Federated Learning on Non-IID Data Silos: An Experimental Study

Qinbin Li*
qinbin@comp.nus.edu.sg
National University of Singapore

Yiqun Diao*
diaoyiqun@sjtu.edu.cn
Shanghai Jiao Tong University

Quan Chen
chen-quan@sjtu.edu.cn
Shanghai Jiao Tong University

Bingsheng He
hebs@comp.nus.edu.sg
National University of Singapore

ABSTRACT

Machine learning services have been emerging in many data-intensive applications, and their effectiveness highly relies on large-volume high-quality training data. However, due to the increasing privacy concerns and data regulations, training data have been increasingly fragmented, forming distributed databases of multiple "data silos" (e.g., within different organizations and countries). To develop effective machine learning services, there is a must to exploit data from such distributed databases without exchanging the raw data. Recently, federated learning (FL) has been a solution with growing interests, which enables multiple parties to collaboratively train a machine learning model without exchanging their local data. A key and common challenge on distributed databases is the heterogeneity of the data distribution (i.e., non-IID) among the parties. There have been many FL algorithms to address the learning effectiveness under non-IID data settings. However, there lacks an experimental study on systematically understanding their advantages and disadvantages, as previous studies have very rigid data partitioning strategies among parties, which are hardly representative and thorough. In this paper, to help researchers better understand and study the non-IID data setting in federated learning, we propose comprehensive data partitioning strategies to cover the typical non-IID data cases. Moreover, we conduct extensive experiments to evaluate state-of-the-art FL algorithms. We find that non-IID does bring significant challenges in learning accuracy of FL algorithms, and none of the existing state-of-the-art FL algorithms outperforms others in all cases. Our experiments provide insights for future studies of addressing the challenges in "data silos".

1 INTRODUCTION

In recent years we have seen fast development of machine learning in many fields including computer vision, natural language processing, recommendation system, data integration and self-learning systems. In data management systems, we have witnessed some promising advancement with leveraging machine learning services, such as learned index structures [9, 46] and learned cost estimation [21, 47]. As such, machine learning services have become emerging data-intensive workloads, such as Ease.ml [40], Machine Learning Bazaar [59] and Rafiki [63]. Those service systems have eased the user pain points from the perspective of machine learning and data management challenges. Despite the success of machine learning services, their effectiveness highly relies on large-volume high-quality training data. However, due to the increasing privacy concerns and data regulations such as GPDR [60], training data have been increasingly fragmented, forming distributed databases of multiple "data silos" (e.g., within different organizations and countries). Due to the deployed data regulations, raw data are usually not allowed to transfer across organizations/countries. For example, a multinational corporation (MNC) provides services to users in multiple nations, whose personal data usually cannot be centralized to a single country due to the data regulations in many countries. For another example, multiple hospitals cannot directly share their data with each other due to regulation policies, although ideally a better model could be obtained if training is performed on the data from all hospitals.

To develop effective machine learning services, it is necessary to exploit data from such distributed databases without exchanging the raw data. While there are many studies working on privacy-preserving data management and data mining [1, 27, 51, 54, 56] in a centralized setting, they cannot handle the cases of distributed databases. Thus, how to conduct data mining/machine learning from distributed databases without exchanging local data has become an emerging topic.

To address the above challenge, we borrow the federated learning (FL) [29, 37, 38, 66] approach from the machine learning community. Originally proposed by Google, FL is a promising solution to enable many parties jointly train a machine learning model while keeping their local data decentralized. Instead of exchanging data and conducting centralized training, each party sends its model to the server, which updates and sends back the global model to the parties in each round. Since their raw data are not exposed, FL is an effective way to address privacy concerns. It has attracted many research interests [7, 22, 31, 36, 39, 45, 64] and been widely used in practice [3, 20, 30]. Thus, we consider FL to develop machine learning services for distributed databases.

One key and common data challenge in such distributed databases is that data distributions in different parties are usually non-identically distributed (non-IID). For example, different areas can have very different disease distributions. Due to the ozone hole, the countries in the Southern Hemisphere may have more skin cancer patients than the Northern Hemisphere. In such a case, the label distributions differ across parties. Another example is that people have different writing styles even for the same world. In such a case, the feature distributions differ across parties. According to previous studies [25, 31, 41], the non-IID data settings can degrade the effectiveness of machine learning services.

*Equal contribution.

There have been some studies trying to develop effective FL algorithms under non-IID data including FedProx [39], SCAFFOLD [31], and FedNova [62]. However, there lacks an experimental study on systematically understanding their advantages and disadvantages, as the previous studies have very rigid data partitioning strategies among parties, which are hardly representative and thorough. In the experiments of these studies, they only try one or two partitioning strategies to simulate the non-IID data setting, which does not sufficiently cover different non-IID cases. For example, in FedAvg [48], each party only has samples of two classes. In FedNova [62], the number of samples of each class in each party follows Dirichlet distribution. The above partitioning strategies only cover the label skewed case. Thus, it is a necessity to evaluate those algorithms with a systematic exploration of different non-IID scenarios.

In this paper, we break the barrier of experiments on non-IID data distribution challenges in FL by proposing NIID-Bench. Specifically, we introduce six non-IID data partitioning strategies which thoroughly consider different cases including label distribution skew, feature distribution skew, and quantity skew. Moreover, we conduct extensive experiments on nine datasets to evaluate the accuracy of four state-of-the-art FL algorithms including FedAvg [48], FedProx [39], SCAFFOLD [31], and FedNova [62]. The experimental results provide insights for the future development of FL algorithms. Last, our code is publicly available [1]. Researchers can easily use our code to try different partitioning strategies for the evaluation of existing algorithms or a new algorithm. We also maintain a leaderboard along with our code to rank state-of-the-art federated learning algorithms on different non-IID settings, which can benefit the federated learning communicty a lot.

Through extensive studies, we have the following key findings. First, we find that non-IID does bring significant challenges in learning accuracy of FL algorithms, and none of the existing state-of-the-art FL algorithms outperforms others in all cases. Second, the effectiveness of FL is highly related to the kind of data skews, e.g., the label distribution skew setting is more challenging than the quantity skew setting. This indicates the importance of having a more comprehensive benchmark on non-IID distributions. Last, in non-IID data setting, instability of the learning process widely exists due to techniques such as batch normalization and partial sampling. This can severely hurt the effectiveness of machine learning services on distributed data silos. More advanced designs need to be investigated.

The contributions of this paper are summarized as follows.

- First, we identify non-IID data distributions as a key and common data challenge in developing effective machine learning services over distributed databases caused by increasing privacy concerns and data regulations.
- Second, we have developed a benchmark with non-IID data partitioning strategies, and conducted an extensive experimental study with four state-of-the-art algorithms, including FedAvg [48], FedProx [39], SCAFFOLD [31], and FedNova [62].
- Third, based on the findings, we discuss the insights and future directions for machine learning services in distributed databases.

[1] https://github.com/Xtra-Computing/NIID-Bench

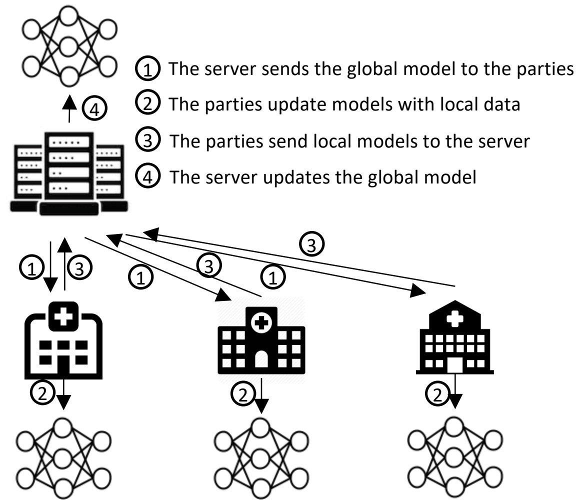


Figure 1: The FedAvg framework.

The remainder of this paper is structured as follows. We introduce the preliminaries in Section 2. We review FL algorithms handling non-IID data in Section 3, and present our non-IID data partition strategies in Section 4. Section 5 present the experimental results, followed by the future research directions in Section 6. We discuss the related work in Section 7, and conclude in Section 8.

2 PRELIMINARIES

2.1 Notations

Let $\mathcal{D} = \{(\mathbf{x}, y)\}$ denote the global dataset. Suppose there are N parties, denoted as $P_1, ..., P_N$. The local dataset of P_i is denoted as $\mathcal{D}^i = \{(\mathbf{x}_i, y_i)\}$, which form the distributed databases. We use w^t and w_i^t to denote the global model and the local model of party P_i in round t, respectively. Thus, w^t is the output model of the machine learn services built from the distributed databases.

2.2 FedAvg

FedAvg [48] has been a de facto approach for FL. The framework of FedAvg is shown in Figure 1. In each round, first, the server sends the global model to the selected parties. Second, each party updates the model with its local dataset. Then, the updated models are sent back to the server. Last, the server averages the received local models as the updated global model. Unlike traditional distributed SGD, the parties update their local model with multiple epochs, which can decrease the number of communication rounds and is much more communication-efficient. However, the local updates may lead to a bad accuracy, as shown in previous studies [25, 31, 41].

2.3 Effect of Non-IID Data

A key challenge in FL is the non-IID data among the parties [29, 37]. Non-IID data can influence the accuracy of FedAvg a lot. Since the distribution of each local dataset is highly different from the global distribution, the local objective of each party is inconsistent with the global optima. Thus, there exists a *drift* in the local updates [31]. The averaged model may also be far from the global optima

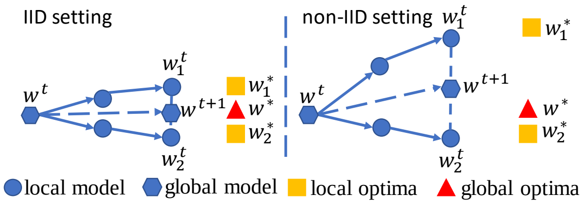


Figure 2: Example of a drift under the non-IID setting.

especially when the local updates are large (e.g., a large number of local epochs) [31, 39, 61, 62]. Eventually, the converged global model has much worse accuracy than IID setting. Figure 2 demonstrates the issue of FedAvg under the non-IID data setting. Under the IID setting, the global optima w^* is close to the local optima w_1^* and w_2^*. Thus, the averaged model w^{t+1} is also close to the global optima. However, under the non-IID setting, since w^* is far from w_1^*, w^{t+1} can be far from w^*. It is challenging to design an effective FL algorithm under the non-IID setting. We will present the FL algorithms on handling non-IID data in the next section.

3 FEDERATED LEARNING ALGORITHMS ON NON-IID DATA

There have been some studies [31, 39, 62] trying to address the drift issue in FL. Here we summary several state-of-the-art and popular approaches as shown in Algorithm 1 (FedAvg [48], FedProx [39], FedNova [62]) and Algorithm 2 (SCAFFOLD [31]). These approaches are all based on FedAvg, and we use colors to mark the parts that specially designed in FedProx (red), SCAFFOLD (blue), and FedNova (orange). Note that the studied approaches have the same objective, i.e., learning an effective global model under the non-IID data setting. There are also other FL studies related to non-IID data setting, such as personalizing the local models for each party [10, 12, 19] and designing robust algorithms against different combinations of local distributions [8, 49, 53], which are out of the scope of this paper.

3.1 FedProx

FedProx [39] improves the local objective based on FedAvg. It directly limits the size of local updates. Specifically, as shown in Line 14 of Algorithm 1, it introduces an additional L_2 regularization term in the local objective function to limit the distance between the local model and the global model. This is a straightforward way to limit the local updates so that the averaged model is not so far from the global optima. A hyper-parameter μ is introduced to control the weight of the L_2 regularization. Overall, the modification to FedAvg is lightweight and easy to implement. FedProx only introduces little computation overhead and does not introduce additional communication overhead. However, one drawback is that users may need to carefully tune μ to achieve good accuracy. If μ is too small, then the regularization term has almost no effect. If μ is too big, then the local updates are very small and the convergence speed is slow.

3.2 FedNova

Another recent study, FedNova [62], improves FedAvg in the aggregation stage. It considers that different parties may conduct

Algorithm 1: A summary of FL algorithms including FedAvg/FedProx/FedNova. We use red and orange colors to mark the part specially included in FedProx and FedNova, respectively.

Input: local datasets $\mathcal{D}^i$, number of parties N, number of communication rounds T, number of local epochs E, learning rate η
Output: The final model w^T

1 **Server executes**:
2 initialize x^0
3 **for** $t = 0, 1, ..., T-1$ **do**
4   Sample a set of parties S_t
5   $n \leftarrow \sum_{i \in S_t} |\mathcal{D}^i|$
6   **for** $i \in S_t$ **in parallel do**
7   send the global model w^t to party P_i
8     $\Delta w_i^t, \tau_i \leftarrow$ **LocalTraining**(i, w^t)
9   For FedAvg/FedProx: $w^{t+1} \leftarrow w^t - \eta \sum_{i \in S_t} \frac{|\mathcal{D}^i|}{n} \Delta w_k^t$
10   For FedNova: $w^{t+1} \leftarrow w^t - \eta \frac{\sum_{i \in S_t} |\mathcal{D}^i| \tau_i}{n} \sum_{i \in S_t} \frac{|D^i| \Delta w_i^t}{n \tau_i}$
11 return w^T
12 **Party executes**:
13 For FedAvg/FedNova: $L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y)$
14 For FedProx: $L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y) + \frac{\mu}{2} \left\| w - w^t \right\|^2$
15 **LocalTraining**(i, w^t):
16 $w_i^t \leftarrow w^t$
17 $\tau_i \leftarrow 0$
18 **for** epoch $k = 1, 2, ..., E$ **do**
19   **for** each batch $\mathbf{b} = \{\mathbf{x}, y\}$ of $\mathcal{D}^i$ **do**
20     $w_i^t \leftarrow w_i^t - \eta \nabla L(w_i^t; \mathbf{b})$
21     $\tau_i \leftarrow \tau_i + 1$
22 $\Delta w_i^t \leftarrow w^t - w_i^t$
23 return $\Delta w_i^t, \tau_i$ to the server

different numbers of local steps (i.e., the number of mini-batches in the local training) each round. This can happen when parties have different computation power given the same time constraint or parties have different local dataset size given the same number of local epochs and batch size. Intuitively, the parties with a larger number of local steps will have a larger local update, which will have a more significant influence on the global updates if simply averaged. Thus, to ensure that the global updates are not biased, FedNova normalizes and scales the local updates of each party according to their number of local steps before updating the global model (see Line 10 of Algorithm 1). FedNova also only introduces lightweight modifications to FedAvg, and little additional computation overhead when updating the global model.

3.3 SCAFFOLD

SCAFFOLD [31] models non-IID as introducing variance among the parties and applies the variance reduction technique [28, 55]. It introduces control variates for the server (i.e., c) and parties (i.e.,

Algorithm 2: The SCAFFOLD algorithm. We use blue color to mark the part specially included in SCAFFOLD compared with FedAvg.

Input: same as Algorithm 1
Output: The final model w^T

1 **Server executes**:
2 initialize x^0
3 $c^t \leftarrow \mathbf{0}$
4 **for** $t = 0, 1, ..., T-1$ **do**
5 Sample a set of parties S_t
6 $n \leftarrow \sum_{i \in S_t} |\mathcal{D}^i|$
7 **for** $i \in S_t$ **in parallel do**
8 send the global model w^t to party P_i $\Delta w_i^t, \Delta c \leftarrow$ **LocalTraining**(i, w^t, c^t)
9 $w^{t+1} \leftarrow w^t - \eta \sum_{i \in S_t} \frac{|\mathcal{D}^i|}{n} \Delta w_k^t$
10 $c^{t+1} \leftarrow c^t + \frac{1}{N} \Delta c$
11 return w^T
12 **Party executes**:
13 $L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y)$
14 $c_i \leftarrow \mathbf{0}$
15 **LocalTraining**(i, w^t, c^t):
16 $w_i^t \leftarrow w^t$
17 $\tau_i \leftarrow 0$
18 **for** epoch $k = 1, 2, ..., E$ **do**
19 **for** each batch $\mathbf{b} = \{\mathbf{x}, y\}$ of $\mathcal{D}^i$ **do**
20 $w_i^t \leftarrow w_i^t - \eta(\nabla L(w_i^t; \mathbf{b}) - c_i^t + c)$
21 $\tau_i \leftarrow \tau_i + 1$
22 $\Delta w_i^t \leftarrow w^t - w_i^t$
23 $c_i^* \leftarrow (i) \nabla L(w_i^t), or (ii) c_i - c + \frac{1}{\tau_i \eta}(w^t - w_i^t)$
24 $\Delta c \leftarrow c_i^* - c_i$
25 $c_i \leftarrow c_i^*$
26 return Δw_i^t, Δc to the server

c_i), which are used to estimate the update direction of the server model and the update direction of each client. Then, the drift of local training is approximated by the difference between these two update directions. Thus, SCAFFOLD corrects the local updates by adding the drift in the local training (Line 20 of Algorithm 2). SCAFFOLD proposes two approaches to update the local control variates (Line 23 of Algorithm 2), by computing the gradient of the local data at the global model or by reusing the previously computed gradients. The second approach has a lower computation cost while the first one may be more stable. Compared with FedAvg, intuitively, SCAFFOLD doubles the communication size per round due to the additional control variates.

3.4 Motivation of this study

Non-IID is a key and common data challenge for developing effective machine learning services in distributed databases. Although previous studies [31, 39, 62] have demonstrated preliminary and promising results over FedAvg on non-IID data, as we will summarize in Table 1 in later section, all above studies have evaluated only one or two non-IID distributions, and tried rigid data partitioning strategies in the experiments. There is still no standard benchmark or a systematic study to evaluate the effectiveness of these FL algorithms. This motivates us to develop a benchmark with more comprehensive data distributions as well as data partitioning strategies, and then we can evaluate the pros and cons of existing algorithms and outline the challenges and opportunities for future machine learning services in non-IID data.

4 SIMULATING NON-IID DATA SETTING

As existing studies only adopt limited partitioning strategies, they cannot represent a comprehensive view of non-IID cases. To bridge this gap, we develop a benchmark named NIID-Bench. Specifically, we need to address two key problems. The first one is on data sets: whether to use real-world non-IID datasets or synthetic datasets. The second one is on how to design comprehensive non-IID scenarios.

For the first problem, we choose to synthetic the distributed non-IID datasets by partitioning a real-world dataset into multiple smaller subsets. Many existing studies [31, 48, 62] use the partitioning approach to simulate the non-IID federated setting. Compared with using real federated datasets [4, 26], adopting partitioning strategies has the following advantages. First, while it is hard to evaluate the imbalance level in real federated datasets, partitioning strategies can easily quantify and control the imbalance level of the local data. Second, partitioning strategies can easily set different numbers of parties to simulate different scenarios, but a real federated dataset usually has a fixed number of data resources. Last, due to the data regulation and privacy concerns, appropriate real federated datasets may not be publicly available [26]. It is more flexible to develop partitioning strategies on existing widely used public datasets, which already have lots of centralized training knowledge as reference, as well as to simulate different non-IID scenarios.

For the second problem, our partitioning strategies are inspired by an existing study [29], which gives a very good and comprehensive summary on non-IID data distributions. Specifically, the article summaries five different cases for non-IID data distributions: (1) label distribution skew; (2) feature distribution skew; (3) same label but different features; (4) same features but different labels; (5) quantity skew. Here the third case is mainly related to vertical FL (the parties share the same sample IDs but different features). We focus on horizontal FL in this paper, where each party shares the same feature space but owns different samples. The fourth case is not applicable in most FL studies, which assume there is a common knowledge $P(y|x)$ among the parties to learn. Thus, we consider label distribution skew, feature distribution skew, and quantity skew as possible non-IID data distribution cases in this paper. Next, we introduce ways to simulate strategies for each case.

4.1 Label Distribution Skew

In label distribution skew, the label distributions $P(y_i)$ vary across parties. Such a case is common in practice. For example, some hospitals are more specialized in several specific kinds of diseases and have more patient records on them. To simulate label distribution

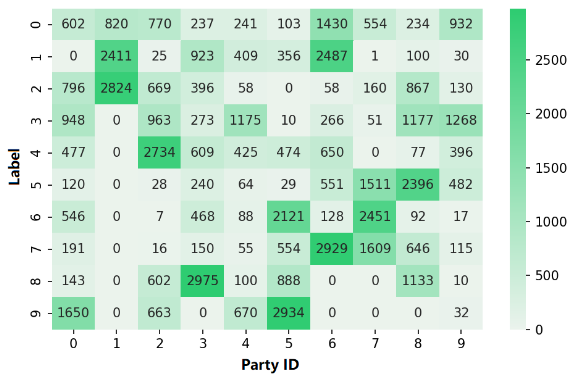


Figure 3: An example of distribution-based label imbalance partition on MNIST [34] dataset with $\beta = 0.5$. The value in each rectangle is the number of data samples of a class belonging to a certain party.

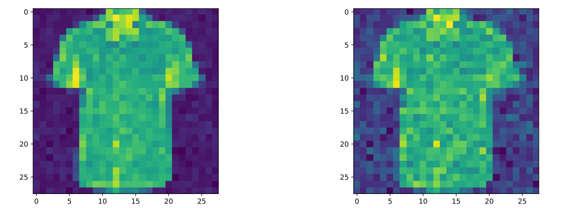
skew, we introduce two different label imbalance settings: quantity-based label imbalance and distribution-based label imbalance.

Quantity-based label imbalance. Here each party owns data samples of a fixed number of labels. This is first introduced in the experiments of FedAvg [48], where the data samples with the same label are divided into subsets and each party is only assigned 2 subsets with different labels. Following FedAvg, such a setting is also used in many other studies [16, 39]. [13] considers a highly extreme case, where each party only has data samples with a single label. We introduce a general partitioning strategy to set the number of labels that each party has. Suppose each party only has data samples of k different labels. We first randomly assign k different label IDs to each party. Then, for the samples of each label, we randomly and equally divide them into the parties which own the label. In this way, the number of labels in each party is fixed, and there is no overlap between the samples of different parties. For ease of presentation, we use $\#C = k$ to denote such a partitioning strategy.

Distribution-based label imbalance. Another way to simulate label imbalance is that each party is allocated a proportion of the samples of each label according to Dirichlet distribution. Specifically, we sample $p_k \sim Dir_N(\beta)$ and allocate a $p_{k,j}$ proportion of the instances of class k to party j. Here $Dir(\cdot)$ denotes the Dirichlet distribution and β is a concentration parameter ($\beta > 0$). This partitioning strategy was first used in [67] and has been used in many recent studies [35, 43, 61, 62]. An advantage of this approach is that we can flexibly change the imbalance level by varying the concentration parameter β. If β is set to a smaller value, then the partition is more unbalanced. An example of such a partitioning strategy is shown in Figure 3. For ease of presentation, we use $p_k \sim Dir(\beta)$ to denote such a partitioning strategy.

4.2 Feature Distribution Skew

In feature distribution skew, the feature distributions $P(x_i)$ vary across parties although the knowledge $P(y_i|\mathbf{x}_i)$ is same. For example, cats may vary in coat colors and patterns in different areas.



(a) add noises from $Gau(0.001)$ **(b) add noises from $Gau(0.01)$**

Figure 4: An example of adding noises on FMNIST [65] dataset. On party P_1, noises sampled from $Gau(0.001)$ are added into its images. On party P_2, noises sampled from $Gau(0.01)$ are added into its images.

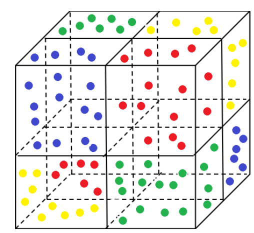


Figure 5: The visualization of our FCUBE dataset. The data points within the upper four cubes have label 0 and within the lower four cubes have label 1. There are a total of eight cubes with four colors. The data points with the same color are assigned to a party.

Here we introduce three different settings to simulate feature distribution skew: noise-based feature imbalance, synthetic feature imbalance, and real-world feature imbalance.

Noise-based feature imbalance. We first divide the whole dataset into multiple parties randomly and equally. For each party, we add different levels of Gaussian noise to its local dataset to achieve different feature distributions. Specifically, given user-defined noise level σ, we add noises $\hat{\mathbf{x}} \sim Gau(\sigma \cdot i/N)$ for Party P_i, where $Gau(\sigma \cdot i/N)$ is a Gaussian distribution with mean 0 and variance $\sigma \cdot i/N$. Users can change σ to increase the feature dissimilarity among the parties. Figure 4 is an example of noise-based feature imbalance on FMNIST dataset [65]. For ease of presentation, we use $\hat{\mathbf{x}} \sim Gau(\sigma)$ to present such a partitioning strategy.

Synthetic feature imbalance. We generate a synthetic feature imbalance federated dataset named FCUBE. Suppose the distribution of data points is a cube in three dimensions (i.e, (x_1, x_2, x_3)) which have two different labels classified by plane $x_1 = 0$. As shown in Figure 5, we divide the cube into 8 parts by planes $x_1 = 0$, $x_2 = 0$, and $x_3 = 0$. Then, we allocate two parts which are symmetric of (0,0,0) to a subset for each party. In this way, feature distribution varies among parties while labels are still balanced.

Real-world feature imbalance. The EMNIST dataset [6] collects handwritten characters/digits from different writers. Then, like [4],

Table 1: The experimental settings in existing studies and our benchmark.

Partitioning strategies		FedAvg	FedProx	SCAFFOLD	FedNova	NIID-Bench
Label distribution skew	quantity-based	✓	✓	✗	✗	✓
	distribution-based	✗	✗	✓	✓	✓
Feature distribution skew	noise-based	✗	✗	✗	✗	✓
	synthetic	✗	✓	✗	✗	✓
	real-world	✗	✓	✗	✗	✓
Quantity skew		✗	✗	✗	✓	✓

it is natural to partition the dataset into different parties according to the writers. Since the character features usually differ among writers (e.g, stroke width, slant), there is a natural feature distribution skew among different parties. Specifically, for the digit images of EMNIST, we divide and assign the writers (and their digits) into each party randomly and equally. Since each party has different writers, the feature distributions are different among the parties. Like [4], we call this federated dataset as FEMNIST.

4.3 Quantity Skew

In quantity skew, the size of the local dataset $|\mathcal{D}^i|$ varies across parties. Although data distribution may still be consistent among the parties, it is interesting to see the effect of the data quantity of each party in FL. Like distribution-based label imbalance setting, we use Dirichlet distribution to allocate different amounts of data samples into each party. We sample $q \sim Dir_N(\beta)$ and allocate a q_j proportion of the total data samples to party P_j. The parameter β can be used to control the imbalance level of the quantity skew. For ease of presentation, we use $q \sim Dir(\beta)$ to denote such a partitioning strategy.

4.4 Summary

Table 1 compares the partitioning strategies in NIID-bench with the experimental settings in existing studies. We can observe that each study only covers partial non-IID cases. Moreover, FedAvg and SCAFFOLD only try a single partitioning strategy. It is impossible to directly compare the results presented in different papers. In contrast, NIID-bench consists of six partitioning strategies, which are more comprehensive and representative for representing different non-IID distributed databases.

5 EXPERIMENTS

To investigate the effectiveness of existing FL algorithms on non-IID data setting, we conduct extensive experiments on nine public datasets, including six image datasets (i.e., MNIST [34], CIFAR-10 [32], FMNIST [65], SVHN [50], FCUBE, FEMNIST [4]) and three tabular datasets (i.e., adult, rcv1, and covtype)[2]. The statistics of the datasets are summarized in Table 2. For the image datasets, we use a CNN, which has two 5x5 convolution layers followed by 2x2 max pooling (the first with 6 channels and the second with 16 channels) and two fully connected layers with ReLU activation (the first with 120 units and the second with 84 units). For the tabular datasets, we use a MLP with three hidden layers. The numbers of hidden units of three layers are 32, 16, and 8. The number of parties is set to 10 by default, except for FCUBE where the number of parties is set to 4. All parties participate in every round to eliminate the effect of randomness brought by party sampling by default [48]. We use the SGD optimizer with learning rate 0.1 for rcv1 and learning rate 0.01 for the other datasets (tuned from $\{0.1, 0.01, 0.001\}$) and momentum 0.9. The batch size is set to 64 and the number of local epochs is set to 10 by default.

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Table 2: The statistics of datasets in the experiments.

Datasets	#training instances	#test instances	#features	#classes
MNIST	60,000	10,000	784	10
FMNIST	60,000	10,000	784	10
CIFAR-10	50,000	10,000	1,024	10
SVHN	73,257	26,032	1,024	10
adult	32,561	16,281	123	2
rcv1	15,182	5,060	47,236	2
covtype	435,759	145,253	54	2
FCUBE	4,000	1,000	3	2
FEMNIST	341,873	40,832	784	10

Benchmark metrics. We use the top-1 accuracy on the test dataset as a metric to compare the studied algorithms. We run all the studied algorithms for the same number of rounds for fair comparison. Specifically, for the experiments in Section 5.1 to Section 5.4, Section 5.5, and Section 5.6, we run the studies algorithms for 50, 100, and 500 rounds, respectively.

5.1 Overall Accuracy Comparison

The accuracy of existing approaches including FedAvg, FedProx, SCAFFOLD, and FedNova under different non-IID data settings is shown in Table 3. For comparison, we also present the results for IID scenarios (i.e., homogeneous partitions). Next we show the insights from different perspectives.

5.1.1 Comparison among different non-IID settings.

Finding (1): The label distribution skew case where each party only has samples of a single class is the most challenging setting, while the feature distribution skew and quantity skew setting have little influence on the accuracy of FedAvg.

From Table 3, we can observe that there is a gap between the accuracy of existing algorithms on several non-IID data settings and on the homogeneous setting. First, among different non-IID data settings, all studied FL algorithms perform worse on the label distribution skew case. Second, in label distribution skew setting, the algorithms have the worst accuracy when each party only has data from a single label. As expected, the accuracy increases as the number of classes in each party increases. Third, for feature distribution skew setting, except for CIFAR-10, existing algorithms

Table 3: The top-1 accuracy of different approaches. We run three trials and report the mean accuracy and standard derivation. For FedProx, we tune μ from $\{0.001, 0.01, 0.1, 1\}$ and report the best accuracy.

category	dataset	partitioning	FedAvg	FedProx	SCAFFOLD	FedNova
Label distribution skew	MNIST	$p_k \sim Dir(0.5)$	98.9%±0.1%	98.9%±0.1%	**99.0%±0.1%**	**99.0%±0.1%**
		$\#C = 1$	29.8%±7.9%	**40.9%±23.1%**	9.9%±0.2%	31.6%±21.2%
		$\#C = 2$	**97.0%±0.4%**	96.4%±0.3%	95.9%±0.3%	96.8%±0.6%
		$\#C = 3$	**98.0%±0.2%**	97.9%±0.4%	96.6%±1.5%	**98.0%±0.4%**
	FMNIST	$p_k \sim Dir(0.5)$	88.1%±0.6%	88.1%±0.9%	**88.4%±0.5%**	88.1%±0.7%
		$\#C = 1$	11.2%±2.0%	**28.9%±3.9%**	12.8%±4.8%	12.6%±4.5%
		$\#C = 2$	**77.3%±4.9%**	74.9%±2.6%	42.8%±28.7%	72.8%±4.1%
		$\#C = 3$	80.7%±1.9%	**82.5%±1.9%**	77.7%±3.8%	81.0%±2.8%
	CIFAR-10	$p_k \sim Dir(0.5)$	68.2%±0.7%	67.9%±0.7%	**69.8%±0.7%**	68.0%±0.3%
		$\#C = 1$	10.0%±0.0%	**12.3%±2.0%**	10.0%±0.0%	10.0%±0.0%
		$\#C = 2$	49.8%±3.3%	**50.7%±1.7%**	49.1%±1.7%	48.9%±1.8%
		$\#C = 3$	**58.3%±1.2%**	57.1%±1.2%	57.8%±1.4%	55.8%±2.3%
	SVHN	$p_k \sim Dir(0.5)$	86.1%±0.7%	86.6%±0.9%	**86.8%±0.3%**	86.3%±0.6%
		$\#C = 1$	11.1%±0.0%	**19.6%±0.0%**	6.7%±0.0%	9.6%±2.5%
		$\#C = 2$	**80.2%±0.8%**	79.3%±0.9%	62.7%±11.6%	72.0%±6.2%
		$\#C = 3$	82.0%±0.7%	82.1%±1.0%	77.2%±2.0%	**82.2%±0.3%**
	adult	$p_k \sim Dir(0.5)$	78.4%±0.9%	**80.5%±0.7%**	76.4%±0.0%	62.0%±7.9%
		$\#C = 1$	**82.5%±2.2%**	76.4%±0.0%	23.6%±0.0%	51.6%±1.5%
	rcv1	$p_k \sim Dir(0.5)$	48.2%±0.7%	**70.3%±13.3%**	64.4%±24.3%	63.6%±26.0%
		$\#C = 1$	**51.8%±0.7%**	**51.8%±0.7%**	**51.8%±0.7%**	**51.8%±0.7%**
	covtype	$p_k \sim Dir(0.5)$	**77.2%±7.4%**	70.9%±0.7%	67.7%±14.9%	79.0%±5.9%
		$\#C = 1$	48.8%±0.1%	**59.1%±2.1%**	49.6%±1.4%	50.4%±1.4%
number of times that performs best			8	10	5	4
Feature distribution skew	MNIST	$\hat{\mathbf{x}} \sim Gau(0.1)$	**99.1%±0.1%**	**99.1%±0.1%**	**99.1%±0.1%**	**99.1%±0.1%**
	FMNIST		89.1%±0.3%	89.0%±0.2%	**89.3%±0.0%**	89.2%±0.1%
	CIFAR-10		68.9%±0.3%	69.3%±0.2%	**70.1%±0.2%**	70.0%±0.2%
	SVHN		**88.1%±0.5%**	**88.1%±0.2%**	**88.1%±0.4%**	**88.1%±0.5%**
	FCUBE	synthetic	**99.8%±0.2%**	**99.8%±0.0%**	99.7%±0.3%	99.7%±0.1%
	FEMNIST	real-world	**99.4%±0.0%**	99.3%±0.1%	**99.4%±0.1%**	99.3%±0.1%
number of times that performs best			4	3	5	2
Quantity skew	MNIST	$q \sim Dir(0.5)$	**99.2%±0.1%**	**99.2%±0.1%**	99.1%±0.1%	99.1%±0.1%
	FMNIST		89.4%±0.1%	**89.7%±0.3%**	88.8%±0.4%	87.0%±3.2%
	CIFAR-10		**72.0%±0.3%**	71.2%±0.6%	62.4%±4.1%	24.4%±15.3%
	SVHN		88.3%±1.0%	**88.4%±0.4%**	11.0%±7.4%	50.9%±29.9%
	adult		82.2%±0.1%	**84.8%±0.2%**	81.6%±4.5%	55.3%±23.6%
	rcv1		96.7%±0.3%	**96.8%±0.4%**	49.0%±1.9%	65.1%±27.4%
	covtype		**88.1%±0.2%**	84.6%±0.2%	63.2%±20.8%	54.2%±2.6%
number of times that performs best			3	5	0	0
Homogeneous partition	MNIST	IID	99.1%±0.1%	99.1%±0.1%	**99.2%±0.0%**	99.1%±0.0%
	FMNIST		89.6%±0.3%	89.5%±0.2%	**89.7%±0.2%**	89.6%±0.2%
	CIFAR-10		70.4%±0.2%	70.2%±0.1%	**71.5%±0.3%**	70.8%±0.1%
	SVHN		**88.5%±0.5%**	**88.5%±0.8%**	88.0%±0.8%	**88.5%±0.5%**
	FCUBE		99.7%±0.1%	99.6%±0.2%	99.8%±0.1%	**99.9%±0.1%**
	FEMNIST		99.3%±0.1%	**99.4%±0.1%**	**99.4%±0.0%**	99.3%±0.0%
	adult		82.6%±0.4%	**84.8%±0.2%**	83.8%±2.5%	82.6%±0.2%
	rcv1		**96.8%±0.4%**	96.6%±0.6%	80.9%±27.8%	96.6%±0.4%
	covtype		87.9%±0.1%	85.2%±0.0%	**88.0%±2.3%**	**88.0%±0.3%**
number of times that performs best			2	3	5	3

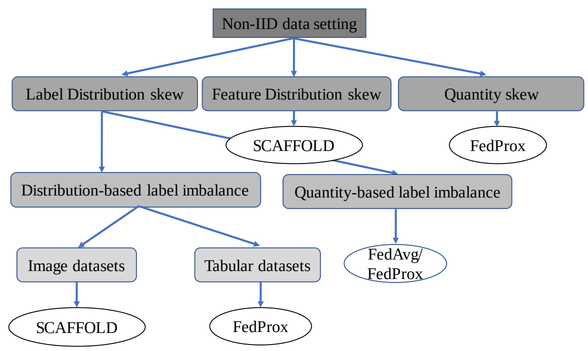


Figure 6: The decision tree to determine the (almost) best FL algorithm given the non-IID setting.

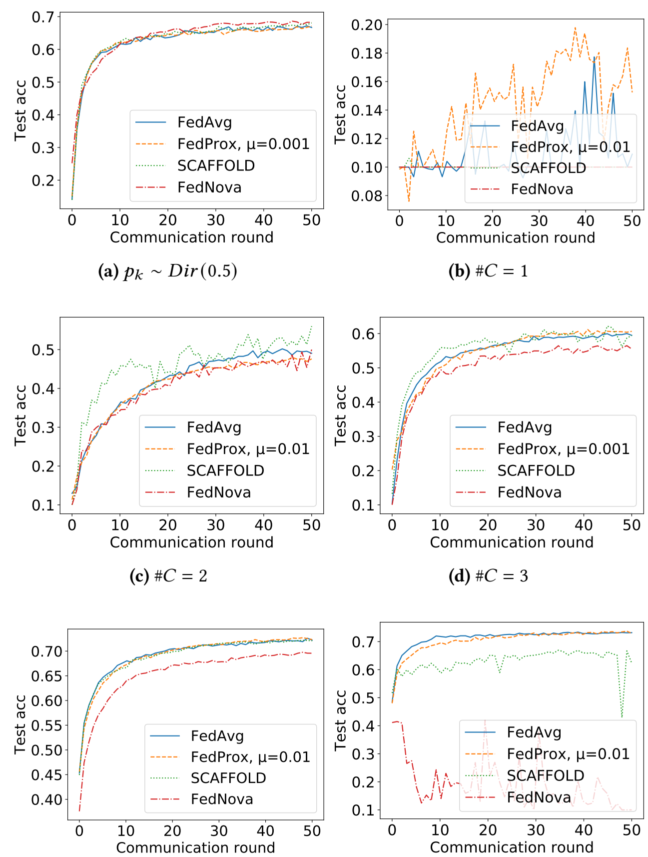
have a very close accuracy compared with the IID setting. Last, in quantity skew setting, FedAvg has almost no accuracy loss. Since the weighted averaging is adopted in FedAvg, it can already handle the quantity imbalance well. Overall, the label distribution skew influences the accuracy of FL algorithms most among all non-IID settings. There is room for existing algorithms to be improved to handle scenarios such as quantity-based label imbalance.

We draw a decision tree to summarize the suitable FL algorithm for each non-IID setting as shown in Figure 6 according to our observations. This decision tree is helpful for users to choose the algorithm for their learning according to the non-IID distribution and the datasets. For example, if the local datasets are likely to have feature distribution skew (e.g., the digits from different writers), then SCAFFOLD may be the best algorithm for FL. If the local datasets have almost the same data distribution but different sizes (e.g., databases with different capacities), then FedProx is likely the appropriate algorithm. If there is no prior knowledge on the local datasets, how to determine the distribution is a challenging problem and more research efforts are needed (see Section 6.1).

5.1.2 Comparison among different algorithms.

Finding (2): No algorithm consistently outperforms the other algorithms in all settings. The state-of-the-art algorithms significantly outperform FedAvg only in several cases.

We have the following observations in aspect of different algorithms. First, in label distribution skew and quantity skew cases, FedProx usually achieves the best accuracy. In feature distribution skew case, SCAFFOLD usually achieves the best accuracy. Second, in some cases (e.g., $p_k \sim Dir(0.5)$, feature distribution skew and quantity skew), the improvement of the three non-IID FL algorithms is insignificant compared with FedAvg, which is smaller than 1%. Third, when $\#C = 1$, FedProx can significantly outperform FedAvg, SCAFFOLD and FedNova. Fourth, for SCAFFOLD, its accuracy is quite unstable. It can significantly outperform the other two approaches in some cases (e.g., $Dir(0.5)$ and $K = 1$ on CIFAR-10). However, it may also have much worse accuracy than the other two approaches (e.g., $K = 1$ and $K = 2$ on SVHN). Last, for FedNova, it does not show much superiority compared with other FL algorithms. Compared with the accuracy of FedAvg on the homogeneous partition, there is still a lot of room for improvement in the non-IID setting.



(a) $p_k \sim Dir(0.5)$ **(b)** $\#C = 1$

(c) $\#C = 2$ **(d)** $\#C = 3$

(e) $\hat{\mathbf{x}} \sim Gau(0.1)$ **(f)** $q \sim Dir(0.5)$

Figure 7: The training curves of different approaches on CIFAR-10.

5.1.3 Comparison among different tasks.

Finding (3): CIFAR-10 and tabular datasets are challenging tasks under non-IID settings. MNIST is a simple task under most non-IID settings where the studied algorithms perform similarly well.

Among nine different datasets, while heterogeneity significantly degrades the accuracy of FL algorithms on CIFAR-10 and tabular datasets, such influence is smaller in other datasets. Among image datasets, the classification task on CIFAR-10 is more complex than the other datasets in a centralized setting. Thus, when each party only has a skewed subset, the task will be more challenging and the accuracy is worse. Also, it is interesting that all the four algorithms cannot handle tabular datasets well in the non-IID setting. The accuracy loss is quite large especially for the label distribution skew case. We suggest that the challenging tasks like CIFAR-10 and rcv1 should be included in the benchmark for distributed data silos.

5.2 Communication Efficiency

Finding (4): FedProx has almost the same convergence speed compared with FedAvg, while SCAFFOLD and FedNova are more unstable in training.

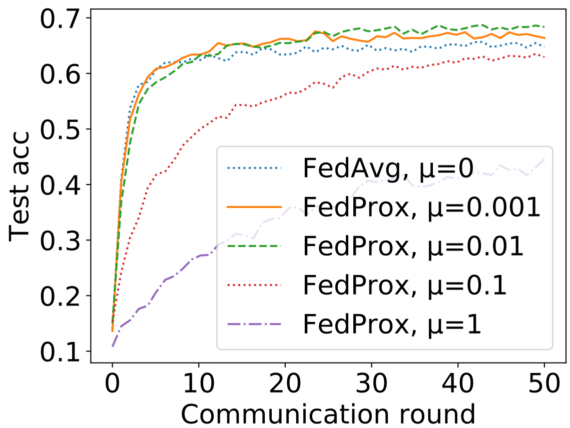


Figure 8: The training curves of FedProx with different μ on CIFAR-10 dataset of $p_k \sim Dir(0.5)$ partition.

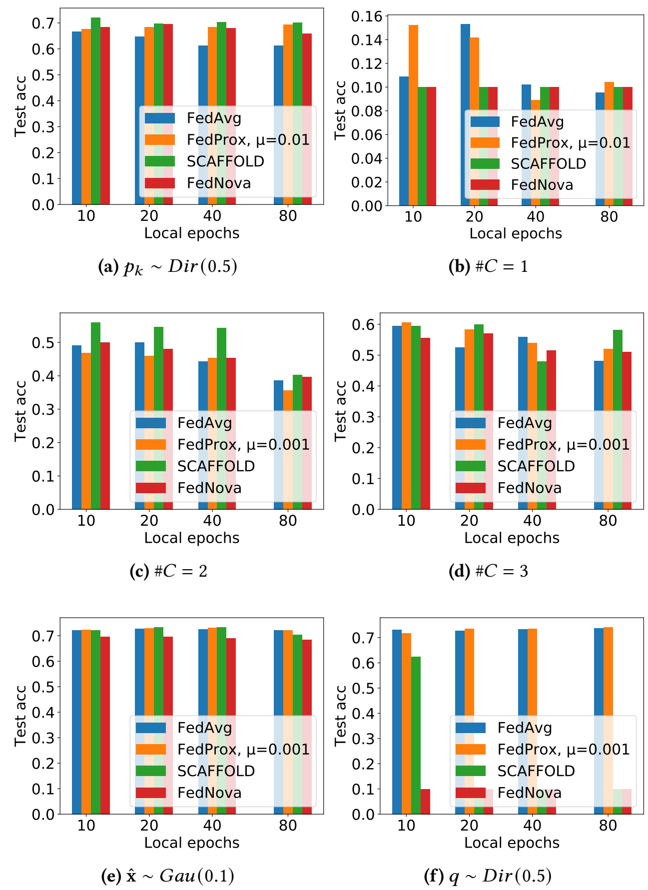
Figure 7 shows the training curves of the studied algorithms on CIFAR-10. For FedProx, we show the curve with the best μ. First, for the $\#C = 1$ setting, FedAvg and FedProx are very unstable, while SCAFFOLD and FedNova even cannot improve as the number of rounds increases. Second, for the $q \sim Dir(0.5)$ setting, FedNova is quite unstable and the accuracy changes rapidly as the number of communication rounds increases. Moreover, FedProx is very close to FedAvg during the whole training process in many cases. Since the best μ is always small, the regularization term in FedProx has little influence on the training. Thus, FedProx and FedAvg usually have similar convergence speed and final accuracy. How to achieve stable learning and fast convergence is still an open problem on non-IID data.

To further understand FedProx, which appears to have a more stable accuracy than SCAFFOLD and FedNova, we vary the hyper-parameter μ in FedProx. The hyper-parameter μ to control the weight of the regularization term. Intuitively, a big μ will slowdown the training process as it limits the size of local updates. A small μ may have little influence on the training as the regularization term can be quite small. In Figure 8, we show the training curves of FedProx with $\mu \in \{0, 0.001, 0.01, 0.1, 1\}$ on CIFAR-10. We can observe that training with a larger μ is indeed slower. However, the final test accuracy may also benefit from a larger μ. Comparing the training curves of $\mu = 0.01$ and $\mu = 0.001$, although the training of $\mu = 0.01$ is slower than $\mu = 0.001$, it has a better accuracy than $\mu = 0.001$ eventually.

5.3 Robustness to Local Updates

Finding (5): The number of local epochs can have a large effect on the accuracy of existing algorithms. The optimal value of the number of local epochs is very sensitive to non-IID distributions.

We vary the number of local epochs from $\{10, 20, 40, 80\}$ and report the final accuracy in Figure 9. On the one hand, we can find that the number of local epochs has a large effect on the accuracy of FL algorithms. For example, when $\#C = 2$, the accuracy of all algorithms generally degrades significantly when the number of



(a) $p_k \sim Dir(0.5)$ (b) $\#C = 1$ (c) $\#C = 2$ (d) $\#C = 3$ (e) $\hat{\mathbf{x}} \sim Gau(0.1)$ (f) $q \sim Dir(0.5)$

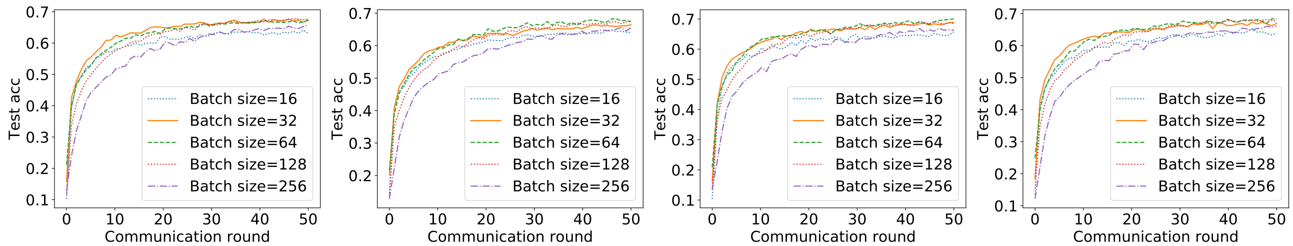
Figure 9: The test accuracy with different numbers of local epochs on CIFAR-10.

local epochs is set to 80. On the other hand, the optimal number of local epochs differ in different settings. For example, when $\#C = 1$ and $\#C = 2$, the optimal number of local epochs is 20 for FedAvg, and is 10 on the settings $p_k \sim Dir(0.5)$ and $\#C = 3$. In summary, existing algorithms are not robust enough against large local updates. Non-IID distributions have to be considered to determine the best number of local epochs.

5.4 Batch Size

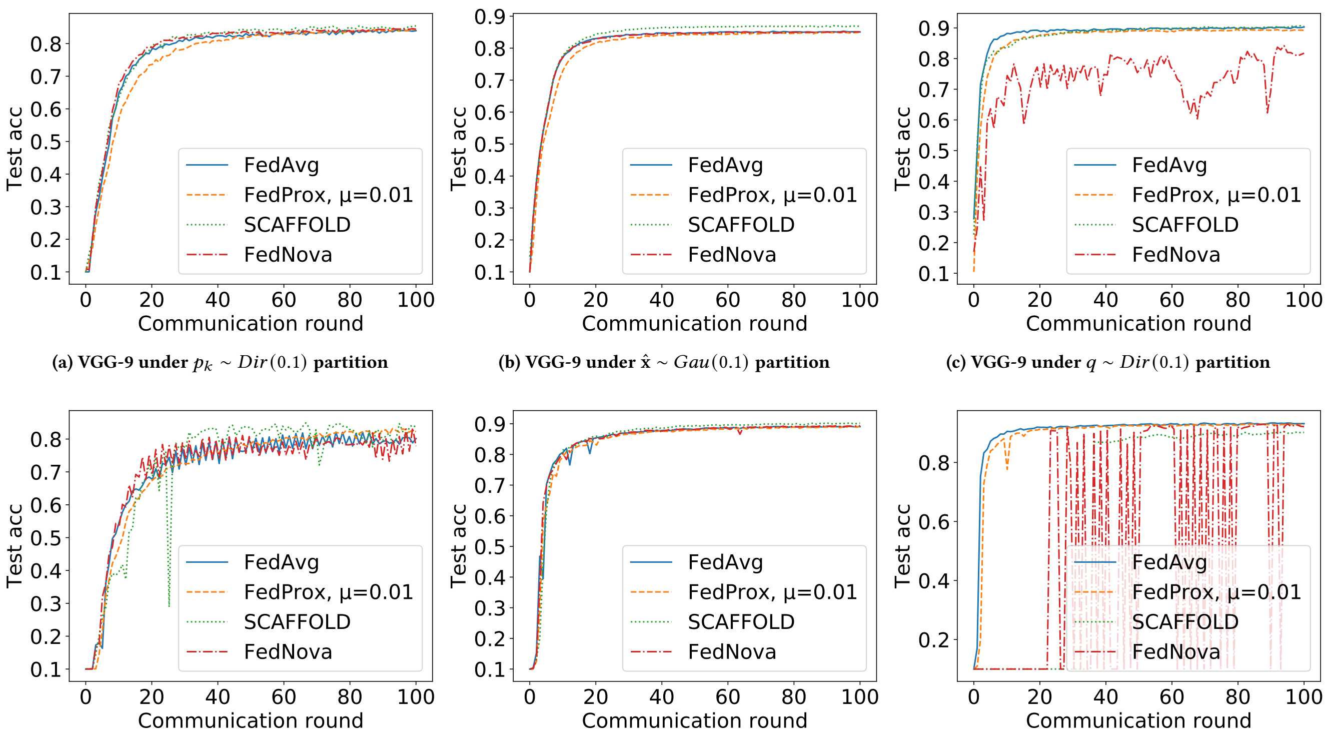
Finding (6): The heterogeneity of local data does not appear to influence the behaviors of different choices of batch sizes.

Batch size is an important hyper-parameter in deep learning. We study the effect of batch size in FL by varying it from 16 to 256 as shown in Figure 10. Like centralized training, a large batch size slows down the learning process. Moreover, four studied algorithms have similar behaviours given different batch sizes. The results demonstrate that there is no clear relationship between the setting of batch size and the heterogeneity of local data. The knowledge of the behaviors of different batch sizes still applies in the non-IID federated setting.



(a) FedAvg **(b) FedProx ($\mu = 0.01$)** **(c) SCAFFOLD** **(d) FedNova**

Figure 10: The training curves of different batch sizes on CIFAR-10 under $p_k \sim Dir(0.5)$ partition.



(a) VGG-9 under $p_k \sim Dir(0.1)$ partition **(b) VGG-9 under $\hat{\mathbf{x}} \sim Gau(0.1)$ partition** **(c) VGG-9 under $q \sim Dir(0.1)$ partition**

(d) ResNet-50 under $p_k \sim Dir(0.1)$ partition **(e) ResNet-50 under $\hat{\mathbf{x}} \sim Gau(0.1)$ partition** **(f) ResNet-50 under $q \sim Dir(0.1)$ partition**

Figure 11: The training curves of VGG-9/ResNet-50 on CIFAR-10 under different partitions.

5.5 Model Architectures

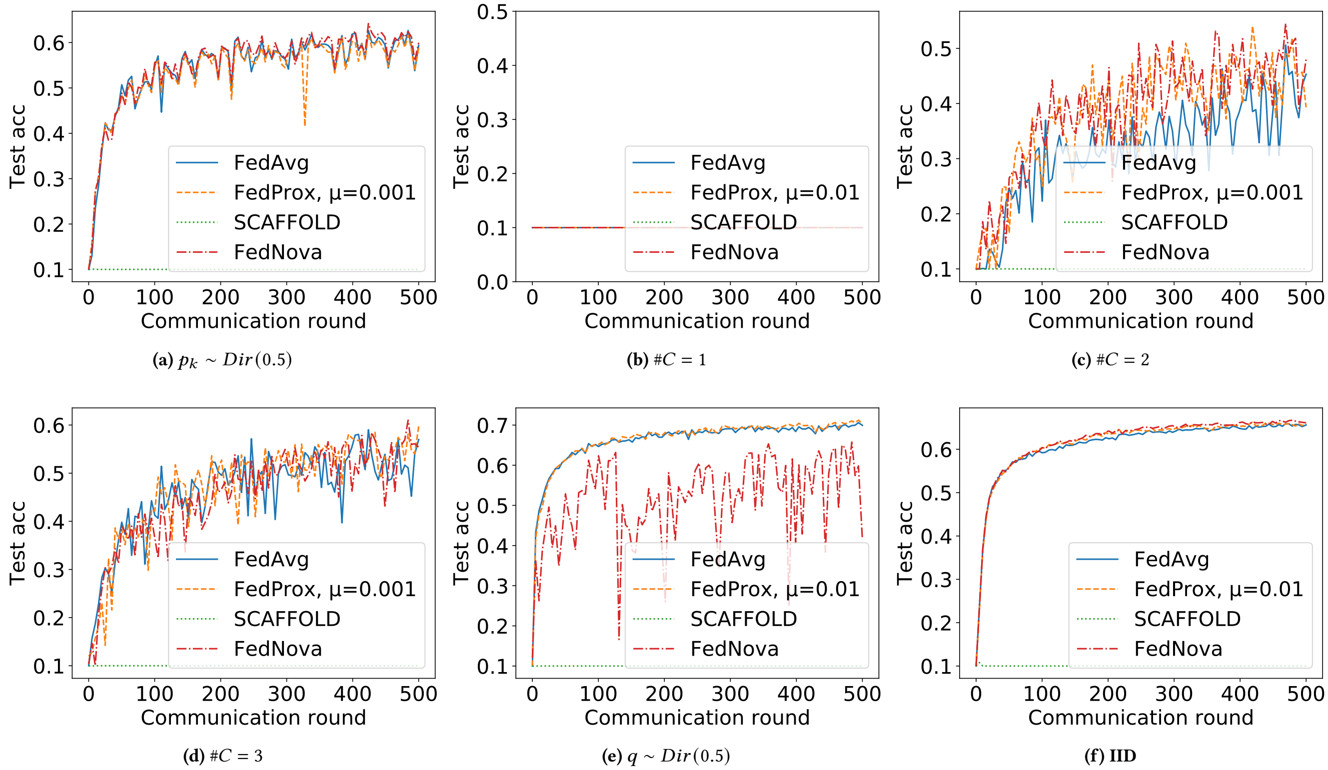
Finding (7): A simple averaging of batch normalization layers introduces instability in non-IID setting.

In the previous experiments, the models we use are simple CNNs and MLPs. Here we try more complex models including VGG-9 and ResNet-50 [24]. The experimental results on CIFAR-10 are shown in Figure 11. Overall, while the final accuracies of using VGG-9 and ResNet-50 are usually close, training a ResNet-50 appears to more unstable. ResNet-50 uses batch normalization to standardize the inputs to a layer. A challenge in training ResNet-50 is how to aggregate the batch normalization layers. While the local batch normalization layers can handle the local distribution well, a simple averaging of these layers may not be able to catch the statistics of global distribution and introduces more instability.

5.6 Scalability

Finding (8): In the partial participation setting, SCAFFOLD cannot work effectively, while the other FL algorithms have a very unstable accuracy during training.

In some scenarios, not all the data silos will participate the entire training process. In such a setting, the sampling technique is usually applied (Line 6 of Algorithm 1). To simulate this scenario, we set the number of parties to 100 and the sample fraction to 0.1. We run experiments on CIFAR-10 and the results are shown in Figure



(a) $p_k \sim Dir(0.5)$ (b) #$C = 1$ (c) #$C = 2$

(d) #$C = 3$ (e) $q \sim Dir(0.5)$ (f) IID

Figure 12: The training curves of different approaches on CIFAR-10 with 100 parties and sample fraction 0.1.

12. We can find that the training curves are quite unstable in most non-IID settings, which further demonstrates the effect of non-IID data compared with Figure 12f. Due to the sampling technique, the local distributions among different rounds can vary, and thus the averaged gradients may have very different directions among rounds. Moreover, we can find that SCAFFOLD has a bad accuracy on all settings. Since the frequency of updating local control variates (Lines 23-25 of Algorithm 2) is low, the estimation of the update direction may be very inaccurate using the control variates.

6 FUTURE DIRECTIONS

From our experimental study, we identify the following promising future directions for data management and federated learning on non-IID distributed databases.

6.1 Opportunities for data management

Integration with learned database systems: Existing learned systems are mostly based on centralized databases, such as learned index structures [9, 46] and learned cost estimation [21, 47]. We believe that, as the concerns on data privacy and data regulation grow, we will see more distributed databases and existing learned systems and algorithms need to be revisited. For example, it could be very interesting to enable federated search and develop learned index structures for multiple "data silos" without exchanging the local data.

Light-weight data techniques for profiling non-IID data: From our experimental study, different non-IID distributions have a large effect on the accuracy and stability of FL algorithms. Thus, it would be helpful if we can know the non-IID distribution in prior before conducting FL. This made a decade of database research relevant, such as data sampling [5] and sketching [17]. Another potential approach is to be made non-IID distributions in the meta data. However, it is still an open problem on how to extend current statistics estimation (such as cardinality estimation) to non-IID distribution.
Non-IID resistant sampling for partial participation: As in Finding (8), the sampling approach can bring instability in FL. Instead of random sampling, selective sampling according to the data distribution features of the parties may significantly increase the learning stability. One inspiration is from the skew resistant data techniques [15, 33], which can be potentially extended to the partial participation in FL training.
Privacy-preserving data mining: Although there is no raw data transfer in FL, the model may still leak sensitive information about the training data due to possible inference attacks [14, 58]. Thus, techniques such as differential privacy [11] are useful to protect the local databases. How to decrease the accuracy loss while ensuring the differential privacy guarantee is a challenge research direction.
Query on Federated Databases: As we focus on distributed databases due to privacy concerns, federated databases [57] also need to be revisited. On the one hand, how to combine the SQL query with machine learning on federated databases is an important problem.

On the other hand, how to preserve the data privacy while supporting both query and learning on federated databases also needs to be investigated.

6.2 Opportunities for better FL design

A Party with a Single Label: From Table 3, the accuracy of FL algorithms is very bad if each party only has data of a single label. This setting is seemingly unrealistic. However, it has many real-world applications in practice. For example, we can use FL to train a speaker recognition model, while each mobile device only has the voices of its single user.

Fast Training: From Figure 7, the training speed of existing FL algorithms are usually close to each other. FedProx, SCAFFOLD, and FedNova do not show much superiority on the communication efficiency. To improve the training speed, researchers can work on the following two directions. One possible solution is to develop communication-efficient FL algorithms with only a few rounds. There are some studies [18, 35] that propose FL algorithms using a single communication round. In their studies, a public dataset is needed, which may potentially limit the applications. Another possible solution is to develop fast initialization approach to reduce the number of rounds while achieving the same accuracy for FL. In the experiments of a previous study [35], they show that their approach is also promising if applied as an initialization step.

Automated Parameter Tuning for FL: FL algorithms suffer from large local updates. The number of local epochs is an important parameter in FL. While one traditional way is to develop approaches robustness to the local updates, another way is to design efficient parameter tuning approaches for FL. A previous paper [7] studies Bayesian optimization in the federated setting, which can be used to search hyper-parameters. Approaches for the setting of number of local epochs need to be investigated.

Towards Robust Algorithms against Different Non-IID Settings: As in Finding (2), no algorithm consistently performs the best in all settings. It is a natural question whether and how we can develop a robust algorithm for different non-IID settings. We may have to first investigate the common characteristics of FL processes under different non-IID settings. The intuitions of existing algorithms are same: the local model updates towards the local optima, and the averaged model is far from the global optima. We believe the design of FL algorithms under non-IID settings can be improved if we can observe more detailed and common behaviours in the training.

Aggregation of Heterogeneous Batch Normalization: From our Finding (7), simple averaging is not a good choice for batch normalization. Since the batch normalization in each party records the statistics of local data distribution, there is also heterogeneity among the batch normalization layers of different parties. The averaged batch normalization layer may not catch the local distribution after sending back to the parties. A possible solution is to only average the learned parameters but leave the statistics (i.e., mean and variance) alone [2]. More specialized designs for particular layers in deep learning need to be investigated.

7 RELATED WORK

There are some existing benchmarks for federated learning [4, 23, 26, 42]. LEAF [4] provides some realistic federated datasets including images and texts. Specifically, LEAF partitions the existing datasets according to its data recourses, e.g., partitioning the data in Extended MNIST [6] based on the writer of the digit or character. OARF [26] proposes federated datasets by combining multiple related real-world public datasets. Moreover, it provides various metrics including utility, communication overhead, privacy loss, and mimics the federated systems in the real world. However, both LEAF and OARF do not provide an algorithm-level comparison. FedML [23] provides reference implementations of federated learning algorithms such as FedAvg, FedNOVA [62] and FedOpt [52]. It integrates many datasets such as FEMNIST and federated Shakespeare [4]. However, it does not provide a systematic summary of the non-IID partitioning. FLBench [42] is proposed for isolated data island scenario. Its framework covers domains including medical, finance, and AIoT. However, currently, FLBench is not open-sourced and it does not provide any experiments.

The above benchmarks do not provide analysis of existing federated learning algorithms on different non-IID settings, which is our focus in this paper. To the best of our knowledge, there is one existing benchmark [44] for the non-IID data setting. However, it only provides two partitioning approaches: random split and split by labels. In this paper, we provide comprehensive partitioning strategies and datasets to cover different non-IID settings.

8 CONCLUSION

As the increasing concerns on data privacy and regulations, there has been a growing interest in exploiting distributed databases (e.g., in different organizations and countries) to improve the effectiveness of machine learning services. In this paper, we study non-IID data as one key challenge in such distributed databases, and develop a benchmark named NIID-bench. Specifically, we introduce six data partitioning strategies which are much more comprehensive than the previous studies. Furthermore, we conduct comprehensive experiments to compare existing algorithms and demonstrate their strength and weakness. We show that none of those approaches can outperform others consistently. This experimental study is a good starting point for building effective machine learning services on distributed databases, and we need more efforts from the data management and machine learning communities to address the challenges from data privacy and regulation constraints.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 2000. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 439–450.

[2] Mathieu Andreux, Jean Ogier du Terrail, Constance Beguier, and Eric W Tramel. 2020. Siloed Federated Learning for Multi-Centric Histopathology Datasets. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Springer, 129–139.

[3] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé M Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards Federated Learning at Scale: System Design. In *SysML*. https://arxiv.org/abs/1902.01046

[4] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097* (2018).

[5] Surajit Chaudhuri, Rajeev Motwani, and Vivek Narasayya. 1998. Random sampling for histogram construction: How much is enough? *ACM SIGMOD Record* 27, 2 (1998), 436–447.

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

[7] Zhongxiang Dai, Bryan Kian Hsiang Low, and Patrick Jaillet. 2020. Federated Bayesian optimization via Thompson sampling. *Advances in Neural Information Processing Systems* 33 (2020).

[8] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2020. Distributionally Robust Federated Averaging. *Advances in Neural Information Processing Systems* 33 (2020).

[9] Jialin Ding, Umar Farooq Minhas, Jia Yu, Chi Wang, Jaeyoung Do, Yinan Li, Hantian Zhang, Badrish Chandramouli, Johannes Gehrke, Donald Kossmann, David Lomet, and Tim Kraska. 2020. ALEX: An Updatable Adaptive Learned Index. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 969–984. https://doi.org/10.1145/3318464.3389711

[10] Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. 2020. Personalized federated learning with Moreau envelopes. *Advances in Neural Information Processing Systems* (2020).

[11] Cynthia Dwork. 2011. Differential privacy. *Encyclopedia of Cryptography and Security* (2011), 338–340.

[12] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. *Advances in Neural Information Processing Systems* 33 (2020).

[13] X Yu Felix, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2020. Federated Learning with Only Positive Labels. *arXiv preprint arXiv:2004.10342* (2020).

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1322–1333.

[15] Sumit Ganguly, Phillip B. Gibbons, Yossi Matias, and Avi Silberschatz. 1996. Bifocal Sampling for Skew-Resistant Join Size Estimation. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (Montreal, Quebec, Canada) *(SIGMOD '96)*. Association for Computing Machinery, New York, NY, USA, 271–281. https://doi.org/10.1145/233269.233340

[16] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).

[17] Anna C Gilbert, Sudipto Guha, Piotr Indyk, Yannis Kotidis, Sivaramakrishnan Muthukrishnan, and Martin J Strauss. 2002. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. 389–398.

[18] Neel Guha, Ameet Talwlkar, and Virginia Smith. 2019. One-shot federated learning. *arXiv preprint arXiv:1902.11175* (2019).

[19] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. 2020. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems* (2020).

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

[21] Shohedul Hasan, Saravanan Thirumuruganathan, Jees Augustine, Nick Koudas, and Gautam Das. 2020. Deep Learning Models for Selectivity Estimation of Multi-Attribute Queries. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1035–1050. https://doi.org/10.1145/3318464.3389741

[22] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge. *Advances in Neural Information Processing Systems* 33 (2020).

[23] Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, et al. 2020. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518* (2020).

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

[25] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335* (2019).

[26] Sixu Hu, Yuan Li, Xu Liu, Qinbin Li, Zhaomin Wu, and Bingsheng He. 2020. The oarf benchmark suite: Characterization and implications for federated learning systems. *arXiv preprint arXiv:2006.07856* (2020).

[27] Nick Hynes, David Dao, David Yan, Raymond Cheng, and Dawn Song. 2018. A demonstration of sterling: A privacy-preserving data marketplace. *Proceedings of the VLDB Endowment* 11, 12 (2018), 2086–2089.

[28] Rie Johnson and Tong Zhang. 2013. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems* 26 (2013), 315–323.

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

[30] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. 2020. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence* (2020), 1–7.

[31] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for on-device federated learning. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR.

[32] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[33] YongChul Kwon, Magdalena Balazinska, Bill Howe, and Jerome Rolia. 2010. Skew-Resistant Parallel Processing of Feature-Extracting Scientific User-Defined Functions. In *Proceedings of the 1st ACM Symposium on Cloud Computing* (Indianapolis, Indiana, USA) *(SoCC '10)*. Association for Computing Machinery, New York, NY, USA, 75–86. https://doi.org/10.1145/1807128.1807140

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

[35] Qinbin Li, Bingsheng He, and Dawn Song. 2020. Model-Agnostic Round-Optimal Federated Learning via Knowledge Transfer. *arXiv preprint arXiv:2010.01017* (2020).

[36] Qinbin Li, Zeyi Wen, and Bingsheng He. 2020. Practical Federated Gradient Boosting Decision Trees.. In *AAAI*. 4642–4649.

[37] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, and Bingsheng He. 2019. A Survey on Federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693* (2019).

[38] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2019. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873* (2019).

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

[40] Tian Li, Jie Zhong, Ji Liu, Wentao Wu, and Ce Zhang. 2018. Ease.Ml: Towards Multi-Tenant Resource Sharing for Machine Learning Workloads. 11, 5 (Jan. 2018), 607–620. https://doi.org/10.1145/3187009.3177737

[41] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJxNAnVtDS

[42] Yuan Liang, Yange Guo, Yanxia Gong, Chunjie Luo, Jianfeng Zhan, and Yunyou Huang. 2020. An isolated data island benchmark suite for federated learning. *arXiv preprint arXiv:2008.07257* (2020).

[43] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble Distillation for Robust Model Fusion in Federated Learning. *Advances in Neural Information Processing Systems* 33 (2020).

[44] Lifeng Liu, Fengda Zhang, Jun Xiao, and Chao Wu. 2020. Evaluation Framework For Large-scale Federated Learning. *arXiv preprint arXiv:2003.01575* (2020).

[45] Yang Liu, Yan Kang, Chaoping Xing, Tianjian Chen, and Qiang Yang. 2020. A Secure Federated Transfer Learning Framework. *IEEE Intelligent Systems* (2020).

[46] Ryan Marcus, Andreas Kipf, Alexander van Renen, Mihail Stoian, Sanchit Misra, Alfons Kemper, Thomas Neumann, and Tim Kraska. 2020. Benchmarking Learned Indexes. *Proc. VLDB Endow.* 14, 1 (Sept. 2020), 1–13. https://doi.org/10.14778/3421424.3421425

[47] Ryan Marcus, Parimarjan Negi, Hongzi Mao, Chi Zhang, Mohammad Alizadeh, Tim Kraska, Olga Papaemmanouil, and Nesime Tatbul. 2019. Neo: A Learned Query Optimizer. *Proc. VLDB Endow.* 12, 11 (July 2019), 1705–1718. https://doi.org/10.14778/3342263.3342644

[48] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629* (2016).

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

[50] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. (2011).

[51] Chaoyue Niu, Zhenzhe Zheng, Fan Wu, Xiaofeng Gao, and Guihai Chen. 2017. Trading data in good faith: Integrating truthfulness and privacy preservation in data markets. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 223–226.

[52] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive Federated Optimization. *arXiv preprint arXiv:2003.00295* (2020).

[53] Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. 2020. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Processing Systems* (2020).

[54] Shariq J Rizvi and Jayant R Haritsa. 2002. Maintaining data privacy in association rule mining. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 682–693.

[55] Mark Schmidt, Nicolas Le Roux, and Francis Bach. 2017. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming* 162, 1-2 (2017), 83–112.

[56] Supreeth Shastri, Vinay Banakar, Melissa Wasserman, Arun Kumar, and Vijay Chidambaram. 2020. Understanding and Benchmarking the Impact of GDPR on Database Systems. *Proc. VLDB Endow.* 13, 7 (March 2020), 1064–1077. https://doi.org/10.14778/3384345.3384354

[57] Amit P Sheth and James A Larson. 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)* 22, 3 (1990), 183–236.

[58] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.

[59] Micah J. Smith, Carles Sala, James Max Kanter, and Kalyan Veeramachaneni. 2020. The Machine Learning Bazaar: Harnessing the ML Ecosystem for Effective System Development. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 785–800.

[60] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* (2017).

[61] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated Learning with Matched Averaging. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BkluqlSFDS

[62] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems* 33 (2020).

[63] Wei Wang, Jinyang Gao, Meihui Zhang, Sheng Wang, Gang Chen, Teck Khim Ng, Beng Chin Ooi, Jie Shao, and Moaz Reyad. 2018. Rafiki: Machine Learning as an Analytics Service System. *Proc. VLDB Endow.* 12, 2 (Oct. 2018), 128–140. https://doi.org/10.14778/3282495.3282499

[64] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. 2020. Privacy preserving vertical federated learning for tree-based models. *Proceedings of the VLDB Endowment* (2020).

[65] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[66] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[67] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian Nonparametric Federated Learning of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR.