Improving FedAvg Algorithm Performance with Non-IID data

Harsh Lahoti, Gaurav Dalvi, Aryan Gupta

1 ABSTRACT

Federated learning allows us to train accurate shared models for resource-constrained devices such as mobile phones and IoT devices. This decentralized paradigm offers significant advantages regarding privacy, security, regulatory compliance, and economic efficiency. While the classic implementation of the FedAvg algorithm works well with IID data, the accuracy reduces drastically, up to ∼40%, when trained on non-IID data. This paper reviews the performance of the classic FedAvg algorithm on well-known datasets while exploring and implementing modifications to the algorithm to improve performance when training models on various non-IID data distributions.

2 INTRODUCTION

Mobile devices have recently become the primary computing resource for a large part of the population, and billions of IoT-enabled devices are set to come online. The data these devices generate is undoubtedly precious, and training effective models on this data is crucial. However, the classic approach of training models on data collected from devices raises concerns over privacy and security. Federated learning solves these problems by training shared models on resource-constrained devices.

Based on stochastic gradient descent (SGD), Federated learning introduced by McMahan et al. [1] has demonstrated empirical solid performance in training deep neural networks. One notable benefit of this approach lies in the separation of model training from the necessity of direct access to the original training data. The working of this approach heavily relies on the training data having an independent and identically distributed (IID) dataset, which helps ensure that the gradient calculate do n the mini-batch is a good approximation of the gradient calculated on the entire dataset. The data on client devices though may not necessarily be IID, which introduces difficulties in training an accurate model.

In section 5, we introduce 'n-class' non-IID data to study the effect of imbalanced data of varying degrees. We observe a significant decline in the accuracy of the CNN trained using the FedAvg algorithm when confronted with highly skewed non-IID data. Specifically, on MNIST datasets, we noted a reduction of approximately 30%, while on CIFAR-10 datasets, the decrease reached as high as 75%.

Addressing this challenge, Section 5 analyzes a data-sharing strategy introduced in [2] to enhance FedAvg's performance with non-IID data. This strategy involves distributing a limited amount of globally shared data, comprising examples from each class. Even though our goal is decentralization of training models, implementing this solution would require keeping a central IID dataset.

Additionally, we explore centrally training a warmup model and distributing these pre-trained weights to the clients. Our experiments demonstrate promising results: on MNIST, we achieve a 20% increase in accuracy, while on CIFAR-10, the improvement reaches 45%.

3 THE PROBLEM WITH NON-IID DATA

In this section, we illustrate the decrease in accuracy when employing FedAvg on non-IID data compared to an IID distribution. We will use MNIST and CIFAR-10 to benchmark our observations; both are image classification datasets with ten output classes.
We have used a CNN comprising two convolutional layers followed by max pooling and ReLU activation for MNIST.
For CIFAR-10, we employed a CNN consisting of four convolutional layers with batch normalization and ReLU activation, interspersed with max-pooling layers. The final layers include two fully connected layers with ReLU activation and dropout.

3.1 Experimental Setup

We evenly distribute the training sets across 20 clients to simulate a distributed learning environment.
In the IID setting, a uniform distribution is assigned to each client over the ten classes, ensuring an equal representation of each class across the clients.
Contrastingly, in the Non-IID setting, we introduce variability by specifying the number of classes "n", which each client will receive. Subsequently, we select "n" classes from the ten available classes for each client, tailoring the data distribution for that client to be specific to its chosen classes.

- *Harsh, Gaurav, and Aryan are with the Department of Computer Science and Engineering, Indian Institute of Technology (Banaras Hindu University), Varanasi 221005 India. {w}@itbhu.ac.in*

This setup allows us to explore the impact of data distribution heterogeneity on model performance and robustness in federated learning scenarios.

The parameter notations utilized for FedAvg are standardized across experiments, as [1].
We have set the following parameters for our federated learning setup:

- Batch size: 10
- Local epochs: 1
- Frequency of central updates/synchronization: 1 per client round (for both datasets)

For the **MNIST** dataset:

- Learning rate: 0.01
- Number of client rounds: 10 (due to its quick convergence)

For the **CIFAR-10** dataset:

- Learning rate: 0.1
- Number of client rounds: 30

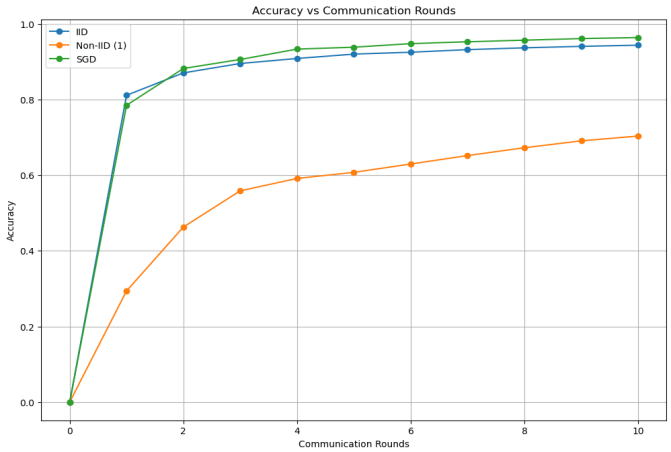
Each dataset's learning rates are optimized individually to enhance convergence and performance. In contrast, the learning rate remains constant across datasets for the Stochastic Gradient Descent (SGD) model.
However, the batch size (B) has been adjusted to be greater than that of FedAvg. This adjustment compensates for the difference in synchronization mechanisms between FedAvg and SGD: FedAvg averages the global model across multiple clients at each synchronization step. These changes ensure a fair comparison when comparing FedAvg with IID data to SGD with shuffled data.

Algorithm 1 Federated Averaging (FedAvg)

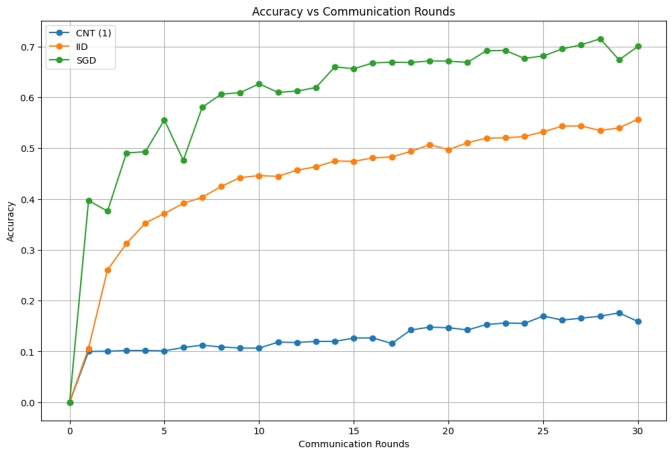
Input: Global model parameters θ, client datasets $\{D_i\}_{i=1}^{N}$, number of communication rounds R
Initialize: θ_0 = Central Params
$t = 1$ to R
  $\theta_t^i \leftarrow$ ClientUpdate(D_i, θ_{t-1}) for all clients $i = 1$ to N
  $\theta_t \leftarrow \frac{1}{N} \sum_{i=1}^{N} n^{(i)} \theta_t^i$ {Aggregate client updates}
Output: Final global model parameters θ_R

Algorithm 2 ClientUpdate(D_i, θ)

Input: Local dataset D_i, current model parameters θ
Initialize: Local model parameters θ_i
$\theta_i \leftarrow \theta$
$t = 1$ to E
  Randomly shuffle D_i
  For each mini-batch (x, y) in D_i
  $\theta_i \leftarrow \theta_i - \eta \cdot \nabla \ell(f_{\theta_i}(x), y)$ {Update local model}
Output: Updated local model parameters θ_i



(a) MNIST



(b) Cifar-10

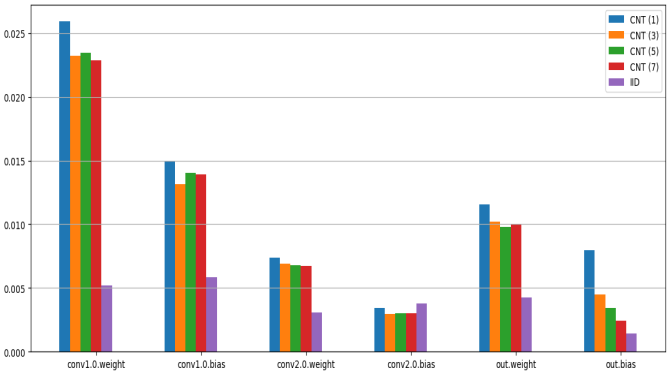
Fig. 1: Test Accuracy

3.2 Experimental Observations

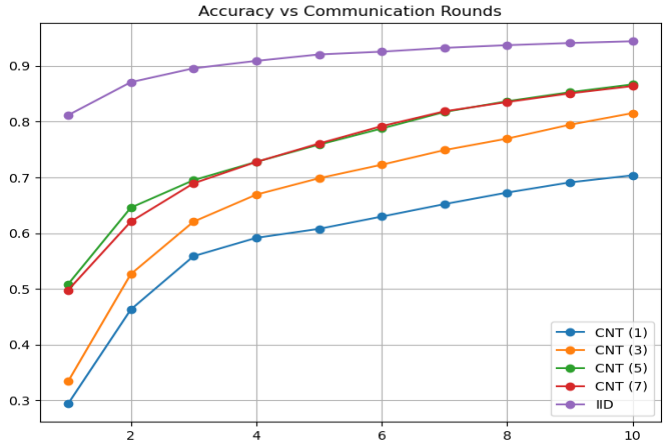
In the IID experiments, FedAvg exhibits convergence curves that significantly coincide with those of SGD across both datasets, resulting in comparable levels of test accuracy. This outcome is consistent with our anticipated findings, [1].
However, the test accuracy drastically reduces when using a non-IID setting. This is likely due to the fact of the diverging weight updates on each client when training on an imbalanced dataset.
Refer Fig. 1

4 Weight Divergence as a Reliable Measure

SGD optimizes models by iteratively updating parameters using gradients computed from local data samples. In non-IID settings, the diversity of data distributions across devices leads to divergent model updates. Consequently, models trained on different devices may exhibit significant discrepancies in learned parameters. Despite aggregation, the models trained on such heterogeneous data distributions may produce conflicting updates, resulting in suboptimal global models.
In this section, we will show that the accuracy of the FedAvg model is closely related to weight divergence between the global dataset and the dataset on the client devices.

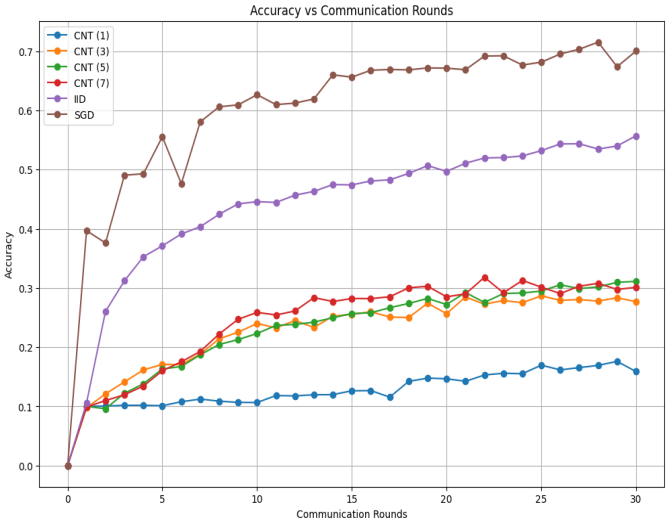


(a) Weight Divergence across different Non-IID settings



(b) Accuracy across different Non-IID settings

Fig. 2: MNIST dataset



(a) Accuracy across different Non-IID settings

Fig. 3: CIFAR-10 dataset

4.1 Logical Arguments

Comparing FedAvg with SGD regarding test accuracy relies heavily on the trained weights. Hence, an excellent approach to view this comparison involves examining the difference in weights relative to those produced by SGD under identical weight initialization. This measure is termed "weight divergence" and can be computed using the following equation:

$$\text{Weight Divergence of layer} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(w_{\text{SGD}}^{(i)} - w_{\text{FEDAVG}}^{(i)}\right)^2} \tag{1}$$

where N represents the total weights in the layer.

Increased weight divergence correlates with decreased model performance. This can be attributed to the fact that weight divergence signifies significant deviations in model parameters across devices, indicating a lack of model convergence. As a result, the aggregated global model may fail to capture important patterns present in the data distribution, leading to reduced accuracy.
Another interpretation,
let $L(\theta)$ denote the loss function associated with the learning task. The expected loss of the global model parameter $\bar{\theta}$ is given by:

$$\mathbb{E}[L(\bar{\theta})] = \int_{\mathcal{X}} L(\bar{\theta}; x)p(x)dx \tag{2}$$

where $\mathcal{X}$ is the input space, and $p(x)$ is the data distribution. When weight divergence increases, the global model parameter $\bar{\theta}$ may deviate significantly from the optimal parameters that minimize $\mathbb{E}[L(\theta)]$, leading to higher expected loss.

4.2 Experimental Validation

An intriguing observation arises: the reduction in performance is more apparent for the (1)-class non-IID data than the (3)-class non-IID data. This justifies that the accuracy of FedAvg might be influenced by the specific data distribution of the clients, particularly its skewness.
The mathematical proof detailing this assertion is available in Section 3.1 of the referenced paper [2]. However, we shall rely on an essential observation stated within the same paper.
Assuming that the Fedavg synchronization is conducted every T steps, then $w_{mT}^{(f)}$ represents the weights calculated after the m-th synchronization step. It is given by the weighted average as follows

$$w_{mT}^{(f)} = \sum_{i=1}^{C} \frac{n^{(i)} w_{(mT)}^{(i)}}{\sum_{j=1}^{C} n^{(j)}} \tag{3}$$

Remark 1
The weight divergence after the m-th synchronization mainly comes from two parts, including the weight divergence after the $(m-1)$-th divergence, i.e.,
$||w_{(m-1)T}^{(f)} - w_{(m-1)T}^{(c)}||$, and the weight divergence induced by the probability distance for the data distribution on client k compared with the actual distribution for the whole population, i.e., $\sum_{i=1}^{C} ||p^{(k)}(y=i) - p(y=i)||$.

(4)

Figure 2 shows that the weight divergence across all layers escalates as the data transitions from IID to progressively

more non-IID states, i.e., from IID towards (1)-class Non-IID. The plot confirms a correlation between weight divergence and the skewness of the data.
Hence, we can interpret the observed reduction in accuracy, discussed in Section 3, through weight divergence, which measures the weight discrepancy between two distinct training processes initiated with the same weight initialization.

5 Exploring Possible Solutions

5.1 Motivation

The test accuracy experiences a notable decline as the weight divergence surpasses a specific threshold. Consequently, for datasets with highly skewed and non-IID data, a potential exists to substantially enhance test accuracy by marginally reducing EMD (Earth Mover's Distance; quantifies weight divergence). Given the lack of control over clients' data, a viable approach is sharing a small subset of global data featuring a uniform class distribution from the cloud to the clients. This strategy seamlessly aligns with the initialization phase of a conventional federated learning setup.
Furthermore, instead of deploying a model with randomized weights, a pre-trained warm-up model can be developed using the globally shared data and then dispatched to the clients. Using the globally shared data can mitigate EMD for the clients, thereby anticipating improved test accuracy.

5.2 Data-Sharing Strategy

A centrally stored dataset G is established in the cloud in the federated learning framework, characterized by a uniform distribution across classes. During the initialization phase of Federated Averaging (FedAvg), a randomly selected portion α of this dataset G is allocated to each participating client. Subsequently, each client's local model undergoes training utilizing the shared data from G and the private data unique to each client.
When dealing with non-IID data, the model may become biased towards certain client distributions. Introducing a common baseline across clients by sharing a subset of IID data helps reduce bias by providing each client with a representative sample of the overall data distribution.
But this comes at the cost of increased resource requirements for maintaining and distributing the larger dataset G.

5.3 Pretrained-Model distribution

A pre-trained global model gives a solid starting point for training models on each client's data, notably when it deviates largely from the standard IID expectations. Moreover, the pre-trained global model distribution serves as an effective regularization tool, by limiting how much local model parameters can change when trained. This solution helps in preventing overfitting and ensures that the model works well across different client datasets, thus making it more robust and easier to scale in federated learning setups.

5.4 Experimental Results

As anticipated, the accuracy plots illustrate a significant improvement when sharing a small subset of IID data(α=10%)

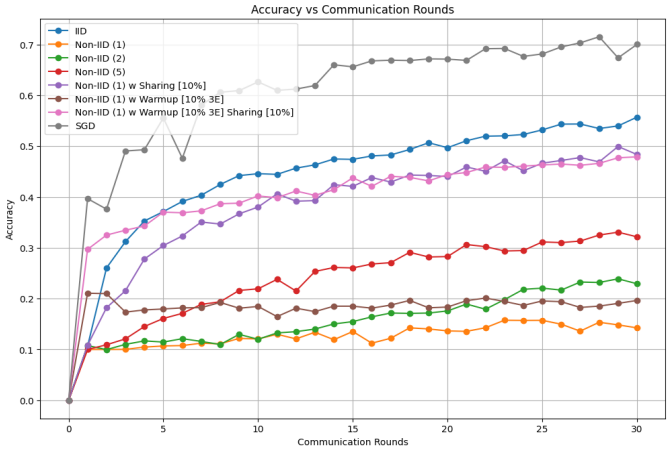
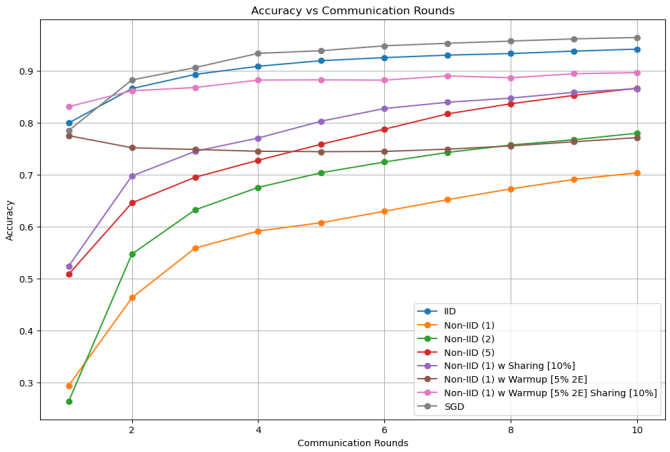
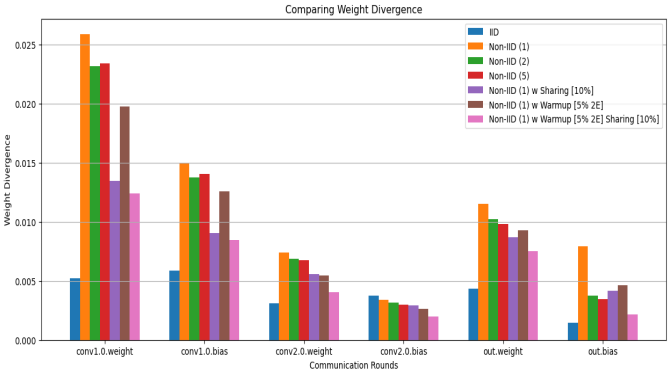


Fig. 4: Comparison amongst different distributions for CIFAR-10



(a) Accuracy



(b) Weight Divergence

Fig. 5: Comparison amongst different distributions for MNIST

to the clients. For MNIST (Figure 5), the accuracy loss diminishes from approximately 30% to around 10%. Similarly, for CIFAR-10 (Figure 4), the accuracy loss decreases from an astounding 80% to ∼30%.
Despite the initial training, the warmup model still struggles with the non-IID distribution of the clients giving ∼65% accuracy loss. This challenge results in performance

comparable to (2)-class non-IID data. Nonetheless, while the warmup model may not be effective in isolation, it is instrumental when combined with the data-sharing strategy. This combination facilitates effective convergence towards the central SGD model, negating the unfavorable effects of non-IID data distributions.

5.5 Implementing Warmup along with Shared-data

In this section, we observe the impact of changing the percentage(α) of the shared subset of IID data while simultaneously varying the fraction(β) of data taken from the total centralized global data, which the warmup model uses to train.

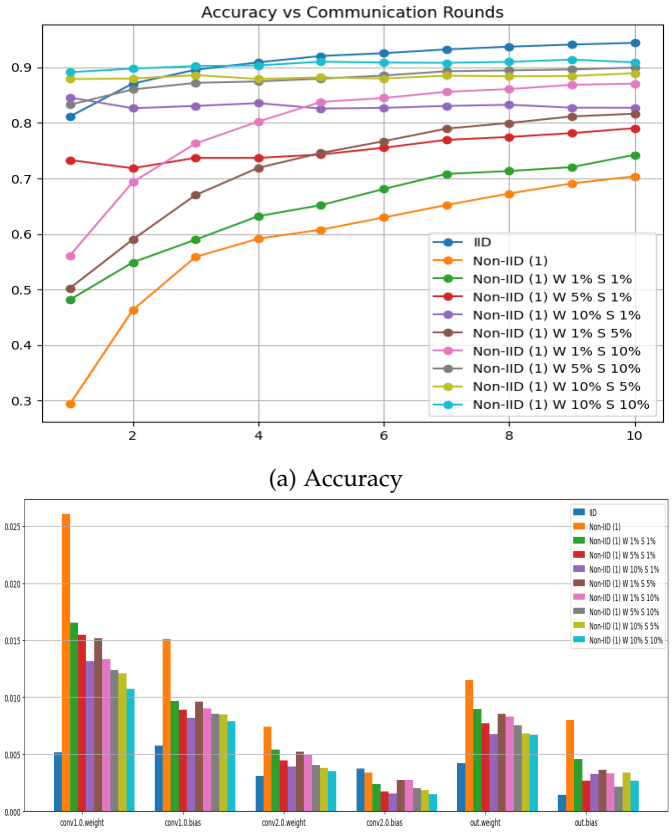
We observe that conducting a warmup phase with only 10% of the data significantly enhanced non-IID accuracy for both the MNIST and CIFAR-10 datasets. Specifically, for MNIST, the increase amounted to approximately 11%, while for CIFAR-10, this improvement soared to $\sim$ 29%.

Similarly, when sharing a mere 10% subset of the global central data, we noticed notable improvements in accuracy. For MNIST, there was an increase of around 18%, whereas for CIFAR-10, the growth reached an impressive 32% compared to the non-IID clients.

Combining both of these solutions has demonstrated significant effectiveness, nearly aligning beautifully with the IID curve. Specifically, the accuracy improvement for the MNIST dataset is approximately 24%, while that for CIFAR-10, it stands at around 35%.

6 Conclusion

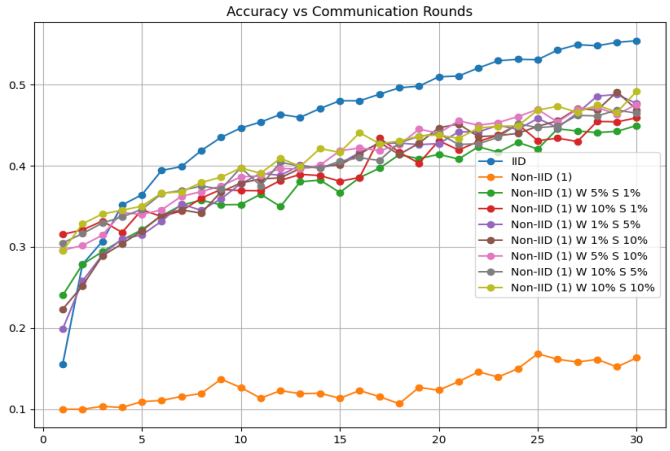
In conclusion, the decentralized approach of Federated Learning is a very efficient technique in using the data generated by millions of emerging IoT devices while simultaneously ensuring the clients' privacy. Unfortunately, the model's efficacy is very sensitive to the data distribution of the edge devices. In this report, we have shown how non-IID data leads to drastic failure within the training of the FedAvg model, using MNIST and CIFAR-10 for benchmarking. We have seen that this reduction in accuracy can be attributed to the difference in data distribution among clients and the overall distribution (known as the Earth Mover's Distance). We then partition the data into different types of non-IID settings to observe the effect of the weight divergence on the model accuracy. As potential solutions, we explore ideas like sharing a subset of iid data and a warmed-up model initialization of clients. We were able to increase the model accuracy by 50% for the CIFAR-10 dataset with only 5-10% globally shared data. There are still quite a few challenges left to mainstream federated learning, but improving model training on non-IID data is critical to progress in this area.



(a) Accuracy

(b) Weight Divergence

Fig. 6: Varying the percentage of warm-up and globally shared data - MNIST



(a) Accuracy

Fig. 7: Varying the percentage of warm-up and globally shared data - CIFAR10

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2023.

[2] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv*, 2018.