Computers and Electrical Engineering 111 (2023) 108900

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A reputation-aware hierarchical aggregation framework for federated learning☆

Monalisa Panigrahi [a,*], Sourabh Bharti [b], Arun Sharma [a]

[a] Indira Gandhi Delhi Technical University for Women, Delhi, India
[b] Munster Technological University, Cork, Ireland

ARTICLE INFO

Keywords:
Distributed learning
Federated learning (FL)
Collaborative learning
Hierarchical FL
Model aggregation in FL
Reputation-aware FL

ABSTRACT

Cross-device federated learning (FL) involves FLClients sharing their model updates to a global server for aggregation, which may result in a single point of failure as it becomes cumbersome for a global server to handle many FLClients. Hierarchical aggregation (HA) places another layer of aggregation (at edge servers) between FLClients and the global server. Although HA reduces the communication cost in aggregation, it does not help reduce the communication cost incurred by resource constrained FLClients while sharing their local models with the edge servers. This paper proposes a novel reputation-aware hierarchical aggregation framework (FedRaHa) that employs a reputation-based method to select clients' updates for aggregation as to minimize unnecessary local update exchanges. FedRaHa is evaluated using benchmark datasets such as MNIST, Fashion-MNIST, and real-world Chest Xray dataset. The results show that FedRaHa achieves the highest accuracy of 86 % and reduces the communication cost by 27.15 % as compared with the state-of-the-art.

1. Introduction

Federated Learning (FL) [1] is the most popular privacy-preserving distributed learning technique proposed by Google, USA in the year 2017. It is a technique for collaborative learning that enables users to jointly learn from each other's data by building a shared model without revealing their raw data to each other. A typical FL process consists of various clients and a centralized server. In each FL round,[1] the global server selects the clients for FL training, known as FLClients. Each FLClient downloads the global model, performs the training with its local data, and shares the trained model parameters (instead of raw data) with the global server. The next step is model aggregation on the server to create a new robust global model for the next round of training. Such collaborative model training preserves the clients' data privacy and minimizes the number of message transfers during the FL process. FL systems can be classified as cross-silo or cross-device depending on the scale of the federation. Cross-device FL setting involves a large number of clients (<100,000), which are typically resource-constrained devices whose availability depends on their current available resources. In a cross-silo FL setting, the number of clients is usually small (i.e., 2–100) and the clients are equipped with rich computational resources and large amounts of data. One of the major research challenges for cross-device FL is increased communication overhead during collaborative model training which leads to increased model convergence time, resource usage, and communication cost. This paper focuses on a cross-device FL setting with the objective of minimizing the communication cost incurred by resource-constrained FLClients during their local model transmissions to the global server.

☆ This paper is for regular issues of CAEE. Reviews processed and recommended for publication to the Editor-in-Chief by Huimin Lu.

* Corresponding author.
E-mail addresses: saytomona@gmail.com (M. Panigrahi), sourabh.bharti@mtu.ie (S. Bharti), arunsharma@igdtuw.ac.in (A. Sharma).

[1] The terms 'FL round' and 'communication round' are used interchangeably throughout the paper.

https://doi.org/10.1016/j.compeleceng.2023.108900
Received 1 April 2023; Received in revised form 31 July 2023; Accepted 31 July 2023
Available online 18 August 2023

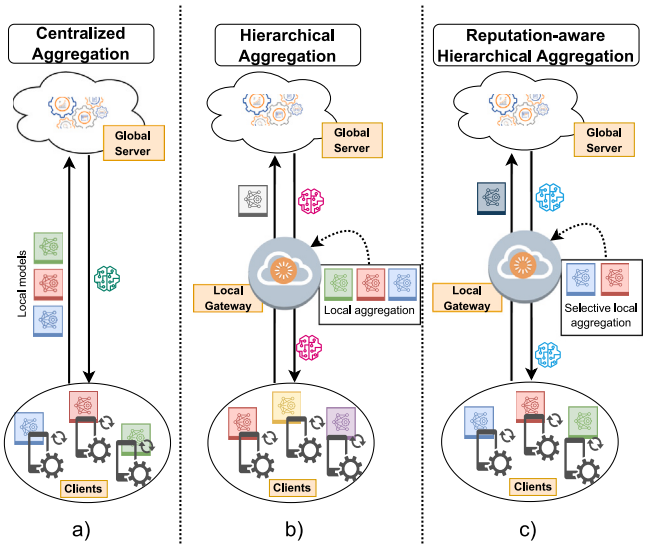


Fig. 1. (a) Centralized, (b) Hierarchical, and (c) Proposed aggregation framework in FL (FedRaHa).

1.1. Issues with existing aggregation approaches

The existing federated learning aggregation frameworks such as proposed in [1,2] are based on a centralized architecture, i.e., the aggregation is performed by the centralized server which acts as an orchestrator for the model training process as shown in Fig. 1(a). As a result of this centralized aggregation, a worn-out server acts as the sole point of contact for all clients, which could result in data loss in the event of a breakdown. Additionally, due to the constrained computational capabilities, point-to-point communication between FLClients and server is not always reliable. Due to the possibility of FLClients' updates being lost during the model exchange, this may impact the correctness of the global model. Additionally, since every FLClient update is transmitted to the global server during each FL round, the communication cost incurred by FLClients rapidly rises. Reputation is employed as a decision variable for client selection in specific proposals, such as [3], before aggregating their local models.

Apart from the centralized aggregation, the edge-based hierarchical aggregation approaches (Fig. 1(b)) such as proposed in [4], and [5] locally aggregate the FLClients' updates at the edge server[2] before sending them to the centralized server, however, such approaches fail to reduce the communication between the FLClients and edge server as FLClients are required to send every model updates to the edge server. In some situations, this can also result in increased model convergence time because of the two-step aggregation.

1.2. Research questions

Fig. 1 shows the difference between centralized, hierarchical, and proposed FL framework (FedRaHa). The centralized aggregation scheme performs only one level of aggregation at the global server. To minimize the communication between FLClients and the global server, the hierarchical aggregation scheme employs two-level aggregation in which the local gateways perform the first-level aggregation of FLClients' updates before sending the aggregated values to the global server (for the second level of aggregation). As FLClients are considered resource-constraint edge devices in a cross-device FL setup, the proposed aggregation framework (FedRaHa) attempts to further reduce the communication between FLClients and the local gateways by investigating the following research questions.

RQ1: Do all FLClients' updates carry equally important information to be shared with the local gateway?

RQ2: Can the communication between FLClients and the local gateway be reduced by selecting the most important updates for first-level aggregation?

RQ3: What will be the impact of this on the global model test accuracy?

1.3. Research contributions

In light of the above discussion, this paper makes the following research contributions:

[2] The terms 'edge server' and 'local gateway' are used interchangeably throughout the paper.

Table 1
Qualitative comparison of FedRaHa with existing aggregation mechanisms.

Mechanism	Aggregation framework	Local aggregation	Global aggregation	Selective update	Resource-aware	Communication cost
[1,2,8]	Centralized	No	Yes	No	No	High
[10]	Centralized	No	Yes	No	No	Moderate
[3,9,11]	Centralized	No	Yes	Yes	No	Moderate
[7,12]	Centralized	No	Yes	No	No	Low
[4,5,13–16]	Hierarchical	Yes	Yes	No	No	Moderate
[17]	Hierarchical	Yes	Yes	Yes	No	Moderate

1. A novel reputation-aware hierarchical aggregation framework (FedRaHa) for cross-device FL with the objective to minimize the communication between FLClients and the local gateway during the FL process.
2. A reputation-aware FLClients' updates selection algorithm to evaluate the effectiveness of clients' updates before aggregation.
3. The performance of FedRaHa is evaluated using two benchmark image datasets: MNIST and Fashion-MNIST, and one real-world Chest Xray dataset and results reveal that FedRaHa outperforms the state-of-the-art in terms of model convergence time, message exchanges, and communication cost, etc.

The remainder of this paper is organized as follows. The related research is discussed in Section 2 and system components and problem definition are discussed in Section 3. The proposed framework FedRaHa is discussed in Section 4. Section 5 focuses on the performance evaluation methodology whereas the simulation results are discussed in Section 6. Finally, Section 7 concludes the paper with future directions.

2. Background and related research

FL is defined as a learning technique that employs a K number of clients with their own training data and heterogeneous compute and communication resources. For an FL round n, the global server selects a fraction (C) of clients from K, i.e., $\lceil K \times C \rceil$ for FL training. All the selected clients train the global model using their own local data and send their model updates to the global server. The global server aggregates all the received model updates and sends the updated model to the next set of selected clients for the next FL round. This process iterates till the model converges or reaches a deadline for the number of FL rounds.

2.1. Review on related work

As the clients in a cross-device FL may grow to a large number, the number of model exchanges between clients and the global server can also grow significantly; contributing to an overburdened global server and increased model convergence time. To this end, various aggregation schemes are proposed in the literature as follows.

2.1.1. Centralized aggregation

In this approach, the client updates are aggregated in a centralized fashion i.e., in one place (global server/cloud). This type of aggregation is also known as one-step aggregation as the aggregation is done only once in each FL round (Fig. 1). The classical FL [1] adopts centralized aggregation that is repeated for several rounds until model convergence. An adaptive aggregation algorithm was proposed in [6] for varying clients' participation and refresh rates. Another aggregation algorithm was proposed in [2] where the accuracy of the local model update is used as an aggregation weight in updating the global model. A Robust Federated Aggregation (RFA) mechanism was proposed in [7] considering the aggregation of updates using the geometric median with a Weiszfeld-type algorithm. The performance of FL with secure aggregation using PySyft was analyzed in [8] and an asynchronous FL mechanism was proposed to improve the performance. A reputation-aware hierarchical aggregation technique was proposed in [3] for mobile platforms, whereas a selective model aggregation approach was proposed in [9] to select and send fine local DNN models to the global server by evaluating the local image quality and computation capability. A quantization-based aggregation approach was proposed in [10] whereas an accuracy threshold adaptive algorithm was proposed in [11] to select high-quality local models for aggregation.

Such centralized aggregation approaches result in high communication cost between clients and the global server as all the client updates are sent to the global server irrespective of their effectiveness in increasing the global model accuracy. This results in an increase in model convergence time and resource usage, making the aggregation process more challenging [1] while aggregating all the clients' updates in every FL round.

2.1.2. Hierarchical aggregation

To overcome the problems faced in centralized aggregation, a two-step aggregation, known as hierarchical aggregation is proposed (shown in Fig. 1). A client-edge-cloud hierarchical FL framework was proposed in [5] in which multiple edge servers performed partial model aggregation, whereas a cluster-based hierarchical aggregation technique for edge computing was proposed in [4]. The problem of edge aggregation interval control and time allocation was formulated as a combined problem in [13] for a hierarchical federated learning system. A selective model aggregation approach was proposed in [17] to select the good quality local models for aggregation, whereas a novel FL mechanism using a Multi-Local and Multi-Global (MLMG) model aggregation technique

Table 2
Key symbols and their definition(s).

Symbol	Definition
K	Total number of clients
C	Fraction of clients to be selected for FL training
N	Total number of FL rounds
L	Total number of local networks
$\mid D^l \mid$	Number of training samples in local gateway l
$\mid D \mid$	Total number of data samples
w^n	Weight of global model in FL round n
G	Total number of local gateways
R	Number of FLClients from lth local gateway
FC_l^n	Set of R FLClients in lth local gateway for FL round n
$(W_l)_{ud}^n$	Set of model updates (ud) from R FLClients for FL round n
$(W_l)_{sud}^n$	Set of selected model updates (sud) for aggregation for FL round n
(D^{fc})	Number of training samples of FLClient fc
W_{fc}	Model update (weight) of FLClient fc
R_{score}^{fc}	Reputation score of FLClient fc
cos_{score}^{fc}	Cosine score of FLClient fc
Ql^{fc}	Test data accuracy of fc's local model
Qt^{fc}	Number of data samples of fc's local data used for training
$\mid correct^{fc} \mid$	Number of correctly predicted test data samples of FLClient fc
$\mid D_{test}^{fc} \mid$	Number of test data samples of FLClient fc
E	Number of epoch per FL round
B	Local batch size
α_1 and α_2	Application/use-case dependent tuning parameters

was proposed in [14] to train the non-IID (non-Independent and Identically Distributed) user data with clustering methods. An edge FL (EdgeFed) method was proposed in [15], in which edge server aggregates the local model updates to improve the learning efficiency and decrease the global communication frequency. A fog node-based aggregation mechanism was proposed in [18] for IoT devices whereas, a cluster-based aggregation technique was proposed in [16]. Table 1. presents the summary of the comparison of proposed FedRaHa with existing aggregation mechanisms. Apart from these categories, a decentralized aggregation approach for FL was proposed in [19] and a multi-center aggregation mechanism was proposed in [20], where multiple global models are learned from data and derive the optimal matching between users and centers.

Synthesis: Due to local aggregation at the local gateway, hierarchical aggregation is able to reduce communication costs when compared with centralized aggregation. However, there is still scope for improvement in terms of communication between resource-constrained FLClients and the local gateways in a cross-device FL setup. This communication can be reduced by checking the effectiveness of FLClients' local updates as to whether all of them contribute to the global model test accuracy.

3. System architecture

In the proposed FedRaHa framework (Fig. 1(c)), there are L local networks managed by a single global server. Each local network consists of an equal number of clients managed by a single local gateway. Let $D = \sum_{l=1}^{L} \mid D^l \mid$ be the training data-set where $\mid D^l \mid$ represents the number of training data samples in local gateway l and $\mid D \mid$ represents the total number of training data samples. Each local gateway selects the clients from its network for FL training, known as FLClients. The FLClients communicate with the global server via their respective local gateways. All the key symbols used throughout the paper are defined in Table 2. The key components of the system are:

- *Global Server*: A single global server acts as a centralized controller for the FL process. The responsibility of a global server includes (1) the creation of the initial global model, (2) communication with local gateways, and (3) aggregation of locally aggregated updates received from the local gateways.
- *Local Gateway*: Local gateways be the controllers of their own local networks, and act as an interface between the clients and the global server. The responsibility of a local gateway includes (1) selection of FLClients, (2) selection of FLClients' whose updates will be included in local aggregation, and (3) aggregation of selected FLClients' updates and sending them to the global server.
- *Clients*: The clients are the end devices, e.g., resource-constrained IoT devices, which are in possession of a particular dataset. The clients are directly connected to their respective local gateways. The responsibilities of the clients include (1) performing FL training on their local datasets, once selected as FLClient by the local gateway, and (2) calculating their respective reputation scores.

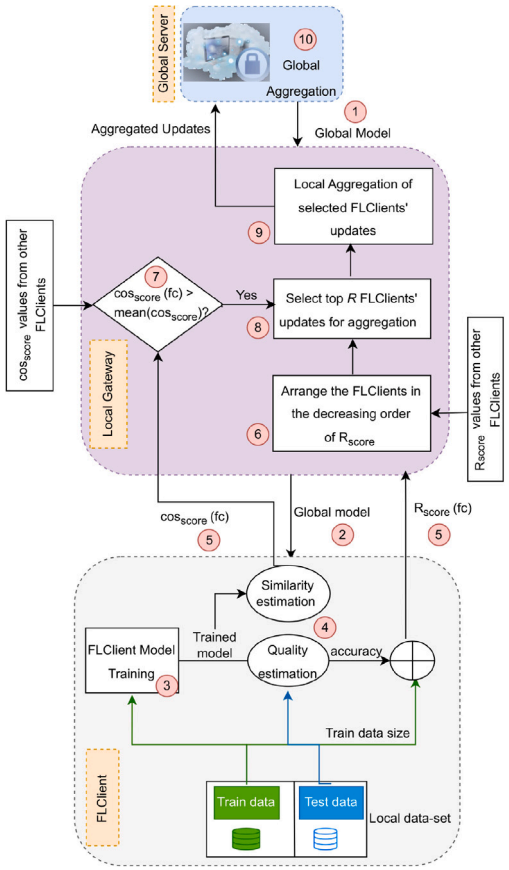


Fig. 2. Flow graph of FedRaHa.

3.1. Problem definition

For FL round n, the global server creates and sends the initial global model with weight w^n to the connected local gateways represented by $G = \{G_1, G_2, G_3, \ldots, G_L\}$ where G be the set of L local gateways. For an FL round n, if the global server requires $\lceil K \times C \rceil$ FLClients, then each local gateway selects R number of FLClients from its local network, where $R = \lceil K \times C \rceil / |G|$. The values of the hyper-parameters K and C are provided by the global server. Let, $FC_l^n = \{FC_{l_1}, FC_{l_2}, FC_{l_3}, \ldots..FC_{l_R}\}$ represents R FLClients in lth local gateway for FL round n and $(W_l)_{ud}^n = \{W_{FC_{l_1}}, W_{FC_{l_2}}, W_{FC_{l_3}}, \ldots..W_{FC_{l_R}}\}$ represents the model updates of its R FLClients for FL round n. If N represents the total number of FL rounds, then the problem of minimizing FLClients' selected model updates $(W_l)_{sud}^n$ addressed by FedRaHa can be defined as:

$$\min \sum_{n=1}^{N} \sum_{l=1}^{L} | (W_l)_{sud}^n | \tag{1}$$

s.t.

$$(W_l)_{sud}^n \subseteq (W_l)_{ud}^n \ \ and \ \ | (W_l)_{sud}^n | \leq R \tag{2}$$

4. FedRaHa: Proposed aggregation framework

The proposed FedRaHa framework with a detailed explanation of every step is presented in this section. The system components of FedRaHa are shown in Fig. 1(c), where the FLClients are directly connected to their respective local gateways, which are connected to the global server. The local gateways are the key enablers for FLClient and their update(s) selection. The process flow of reputation-aware FLClient update selection in FedRaHa is shown in Fig. 2. The overall FL process enabled by FedRaHa is explained in Protocol 1 whereas the computational steps involved in FedRaHa are depicted in Algorithm 1. FedRaHa selects the most important FLClients' updates for aggregation with the objective to minimize the communication cost between FLClients and the local gateways during the FL process. The importance of a FLClient's update is estimated in terms of its reputation score and its local model's

Table 3
Key simulation parameters with values.

Parameter	Value
K	100
C	0.5 (MNIST/Fashion-MNIST), 0.1 (Chest Xray)
G	5 (MNIST/Fashion-MNIST), 2 (Chest Xray)
N	50 (MNIST/Fashion-MNIST), 10 (Chest Xray)
E	5
B	100 (MNIST/Fashion-MNIST), 50 (Chest Xray)
α_1 and α_2	0.5
Learning rate	0.0001 (MNIST,Fashion-MNIST), 0.0005 (Chest Xray)
Effective capacitance	10^{-28}
CPU cycle	2–3
CPU frequency	1–2
Total round time	8 min
Initial energy of IoT device	7000 J
Energy/bit consumed by the transmitter electronics	50×10^{-9} J/bit
Energy consumed in the transmit op-amp	100×10^{-12} J/bit/m^2
Transmission range of IoT device	30
Power index	2
Bit rate	1 Mbit/s
Data rate of IoT device	1–3 Mbps

similarity with the global model in the current round (Section 4.2.1). The reputation score estimation and similarity calculation are performed by individual FLClients and the resulting information is shared with their respective local gateways (Fig. 2). Based on the received information, local gateways select the most reputed FLClients' updates for local aggregation which helps to reduce the message exchanges (i.e., communication cost) without impacting the global model test accuracy (Fig. 5).

4.1. FLClient selection

Being the first step of FL, the FLClient selection in the proposed FedRaHa is performed by respective local gateways. The clients with enough computational resources to complete the training before a deadline are selected as FLClients [21]. For every FL round n, every local gateway selects R FLClients. To this end, the energy and time model to estimate the consumed energy and training time, respectively are adopted from [21] and the values of hyper-parameters used in the simulation are presented in Table 3. As shown in Algorithm 1 (Step 8), the FLClients train the model using their local dataset (D^{fc}) and store the updated weights in W_{fc}.

4.2. Reputation-aware FLClient update selection

For each FL round, this is a two-step process for reputation-aware FLClient update selection at local gateways. In the first step, each FLClient (fc) estimates its reputation score (R^{fc}_{score}) and shares it with its respective local gateway. To minimize the communication cost between FLClients and the local gateway, only the most reputed FLClient updates are selected for aggregation in the second step.

4.2.1. Reputation and similarity estimation

For an FL round n, the reputation of an FLClient is the measure of its local model performance as the performance of the globally trained model largely depends upon the performance of the FLClients' locally trained models. The reputation score (R^{fc}_{score}) of a FLClient fc, is modeled as a function of (1) the quality (Ql^{fc}) of its local model update and (2) quantity (size) (Qt^{fc}) of its dataset contributed in the current round. It is assumed that all the FLClients are trusted and use correct training dataset size information for reputation estimation. Ql^{fc} is measured in terms of its accuracy against the test data whereas the Qt^{fc} is the number of data samples of fc's local data used for training in the current round. Since FLClients can contribute a different number of samples, the normalized samples are taken for a fair comparison.

$$R^{fc}_{score} = \alpha_1 Ql^{fc} + \alpha_2 Qt^{fc} \tag{3}$$

While Ql^{fc} represents the test data accuracy of fc's local model, which can be calculated as $\frac{|correct^{fc}|}{|D^{fc}_{test}|}$ where, $|\ correct^{fc}\ |$ is the number of correctly predicted test data samples of fc, $|\ D^{fc}_{test}\ |$ is the number of test data samples of fc. Qt^{fc} is calculated as $\frac{|D^{fc}|}{|D^{l}|}$ where, $|\ D^{fc}\ |$ is the number of training data samples of fc, $|\ D^{l}\ |$ is the total collective number of training data samples in the local gateway $l = \sum_{r=1}^{R} |\ D^{fc_l r}\ |$. α_1 and α_2 are application/use-case dependent tuning parameters that can be used to give preference to the quality (Ql) and quantity (Qt) of the training data respectively, in order to estimate the reputation accordingly. This work considers the values of α_1 and α_2 as constants (i.e., 0.5 & 0.5) to register equal importance to both Ql and Qt for all FLClients. However, these values can be tuned to support use cases that require giving more weightage to Ql or Qt of the FLClient by assigning different values to α_1 and α_2 such that $\alpha_1 + \alpha_2 = 1$.

Algorithm 1 Reputation-aware Hierarchical Aggregation Algorithm

1: initialize $w^0, (W_l)^n_{ud} = \emptyset, (W_l)^n_{sud} = \emptyset$
2: **for** every round N = 1,2,3, ... n **do**
3: global server sends w^n to L local gateways (G)
4: **for** every G_l in G **do**
5: // FLClient Selection
6: G_l selects and sends w^n to R FLClients (FC^n_l)
7: **for** every fc in FC^n_l **do**
8: $W_{fc} \leftarrow FCUpdate(fc, w^n)$ //FLClient Local Training
9: $(W_l)^n_{ud} = (W_l)^n_{ud} \cup W_{fc}$ //FLClients' Updates
10: calculates R^{fc}_{score} and cos^{fc}_{score} using eq. 3 and 4 respectively
11: sends R^{fc}_{score} and cos^{fc}_{score} to G_l
12: **end for**
13: $RFC_l \leftarrow$ sort fc in the decreasing order of R^{fc}_{score}
14: **for** every $fc \in RFC_l$ **do**
15: **if** $cos^{fc}_{score} \geq mean(cos_{score})$ **then**
16: $w \leftarrow W_{fc}$
17: **end if**
18: $(W_l)^n_{sud} = (W_l)^n_{sud} \cup w$ //FLClients' Updates Selection
19: **end for**
20: $I \leftarrow |\,(W_l)^n_{sud}\,|$
21: $W^n_l \leftarrow \sum_{i=1}^{I} \frac{|D^{FC_{l_i}}|}{|D^l|} W^n_{FC_{l_i}}, \forall W^n_{FC_{l_i}} \in (W_l)^n_{sud}$ //Local Aggregation
22: **end for**
23: $W^{n+1} \leftarrow \sum_{l=1}^{L} \frac{|D^l|}{|D|} W^n_l$ //Global Aggregation
24: **end for**
25: FCUpdate (fc, w) :
26: $B \leftarrow$ split D^{fc} into batches of size B
27: **for** every epoch e from 1 to E **do**
28: **for** batch $b \in B$ **do**
29: $w \leftarrow w - \eta \delta l(w; b)$
30: **end for**
31: **end for**
32: return w to the local gateway

In addition to the reputation score (R_{score}), every FLClient also estimates the similarity between its local model and the global model for the current round. This similarity score (cos_{score}) is measured using the cosine similarity. *Cosine similarity is a popular metric used to measure the similarity between two vectors in an inner product space, which calculates the cosine of the angle between these two vectors, indicating the degree to which they align in the same direction and establish a similarity relationship.* If the client's local model and global model are considered as two vectors, then their cosine similarity helps in determining the direction in which the model parameters are moving towards the global optimum and can be calculated as:

$$cos^{fc}_{score} = \frac{\langle \Delta W^n_{fc}, \Delta W^n \rangle}{\|\Delta W^n_{fc}\| \|\Delta W^n\|} \tag{4}$$

where, W^n_{fc} and W^n represent the weight parameters of fc's local model and the global model, respectively for round n. A large cos^{fc}_{score} value indicates that the local model gradients have a direction similar to the global model gradients. Both R_{score} and cos_{score} for every FLClient are shared with their respective local gateways.

4.2.2. FLClient update selection

Every local gateway arranges its respective FLClients in the decreasing order of their R_{score} in the set RFC_l (Algorithm 1). For FLClients' updates selection, the local gateway further selects the most important updates by sequentially selecting FLClients from RFC_l, whose similarity scores are greater than or equal to the average similarity score of the local network. The sequential selection terminates when the local gateway reaches the maximum limit i.e., R. This is depicted by Steps 15 and 18 of Algorithm 1.

4.3. Aggregation

FedRaHa takes advantage of the hierarchical aggregation and employs two-step aggregation to minimize the communication between the local gateways and FLClients during the FL process. Each local gateway performs the local aggregation of selected

FLClients' updates (Algorithm 1, Step 21) and shares the aggregated updates with the global server for global aggregation (Algorithm 1, Step 23). This minimizes the message exchanges between FLClients and local gateways which results in less resource consumption of the resource-constrained FLClients.

Protocol 1 Federated model training protocol in action with FedRaHa

1: *Initialization*: The global server generates an initial model and distributes it to the local gateways.
2: *Distributed FLClient Selection and Model Distribution*: Each local gateway selects maximum R resource-efficient FLClients and forwards the global model to the selected FLClients inside their network.
3: *FLClients' Update*: FLClients update the shared model.
4: *FLClients' Upload* : Each FLClient calculates its reputation score and cosine score and shares them with the local gateway. The local gateway employs a reputation-aware approach to select the appropriate FLClients' update for aggregation.
5: *Local Aggregation*: Each local gateway aggregates the selected FLClients' update.
6: *Global Aggregation*: The global server receives the locally aggregated updates through the local gateways, and performs global aggregation as shown in Algorithm 1 (Step 23).
7: Steps 2 to 6 are repeated for several rounds until the model achieves the desired performance or the final deadline.

5. Performance evaluation

This section discusses the simulation settings, the dataset and distribution settings, the global model architecture, the baseline approaches, and the evaluation criteria.

5.1. Simulation setting

PySyft[3] is used to simulate FedRaHa, and the clients/local gateways communicate with one another as virtual workers. The simulation setup takes into account a single global server, five local gateways for MNIST/Fashion-MNIST datasets and two local gateways for Chest Xray dataset, and 100 clients (20 clients (MNIST/Fashion-MNIST) and 50 clients (Chest Xray) in each local network). Each client is assumed as a resource-constrained IoT device and it is anticipated that each local gateway would choose the same number of clients (R) for FL training from its local network in each round. The simulation is performed in Google Colab using TPU and 12 GB RAM. Stochastic gradient descent (SGD) model optimizer is used to update the model parameters during training and negative log-likelihood (nll) loss function is used in the experiment. The key simulation parameters are included in Table 3., together with the values that were taken into account during the experiments.

5.2. Dataset and data distribution settings

In order to perform extensive testing of the proposed FedRaHa, the two benchmark datasets: MNIST [22] and Fashion-MNIST [23] are used for the experiment. Both MNIST and Fashion-MNIST consist of 60,000 training images and 10,000 testing images of handwritten digits and fashion products respectively. The training dataset of MNIST and Fashion-MNIST are distributed among 100 clients in IID (Independent and Identically Distributed) and Non-IID settings. In the IID setting, an equal number of images are assigned to every client randomly from the whole training dataset. In the Non-IID setting as adopted from [1], the whole training dataset of 60,000 images are first sorted by digit label and 2000 groups with 30 sample images per group are formed, out of which clients are assigned sample images from 20 groups. This Non-IID setting ensures that each client can get sample images from a maximum of 2-digit labels, which helps in validating the proposed approach by accurately replicating a real-world scenario where clients may possess entirely distinct datasets. For an FL round n, C is taken as 0.5, and maximum $\lceil K \times C \rceil = 50$ clients in total, i.e., 10 clients from each local network can be selected for FL training.

5.3. Global model architecture

The convolutional neural network used by the global model has two 5×5 convolution layers. There are 20 and 50 channels in each of the convolution layers, respectively [24]. ReLU activation function is used to activate each layer, followed by 2×2 max pooling and two fully-connected layers.

[3] https://docs.openmined.org/pysyft/.

5.4. Baseline approaches

The baseline approaches used for comparative evaluation are:

1. Classical FL protocol [1], where all the FLClients' updates are sent to the global server for aggregation in every FL round. This approach is referred to as Fed-SAVG.
2. Hierarchical FL [5], where all the FLClients' updates are sent to their respective local gateways for local aggregation. Then, all the locally aggregated updates are sent to the global server for final aggregation in every FL round. This approach is referred to as FedH-GSAVG.
3. Hierarchical FL [5] with server aggregation is a variation of FedH-GSAVG, where all the FLClients' updates are sent to the global server through the respective local gateway for aggregation as in Fed-SAVG. This approach is referred to as FedH-SAVG, which follows a hierarchical structure only for communication but performs only one step aggregation, i.e., at the global server. FedH-SAVG is considered to show the effect of two-step aggregation over one-step aggregation.

5.5. Evaluation criteria

The performance of the FedRaHa framework is assessed using the following criteria:

1. **Model convergence time:** It is usually measured in the form of time taken for the model to achieve an ideal loss during training. Faster global model convergence is beneficial for the resource-constrained FLClients in a cross-device FL setting (to answer *RQ1*).
2. **Number of FL rounds to achieve the desired test data accuracy:** It is important for an aggregation framework to train a model within minimum number of FL rounds without compromising on its accuracy (to answer *RQ1* & *RQ3*).
3. **Number of message exchanges during FL process:** If there are more message transfers throughout the FL process between the FLClients and local gateways/global server, the FLClients quickly exhaust their available resources, increasing the cost of computation. Thus, the number of message transfers should be minimized in each FL round while retaining the global model's test data accuracy (to answer *RQ2*).
4. **Communication cost during FL process:** The network bandwidth is also an important resource in a cross-device FL setting. Therefore, redundant and unnecessary information exchange should be avoided for efficient bandwidth utilization in the framework (to answer *RQ2*).

6. Results and analysis

This section discusses the simulation results obtained through the experimental analysis of the proposed FedRaHa on the evaluation criteria discussed in Section 5.5. A thorough comparative performance analysis of FedRaHa with baseline approaches is also discussed in detail.

6.1. Model convergence time

Fig. 3 shows the comparative analysis of all the approaches in terms of the number of FL rounds an FLClient needs to complete in order to converge to an optimal loss value. The results show that FedRaHa and FedH-SAVG take minimum and maximum time, respectively to converge as compared to all the approaches in both IID and Non-IID settings. This is due to the fact that FedRaHa makes resource-aware FLClient selection which helps minimize FLClient(s) drop-out rate during the FL process. Minimum drop-outs prevent information loss and thus help expedite model convergence. In addition, the provisions in FedRaHa to select and aggregate the most important FLClients' model updates minimize the total time taken by an FL round; resulting in faster convergence. This answers the *RQ1* listed in Section 1.2 that all local model updates are not equally important and selecting the most important local model updates for the aggregation helps in minimizing the global model convergence time as compared to all other approaches. On the other hand, the baseline approaches select the FLClients without taking into account the clients' restricted resources, which leads to a higher FLClients drop-out rate during the FL process. Also, aggregating all the client updates increases the time taken by an FL round and contributes to slower global model convergence in baseline approaches.

6.2. Number of FL rounds to achieve the desired test data accuracy

The results shown in Fig. 4 depict the comparative analysis of all four approaches in terms of the total number of FL rounds needed by the FLClients to reach desired global model test data accuracy. It is evident from the results that FedRaHa gives the highest accuracy i.e., 86% as compared to the other three approaches for both IID and Non-IID settings. This result further re-asserts the fact that FedRaHa is able to achieve a faster global model convergence without suffering on the accuracy aspect. This answers the *RQ1* & *RQ3* listed in Section 1.2 that not all local model updates carry equal importance and selecting the most important local model updates for the aggregation not only maintains the global model test data accuracy but also achieves it in less number of FL rounds as compared to the baseline approaches. Achieving desired test data accuracy level in fewer FL rounds also results in reduced computation and communication tasks on the part of resource-constrained FLClients which is beneficial for cross-device FL setup.

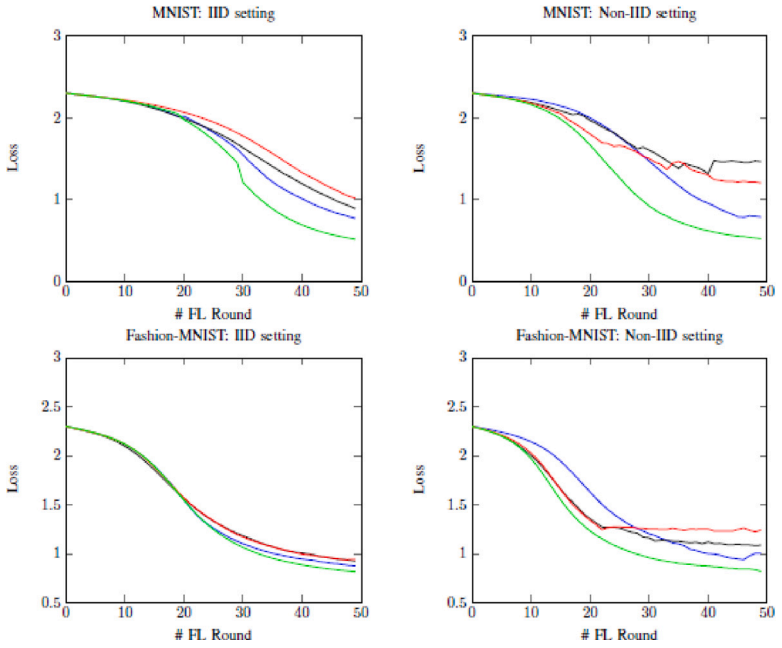


Fig. 3. Convergence time analysis of MNIST and Fashion-MNIST dataset in IID and Non-IID setting. Fed-SAVG [1] ——, FedH-GSAVG [5] ——, FedH-SAVG [5] ——, FedRaHa ——.

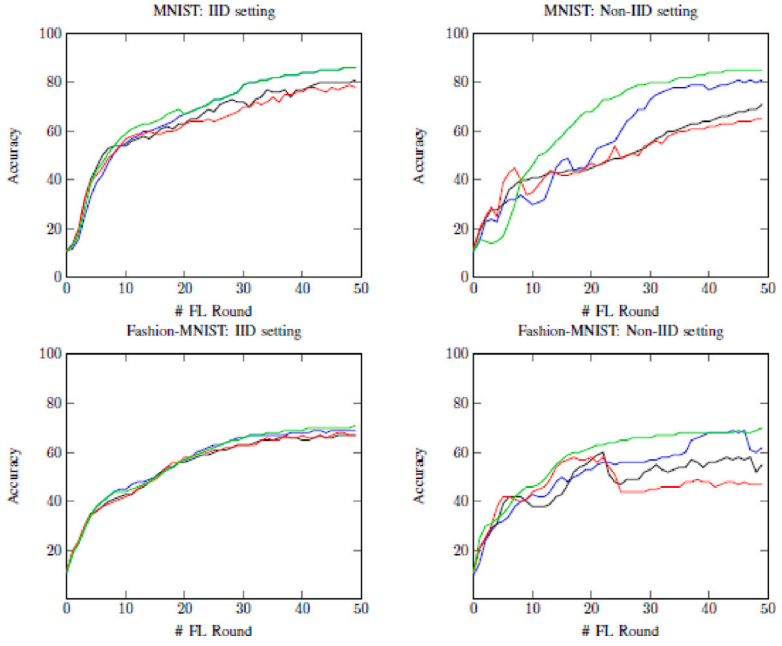


Fig. 4. Accuracy analysis of MNIST and Fashion-MNIST dataset in IID and Non-IID setting. Fed-SAVG [1] ——, FedH-GSAVG [5] ——, FedH-SAVG [5] ——, FedRaHa ——.

6.3. Number of message exchanges during FL process

The results shown in Fig. 5 depict the comparative analysis of all four approaches in terms of the total number of message exchanges between FLClients and the local gateways/global server in each FL round during the FL process. The cumulative number of message exchanges to reach a defined test data accuracy is observed for 50 FL rounds. Message exchanges include global model distribution messages between the global server/local gateways and FLClients and local model update messages between FLClients

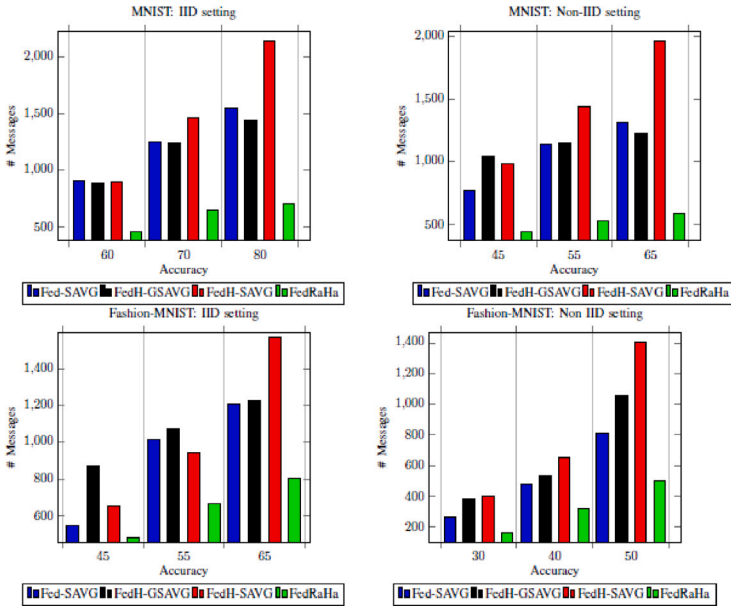


Fig. 5. Analysis of the number of message exchanges of MNIST and Fashion-MNIST dataset in IID and Non-IID setting.

and local gateways/global server. Each bar in the bar graph of Fig. 5 shows the cumulative number of message exchanges between the global server/local gateways and FLClients to reach the desired test data accuracy using a particular approach throughout the FL process. The results reveal that FedRaHa and FedH-SAVG employ minimum and maximum number of message exchanges, respectively, to reach the desired test data accuracy compared to other approaches in both IID and Non-IID settings. This is because FedRaHa eliminates the unimportant FLClient updates before first-level aggregation, which reduces the communication between FLClients and the local gateways and results in an overall reduction in the number of message exchanges. This answers the *RQ2* listed in Section 1.2 that unimportant local client updates may be eliminated by selecting the important local model updates for first-level aggregation which reduces the communication between FLClients and local gateways. On the other hand, the baseline approaches contribute more number of message exchanges as every local update from each FLClient is sent to the local gateways/global server for aggregation. The less number of message exchanges in FedRaHa not only minimizes the communication cost (Fig. 6) but also reduces the bandwidth consumption.

6.4. Communication cost during FL process

The results shown in Fig. 6 depict the comparative analysis of all four approaches in terms of communication cost per FL round during the FL process. The communication cost incurred in an FL round is quantified as the total volume of data (measured in MB) transmitted in the global model distribution messages between global server/local gateways and FLClients and local model update messages between FLClients and local gateways/global server. It is evident from the result that FedRaHa and FedH-SAVG observe minimum and maximum communication cost respectively in both IID and Non-IID settings. This answers the *RQ2* listed in Section 1.2 that the provision of selective local model updates in FedRaHa minimizes the communication inside the local network which in turn reduces the communication cost during the FL process as explained in Section 6.3. On the other hand, for the baseline approaches all local model updates are sent to the global server and resulting in increased communication cost.

6.5. Results on real-world dataset

Further experiments on a Chest Xray dataset [25] are performed to observe the performance of FedRaHa in a real-world scenario with a heterogeneous dataset. The Chest Xray dataset contains 5856 Xray images divided in two classes : Normal and Pneumonia. The Chest Xray images of 1 to 5 year old pediatric patients were selected from Guangzhou Women and Children's Medical Center for this dataset. The training set contains 5232 images and the testing set contains 624 images which are equally divided among 100 clients in our simulation setting. The evaluation criteria for these experiments is same as discussed in Section 5.5 and the values of hyper-parameters used in the simulation are presented in Table 3. Fig. 7(a) shows the comparative analysis of all the approaches in terms of the number of FL rounds an FLClient needs to complete in order to converge to an optimal loss value. The results show that FedRaHa and FedH-SAVG take minimum and maximum time, respectively to converge as compared to all the approaches. The results shown in Fig. 7(b) depict the comparative analysis of all four approaches in terms of the total number of FL rounds needed by the FLClients to reach desired global model test data accuracy. It is evident from the results that FedRaHa gives the highest accuracy i.e., 79%

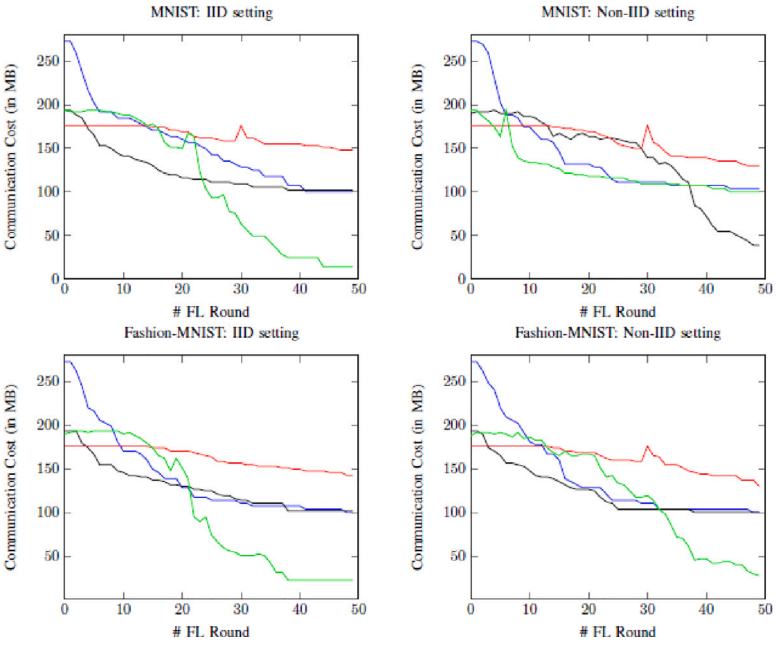
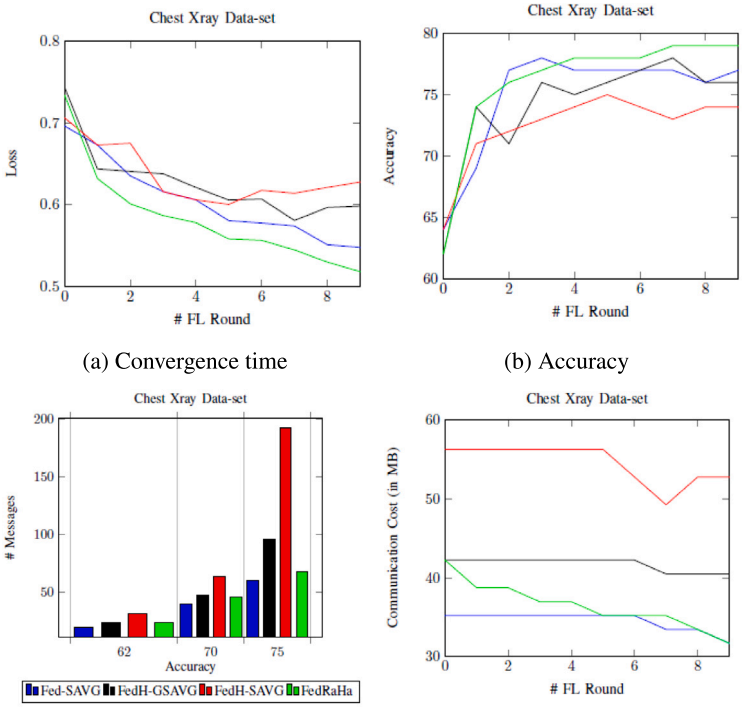


Fig. 6. Analysis of communication cost of MNIST and Fashion-MNIST dataset in IID and Non-IID setting. Fed-SAVG [1] ——, FedH-GSAVG [5] ——, FedH-SAVG [5] ——, FedRaHa ——.



(a) Convergence time (b) Accuracy

(c) Number of message exchanges (d) Communication cost

Fig. 7. Comparative analysis of FedRaHa using Chest Xray dataset: Fed-SAVG [1] ——, FedH-GSAVG [5] ——, FedH-SAVG [5] ——, FedRaHa ——.

as compared to the other three approaches. The results shown in Fig. 7(c) depict the comparative analysis of all four approaches in terms of the total number of message exchanges between FLClients and the local gateways/global server in each FL round during the FL process. The cumulative number of message exchanges to reach a defined test data accuracy is observed for 10 FL rounds.

The results reveal that FedRaHa and FedH-SAVG employ minimum and maximum number of message exchanges, respectively, to reach the desired test data accuracy compared to other approaches. The results shown in Fig. 7(d) depict the comparative analysis of all four approaches in terms of communication cost per FL round during the FL process. It is evident from the result that Fed-SAVG and FedRaHa observe minimum communication cost, whereas FedH-SAVG observes maximum communication cost.

7. Conclusion and future work

A reputation-aware hierarchical aggregation framework (FedRaHa) for FL is proposed in this paper. FedRaHa provides a new perspective to the aggregation in FL by considering the effectiveness of FLClients' updates before considering them for global model aggregation. FedRaHa provides methods to estimate the effectiveness of the FLClient update and it has been observed that not all FLClients contribute equally towards the development of a global model which can be generalized to a large number of devices, and discarding less effective updates leads to saving scarce communication and computational resources of the FLClients without compromising on the global model test accuracy. The performance of FedRaHa is compared with existing centralized and hierarchical aggregation mechanisms such as Fed-SAVG, FedH-GSAVG, and FedH-SAVG on parameters such as communication cost and global model test accuracy. FedRaHa improves test accuracy by 6.17% in MNIST IID, 4.94% in MNIST Non-IID, 2.9% in Fashion-MNIST IID, 12.9% in Fashion-MNIST Non-IID and 2.6% in Chest Xray dataset. FedRaHa reduces the number of message exchanges by 50% in MNIST IID, 52.37% in MNIST Non-IID, 33.22% in Fashion-MNIST IID, 30% in Fashion-MNIST Non-IID, and 29.17% in Chest Xray dataset. Moreover, FedRaHa reduces communication cost by 13% in MNIST IID, 9.83% in MNIST Non-IID, 27.15% in Fashion-MNIST IID, 13% in Fashion-MNIST Non-IID, and 11.51% in Chest Xray dataset. Therefore, it is concluded that FedRaHa is able to reduce the model convergence time, the number of FL rounds to reach the desired test data accuracy, the number of message exchanges, and communication costs during the FL process. The experiments can be extended by implementing the proposed approach in a real-world setting and having online data generated from different clients while they train the global model. Authors are planning to experiment on some real-world use-cases of the proposed approach as part of their future work.

CRediT authorship contribution statement

Monalisa Panigrahi: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Sourabh Bharti:** Supervision, Methodology, Validation, Writing – review & editing. **Arun Sharma:** Supervision, Writing - review.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: 20th International conference on artificial intelligence and statistics (AISTATS). 2017, p. 1273–82.
[2] Xu R, Feng X, Zheng H. Robust model aggregation for federated learning with heterogeneous clients. In: 2021 7th International conference on computer and communications (ICCC). IEEE; 2021, p. 1606–10.
[3] Wang Y, Kantarci B. Reputation-enabled federated learning model aggregation in mobile platforms. In: ICC 2021-IEEE international conference on communications. IEEE; 2021, p. 1–6.
[4] Wang Z, Xu H, Liu J, Huang H, Qiao C, Zhao Y. Resource-efficient federated learning with hierarchical aggregation in edge computing. In: IEEE INFOCOM 2021-IEEE conference on computer communications. IEEE; 2021, p. 1–10.
[5] Liu L, Zhang J, Song S, Letaief KB. Client-edge-cloud hierarchical federated learning. In: ICC 2020-2020 IEEE international conference on communications (ICC). IEEE; 2020, p. 1–6.
[6] Jin H, Yan N, Mortazavi M. Simulating aggregation algorithms for empirical verification of resilient and adaptive federated learning. In: 2020 IEEE international conference on big data computing, applications and technologies (BDCAT). IEEE; 2020, p. 124–33.
[7] Pillutla K, Kakade SM, Harchaoui Z. Robust aggregation for federated learning. 2019, arXiv preprint arXiv:1912.13445.
[8] Lin P-S, Kao M-C, Liang W-Y, Hung S-H. Performance analysis and optimization for federated learning applications with pysyft-based secure aggregation. In: 2020 International computer symposium (ICS). IEEE; 2020, p. 191–6.
[9] Ye D, Yu R, Pan M, Han Z. Federated learning in vehicular edge computing: A selective model aggregation approach. IEEE Access 2020;8:23920–35.
[10] Chen S, Shen C, Zhang L, Tang Y. Dynamic aggregation for heterogeneous quantization in federated learning. IEEE Trans Wireless Commun 2021;20(10):6804–19.
[11] Geng D, He H, Lan X, Liu C. An adaptive accuracy threshold aggregation strategy based on federated learning. In: 2021 IEEE 2nd international conference on big data, artificial intelligence and internet of things engineering (ICBAIE). IEEE; 2021, p. 28–31.

[12] Sannara E, Portet F, Lalanda P, German V. A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison. In: 2021 IEEE international conference on pervasive computing and communications (PerCom). IEEE; 2021, p. 1–10.
[13] Xu B, Xia W, Wen W, Zhao H, Zhu H. Optimized edge aggregation for hierarchical federated learning. In: 2021 IEEE 94th vehicular technology conference (VTC2021-Fall). IEEE; 2021, p. 1–5.
[14] Qin Y, Kondo M. Mlmg: Multi-local and multi-global model aggregation for federated learning. In: 2021 IEEE international conference on pervasive computing and communications workshops and other affiliated events (PerCom Workshops). IEEE; 2021, p. 565–71.
[15] Ye Y, Li S, Liu F, Tang Y, Hu W. EdgeFed: Optimized federated learning based on edge computing. IEEE Access 2020;8:209191–8.
[16] Wang Z, Xu H, Liu J, Xu Y, Huang H, Zhao Y. Accelerating federated learning with cluster construction and hierarchical aggregation. IEEE Trans Mob Comput 2022.
[17] Qin Y, Matsutani H, Kondo M. A selective model aggregation approach in federated learning for online anomaly detection. In: 2020 International conferences on internet of things (IThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData) and IEEE congress on cybermatics (Cybermatics). IEEE; 2020, p. 684–91.
[18] Liu Y, Dong Y, Wang H, Jiang H, Xu Q. Distributed fog computing and federated learning enabled secure aggregation for IoT devices. IEEE Internet Things J 2022.
[19] Malladi V, Li YJ, Siddula M, Seoand D, Huang Y. Decentralized aggregation design and study of federated learning. In: 2021 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI). IEEE; 2021, p. 328–37.
[20] Xie M, Long G, Shen T, Zhou T, Wang X, Jiang J, Zhang C. Multi-center federated learning. 2020, arXiv preprint arXiv:2005.01026.
[21] Panigrahi M, Bharti S, Sharma A. FedDCS: A distributed client selection framework for cross device federated learning. Future Gener Comput Syst 2023.
[22] Deng L. The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process Mag 2012;29(6):141–2.
[23] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. 2017, arXiv preprint arXiv:1708.07747.
[24] Panigrahi M, Bharti S, Sharma A. An exhaustive investigation on resource-aware client selection mechanisms for cross-device federated learning. In: Proceedings of the 2022 fourteenth international conference on contemporary computing. 2022, p. 67–73.
[25] Kermany DS, Goldbaum M, Cai W, Valentim CC, Liang H, Baxter SL, McKeown A, Yang G, Wu X, Yan F, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. cell 2018;172(5):1122–31.

Monalisa Panigrahi received an M.Tech degree in information technology from the Indian Institute of Information Technology, Allahabad, India. She is currently working towards a Ph.D. degree in information technology, at Indira Gandhi Delhi Technical University for Women (IGDTUW), Delhi, India. Her research interests include federated learning, the Internet of Things (IoT) and wireless sensor network.

Sourabh Bharti (Member, IEEE) based at Nimbus Research Centre, Munster Technological University, Cork, Ireland. He is also an Associate Investigator at CONNECT research center, in Ireland and is the lead for a number of nationally-funded research projects in the area of common dataspaces, AI on Edge, and distributed machine learning.

Arun Sharma (Senior Member, IEEE) is a Professor in the Department of Information Technology, Indira Gandhi Delhi Technical University for Women (IGDTUW), Delhi, India. He received his Ph.D from Thapar University in 2009. He has a teaching experience of more than 25 years. His research interests include Machine Learning, Data Science and Software Engineering.