FedSim: similarity guided model aggregation for federated learning.

PALIHAWADANA, C., WIRATUNGA, N., WIJEKOON, A. and KALUTARAGE, H.

2022

FedSim: Similarity Guided Model Aggregation for Federated Learning

Chamath Palihawadana[a,*], Nirmalie Wiratunga[a], Anjana Wijekoon[a], Harsha Kalutarage[a]

[a]*School of Computing, Robert Gordon University, Aberdeen, UK*

Abstract

Federated Learning (FL) is a distributed machine learning approach in which clients contribute to learning a global model in a privacy preserved manner. Effective aggregation of client models is essential to create a generalised global model. To what extent a client is generalisable and contributing to this aggregation can be ascertained by analysing inter-client relationships. We use similarity between clients to model such relationships. We explore how similarity knowledge can be inferred from comparing client gradients, instead of inferring similarity on the basis of client data which violates the privacy-preserving constraint in FL. The similarity-guided *FedSim* algorithm, introduced in this paper, decomposes FL aggregation into local and global steps. Clients with similar gradients are clustered to provide local aggregations, which thereafter can be globally aggregated to ensure better coverage whilst reducing variance. Our comparative study also investigates the applicability of *FedSim* in both real-world datasets and on synthetic datasets where statistical heterogeneity can be controlled and studied systematically. A comparative study of *FedSim* with state-of-the-art FL baselines, *FedAvg* and *FedProx*, clearly shows significant performance gains. Our findings confirm that by exploiting latent inter-client similarities, *FedSim*'s performance is significantly better and more stable compared to both these baselines.

Keywords: Federated Learning, Model Aggregation, Similarity, Clustering

*Corresponding author

Email address: c.palihawadana@rgu.ac.uk (Chamath Palihawadana)

1. Introduction

Federated Learning (FL) is a Machine Learning (ML) paradigm which learns from distributed clients to collaboratively train a global model in a privacy preserving manner without transferring local data to a central server [1]. FL is typically applied in two types of application scenarios: cross-device and cross-silo. In a cross-device applications it is common to include a large number of clients, such as in keyboard next-word predictions [2], emoji prediction [3] and wake word detection [4, 5, 6, 7]. In contrast with cross-silo applications the number of clients are limited but each client typical has a large repository of data. Here typical applications include those in healthcare and banking, where a small number of institutions have a business incentive to train a shared model [8, 9, 10, 11, 12, 13]. In the cross-device settings of FL, intermittent availability of devices and the need to work with smaller client datasets remains an interesting challenge for model training. This calls for novel aggregation methods that can adapt to clients with small data with intermittent availability.

In traditional ML algorithms, the assumption of data being Independent and Identically Distributed (IID) is an important prerequisite. However, in the FL setting, this assumption is not held due to the distributed nature of data and the diversity of clients. This has created a demand for distributed training strategies suited to non-IID settings. Federated Averaging (*FedAvg*) is the most widely used algorithm for learning a generalised global model in non-IID settings [1]. *FedAvg* is an incremental and distributed stochastic gradient descent (SGD) based model optimisation strategy commonly used in the cross-device application scenarios. At each FL round, locally optimised client models are aggregated to create a new and improved global model at a central location and is repeated as many rounds until model convergence or application imposed resource constraints are reached.

In many applications, there are commonalities to be found amongst clients (e.g. similarity in demographics, interests, subjectivity). For example, Figure 1 illustrates the similarities (using the Kolmogorov-Smirnov statistic for distribution comparison) between clients using three datasets commonly used in FL research [1, 14, 15]. Exploiting these pairwise client similarities could in turn help reduce computational costs (e.g. number of FL communication rounds) needed to achieve comparable convergence. Similarity knowledge can also be used to discover divergent clients, therein help temper their influence on the global aggregation (i.e. reduce variation). In this paper we investigate this research hypothesis and demonstrate that exploiting inter-client similarities leads to better performance in an FL setting than current benchmarks.

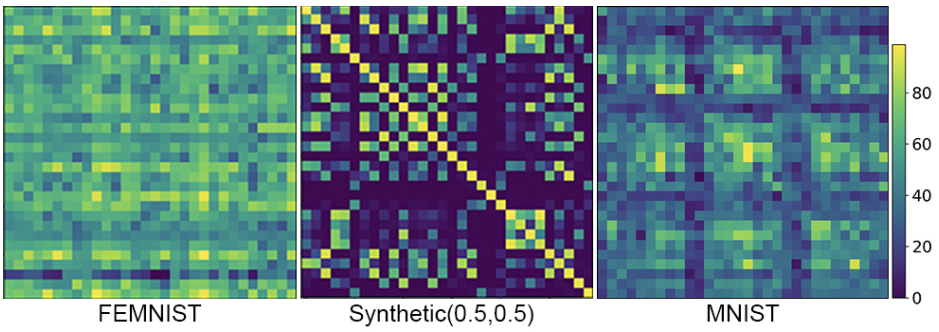

Figure 1: Pairwise similarity between clients in FL datasets

To this end, we make the following contributions:

- introduce *FedSim*, a pairwise client similarity guided model aggregation strategy for FL, using client gradients to compute similarity;
- conduct a comparative evaluation of *FedSim*, against the state-of-the-art *FedAvg* [1] and *FedProx* [14] to study performance and training stability using 4 real-world datasets and 6 synthetic datasets; and
- provide two novel datasets for FL research, extracted from two publicly available datasets: Fed-MEx and Fed-Goodreads.

Our evaluation results suggests that by exploiting latent similarity knowledge between clients, *FedSim* is able to significantly improve performance over other baselines and that training performance is also seen to be more stable. These findings are further verified, analysed and insights drawn to understand: the performance improvements observed with *FedSim* with respect to the quality of client cluster analysis; statistical significance in performance improvements with increasing FL training; impact of similarity guided vs. random clustering for *FedSim*; generalisability of *FedSim* with a variety of popular classifier models; and measuring non-IIDness in a privacy preserved manner to validate the impact of *FedSim* at different levels of statistical heterogeneity.

The rest of this paper is organised as follows. Section 2 investigates the related literature in FL. Next, the novel similarity guided model aggregation strategy *FedSim* is presented in Section 3. Section 4 presents the datasets, evaluation methodology and performance measures. Results and discussion with real-wolrd datasets are presented in Sections 5 followed by a comparative study with synthetic data in Section 6. The conclusions and plans for future work are presented in Section 7.

2. Related Work

In this section, recent research literature in five areas of interest are investigated: FL algorithms; use of similarity in FL algorithms; security and efficiency improvements with similarity; measuring of statistical heterogeneity in an FL setting; and FL evaluation methodologies.

2.1. Federated Learning

FedAvg [1] is the most widely used FL algorithm in recent literature to learn a global model under the client privacy preserving mandate. The aim of *FedAvg* is to create an effective global model with wider coverage of participating clients. At the beginning, the server randomly initialises the global model, w_0, and communicates it to the clients. Thereafter, the global model is continuously optimised over many communication rounds. A communication round t in *FedAvg* is aimed at distributing and updating a global model which can thereafter be progressively improved with increasing rounds until convergence. Each round involves sampling K clients and these selected clients will receive the most recent global model. Note that the sampling is applied to clients present in a given round and in a real-world scenario these clients can be intermittent. Thereafter each client performs a local update using SGD for E number of epochs with batch size B of its training data. Once the local update is completed, the updated weights are communicated to the server. These updates are weighted by the client sample size n_k and aggregated to obtain the updated global model. The weighted aggregation step is shown in Equation 1, where K is the clients selected in a round, w^k_{t+1} the updated weights of a client, n the total data sample size from all clients, n_k the sample size of the k client and w_{t+1} the updated weights of the global model.

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w^k_{t+1} \tag{1}$$

Once the updated global model w_{t+1} is computed it is re-distributed to all clients to benefit from the refined model on completion of a single FL round.

In a highly heterogeneous FL setting, regularisation can be used to recover from aggregations that might harmfully divert the global model. *FedProx* [14] is a framework that addresses two such heterogeneity challenges encountered in an FL setting. Firstly with system heterogeneity *FedProx* addresses real-world application challenges such as when clients are unable to complete local training or clients might not be available throughout training. Secondly with statistical heterogeneity it addresses situations when the client's local training samples are from

highly non-IID data distributions. In standard *FedAvg* a client is required to complete the full FL computation round (involving several local epochs) or else they are considered as dropouts. In contrast with the *FedProx* algorithm such clients are considered to be stragglers that are allowed to communicate partial updates based on their status. In order to handle partial updates, *FedProx* uses an inexactness measure γ for each client in each round. γ measures how much computation is completed to optimise the local model. For statistical heterogeneity *FedProx* proposes a proximal term (μ) to the local update objective. *FedProx* changes the local update objective h_k as in Equation 2. F_k is the local objective function of a client k. Note that *FedAvg* is a case when $\mu = 0$ in *FedProx*, when the local update is SGD and when γ is a constant for all clients.

$$h_k = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2 \tag{2}$$

FedProx has shown significant stabilisation in training, relative to *FedAvg* in the presence of statistical heterogeneity. It is worth noting that this comparative performance advantage is not observed unless there are a majority of clients who are considered as stragglers. In our comparative studies, we benchmark *FedSim* against both *FedAvg* and *FedProx* to empirically evaluate the performances achieved by our similarity guided model aggregation strategy.

2.2. Similarity in Federated Learning

Pairwise client similarity can help recognise divergent clients. Similarity based weighting schemes are increasingly proving advantageous in other areas of machine learning (e.g. Transformers in NLP and image vision applications [16]). With FL, the need for privacy automatically rules out any considerations of using similarity weighted aggregation at the data level.

Clustered Federated Learning (CFL) proposed in [15] is a Federated Multi-Task Learning framework which groups clients into clusters with similar data distributions. CFL is a post-processing algorithm which begins after the training phase of FL is completed and the global model is converged. CFL focuses on creating specialised models for a set of clients which can benefit from their data distribution similarities. To this end, it uses client gradients as a basis for similarity computation for post-FL adaptation of the global model by similar client groups. CFL computes bi-partition (branching) of the global model until there is a specialised model for each branch containing one or more clients. In our work, we use gradients instead of data to gauge pairwise client similarities, however this is done to improve model aggregation during FL training, and not as a post-FL

processing strategy, hence *FedSim* is different from CFL. Nonetheless CFL can still be applied following *FedSim*'s FL training phase.

In meta-learning research, like in FL, a combination of independent gradient descent steps carried out by different tasks (analogous to clients) are aggregated to form a meta-model (analogous to the global model) for distribution at each round. Unlike FL, the focus is not on satisfying privacy constraints, but on maximising the adaptability of the model to new tasks. Interestingly, in meta-learning, organising tasks considering semantic similarities has resulted in superior learning with applications that rely on inter-client relationships such as personalised activity recognition [17] and personalised conversation generation [18]. Here we explore whether *FedSim* can also be applied to personalised settings, but in a FL context. More specifically using FEMNIST and two new datasets, Fed-MEx and Fed-Goodreads, we mimic the need for personalisation where each client can be viewed as having data from a single person.

2.3. Security and Efficiency with Similarity in FL

In FL applications security plays a major role where by default privacy preservation of data is central to the concept of model learning. However when communicating client information, such as weights, gradients and meta data, there is an obvious vulnerability to cyber threats. Hashing has been successfully used in distributed ML applications for secure communication, specifically where locality hashing is used to also preserve similarity knowledge between patients in health applications [12]. Here the distributed ability to learn hash codes which are context-specific and representative of patients across different institutions is also very relevant to FL. Specifically in [19], a cross-siloed FL setting was able to train boosted decision trees using a pre-processing stage where each client computes hash values to build global hash tables. Unlike with the cross-device FL setting, clients communicate learnt models with other clients (instead of sending it to a server) in a sequential manner (from client to client) where all clients are expected to be available during training with large amounts of data at their disposal. However the use of locality sensitive hashing is relevant also to a cross-device settings (and our work), as such it can be adopted as a means to securely communicate information in most FL settings. Contextually relevant hashing can also be adopted to reduce dimensionality for similarity computation as an alternative to general purpose reduction methods (e.g. singular value decomposition [20], locality sensitive hashing [21]). In our work, we use principle component analysis for this purpose.

2.4. Measuring Statistical Heterogeneity

Measuring the degree of statistical heterogeneity (i.e. Non-IIDness) of clients provides insights on generalisability of models in an FL setting. A recent review identified several factors that affect statistical heterogeneity [13]:

- **Feature distribution skew** is when the data distributions of clients are different and can be varied due to different personalisation nuances (e.g. a letter can be written in different ways).
- **Class distribution skew** is when the distribution of classes are varied across clients. Some clients can have a small subset of classes and some can have a mix of many classes (e.g. certain words can be used by a certain group of people only).
- **Quantity shift** is when different clients have different amounts of data. The amounts can be vastly varied among the client population.
- **Concept shift and concept drift** occur in FL settings when clients identify similar data instances by different class labels, and when clients label different data samples with the same class label. In such scenario the non-IID assumption is extreme and FL is adversely impacted.

In the real-world, a mixture of these factors are to be expected. Several metrics attempt to quantify non-IIDness of a client by comparing its data at the class level with others [22, 23]. A global Non-IID Index (NI) was introduced in [23] to extract a class level embedding using a pre-trained deep learning architecture (ConvNet) using all train and test data obtained from clients. Here an average embedding is generated for each class using the train and test data separately. The final NI measure is the sum of the normalised differences between the train and test average embeddings in each class. Reliance of client data to generate embeddings limits the applicability of this metric in an FL setting where privacy must be preserved. An adaptation of this NI algorithm, Client-wise Non-IID Index (CNI), considers class-level embeddings at local clients (instead of globally) [22]. CNI considers the degree of distribution shift at each client instead of between train and test sets of all clients. It captures feature distribution, label distribution and quantity shift. However CNI can only be used to study client heterogeneity using image datasets. The *PNI* measure proposed in this paper, instead can be applied on non-image data while preserving privacy. Accordingly we use *PNI* to draw insights from evaluation results and to understand performance behaviours across the different real and synthetic datasets used in our work.

Method	Evaluation Datasets	Baselines	Evaluation Metrics
FedAvg [1]	MNIST, FEMNIST CIFAR-10, Shakespeare	FedAvg	Communication Rounds
FedProx [14]	Synthetic, MNIST FEMNIST, SENT140 Shakespeare	FedAvg	Stabilisation (visually on train loss) Accuracy
LotteryFL [22]	MNIST, FEMNIST CIFAR-10	FedAvg	Accuracy Communication Cost
IFCA [24]	Synthetic, MNIST CIFAR-10	Global model Local model	Accuracy
SCAFFOLD [25]	Synthetic, FEMNIST	FedAvg FedProx	Communication Rounds
LEAF [26]	FEMNIST, Sent140 Celeba, Reddit Shakespeare	FedAvg	Accuracy Communication Cost Amount of FLOPS

Table 1: Datasets and evaluation metrics found in FL literature

2.5. Performance Measures

Table 1 presents a comparison of datasets used in previous work along with the metrics used to evaluate proposed methods. Compared to conventional ML, evaluating the performance of FL algorithms must consider criteria relevant to both local and global learning perspectives. Communication rounds, is a common criteria, which is interpreted as the number of rounds needed to achieve a target accuracy or convergence [25, 1]. Communication cost refers to the amount of network bandwidth consumed to achieve a target accuracy [22, 26]. Aggregation of test accuracy from each client on the distributed global model has been formed into a performance measure in [24, 14]. Essentially, the choice of evaluation criteria depends on the hypothesis being examined. For instance, aggregation methods aim to increase the global model's test accuracy, whilst communication reduction techniques aim to reduce communication costs. The focus of this paper is an improved aggregation mechanism, therefore we use global model test accuracy, supported by a comparative accuracy fluctuation analysis metric over communication rounds to evidence performance stability.

3. Methods

FedSim favours similar clients when formulating locally specialised models. It must also create a generalised global model that can cover useful differences among these specialised models. Use of similarity knowledge in clustering helps to improve coverage and identify representative clients. In non-IID settings improving client coverage will help generalisability of the global model, whilst the use of similarity will help identify and reduce potentially harmful influences on global aggregations by divergent clients. In the rest of this section we present a detailed description of the *FedSim* algorithm and propose a measure of statistical heterogeneity.

3.1. Federated Learning with FedSim

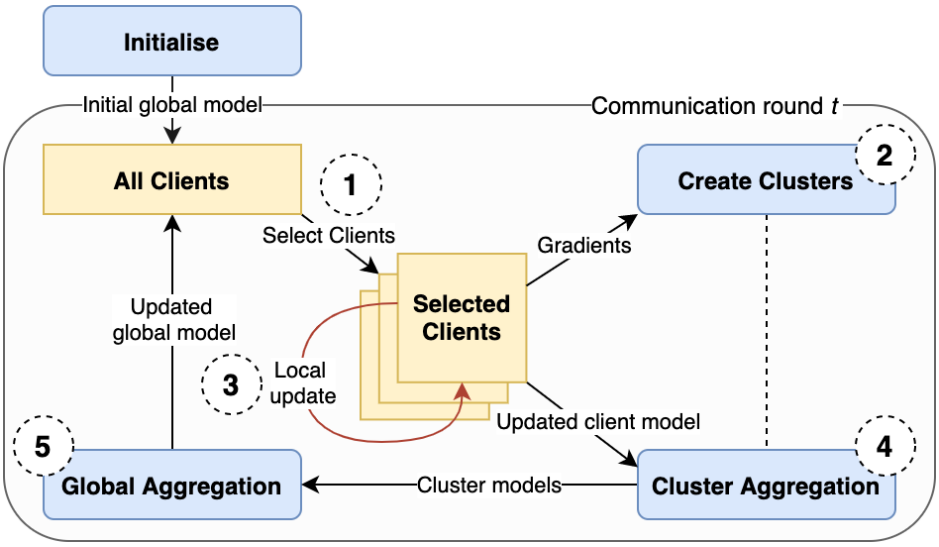

Figure 2: High-level approach of the proposed *FedSim* algorithm

A high-level view of the proposed *FedSim* algorithm is illustrated in Figure 2. Following the distribution of a randomly initialised global model to clients, a *FedSim* communication round has the following steps: client sampling, clustering, local updates cluster aggregation and global aggregation.

Step 1 due to communication constraints and intermittent client availability a subset of clients are randomly sampled to participate in each FL round.

Step 2 selected clients are clustered based on their local gradients without sharing each other's privately held data. These gradients are calculated based on error from the most recently distributed global model.

Step 3 selected clients continue to update their model weights (using an optimisation method such as SGD) to produce their local models.

Step 4 locally updated client weights are combined within each cluster using a weighted average to form a specialised, representative model for the cluster. Here weights are a function of a client's sample size given the cluster.

Step 5 a global aggregation step is used to combine the specialised models to generate the new global model

Follow on sections discuss these *FedSim* steps in detail providing a theoretical underpinning and with reference to Algorithm 1.

Algorithm 1 *FedSim* Algorithm

Require: w_0 initial global model, $\mathcal{K}$ clients, *n_clusters*

for t=1,2,.. **do**
  Broadcast w_t to all clients
  Select $\mathcal{S}$ clients where $\mathcal{S} \subset \mathcal{K}$
  $\mathcal{C} \leftarrow$ Clustering(S, *n_clusters*) *(Algorithm 2)*
  for all $c \in \mathcal{C}$ **do**
    for all $k \in c$ **do**
      $w_t^k \leftarrow$ updates w_t using SGD
    end for
    $\bar{w}_t^c \leftarrow$ ClusterAggregation($w_t^1, w_t^2, ..., w_t^{|c|}$) (Eq.13)
  end for
  $w_{t+1} \leftarrow$ GlobalAggregation($\bar{w}_t^1, \bar{w}_t^2, ..., \bar{w}_t^{|\mathcal{C}|}$) (Eq. 14)
end for

3.2. Federated Optimisation with Clusters

Mathematically, a ML optimisation problem aims to minimise an objective function, f, which is defined as follows for an instance, i, with weights, w:

$$\min_w f(w^*) \quad \text{where} \quad f(w) = \mathbb{E}(f_i(w)) \tag{3}$$

Here f is a function of the model error (e.g. $f_i(w) = (\hat{y}_i - y_i)^2$) and $\mathbb{E}$ is the expected value. In a FL system with K clients indexed by k, each with n_k data instances, the objective function for a client k is:

$$F_k = \mathbb{E}(f_i(w); \forall i \in k) = \frac{1}{n_k} \sum_{i=1}^{n_k} f_i(w) \tag{4}$$

Let the objective function of ***FedAvg***, for a set of selected clients, K, in a given round be:

$$F_O = \mathbb{E}(F_k; \forall k \in K) \tag{5}$$

Suppose we used a clustering algorithm (such as k-means), to create a client cluster, c, containing a set of S clients, then we can define an objective function for that cluster as:

$$F_c = \mathbb{E}(F_k; \forall k \in S) \tag{6}$$

Accordingly, for a FL system with ***FedSim***, having a set of clusters, C, let the objective function be:

$$F_G = \mathbb{E}(F_c; \forall c \in C) \tag{7}$$

Furthermore, to prove the impact of similarity based clustering, let F_R be the objective function of a randomly formed cluster (R) of the same size as that of cluster c (as in Equation 6). Then:

$$F_R = \mathbb{E}(F_k; \forall k \in R) \tag{8}$$

Equation 4 is a function of model error, which increases with increasing variance in data points. When clients are tightly clustered using similarity based measures as in Equation 6, it is expected that the error variance in a single cluster will be lower than that of a cluster that is formed randomly. Hence when the weight parameters tends towards optimal values, we expect:

$$F_c \leq F_R \tag{9}$$

$$\mathbb{E}(F_c) \leq \mathbb{E}(F_R) \tag{10}$$

Note that in ***FedAvg*** clients are randomly included within a single cluster, therefore from Equation 5 and 7, we expect:

$$F_G \leq F_O; \text{ where } F_G = \mathbb{E}(F_c; \forall c \in C) \text{ and } F_O = \mathbb{E}(F_k; \forall k \in K) \tag{11}$$

Now suppose we introduce an arbitrary constant λ as follows:

$$F_G + \lambda = F_O \tag{12}$$

such that in Equation 12, λ acts as a regularisation term, that varies with cluster settings (e.g. number of clusters, clustering method, similarity metric, dimensionality). This would then suggest that by using an informed clustered approach we can obtain a more regularised objective function which would produce improved performance compared to a FL method with just a single cluster. Indeed when all clients belong to a single cluster, then Equation 7 is identical to *FedAvg* (F_O) where, $|C| = 1, n_c = n$ and λ becomes 0. When $|C| > 1$, we expect clustering to provide a better representation of the federated problem space by ensuring that the influence of similar clients do not dominate the aggregation of parameters in federated learning. Essentially this helps to reduce potential variance in the weight aggregation (averaging) step. Our idea is to cluster clients according to the similarity of client parameters, which in turn acts as a proxy to similarity on the basis of client data.

3.2.1. Initialisation

The initialisation step of *FedSim* is identical to *FedAvg* where a global model is initialised with random weights, w_0. Here, the global model (and corresponding local models) is selected to meet the requirements of the reasoning task (i.e. next word suggestion or character recognition). Commonly it is a neural architecture where w_0 are its model parameters.

3.2.2. Clustering

In communication round, t, a set of clusters, $\mathcal{C}$, are created with a subset of clients, $\mathcal{S}$, randomly sampled from the set of all clients, $\mathcal{K}$, where $\mathcal{S} \subset \mathcal{K}$. This clustering process (see Algorithm 2) is triggered in each *FedSim* round as shown in Algorithm 1, line 4. A client, k, is represented using the gradient vector, g_k, obtained using the client's training data error of the recently distributed global model, w_t. Once client gradients are communicated to the server, a clustering algorithm (such as *kmeans++*), is used to create $|\mathcal{C}|$ number of clusters (i.e. $n\_clusters$) based on similarity of client gradient vectors. Clustered clients perform SGD to create their locally updated models (w_t^k for client k).

Clients are sampled without replacement at each round (as in *FedAvg*) which helps with applications having intermittent clients. Accordingly, the clustering step in *FedSim* needs to be repeated at each round. Computational cost of each

communication round can increase exponentially due to pairwise similarity calculations for clustering. We alleviate this cost in two ways: sampling a few clients in each round; and applying dimensionality reduction (in this work we use Principal Component Analysis (PCA)) to the gradient vectors.

Algorithm 2 *FedSim* Clustering Method

Require: $\mathcal{S}$ clients, *n_clusters*, w_t model
1: **for all** $k \in \mathcal{S}$ (selected clients in round t) **do**
2: $g_k \leftarrow$ compute gradients for w_t using SGD on local data
3: **end for**
4: $G' \leftarrow g_1 ... g_{|\mathcal{S}|}$, where gradients g_k received from each client k $\in S$
5: $G \leftarrow$ dimensionality_reduction(G') ; e.g. PCA
6: $\mathcal{C} \leftarrow$ client_clustering(*n_clusters*, G') ; e.g. K-Means++
7: **return** $\mathcal{C}$

Use of gradients vectors ensures semantically meaningful private data is not communicated when similarity is computed. We also expect that for similarity, gradients will capture latent patterns of a client's data w.r.t. model error. Accordingly, using gradient vectors to derive similarity between clients ensures that *FedSim* is able to reason about client similarity without exposing client data or client's meta-data to the server or other clients. Other forms of locality sensitive hashing methods can also be adopted to further secure the communication of similarity information.

As stated in Section 3.2, when $n\_clusters = 1$, Algorithm 1 is equivalent to the baseline *FedAvg*; where cluster aggregation in step 9 is applied to all clients involved in a given FL round using the aggregation in Equation 1; and step 11 becomes redundant (average for a single cluster). In addition to $n\_clusters = 1$, the *FedProx* algorithm, also combines a proximal regularisation term in step 7.

3.2.3. Cluster Aggregation

Purpose of cluster aggregation is to combine local models to create a representative model for each cluster. For a given cluster, c, at round, t, a cluster model is formed as:

$$\bar{w_t^c} \leftarrow \sum_{k \in c} \frac{n_k}{n} w_t^k \tag{13}$$

Here n_k is the sample size of client k, n the total number of samples and w_t^k the updated local model of client k (i.e. after locally updating the global model w_t).

The weighting by sample size, is borrowed from the original *FedAvg* aggregation. At the end of cluster aggregation, we obtain $|\mathcal{C}|$ clusters (i.e. $|\mathcal{C}| < |S|$), with each representing a specialisation over a distinct set of similar clients. Note that from Equation 12, the cluster settings in terms of; the number of clusters, and the metric space for similarity computations, all contribute to λ regularisation term.

3.2.4. Global Aggregation

The main objective of FL settings is to learn a global model that is generalisable to clients. In *FedSim* the new global model for distribution in round, $t+1$, is created as:

$$w_{t+1} \leftarrow \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \bar{w}_t^c \tag{14}$$

which is an average over all cluster models $\bar{w}_t^c$. Here all clusters are considered equally important to ensure equal coverage of the federated cluster space $\mathcal{C}$.

3.3. Measuring Statistical Heterogeneity

In order to study the relationship between similarity and non-IIDness, we consider a privacy preserving data characterisation measure (PNI) which we will use to help analyse the impact of statistical heterogeneity on FL performance. Essentially we expect that such a measure of non-IIDness should increase with increasing statistical heterogeneity.

Model error captures to what extent a model fails to generalise to its underlying data, which in turn can be due to the heterogeneity of the feature distributions. Here a non-IIDness measure can be defined as a function of error terms. Unlike raw prediction data, using derived information like model error is privacy preserving (and preferable over measures that rely on having access to raw data).

Root mean square error (RMSE) is selected as the error function for the proposed *PNI* measure. RMSE calculations requires access to raw data (actual and predicted class data), which for privacy reasons should not be communicated to the server. In order to overcome the privacy concern, a model, w_{rnd}, is initialised at the server with random weights and communicated to all clients. Here w_{rnd} is considered to be a neural model suitable for the reasoning task of clients. Each client i will predict labels $\hat{y}_i$ for all its training data D_i (Equation 15).

$$\hat{y}_i \leftarrow predict(w_{rnd}, D_i) \tag{15}$$

Each client will then use the predicted labels to calculate RMSE with respect to the actual label y and for n number of local samples (Equation 16).

Once computed locally, $RMSE_i$ will be communicated to the server.

$$RMSE_i = \sqrt{\sum_{n}^{D_i} \frac{(\hat{y}_n - y_n)^2}{n}} \tag{16}$$

Here we can use paired differences between model error, to measure statistical heterogeneity of a client, i:

$$PNI_i = \left\| RMSE_i - RMSE_{j \neq i} \right\|_2 \tag{17}$$

where differences in RMSE between client, i, and all other clients, $j \neq i$, are computed using the Euclidean norm (L2 distance). The Privacy-preserving Non-IID Index (*PNI*), for the federation of all clients is calculated as an average over client *PNI* values:

$$PNI = \frac{1}{|C|} \sum_{i}^{C} PNI_i \tag{18}$$

4. Evaluation

The aim of the evaluation is to study the utility of the similarity guided federated learning algorithm (*FedSim*) through a comparative study involving the following baseline algorithms:

- ***FedAvg*** is the most popular and widely used FL approach available today [1].
- ***FedProx*** [14] improves stability of *FedAvg* in non-IID settings and is considered state-of-the-art for both statistical and system heterogeneous settings.
- ***FedSim*** is our algorithm proposed in this paper (see Section 3.1)

The remainder of this section will present the datasets, implementation details and evaluation methodology used in this paper.

4.1. Datasets

FedSim is evaluated using 4 real-world datasets and 6 synthetic datasets. We have considered commonly used publicly available federated datasets for our experiments to enable reproducible results. Also, we have curated two novel federated datasets from the Goodreads [27] and MEx datasets [28] both of which are available to download from GitHub.

4.1.1. Real-world Datasets

The four real-world datasets considered in the experiment setup are MNIST, FEMNIST, Fed-MEx and Fed-Goodreads. Real-world datasets exhibit a mixture of feature distribution skew, class distribution skew and quantity shift to characterise statistical heterogeneity (see Section 2.4).

MNIST is a handwritten digit recognition dataset adapted in the FL setting. We reuse the FL setting proposed by [14] where there are 69,035 data samples of 10 classes distributed among 1000 clients and each client has samples for only 2 classes. A data sample is an image of size 28×28 and the number of samples per client follows a power law.

FEMNIST is a 62-class handwritten character recognition dataset. A subset of FEMNIST representing 26 classes is used in the FL setting as proposed by [14]. A total of 18,345 samples are distributed among 200 clients and each client has samples only for 5 classes. A data sample is an image of size 28×28 and the number of samples per client follows a power law.

Fed-MEx is a novel FL dataset produced with this work. MEx is a publicly available exercise recognition dataset collected with 30 subjects performing 7 different physiotherapy exercises [28][1]. The Fed-MEx dataset has 934 data samples from the pressure mat subset of the MEx dataset. Each client has a random amount of samples for only 2 exercise classes. A pressure mat data sample contains a sequence of heat maps (size $5 \times 16 \times 16$) recorded for 5 seconds with $1Hz$ frequency. MEx has previously been used for personalised activity recognition research [17] and forms an interesting contrast to the other image and text datasets. The federated version of this dataset and the generation code is published on GitHub[2].

Fed-Goodreads is a novel FL dataset produced in this work. Goodreads[3] is a publicly available dataset and commonly used for text classification with DL due to its large volume [27, 29, 30, 31]. Fed-Goodreads contains the book reviews subset with parsed spoiler tags is used (1.38m reviews) to perform a binary classification task of predicting if a review sentence contains a spoiler or not. This dataset provides an ideal peronsalised setting

[1] https://archive.ics.uci.edu/ml/datasets/MEx

[2] https://github.com/chamathpali/Fed-MEx

[3] https://sites.google.com/eng.ucsd.edu/ucsdbookgraph

for a federated dataset as the data is organised by individual users, where a user will have different quantities of data and different users have different patterns of writing sentences. Fed-Goodreads contains 100 unique clients. The number of samples per client is limited to 2-10 to enforce statistical heterogeneity and each data sample contains 2517 features. The variation in text vocabulary in relation to the binary classification makes this a challenging text classification task. The curated FL version of the dataset and the generation code is published on GitHub[4].

4.1.2. Synthetic Datasets

Synthetic datasets are generated using the approach described in [32] and widely used in FL experiments [14, 33]. Two parameters α and β control the statistical heterogeneity of the generated data: increasing α increases class distribution skew by controlling the class generator model; and increasing β increases feature distribution skew. Both parameters together will impact levels of concept shift where the relationship between features and classes can change from client to client. Data samples, each with 60 features are generated using different α and β values to obtain 6 datasets with varying degrees of statistical heterogeneity. Each dataset has 10 classes and 30 clients. The number of samples per client follows a power law with which quantity skew is controlled. A synthetic dataset is denoted by the *synthetic*(α, β) notation. *synthetic IID* dataset is generated using identical distribution for features and classes across clients. All statistical heterogeneity factors (in Section 2.4) feature in synthetic non-IID datasets.

4.2. Experiment Details

All datasets present classification tasks which we initially model using Multinomial Logistic Regression. A flattened feature vector is used as the input. Input sizes for image data, Fed-MEx, Fed-Goodreads and *synthetic(*α, β*)* are 784, 1280, 2517 and 60. We also select following hyper-parameters: number of epochs for local update as 20; and batch size for local update as 10. Learning rates used for local update are 0.03, 0.003, 0.01, 0.3 and 0.01 for MNIST, FEMNIST, Fed-MEx, Fed-Goodreads and *synthetic(any)* respectively. Number of communication rounds are limited after convergence or at maximum 500 rounds. Selected communications rounds for MNIST, FEMNIST, Fed-MEx, Fed-Goodreads and *synthetic(*α, β*)* are 30, 500, 200, 250 and 100 respectively. All datasets maintain

[4]https://github.com/chamathpali/Fed-Goodreads

a client's train and test data split of 80% and 20%. Hyper-parameters mentioned above for MNIST, FEMNIST and *synthetic(α, β)* were adapted from [14] to ensure comparability and reproducibility.

Dataset	Features	Learning rate	Total clients	Com. rounds	Clients per round	Number of clusters
MNIST	784	0.03	1,000	30	20	5
FEMNIST	784	0.003	200	500	20	9
Fed-MEx	1280	0.01	30	200	10	3
Fed-Goodreads	2517	0.3	100	250	20	11
synthetic(any)	60	0.01	30	100	10	5

Table 2: Hyper-parameter details

Additionally, we explore two hyper-parameters specifically for *FedSim*: the number of clients per round and the number of clusters. These are two key factors to successfully discover latent similarity properties among clients. We explored following values: 10, 20, and 30 clients per round while keeping cluster size constant at 5; and 3, 5, 7, 9, 11 cluster sizes while keeping clients per round at 10 and 20. We find 20, 20, 10, 20, 10 as the most optimal number of clients per round and 5, 9, 3, 11, 5 are the most optimal cluster sizes for MNIST, FEMNIST, Fed-MEx, Fed-Goodreads and *synthetic(any)* datasets respectively. Hyper-parameter are summarised in Table 2.

All experiments are implemented in Python using the TensorFlow [34] libraries for Machine Learning. Experiments are performed on a MacBook Pro with 1.7 GHz Quad-Core Intel Core i7 processor and 16GB RAM memory and on 8 NVIDIA Tesla P100 SXM2-16GB GPUs. Average time elapsed for a communication round with MNIST, FEMNIST, Fed-MEx and Fed-Goodreads in milliseconds are as follows: 2929.3, 4433.5, 759.9 and 541.8 for *FedAvg*; 1423.7, 5855.8, 949.9 and 716.7 for *FedProx* and; 2858.3, 4443.4, 1049.5 and 646.8 for *FedSim*. The comparison of average time taken per communication round is presented in Table 3. The average time elapsed for a communication round of *FedSim* is nearly comparable on all datasets to other two methods. The time taken for a round is varied in each dataset due to its data size and experiment configuration (e.g. local epochs, number of clients selected per round). The source code for the experiment setup is available on GitHub [5].

[5]https://github.com/chamathpali/FedSim

Dataset	*FedAvg*	*FedProx*	*FedSim*
MNIST	2929.3	1423.7	2858.3
FEMNIST	4433.5	5855.8	4443.4
Fed-MEx	759.9	949.9	1049.5
Fed-Goodreads	541.8	716.7	646.8

Table 3: Comparison of average time taken per communication round in milliseconds

4.3. Performance Measures

The primary performance measure is the test accuracy of the global model against each client's test data. At the end of a communication round, once the global model is updated using the global aggregation step, it is communicated to all clients to evaluate using their test data. The final test accuracy of an algorithm, at any given round, is the mean of all test client accuracy measures weighted by the client's test set size.

In comparative evaluations, a mean performance improvement is calculated for *FedSim* and *FedProx* over the *FedAvg* baseline as a quantitative measure. The cumulative difference of test accuracy measures between two algorithms are averaged over the number of rounds to obtain the mean performance improvement as a percentage. We highlight the algorithm that records the highest improvement with bold text. In addition, comparison of test accuracy measures over the communication rounds are plotted to provide a qualitative measure of performance stability over the baselines.

4.4. Statistical Significance

Statistical significance helps quantify whether an outcome of an experiment is random or likely due to the factor of interest. Therefore, a one-tailed hypothesis test with a significance level of 0.05 was carried out to determine if *FedSim* performed better than *FedAvg* and *FedProx*. The experiments performed on all the datasets were carried out with 35 random seeds (from 0 to 34 incremented by 1) to empirically demonstrate the significance. Repetition of the same experiment with different random seeds helps to reduce the sampling error of our experiments.

5. Results and Discussion with Real-world Data

Figure 3 presents performance results for the three algorithms with increasing number of rounds on four real-worlds datasets. *FedSim* reaches higher performance on all datasets with pronounced improvements in FEMNIST, Fed-MEx

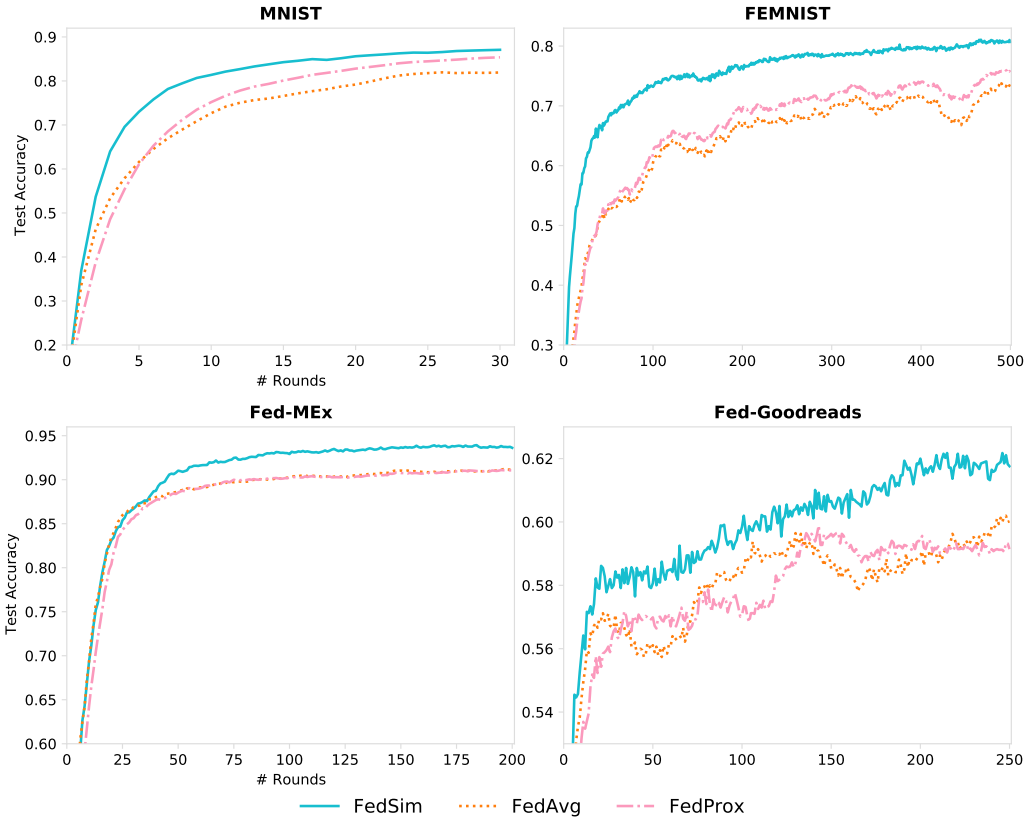

Figure 3: Comparison of performances over communication rounds

and Fed-Goodreads. All algorithms show stable convergence with increasing rounds on 3 datasets (MNIST, FEMNIST and Fed-MEx) with *FedSim* showing earlier convergence and greater stability on FEMNIST and Fed-MEx. *FedAvg* and *FedProx* have very similar convergence graphs on Fed-MEx but *FedSim* achieves greater performance over these baselines from approximately round 30 onwards. We observed convergence on Fed-Goodreads at round 250 when changes to loss did not exceed 8e-4 for the last 50 rounds. Although results with this dataset were less stable compared to the other datasets, *FedSim*'s performance was consistently better than the baselines. This is expected in a dataset like Fed-Goodreads where statistical heterogeneity is high (e.g. there are clients with one training instance and others can have over 5, furthermore vocabulary overlap is low due to significant variation in word usage). Nevertheless, in comparative terms *FedSim* manages to maintain relatively higher accuracy with a more stable performance graph.

However it was surprising that *FedProx* which was introduced with the aim to improve stability over *FedAvg*, had performed poorly (notably on FEMNIST).

Dataset	*FedSim* improvement over *FedAvg* (%)	*FedProx* (%)
FEMNIST	**11.11**±2.88	**9.08**±3.63
MNIST	**7.32**±2.69	**5.65**±4.35
Fed-MEx	**2.08**±1.26	**2.68**±0.78
Fed-Goodreads	**1.86**±0.73	**1.98**±0.64

Table 4: Comparison of overall performance improvements of *FedSim* over baselines

Table 4 lists the averaged accuracy percentages improvement gains achieved by *FedSim* over each of the two baselines. Here *FedSim* has significantly out performed both baselines on all four datasets (MNIST, FEMNIST, Fed-MEx and Fed-Goodreads). With FEMNIST this improvement is very pronounced, and to explore this further, we perform a cluster analysis using all clients for each dataset. The two most significant PCA components from the gradients are used to repre-

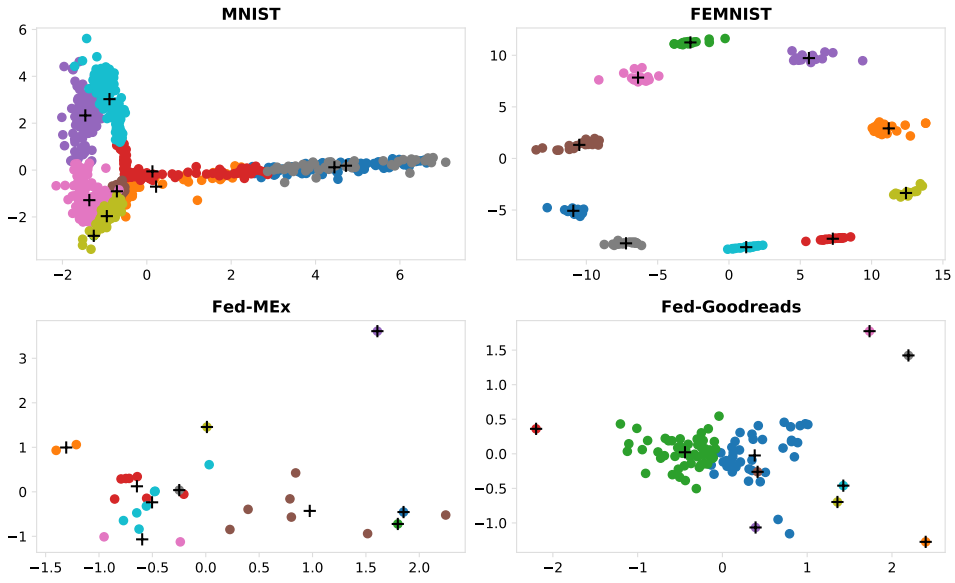

Figure 4: An example clustering of clients in a single trial

sent clients and to cluster them using K-Means (with $n\_clusters = 10$). Figure 4

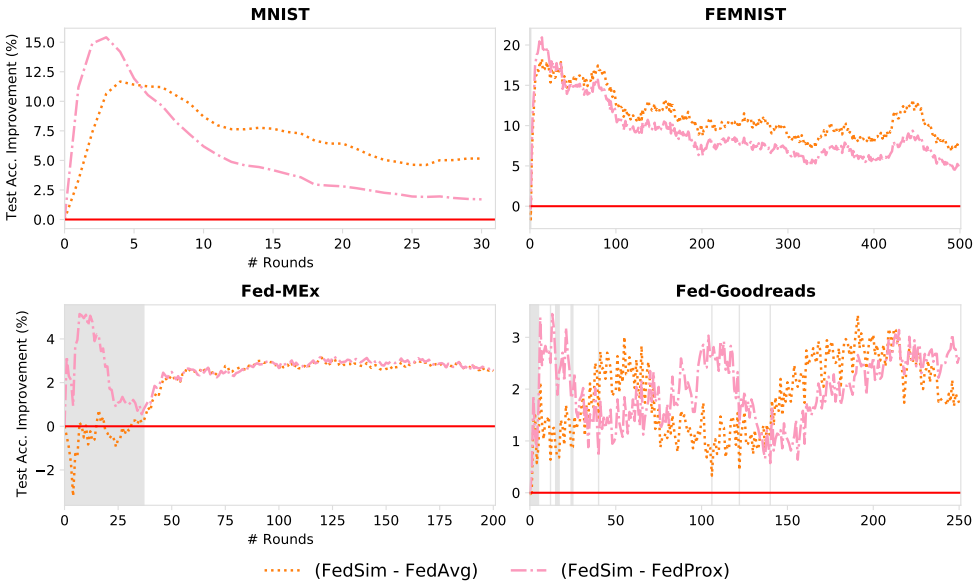

Figure 5: Analysis of accuracy improvements of *FedSim* compared to *FedAvg* and *FedProx* of experiments in Figure 3

presents a two-dimensional mapping of the clustering showing the clients and their cluster memberships (using 10 colours for the different clusters). It is apparent that FEMNIST has well defined clusters, having both higher intra-cluster similarity (density) and greater inter-cluster distance (separability), compared to the other datasets. Similar observations can be made of the MNIST clustering, where although clusters are densely formed, their separability is less pronounced, compared to that of FEMNIST. In contrast Fed-MEx lacks well-defined clusters, which possibly explains the significant yet smaller improvements observed with *FedSim*. Like with MNIST, in Fed-Goodreads we can observe reasonable clustering but with weaker separability between clusters. Here our analysis of clustering suggests that, as expected, *FedSim* is able to exploit similarity in client learning and that these similarities can be captured by comparing their gradients.

In Figure 5, we highlight in gray, any communication rounds in which *FedSim* failed to significantly outperform at least one of the baselines (significance level=0.05). Values below zero indicate negative performance against a baseline and grey vertical lines denote areas of no statistical significance. Significance testing results show that with a majority of increasing rounds, *FedSim*'s performance is superior to the baselines on all of the datasets. For instance with MNIST and

FEMNIST, significance was observed after the first communication round. Whilst with Fed-MEx significance was achieved after 37 rounds and thereafter maintaining significance until the 200th round. Similar observations were noted with Fed-Goodreads where in a majority of the rounds (i.e. 94% of the rounds), *FedSim* had achieved significant improvements. In comparison to FEMNIST, Fed-MEx and Fed-Goodreads datasets where cluster separability is not as pronounced (see Figure 4), *FedSim* achieves only minor improvements while maintaining statistical significance.

5.1. Generalisability Over Different Learning Models

In order to analyse model generalisability of *FedSim*, further experiments were conducted with alternative neural classifiers. A 2-D Convolutional neural network (CNN) was used for MNIST and FEMNIST hand-written digit classification tasks, a multi-layer neural network (MLP) with 3 hidden layers for the Fed-MEx dataset and a single layer recurrent neural network (RNN) for the Fed-Goodreads dataset. Details of these architectures appear in Table 5. We use the same experimental setup and *FedSim* hyper-parameter configuration as discussed in Section 4 (see also Table 2) with the exception of number of local epochs (10), and learning rate (0.0001) for Fed-Goodreads dataset. These exceptions were necessary to reduce the risk of overfitting, which has been a problem when training RNNs on the Fed-Goodreads dataset.

Dataset	Model	Architecture
FEMNIST	CNN-2D	$conv2d(3,3)64 \rightarrow maxpool(2,2) \rightarrow$
MNIST		$conv2d(3,3)64 \rightarrow maxpool(2,2) \rightarrow dense(2048)$
Fed-MEx	MLP-3	$dense(1280) \rightarrow dense(640) \rightarrow dense(120)$
Fed-Goodreads	RNN	$rnn(128)$

Table 5: Alternative model architectures used in the *FedSim* generalisability study

Figure 6 plots the test accuracy over FL rounds using the CNNs, MLP and RNN models. It is also evident that *FedSim* maintains similar accuracy improvements and learning stability with a majority of the neural models, which suggests that *FedSim* is model agnostic. *FedSim* accuracy improvements with FEMNIST is small, because the CNN model has converged quickly, however the model maintains better stability with *FedSim* (compared with *FedAvg* and *FedProx*). Results with Fed-Goodreads has been disappointing with none of the RNN models managing to achieve the previous accuracy levels achieved with the simpler logistic

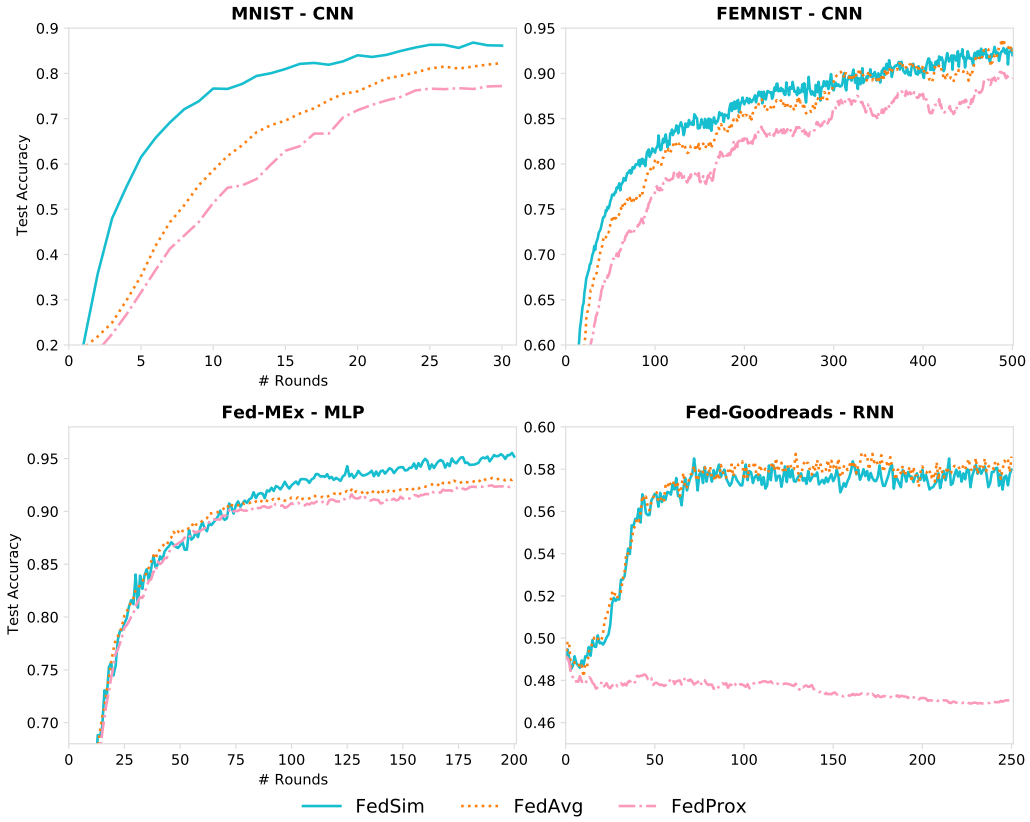

Figure 6: Comparison of performances over communication rounds with real-world datasets for different neural classifiers

regression (in Figure 3). We found that anything other than a simpler regression model led to overfitting with this dataset. Changing the optimiser to Adam [35] (from SGD) helped to a certain extent. Whilst it is clear from these experiments that the logistic regression model is best for the Fed-Goodreads dataset, we are still able to demonstrate that *FedSim*'s performance is comparable to *FedAvg*. Note that *FedProx* was badly impacted due to its inability to use the Adam optimiser (due to its use of partial updates from straggler clients) which explains its poor performance (we used the original optimiser recommended by the authors).

Results in Table 6 shows *FedSim* having an overall accuracy improvement on most of the datasets (with the exception of Fed-Goodreads) compared to results seen with the logistic regression model (in Figure 3). According to these results it is evident that the highest improvement with *FedSim* is gained on the MNIST

Dataset	Model	*FedSim* improvement over *FedAvg* (%)	*FedProx* (%)
FEMNIST	CNN-2D	**1.42**±1.52	**5.05**±2.75
MNIST	CNN-2D	**11.69**±7.58	**17.25**±8.26
Fed-MEx	MLP-3	**0.89**±1.29	**1.79**±1.28
Fed-Goodreads	RNN	-0.32 ±0.42	**8.88**±2.87

Table 6: Comparison of overall performance improvements of *FedSim* over baselines with real-world datasets with different model architectures

dataset which is 11.69% over *FedAvg* and 17.25% over *FedProx*. The visual presentation in Figure 6 supports this improvement on MNIST. Both FEMNIST and Fed-MEx have also gained an overall accuracy improvement with *FedSim*. With these results we can empirically prove that the proposed method is model agnostic and can be used with different model architectures in practical use cases.

5.2. Similarity Guided vs. Random Clustering

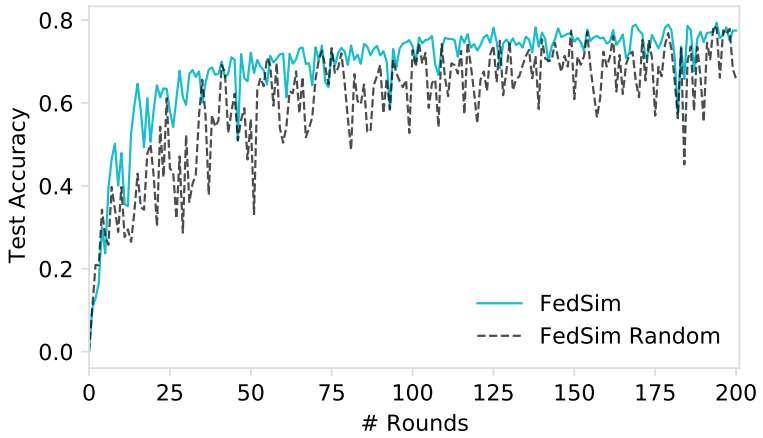

Figure 7: Comparison of similarity guided clustering vs random clustering on FEMNIST

To investigate the effect of exploiting similarity knowledge, a closer examination of *FedSim* was carried out by comparing a random cluster creation approach which assigns clients in a round-robin manner. The large gains observed on FEMNIST with *FedSim* can be explained by this comparison in Figure 7 which clearly demonstrates the benefit of using similarity knowledge for model aggregation. This empirically proves our expectation previously stated in Equation 10.

Interestingly (but not surprisingly) we also found that random clustering outperforms similarity clustering with extreme non-IID datasets (*synthetic* (0.75,0.75) and (1,1)) where there is likely to be no useful similarity knowledge to exploit.

5.3. Dimensionality Reduction with PCA

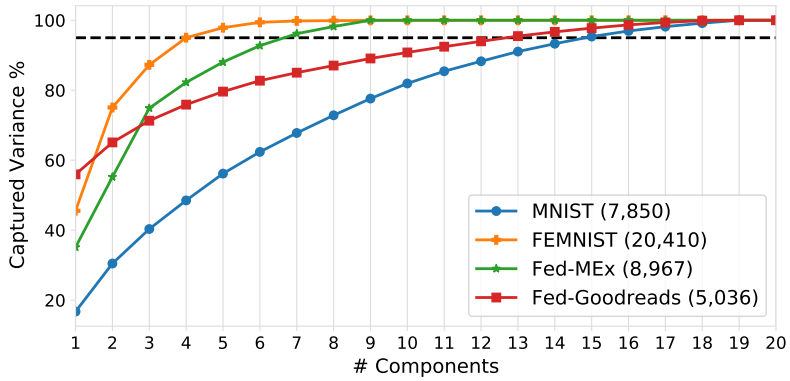

Figure 8: Analysis of variance captured by the number components when using PCA

As discussed in section 3.1, dimensionality reduction with PCA is utilised to minimise computation costs when computing similarity based clusters. Figure 8 explains the level of compression that can be achieved with PCA applied to each real-world dataset. The dotted line indicates the number of PCA coefficients that captured 95% of the variance of original gradients and are selected for clustering. We find that PCA reduces the gradient vector size to a value between 4 and 15 with all datasets. The impact of dimensionality reduction to minimise computational cost is quantified by comparing the time elapsed for a communication round in *FedSim* with and without PCA. We found that a *FedSim* communication round with MNIST, FEMNIST, Fed-MEx and Fed-Goodreads is 230.0, 469.1, 19.3 and 92.9 milliseconds, faster on average compared to without PCA. This improvement will have a significant impact on performance in production environments.

6. Comparative Study with Synthetic Data

A further investigation was carried out to understand the performance of *FedSim* and baselines on different levels of controlled IID-ness. For this purpose we use the five synthetic datasets with varying statistical heterogeneity situations. For instance by increasing α we increase the class distribution shift by varying the

standard deviation for sampling the weights that control the class label generation model. Similarly with increasing β we are able to shift the feature distributions between clients and thereby increase the levels of Non-IIDness through features. However in practice we expect that real-world FL datasets are unlikely to be completely Non-IID (e.g. α and β equals one) in terms of having clients that have both unique feature and class distributions. In contrast the IID dataset will have no feature distribution shift and will use the same class label generation model to generate client data, resulting in highly similar clients.

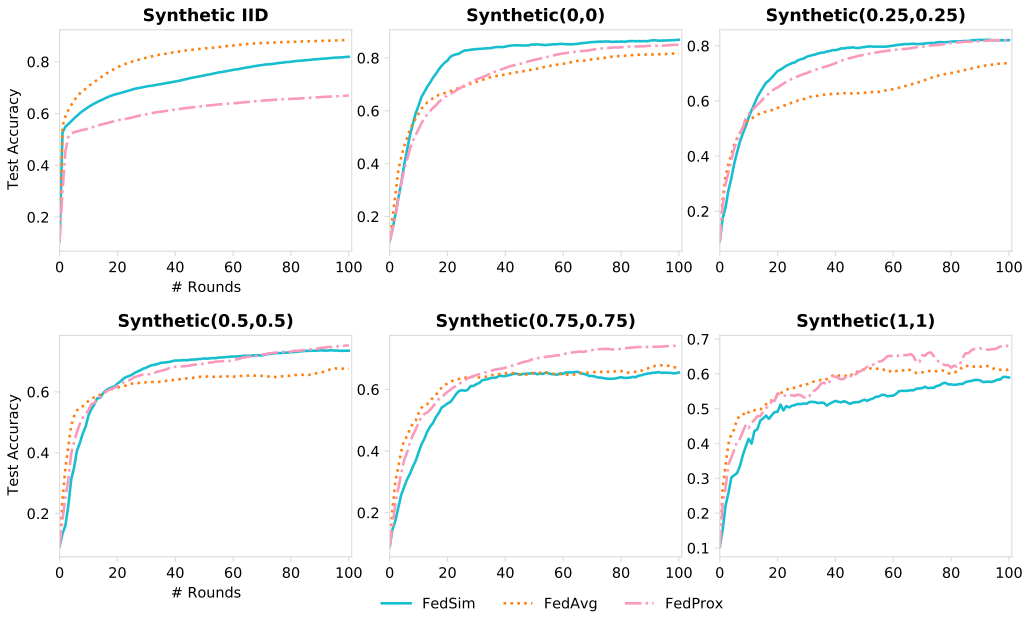

Figure 9: Comparison of performances over communication rounds with *synthetic* datasets to study the effect of statistical heterogeneity and similarity

Figure 9 plots the test accuracy measures over the communication rounds to investigate the performance stability of *FedSim* compared to *FedProx* and *FedAvg*. The similarity guided, *FedSim*, has achieved increased performance stability on *synthetic* datasets (0,0), (0.25,0.25) and (0.5,0.5) which are considered to be moderately non-IID. For datasets (0.75,0.75), (1,1) that are extremely non-IID, *FedSim* fails to outperform *FedProx*, however *FedSim* is significantly stable compared to *FedAvg*. Similarly, in IID setting, *FedSim* fails to outperform *FedAvg* however *FedSim* significantly outperforms *FedProx*.

A summary of results are presented in Table 7 which shows the mean test ac-

Synthetic Dataset	*FedSim* improvement over *FedAvg* (%)	*FedSim* improvement over *FedProx* (%)
Synthetic IID	-8.92 ±2.04	**11.99**±2.90
Synthetic(0,0)	**6.93**±4.78	**5.83**±4.17
Synthetic(0.25,0.25)	**11.21**±6.39	**1.87**±3.01
Synthetic(0.5,0.5)	**3.61**±6.23	-0.06±2.47
Synthetic(0.75,0.75)	-3.23±4.16	-6.22±2.74
Synthetic(1,1)	-6.18±2.83	-6.98±2.83

Table 7: Comparison of overall performance improvements of *FedSim* over baselines

curacy improvement (as a percentage) achieved by *FedSim*, over *FedAvg* and *FedProx* over 35 trials each involving 100 communication rounds. *FedProx*, which is optimised for non-IID settings, has significantly poor performance in IID settings (-20.92% with *synthetic IID* dataset). In comparison, *FedSim* has only a 8.92% drop in performance over *FedAvg* compared to that of 11.99% with *FedProx*. We expect that the IID situation is unlikely to benefit from clustering since all clients are likely to be similar and could instead be treated as members of one and the same cluster. *FedProx* achieves performance improvements with all 5 non-IID datasets compared to *FedAvg*. Similarly, 3 of the 5 non-IID datasets record performance improvements with *FedSim*. Notably, with the most extreme non-IID synthetic dataset, *FedSim* has failed to outperform *FedAvg* and records a performance reduction of 6.18%.

Overall, moderate non-IID settings benefit from a similarity guided approach to FL. In addition, *FedSim* performs comparably well in an IID setting as well as in an extreme non-IID setting. In moderate non-IID settings, *FedSim* exploits the latent similarities between clients to achieve additional performance improvement. However in the IID setting, *FedAvg*'s improvement over *FedSim* can be explained by observing that *FedSim* with $|C| = 1$ is equivalent to *FedAvg*. This suggests that when inter-client similarities are high it is better to form fewer clusters for aggregation. Currently *FedSim* maintains a fixed cluster size and instead would need to reduce the number of clusters (in an IID setting) to achieve comparable performance with *FedAvg*. In an extreme non-IID setting, the similarities are minimal to non-existent, hence *FedProx* using a proximity regularisation in its weight update step is better able to adapt to the setting achieving better performance.

Finally in order to study if *PNI* correlates with any known factors that cause statistical heterogeneity we use the methods described in [32], to create 11 syn-

thetic Non-IID datasets by changing α incrementally from 0 to 1 (α controls the variation in class distributions among clients). Then 3 variants of each dataset are created for β values 0, 0.5 and 1 (β controls the variation in feature distributions). Figure 10 plots the mean *PNI* values obtained for the 11 datasets and their vari-

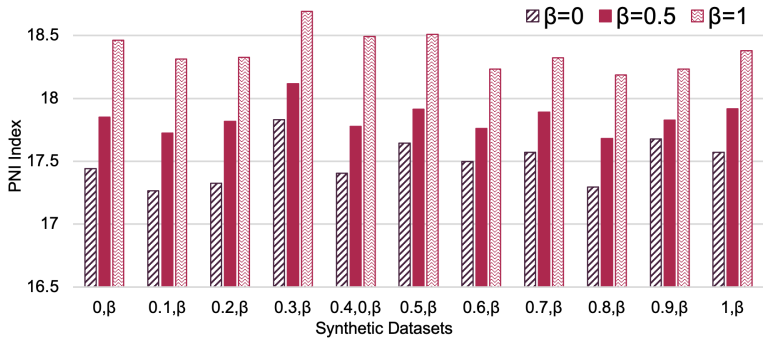

Figure 10: Comparison of *PNI* values across different feature distributions

ants from 100 repeated experiments (with 100 random seeds). Overall, *PNI* values consistently increased with β which demonstrates that *PNI* is capturing the heterogeneity in feature distributions among clients. It is reassuring to find that the *PNI* measure validated in this controlled setting with synthetic data, has increasing values with increasingly heterogeneous feature distributions. This suggests that *PNI* could in future be used to characterise real-world datasets where feature distribution are not explicitly controlled.

7. Conclusion

In this work, we introduced *FedSim*, an aggregation strategy to take advantage of inter-client relationships, modelled as pairwise similarity in gradients without sharing client data. A comprehensive evaluation on multiple application domains using real-world (including two new datasets) and synthetic datasets demonstrated that *FedSim* outperforms *FedAvg* and *FedProx* baselines, when similarity knowledge is harnessed. Results with real-world datasets confirmed that *FedSim* captured the similarity knowledge in clients to improve model aggregation leading to significantly better performance. Our findings also confirm the generalisability of *FedSim* with alternative neural models and optimisation algorithms.

In order to explore which settings are best suited to *FedSim*, we carried out experiments with six synthetic datasets with IID and multiple variants of non-IID distributions. Significant performance improvements were observed with *FedSim*

on multiple variants of synthetic datasets except with the IID and extreme non-IID settings. These findings suggest that switching between different aggregation policies can help to address changing levels of statistical heterogeneity or alternatively be able to change $n\_clusters$ dynamically to harness different levels of similarity relationships in client data. In future work, we will explore strengthening security mechanisms where similarity information can be captured without the risk of exposing privacy and to introduce a method to dynamically switch aggregation methods.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[2] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction, arXiv preprint arXiv:1811.03604 (2018).

[3] S. Ramaswamy, R. Mathews, K. Rao, F. Beaufays, Federated learning for emoji prediction in a mobile keyboard, arXiv preprint arXiv:1906.04329 (2019).

[4] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, J. Dureau, Federated learning for keyword spotting, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 6341–6345.

[5] M. B. Hoy, Alexa, siri, cortana, and more: an introduction to voice assistants, Medical reference services quarterly 37 (1) (2018) 81–88.

[6] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, J. Dureau, Federated learning for keyword spotting, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 6341–6345.

[7] A. Hard, K. Partridge, C. Nguyen, N. Subrahmanya, A. Shah, P. Zhu, I. L. Moreno, R. Mathews, Training keyword spotting models on non-iid data with federated learning, arXiv preprint arXiv:2005.10406 (2020).

[8] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, W. Shi, Federated learning of predictive models from federated electronic health records, International journal of medical informatics 112 (2018) 59–67.

[9] J. Lee, J. Sun, F. Wang, S. Wang, C.-H. Jun, X. Jiang, Privacy-preserving patient similarity learning in a federated environment: development and analysis, JMIR medical informatics 6 (2) (2018) e20.

[10] D. Gao, C. Ju, X. Wei, Y. Liu, T. Chen, Q. Yang, Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography, arXiv preprint arXiv:1909.05784 (2019).

[11] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, W. Shi, Federated learning of predictive models from federated electronic health records, International journal of medical informatics 112 (2018) 59–67.

[12] J. Lee, J. Sun, F. Wang, S. Wang, C.-H. Jun, X. Jiang, Privacy-preserving patient similarity learning in a federated environment: development and analysis, JMIR medical informatics 6 (2) (2018) e20.

[13] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, arXiv preprint arXiv:1912.04977 (2019).

[14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, arXiv preprint arXiv:1812.06127 (2018).

[15] F. Sattler, K.-R. Müller, W. Samek, Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints, IEEE Transactions on Neural Networks and Learning Systems (2020).

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[17] N. Wiratunga, A. Wijekoon, K. Cooper, Learning to compare with few data for personalised human activity recognition, in: International Conference on Case-Based Reasoning, Springer, 2020, pp. 3–14.

[18] A. Madotto, Z. Lin, C.-S. Wu, P. Fung, Personalizing dialogue agents via meta-learning, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5454–5459.

[19] Q. Li, Z. Wen, B. He, Practical federated gradient boosting decision trees, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 4642–4649.

[20] M. E. Wall, A. Rechtsteiner, L. M. Rocha, Singular value decomposition and principal component analysis, in: A practical approach to microarray data analysis, Springer, 2003, pp. 91–109.

[21] L. Paulevé, H. Jégou, L. Amsaleg, Locality sensitive hashing: A comparison of hash function types and querying mechanisms, Pattern recognition letters 31 (11) (2010) 1348–1358.

[22] A. Li, J. Sun, B. Wang, L. Duan, S. Li, Y. Chen, H. Li, Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets (2020). arXiv:2008.03371.

[23] Y. He, Z. Shen, P. Cui, Towards non-i.i.d. image classification: A dataset and baselines (2019). arXiv:1906.02899.

[24] A. Ghosh, J. Chung, D. Yin, K. Ramchandran, An efficient framework for clustered federated learning, arXiv preprint arXiv:2006.04088 (2020).

[25] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A. T. Suresh, Scaffold: Stochastic controlled averaging for federated learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 5132–5143.

[26] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, A. Talwalkar, Leaf: A benchmark for federated settings, arXiv preprint arXiv:1812.01097 (2018).

[27] M. Wan, R. Misra, N. Nakashole, J. J. McAuley, Fine-grained spoiler detection from large-scale review corpora, in: A. Korhonen, D. R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 2605–2610. doi:10.18653/v1/p19-1248.
URL https://doi.org/10.18653/v1/p19-1248

[28] A. Wijekoon, N. Wiratunga, K. Cooper, K. Bach, Learning to recognise exercises in the self-management of low back pain, in: Florida Artificial Intelligence Research Society Conference, 2020.
URL https://aaai.org/ocs/index.php/FLAIRS/FLAIRS20/paper/view/18460

[29] A. A. Zuccala, F. T. Verleysen, R. Cornacchia, T. C. Engels, Altmetrics for the humanities: Comparing goodreads reader ratings with citations to history books, Aslib Journal of Information Management (2015).

[30] M. Thelwall, Reader and author gender and genre in goodreads, Journal of Librarianship and Information Science 51 (2) (2019) 403–430.

[31] S. K. Maity, A. Panigrahi, A. Mukherjee, Analyzing social book reading behavior on goodreads and how it predicts amazon best sellers, in: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Springer, 2018, pp. 211–235.

[32] O. Shamir, N. Srebro, T. Zhang, Communication-efficient distributed optimization using an approximate newton-type method, in: International conference on machine learning, 2014, pp. 1000–1008.

[33] M. Duan, D. Liu, X. Ji, R. Liu, L. Liang, X. Chen, Y. Tan, Fedgroup: Ternary cosine similarity-based clustered federated learning framework toward high accuracy in heterogeneity data, arXiv preprint arXiv:2010.06870 (2020).

[34] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), 2016, pp. 265–283.

[35] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).