-of-the-art personalized methods outperform standard federated learning methods, and those with fine-tuning are often sufficient to deal with data heterogeneity.

The standard deviation between clients indicates the difference in accuracy between clients. Thus, if the std is smaller, clients achieve similar accuracy, i.e., fairly provide accurate models to clients. Among existing methods, FedAvg+FT and FedMe+FT achieved the best average rank. We can also see that fine-tuning often decreases standard deviations between clients, and thus it contributes to providing suitable models for each client.

Convergence speed: Fig. 1 shows the validation accuracy of each global communication round. The validation accuracy is the average accuracy at each epoch of the five experiments. Since each client evaluates its model by its own validation data after training its model and before aggregating models, the accuracy of each method is equivalent to that after fine-tuning.

From Fig. 1, we can see that FedAvg and Ditto are stable and converge quickly for all datasets. On the other hand, we can see that FedMe has the highest average rank but loses in convergence speed to FedAvg and Ditto. From this result, we can find that the methods with the highest accuracy and the fastest convergence are different.

Training time: We evaluate run time on the training phase in each method. Fig. 2 shows the average run time per global communication round. We note that the run time is the average of ten global communication rounds.

From Fig. 2, we can see that FedAvg has the smallest training time for all datasets. FedMe and Ditto have a large training time than the other methods. pFedMe spends similar training time to the other methods on FEMNIST and Sent140, while it spends much larger time than the other methods on Shakespeare, MNIST, and CIFAR-10. pFedMe has large training time for clients, so when the volume of local data increases, its training time increases.

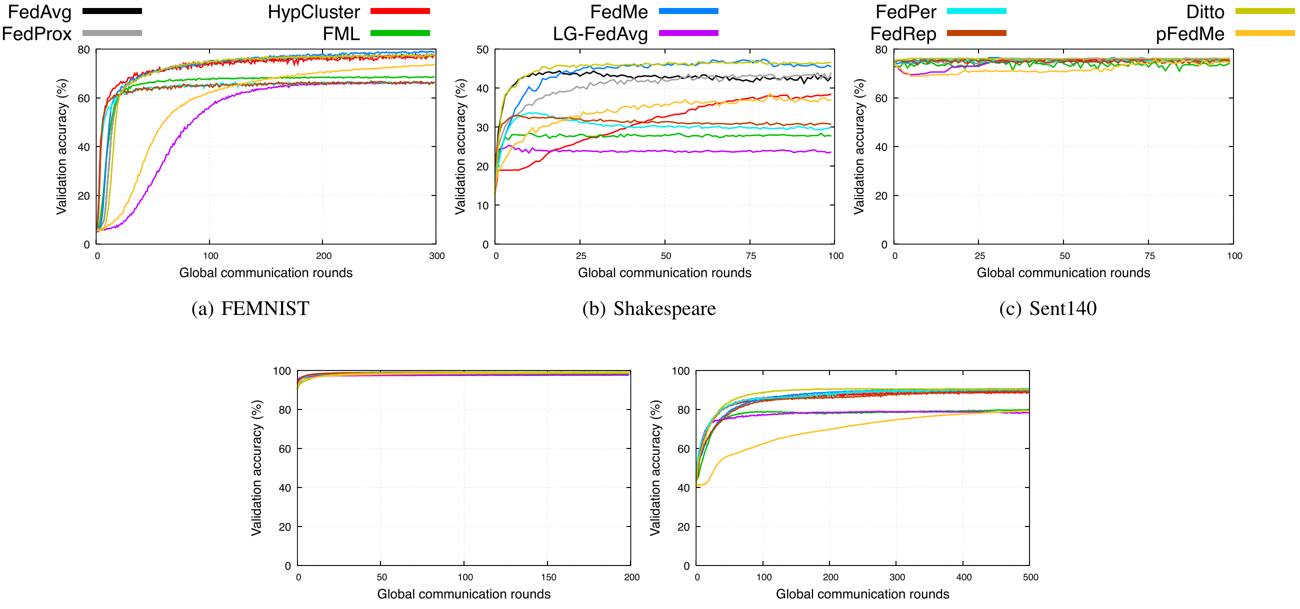
Communications traffic: We evaluate communications traffic on the training phase in each method. Since each method exchanges models between the server and client, communications traffic is compared by the size of model parameters sent per global communication round. Table 6 shows the communications traffic per global communication round.

From Table 6, we can see that FedMe has the largest communication traffic. This is because FedMe has the extra model transmission compared with the other methods. FedPer, FedRep, and LG-FedAvg have smaller communication traffic than other methods because these three methods send only a part of the model between the server and the clients. LG-FedAvg has the smallest communication traffic among them because the output side of the model has a smaller number of model parameters than the input side of the model.

D. IMPACT OF EXPERIMENTAL SETTINGS ON ACCURACY

In this section, we compare the accuracy of each method in different experimental settings.

Impact of the degree of data heterogeneity: Table 7 shows the accuracy when we vary the degree of data heterogeneity. A smaller α_{label} indicates a larger degree of data heterogeneity (i.e., close to the class-based splitting). On the other hand, a



(a) FEMNIST (b) Shakespeare (c) Sent140

(d) MNIST (e) CIFAR-10

FIGURE 1. Validation accuracy over time of various methods.

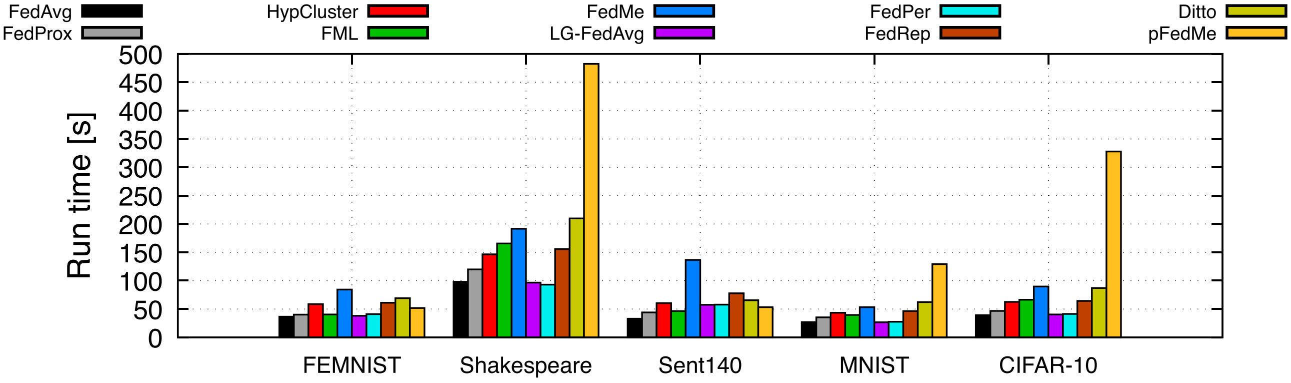


FIGURE 2. Training time per global communication round.

TABLE 6. Communication Traffic: The Number of Model Parameters Communicated Between the Server and Clients Per Round

	FEMNIST		Shakespeare		Sent140		MNIST		CIFAR-10	
FedAvg	2413180	(1×)	1645140	(1×)	161344	(1×)	2399764	(1×)	19870868	(1×)
FedProx	2413180	(1×)	1645140	(1×)	161344	(1×)	2399764	(1×)	19870868	(1×)
HypCluster	3619770	(1.5×)	2467710	(1.5×)	242016	(1.5×)	3599646	(1.5×)	29806302	(1.5×)
FML	2413180	(1×)	1645140	(1×)	161344	(1×)	2399764	(1×)	19870868	(1×)
FedMe	6032950	(2.5×)	4112850	(2.5×)	403360	(2.5×)	5999410	(2.5×)	49677170	(2.5×)
LG-FedAvg	15996	(0.007×)	46260	(0.028×)	25644	(0.159×)	2580	(0.001×)	1060884	(0.053×)
FedPer	2397184	(0.993×)	1598880	(0.972×)	161300	(1×)	2397184	(0.999×)	18809984	(0.947×)
FedRep	2397184	(0.993×)	1598880	(0.972×)	161300	(1×)	2397184	(0.999×)	18809984	(0.947×)
Ditto	2413180	(1×)	1645140	(1×)	161344	(1×)	2399764	(1×)	19870868	(1×)
pFedMe	2413180	(1×)	1645140	(1×)	161344	(1×)	2399764	(1×)	19870868	(1×)

TABLE 7. Accuracy v.s. Degree of Data Heterogeneity

	MNIST					CIFAR-10				
	α_{label}=5.0	α_{label}=1.0	α_{label}=0.5	α_{label}=0.1	Average rank	α_{label}=5.0	α_{label}=1.0	α_{label}=0.5	α_{label}=0.1	Average rank
FedAvg	98.95±0.04	98.89±0.09	98.90±0.10	98.61±0.24	7.0	86.76±0.40	86.20±0.66	86.05±0.48	80.04±2.89	9.5
FedAvg+FT	98.94±0.08	**99.12±0.06**	**99.23±0.09**	99.52±0.18	**1.8**	86.67±0.75	88.56±1.32	89.59±0.94	94.45±0.97	4.5
FedProx	98.93±0.05	98.90±0.06	98.87±0.06	98.61±0.23	7.8	86.79±0.21	86.26±0.39	86.01±0.38	80.86±2.02	9.0
FedProx+FT	**98.98±0.06**	99.07±0.10	99.20±0.10	**99.54±0.14**	**1.8**	86.17±0.46	88.51±0.65	89.76±0.62	94.57±1.11	4.5
HypCluster	98.87±0.16	98.50±0.58	98.90±0.09	98.38±0.26	9.5	84.93±0.45	84.45±0.64	85.21±1.22	82.14±1.88	11.0
HypCluster+FT	98.81±0.09	98.71±0.64	99.15±0.12	99.40±0.12	6.3	84.13±0.49	86.52±1.08	88.54±1.42	93.92±1.20	8.0
FML	97.79±0.16	96.92±1.79	98.26±0.16	98.01±1.83	13.0	68.89±0.89	75.59±1.50	79.89±1.44	91.16±2.29	11.3
FedMe	98.72±0.11	98.73±0.20	98.92±0.14	98.89±0.25	8.0	87.01±0.45	87.98±0.68	88.15±0.52	82.79±8.33	7.0
FedMe+FT	98.84±0.09	98.93±0.24	99.17±0.07	99.46±0.12	4.5	**87.73±0.45**	**89.60±0.74**	**90.96±0.84**	94.50±1.28	**1.5**
LG-FedAvg	97.08±0.11	96.28±1.61	97.80±0.16	97.77±1.94	14.0	67.66±0.76	74.26±1.63	78.53±1.57	90.93±2.28	12.5
FedPer	98.81±0.07	97.80±1.71	99.11±0.08	98.84±0.98	8.5	86.25±0.79	88.26±0.73	90.00±0.83	93.97±1.70	5.0
FedRep	98.71±0.08	97.78±1.82	99.06±0.07	98.86±1.05	9.5	85.25±0.55	86.81±0.98	88.96±0.48	93.55±1.69	7.3
Ditto	98.89±0.08	99.05±0.15	99.22±0.06	98.87±1.44	4.3	87.52±0.34	89.22±0.32	90.41±0.67	**94.82±1.06**	1.8
pFedMe	98.67±0.09	98.69±0.19	98.96±0.05	99.21±0.09	8.5	69.93±1.08	63.16±29.82	79.46±2.08	86.23±3.48	12.3
Local Data Only	96.73±0.14	96.02±1.58	97.60±0.28	97.76±1.54	-	59.03±0.34	66.74±1.32	73.17±1.55	88.85±2.92	-
Centralized	98.85±0.06	98.90±0.02	98.89±0.05	98.83±0.13	-	85.68±0.70	85.62±0.75	85.96±0.54	85.84±0.74	-
Centralized+FT	99.02±0.08	99.11±0.10	99.27±0.08	99.37±0.16	-	87.27±0.31	88.99±0.67	90.80±0.92	95.59±1.10	-

The bold values indicate the highest accuracy among federated learning methods.

TABLE 8. Accuracy v.s. Number of Clients

	MNIST					CIFAR-10				
	$\lvert S\rvert = 5$	$\lvert S\rvert = 10$	$\lvert S\rvert = 20$	$\lvert S\rvert = 100$	Average rank	$\lvert S\rvert = 5$	$\lvert S\rvert = 10$	$\lvert S\rvert = 20$	$\lvert S\rvert = 100$	Average rank
FedAvg	98.68±0.13	98.82±0.14	98.90±0.10	98.67±0.08	9.3	87.33±0.56	86.62±1.11	86.05±0.48	81.60±0.50	8.8
FedAvg+FT	99.20±0.16	**99.26±0.06**	**99.23±0.09**	98.87±0.09	**1.8**	90.18±1.05	90.89±0.51	89.59±0.94	82.61±0.58	4.0
FedProx	98.65±0.11	98.81±0.14	98.87±0.06	98.66±0.09	10.5	87.82±0.94	86.80±0.70	86.01±0.38	81.55±0.53	8.8
FedProx+FT	99.17±0.14	99.24±0.11	99.20±0.10	**98.89±0.12**	2.8	90.22±1.04	91.16±0.54	89.76±0.62	82.24±0.83	3.5
HypCluster	98.86±0.20	98.86±0.08	98.90±0.09	98.62±0.08	9.0	86.27±0.79	85.71±1.22	85.21±1.22	80.38±1.01	10.8
HypCluster+FT	99.12±0.15	99.11±0.11	99.15±0.12	98.84±0.14	5.5	88.52±1.00	88.60±0.92	88.54±1.42	81.65±0.29	6.8
FML	98.84±0.31	98.71±0.13	98.26±0.16	95.31±0.35	12.5	84.01±7.31	83.80±0.56	79.89±1.44	65.97±0.68	12.3
FedMe	98.93±0.14	98.87±0.12	98.92±0.14	97.93±0.20	9.3	89.68±0.65	89.15±1.11	88.15±0.52	76.55±1.21	8.0
FedMe+FT	99.23±0.15	99.19±0.12	99.17±0.07	98.35±0.19	4.5	91.81±0.85	**91.93±0.24**	**90.96±0.84**	81.23±0.47	3.0
LG-FedAvg	87.10±4.49	96.46±3.97	97.80±0.16	94.30±0.41	14.0	85.47±2.12	83.01±1.01	78.53±1.57	64.85±1.00	13.0
FedPer	99.15±0.20	99.21±0.08	99.11±0.08	97.96±0.26	6.0	90.27±1.81	90.33±0.85	90.00±0.83	81.61±0.80	4.0
FedRep	99.13±0.26	99.11±0.07	99.06±0.07	97.86±0.27	8.0	89.57±1.58	89.52±0.60	88.96±0.48	80.51±0.69	7.0
Ditto	**99.27±0.17**	99.25±0.08	99.22±0.06	98.19±0.24	3.3	**92.15±0.76**	91.59±0.40	90.41±0.67	**82.79±0.68**	**1.5**
pFedMe	99.14±0.17	99.08±0.09	98.96±0.05	97.89±0.19	8.3	81.08±2.73	82.13±2.53	79.46±2.08	58.41±1.63	13.8
Local Data Only	98.78±0.27	98.28±0.12	97.60±0.28	93.86±0.37	-	82.23±2.26	78.31±1.26	73.17±1.55	59.76±1.03	-
Centralized	98.83±0.06	98.91±0.06	98.89±0.05	98.81±0.06	-	85.93±0.28	86.24±0.60	85.96±0.54	86.07±0.55	-
Centralized+FT	99.21±0.10	99.24±0.03	99.27±0.08	99.06±0.06	-	90.16±1.13	90.91±0.36	90.80±0.92	90.62±0.40	-

The bold values indicate the highest accuracy among federated learning methods.

larger α_{label} indicates a smaller degree of data heterogeneity (i.e., close to the random splitting).

From Table 7, we can see that the accuracy of FedAvg and FedProx decreases as the degree of data heterogeneity increases. On the other hand, we can see that the accuracy of personalized federated learning methods tends to increase as the degree of data heterogeneity increases. As the degree of data heterogeneity increases, the clients can easily build personalized models that fit their local data. We can find that data heterogeneity works positively for personalized federated learning.

We can also see that FedAvg+FT and FedProx+FT have the highest average rank on MNIST, and FedMe+FT has the highest average rank on CIFAR-10. This result indicates that the standard federated learning methods with fine-tuning are often sufficient to deal with the data heterogeneity.

Impact of the number of clients: Table 8 shows the accuracy of varying the number of clients. From Table 8, we can see that the accuracy decreases significantly as the number of clients increases. As the number of clients increases, it becomes more difficult to aggregate the model on the server, resulting in decreasing accuracy. FedAvg+FT has the highest average rank for MNIST, and Ditto has the highest average rank for CIFAR-10. This result indicates that a larger number of clients is more challenging, while we can design robust methods for a different number of clients.

Impact of the total number of data samples: Table 9 shows the accuracy when we vary the total number of data samples. From Table 9, we can see that the accuracy decreases as the total number of data samples decreases. This is because clients do not have sufficient data samples to train their models when the number of data samples is small. The ranks of methods

TABLE 9. Accuracy v.s. Total Number of Data Samples

	MNIST					CIFAR-10				
	D=0.25	D=0.5	D=0.75	D=1.0	Average rank	D=0.25	D=0.5	D=0.75	D=1.0	Average rank
FedAvg	97.76±0.30	98.49±0.26	98.70±0.07	98.90±0.10	8.5	70.82±1.58	79.99±0.41	83.75±0.71	86.05±0.48	9.5
FedAvg+FT	**98.41±0.19**	98.90±0.24	**99.13±0.06**	**99.23±0.09**	**1.3**	77.27±2.19	84.76±0.23	87.75±1.09	89.59±0.94	4.0
FedProx	97.79±0.30	98.56±0.24	98.68±0.08	98.87±0.06	8.8	71.13±1.35	79.90±0.42	83.84±0.73	86.01±0.38	9.5
FedProx+FT	98.36±0.23	**98.98±0.17**	**99.13±0.05**	99.20±0.10	1.8	76.75±1.71	84.61±0.47	88.17±1.08	89.76±0.62	4.0
HypCluster	97.20±0.20	98.44±0.23	98.66±0.15	98.90±0.09	10.5	70.38±0.84	78.83±0.61	82.59±1.39	85.21±1.22	11.0
HypCluster+FT	97.73±0.21	98.66±0.28	98.94±0.13	99.15±0.12	6.0	73.84±2.30	82.28±0.75	85.95±1.35	88.54±1.42	7.3
FML	95.53±1.94	97.23±0.30	97.90±0.12	98.26±0.16	13.0	65.86±3.54	72.99±0.86	77.38±1.88	79.89±1.44	12.3
FedMe	97.83±0.38	98.47±0.13	98.63±0.13	98.92±0.14	9.0	71.43±3.49	83.50±1.15	86.45±1.10	88.15±0.52	7.3
FedMe+FT	98.11±0.45	98.78±0.19	99.01±0.08	99.17±0.07	3.8	**78.21±1.32**	**86.40±0.63**	**89.43±0.93**	**90.96±0.84**	**1.0**
LG-FedAvg	94.58±2.11	96.54±0.25	97.35±0.18	97.80±0.16	14.0	66.20±3.12	72.26±0.97	76.38±2.22	78.53±1.57	13.3
FedPer	97.00±1.69	98.62±0.25	98.99±0.10	99.11±0.08	7.3	77.39±2.53	85.26±0.86	88.32±0.47	90.00±0.83	2.3
FedRep	97.10±1.65	98.60±0.09	98.94±0.08	99.06±0.07	7.8	75.65±1.98	83.82±0.72	87.36±0.96	88.96±0.48	5.5
Ditto	98.01±0.35	98.86±0.20	99.09±0.08	99.22±0.06	3.0	73.41±2.72	83.40±1.25	88.18±0.71	90.41±0.67	4.8
pFedMe	97.58±0.23	98.39±0.27	98.67±0.14	98.96±0.05	9.8	60.95±3.73	71.85±1.05	77.00±1.66	79.46±2.08	13.5
Local Data Only	94.17±1.98	96.07±0.26	97.05±0.14	97.60±0.28	-	60.34±4.30	66.50±1.01	70.51±2.61	73.17±1.55	-
Centralized	97.72±0.28	98.56±0.19	98.77±0.10	98.89±0.05	-	74.67±1.40	80.53±0.75	83.97±0.98	85.96±0.54	-
Centralized+FT	98.27±0.13	98.95±0.20	99.12±0.05	99.27±0.08	-	83.21±0.76	87.06±0.58	89.64±1.03	90.80±0.92	-

The bold values indicate the highest accuracy among federated learning methods.

do not change much, so the number of data samples does not significantly impact deciding the superiority of methods.

E. SUMMARY

We summarize our experimental results as follows:

- There is a trade-off between accuracy, communication traffic, and training time. For example, FedMe is accurate in various experimental settings but reports large communication traffic and training time. Therefore, it is essential to report not only accuracy but also communication traffic and training time.
- The standard federated learning methods with fine-tuning work well for data heterogeneity. In particular, in easy-to-learn datasets such as MNIST, they outperform the personalized federated learning methods.
- In a large degree of heterogeneity, we observed higher accuracy of federated learning methods. These characteristics should be considered when developing and evaluating new federated learning methods.
- The number of clients has a large impact on the accuracy, so it is important to evaluate the performance in various settings. On the other hand, since the size of the dataset does not have a large impact, it is not essential to evaluate its impact.
- Our experimental settings can reveal the pros and cons of existing methods. So, our settings can evaluate the performance of PFL methods fairly.

V. OPEN ISSUES

We discuss open issues of personalized federated learning.

A. HYPER-PARAMETER SEARCH

It is difficult to tune hyper-parameters in (personalized) federated learning. Even when we select the best hyper-parameters using the whole dataset, it takes a large time to select them. There are two types of hyper-parameters; client and global settings. For the former, each client possibly selects their best hyper-parameters if the hyper-parameters only affect their personalized models, such as the number of local epochs and learning rate. However, their personalized models affect other personalized models, so it may cause the deterioration of the performance of other personalized models. We need to avoid selfishly selecting hyper-parameters, so it is beneficial to develop hyper-parameter tuning methods that improve the performance of all personalized models.

For the latter, some hyper-parameters of the methods are shared among clients to build personalized models. For example, in methods with clustering, the optimal number of clusters may be different from each client, and in methods with model-decoupling, optimal server and client-side models may be different. The server generally cannot collect the accuracy of personalized models due to privacy concerns. So, the server needs to select the hyper-parameters from their personalized models and/or other non-privacy information.

Therefore, we need efficient and effective hyper-parameter search methods.

B. HETEROGENEOUS CLIENTS

In our experimental studies, we assume that all clients are the same device. However, it is often not true, in particular, federated learning among mobile clients, for example, people use smartphones and tablets with different computing resources and communication bandwidth. Setting on heterogeneous clients assumes that each client has different devices [1]. This setting follows the real-world application because devices are generally different across clients. Therefore, some devices cannot store large size of models due to the memory space, and other devices may take a long time to train their personalized models.

Existing PFL methods often assume that each client has the same device and consider only the accuracy performance of each client. As we show the training time and communication traffic of existing methods, some methods take large (or small) training time and model sizes. We need PFL methods that adaptively select models and training methods according to client devices.

C. BENCHMARKING SETTING

Many PFL methods have been proposed, but they are evaluated in different datasets and metrics. There is no de facto standard on the evaluation setting on (personalized) federated learning. In addition, the metrics differ across existing studies, for example, accuracy, communication traffic, and training time. In our experimental results, FedMe and Ditto often achieved good accuracy, but it takes a longer training time compared with other methods. Furthermore, fairness (e.g., group and individual fairness) has become important recently in machine learning fields. So, we need to consider additional metrics that are not used in existing studies.

Furthermore, new federated learning settings have been studied recently. For example, federated class-incremental learning [16], [17] assumes that the number of classes increases on demand (e.g., increases the predicted target of diseases, e.g., COVID-19). We need to benchmark the performance on recent new settings for further studies.

VI. CONCLUSION

In this article, we empirically evaluated personalized federated methods in various experimental settings. Our experimental results showed several key findings: First, no method consistently outperformed the others in all the datasets. Second, standard federated learning with fine-tuning was accurate compared with most personalized federated learning methods. Third, the large degree of data heterogeneity improved the accuracy of personalized federated learning methods. We opened our Jupyter notebook-based tool `FedMeasure` to facilitate experimental studies. We hope that our experimental results help to develop and evaluate new federated learning methods.

Limitations and future work: This study has three limitations. First, despite 17 methods (ten federated learning, four variants, and three non-federated learning methods) and five datasets were used in this study, which are comprehensive compared with previous ones, we also note that there are numerous other federated learning methods (e.g., [24], [26], [37], [40], [52]) and datasets. Second, to study the impact of the data heterogeneity, we controlled the label distribution skew but did not investigate the impact of other types of skews, such as quantity skew, in which each client has a different number of data samples, and feature distribution skew, in which the clients' data share the same labels but vary in features. Third, we varied the number of clients, the total number of data samples, and the degree of data heterogeneity, whereas other parameters, such as client-participant ratio, the number of local epochs, and model architectures, were not varied. In future work, we plan to enrich our benchmark tool by adding datasets and methods to find further insights.

REFERENCES

[1] A. M. Abdelmoniem, C.-Y. Ho, P. Papageorgiou, and M. Canini, "Empirical analysis of federated learning in heterogeneous environments," in *Proc. 2nd Eur. Workshop Mach. Learn. Syst.*, 2022, pp. 1–9.

[2] D. A. E. Acar et al., "Debiasing model updates for improving personalized federated training," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 21–31.

[3] I. Achituve, A. Shamsian, A. Navon, G. Chechik, and E. Fetaya, "Personalized federated learning with Gaussian processes," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2021, pp. 8392–8406.

[4] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," 2019, *arXiv:1912.00818*.

[5] M. Asad, A. Moustafa, and T. Ito, "Fedopt: Towards communication efficiency and privacy preservation in federated learning," *Appl. Sci.*, vol. 10, no. 8, 2020, Art. no. 2864.

[6] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," 2020, *arXiv:2007.14390*.

[7] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–9.

[8] S. Caldas et al., "LEAF: A benchmark for federated settings," 2018, *arXiv:1812.01097*.

[9] Z. Chai et al., "TiFL: A tier-based federated learning system," in *Proc. 29th Int. Symp. High-Performance Parallel Distrib.*, pp. 125–136, 2020.

[10] D. Chen, D. Gao, W. Kuang, Y. Li, and B. Ding, "pFL-bench: A comprehensive benchmark for personalized federated learning," in *Proc. NeurIPS Datasets Benchmarks Track*, 2022.

[11] H.-Y. Chen and W.-L. Chao, "FedBE: Making bayesian model ensemble applicable to federated learning," in *Proc. Int. Conf. Learn. Representations*, 2021.

[12] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 2089–2099.

[13] L. Corinzia, A. Beuret, and J. M. Buhmann, "Variational federated multi-task learning," 2019, *arXiv:1906.06268*.

[14] Z. Dai, B. K. H. Low, and P. Jaillet, "Federated Bayesian optimization via thompson sampling," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 9687–9699.

[15] E. Diao, J. Ding, and V. Tarokh, "HeteroFL Computation and communication efficient federated learning for heterogeneous clients," in *Proc. Int. Conf. Learn. Representations*, 2021.

[16] J. Dong et al., "Federated class-incremental learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10164–10173.

[17] J. Dong, D. Zhang, Y. Cong, W. Cong, H. Ding, and D. Dai, "Federated incremental semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3934–3943.

[18] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 826–838, 2004.

[19] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 3557–3568.

[20] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtárik, "Lower bounds and optimal algorithms for personalized federated learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 2304–2315.

[21] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large CNNs at the edge," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 14068–14080.

[22] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[23] B. Huang, X. Li, Z. Song, and X. Yang, "FL-NTK: A neural tangent kernel-based framework for federated learning analysis," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4423–4434.

[24] Y. Huang et al., "Personalized cross-silo federated learning on non-IID data," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 7865–7873.

[25] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," 2019, *arXiv:1909.12488*.

[26] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[27] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, "Adaptive gradient-based meta-learning methods," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5917–5928.

[28] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, *arXiv:1910.03581*.

[29] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," in *Proc. IEEE Int. Conf. Data Eng.*, 2022, pp. 965–978.

[30] Q. Li, Z. Wen, and B. He, "Practical federated gradient boosting decision trees," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 4642–4649.

[31] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6357–6368.

[32] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.

[33] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-IID data," in *Proc. Int. Conf. Learn. Representations*, 2019.

[34] P. P. Liang, T. Liu, L. Ziyin, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," 2020, *arXiv:2001.01523*.

[35] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2020, pp. 2351–2363.

[36] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," 2020, *arXiv:2002.10619*.

[37] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal, "Federated multi-task learning under a mixture of distributions," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2021, pp. 15434–15447.

[38] K. Matsuda, Y. Sasaki, C. Xiao, and M. Onizuka, "FedME: Federated learning via model exchange," in *Proc. SIAM Int. Conf. Data Mining*, 2022, pp. 459–467.

[39] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 26th Int. Conf. Art. Intell. Statist.*, 2017, pp. 1273–1282.

[40] K. Ozkara, N. Singh, D. Data, and S. Diggavi, "QuPed: Quantized personalization via distillation with applications to federated learning," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2021, pp. 3622–3634.

[41] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[42] S. Reddi et al., "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Representations*, 2021.

[43] J. Ren, X. Shen, Z. Lin, R. Mech, and D. J. Foran, "Personalized image aesthetics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 638–647.

[44] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.

[45] T. Shen et al., "Federated mutual learning," 2020, *arXiv:2006.16765*.

[46] N. Shoham et al., "Overcoming forgetting in federated learning on non-IID data," 2019, *arXiv:1910.07796*.

[47] K. Singhal, H. Sidahmed, Z. Garrett, S. Wu, J. Rush, and S. Prakash, "Federated reconstruction: Partially local federated learning," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2021, pp. 11220–11232.

[48] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.

[49] B. Sun, H. Huo, Y. Yang, and B. Bai, "PartialFed: Cross-domain personalized federated learning via partial initialization," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2021, pp. 23309–23320.

[50] C. T. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21394–21405.

[51] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Representations*, 2020.

[52] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2020, pp. 7611–7623.

[53] Z. Wang et al., "Federatedscope-GNN: Towards a unified, comprehensive and efficient package for federated graph learning," in *Proc. Conf. Knowl. Discov. Data Mining*, 2022, pp. 4110–4120.

[54] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, 2020.

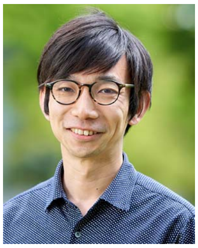
[55] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7252–7261.

[56] J. Zhang, S. Guo, X. Ma, H. Wang, W. Xu, and F. Wu, "Parameterized knowledge transfer for personalized federated learning," in *Proc. Int. Adv. Conf. Neural Inf. Process. Syst.*, 2021, pp. 10092–10104.

[57] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," in *Proc. Int. Conf. Learn. Representations*, 2021.

[58] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 12878–12889.

KOJI MATSUDA received the B.E. degree from Osaka University, Suita, Japan, in 2021. He is currently a Student with the Graduate School of Information Science and Technology, Osaka University. His research focuses on federated learning.



YUYA SASAKI received the B.E., M.E, and Ph.D. degrees from Osaka University, Suita, Japan, in 2009, 2011, and 2014, respectively. He is currently an Assistant Professor with the Graduate School of Information Science and Technology, Osaka University. His research interests include spatio-temporal/graph data management and analysis and applied data science to other fields such as chemical and medical science.

CHUAN XIAO received the B.E. degree from Northeastern University, Shenyang, China, in 2005, and the Ph.D. degree from the University of New South Wales, Sydney, NSW, Australia, in 2010. He is currently an Associate Professor with the Graduate School of Information Science and Technology, Osaka University, Suita, Japan, and a guest Associate Professor with the Graduate School of Informatics, Nagoya University, Nagoya, Japan. His research interests include data cleaning, data integration, data lake management, and textual databases.

MAKOTO ONIZUKA is currently a Professor (Executive Assistants to the President) with the Graduate School of Information Science and Technology, Osaka University, Suita, Japan. He is the leader of Big Data Engineering Laboratory and conducts research on graph mining algorithms and AI-driven database query optimization techniques.