

Received 7 March 2023; revised 29 May 2023 and 19 July 2023; accepted 22 July 2023.  
Date of publication 7 August 2023; date of current version 24 August 2023.

The associate editor coordinating the review of this article and approving it for publication was G. Yu.

Digital Object Identifier 10.1109/TMLCN.2023.3302811

# Incentivization and Aggregation Schemes for Federated Learning Applications

VIKAS HASSIJA<sup>1</sup>, VAIBHAV CHAWLA<sup>2</sup>, VINAY CHAMOLA<sup>3,4</sup> (Senior Member, IEEE),  
AND BIPLAB SIKDAR<sup>5</sup> (Senior Member, IEEE)

<sup>1</sup>School of Computing, National University of Singapore, Singapore 119077

<sup>2</sup>Avalara Technologies Pvt Ltd., Noida 201301, India

<sup>3</sup>Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani Campus, Pilani 333031, India

<sup>4</sup>APPCAIR, BITS-Pilani, Pilani Campus, Pilani 333031, India

<sup>5</sup>Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077

CORRESPONDING AUTHORS: V. Chamola (vinay.chamola@pilani.bits-pilani.ac.in) and B. SIKDAR (bsikdar@nus.edu.sg)

This work was supported by the National Research Foundation, Singapore, and Infocomm Media Development Authority under its Future Communications Research and Development Program under Project FCP-NUS-TG-2022-001.

**ABSTRACT** Currently, the data collected by the Internet of Things (IoT) still relies on the cloud-centric data aggregation and processing approach for preparing machine learning models. This approach puts the privacy of the participants at risk. In this paper, federated learning (FL) is proposed for privacy-preserving collaborative model training on data distributed across IoT users. To motivate participants, we must incentivize the whole process by rewarding each participant for their contribution to the training process of the federated learning model. The process of collective training takes place over a long duration of time and multiple iterations. However, participants in the training process may have varying levels of willingness to participate (WTP) and may contribute duplicate or poor-quality data. Therefore, in each iteration, participants must be rewarded based on their contribution in that specific iteration. In this paper, a methodology to reward each participant based on their contribution and a model aggregation technique are proposed. The aggregation technique uses Polyak-averaging to aggregate weights of local models, with the weightage assigned to each local model being proportional to its accuracy on the test dataset. Performance evaluation shows that the federated learning model formed using our aggregation approach achieves the performance level of machine learning as we perform more iterations and performs slightly better than the model formed using the FedAvg algorithm. Additionally, our incentivization methodology provides better performance-based rewards compared to other profit-sharing schemes.

**INDEX TERMS** Federated learning (FL), Internet of Things (IoT), participation-based earning, ensemble learning.

## I. INTRODUCTION

WITH recent developments in the applications of the Internet of Things (IoT) combined with Artificial Intelligence (AI), numerous opportunities for innovation and rejuvenation have opened up for traditional industries. Using data from a relatively small number of volunteers spread across a vast region, we can create a generalized model accessible across the devices of a large number of users. Conventionally, to harness the benefits of IoT, the data collected through volunteers' smart IoT devices is transmitted to the cloud to create a centralized machine learning model [1], [2], [3]. This centralized model is then made

available as an endpoint for performing inferencing tasks, thereby generating revenue for the model owner. Sending raw data to the remote cloud exposes the data to attacks and incurs larger bandwidth costs and transmission latency. Moreover, sending data to the cloud fails to provide quality of service (QoS) and a satisfactory user experience due to network congestion and traffic [4]. Additionally, since the transmitted data inevitably contains private information, it can be misused for targeted advertisements and other malicious activities such as burglary [5], [6], [7].

To address data privacy concerns, it is preferable to keep raw data on the local device. However, for an effective AI

model, a large amount of data must be available. Federated Learning (FL), proposed by Google in 2016 [8], provides an alternative to the cloud-centric approach by allowing models to be trained on local devices, with only the model weights needing to be updated on the cloud. This approach involves the existence of two types of models: a global model and multiple local models. Local models are prepared by training on local data, and their weights are transmitted to the cloud where they are aggregated to form the global model [9]. In the next iteration, the global model's weights are sent back to each local device, where they are recalculated using the local data, thus creating a new local model. To prevent battery issues, local devices may perform model training during idle periods, and users' devices can contribute to training and sharing the model while being charged [10]. This collaborative approach allows many users to contribute data for creating a global model while keeping the raw data on the local device [11], [12], [13].

The focus of this paper is on the FL solution, where user participation is a significant concern. Different users may have varying willingness to participate (WTP) at different times. Therefore, an incentivization scheme needs to be developed to ensure sustainable participation with a dynamic WTP, rewarding participants based on their contributions. Furthermore, existing solutions are vulnerable to malicious behavior by local devices [14], [15]. Malicious devices can undermine the accuracy of the global model by providing fake data and altering the weights during the training process, potentially rendering the global model unusable [16], [17]. The lightweight nature of the training devices, not primarily designed for deep learning, makes perpetual training detrimental to their sustainability [18]. Moreover, current solutions do not fully prevent privacy breaches. Although the training phase is decoupled from centralized data management, model inversion attacks can be performed against gradients or uploaded model parameters to gain access to private raw data [19]. Additionally, divergent data supplied by different IoT devices can slow down the aggregation process even with well-defined initial parameters [20], [21], [22]. The data provided by local devices also depends on communication latency and computational ability, leading to imbalanced data distribution [14]. Therefore, a central server should discard the local straggler's model during the training process, ensuring that its weights are not considered in altering the weights of the global model.

In this paper, we propose a scheme for intelligent aggregation of local models and incentivization for participants with a variable WTP, ensuring robustness to dynamic participant behavior, including participant dropouts. We design a contract-theoretic incentive mechanism that tracks the contributions of participants, allowing the model owner to reward them based on their level of contribution towards creating a global model. The major contributions of this paper can be listed as follows:

- 1) The proposed incentivization model can be applied to any FL application that aims to incentivize

participation in the process, thereby addressing the privacy concerns of the cloud-centric approach.

- 2) The proposed incentivization model quantifies the willingness to participate by considering the quantity of data contributed, taking into account the system resources of the device and the impact of the contribution on the global model.
- 3) The proposed incentivization model offers improved performance-based rewards compared to other profit-sharing schemes while also encouraging long-term participation.
- 4) The proposed aggregation model prevents the weights contributed by underperforming participants from aggregating with the global model.

## II. RELATED WORK

Notable contributions have previously been made in the field of edge computing and FL. Initially, edge-computing-based solutions were proposed [23] to alleviate the burden on networks and cloud servers due to the increasing number of devices and to enhance privacy. To leverage the high bandwidth and low latency between the edge server and the client, computation offloading [24] was proposed to divide and offload complex computing tasks execution from mobile devices to the edge server.

FL has been proposed as a means to preserve user privacy while collaboratively forming a model on distributed devices. The authors in [18] proposed optimization algorithms for FL based on edge computing. Islam et al. [25] proposed differential privacy techniques to prevent model inversion attacks.

Several incentive-based mechanisms have been proposed in order to encourage user participation. The authors in [26], [27] made the optimistic assumption that all devices would unconditionally comply when asked to participate, which is impractical due to the resource costs incurred by participants in training a model. Irfan et al. in [28] explored the leader-follower approach, where the leader auctions for followers to form coalitions. The authors in [29] proposed a contract theory that maps contributed resources to appropriate rewards. In [30], the interaction between devices and the owner was modeled as a Stackelberg game, where the model owner purchases the services provided by the devices. A Deep Reinforcement Learning (DRL) based approach that learns system states from historical training records was proposed in [31].

Previous works on developing a robust incentivization model did not take variable WTP into consideration and mostly relied on the assumption that participants would exhibit the same behavior in the future as they did historically. However, in a highly dynamic world, this is practically impossible. The proposed incentivization and aggregation models consider each contribution as a standalone entity and reward participants based on their individual contributions. Furthermore, Table 1 lists the major contributions, strengths, weaknesses, and recent works related to devising incentivization models.

TABLE 1. Related work

Reference	Year	Contributions	Strength	Weakness
M. Irfan and A. Farooq [28]	Jan 2016	Leader-follower approach with coalitions	Deals with varying capacity to work of participants	Does not take the quality of data contributed into consideration
H. Kim et al. [27]	Jun 2019	Contributes BlockFL, a blockchain-based FL architecture where blockchain verifies data exchange between local devices and provides rewards	Overcomes single point failures and promotes federation of devices by rewarding proportional to training sample sizes	Considers participation without any incentive
J. Kang et al. [29]	Oct 2019	Presents a contract theory based on accuracy-related local data	Attracts data owners with larger-accuracy local data by providing better rewards in comparison to others	Does not take variable WTP into consideration
M. Shayan et al. [26]	Dec 2019	Combines prior techniques into one to execute machine learning involving multiple parties in a highly distributed environment in a private and secure manner	Does not rely on specialized hardware or trusted execution environments	Considers participation without any incentive and vulnerable to attacks which may lead to data leakage from aggregated model
Y. Zhan et al. [31]	Jan 2020	Implements a DRL based approach which learns from historical training data	Parameter server and edge nodes can adjust their strategies dynamically in order to optimize their interests	Needs access to historical training records and does not consider variable WTP
Sarikaya and O. Ercetin [30]	Mar 2020	Proposes a Stackelberg game strategy between model owner and devices	Takes into consideration that other tasks are performed along with learning tasks. Distributes incentives to the participating workers to improve the convergence time of the learning task	Does not quantify how the change in weights of the global model upon training on each local device impacts the global model
Md Sadman Siraj et al. [32]	Nov 2022	Proposes a location-based federated learning model and incentive mechanism based on Generalized Colonel Blotto strategy	Works well for a smart city scenario characterized by different mobility patterns	Assumes that the importance level of the node depends on its location, which correlates to the quality of the raw data available
Proposed Model	Mar 2023	Proposes a Polyak averaging aggregation approach and contract-based incentivization methodology	The incentivization mechanism proposed takes into consideration the variable WTP and quantifies the quality of contribution at each iteration while rewarding the participants	Relies on negative change in accuracy on the test dataset to penalize participants for contributing duplicate and poor-quality data

### III. SYSTEM MODEL

In this section we explain the proposed model in detail. The model is divided into two parts, namely, the incentivization model and the aggregation model. The incentivization model focuses on rewarding each participant based on their contribution, while the aggregation model presents a method to combine local models into a new global model.

#### A. INCENTIVIZATION METHOD

The proposed incentivization model in this paper involves two main entities: the model owner and the participant.

Participants can be categorized into two types based on their incentive to participate in the process, referred to as Type I and Type II participants.

Type I participants contribute their data to create a model and seek monetary incentives from the process, while Type II participants utilize the prepared model in their daily activities, gaining non-monetary benefits. Since Type II participants only use the model without contributing their data, they are not provided with monetary benefits. The focus of the incentivization scheme proposed in this paper is on Type I participants.

Existing solutions using Federated Learning (FL) assume that Type I participants have the same willingness to participate (WTP) throughout the model creation process. Therefore, they are rewarded equally for all their contributions, regardless of the quantity and quality of the data they provide.

The proposed scheme utilizes a contract to record every contribution made by participants during the process. The contract is updated by evaluating the impact of each participant's local model's weights on the accuracy of the global model after aggregation. Initially, Type I participants will be contacted by the model owner, and upon agreeing to the terms and conditions, they will need to provide their device's IMEI or equivalent identification number, which will be used for their contributions. If a participant changes their device, they will need to register again with the new device.

It is assumed that the model owner will contact participants in a way that ensures there is no class imbalance in the classification problem. The system resources of the contributing device can be determined using the identification number and measured relative to a standard device. The entire process is divided into two phases. The first phase concludes with the preparation of the initial global model, while the second phase encompasses all future requests for updating the global model. Figure 1 illustrates the practical working procedure of the proposed methodology. The diagram consists of three parts: the first part includes the global model and its owner, the second part represents the cloud, and the third part represents the devices of Type I participants. The cloud maintains two crucial components of the proposed methodology: the test dataset and the contract. The test dataset is a pre-prepared dataset for a similar problem, stored in the cloud to calculate rewards based on the quality of data contributed in each iteration. Upon receiving the weights of the local model and the amount of contributed data from all the participants, the contract calculates the rewards, and the aggregation process begins in the cloud. The global model is updated after each iteration, benefiting from a larger amount of data compared to the previous iteration, which enhances its robustness. Since the test dataset and the contract stored in the cloud do not contain any sensitive participant data, the proposed methodology does not pose a risk to participant privacy. The three types of devices shown on the rightmost part of the figure indicate that participants are evaluated based on the quantity and quality of their contributions, without bias, irrespective of the size and system resources of their devices. Once selected by the model owner, each participant starts collecting data on their device. Upon the model owner's request to update the global model, the participant initiates local model formation using the collected data, and the weights of the local model are sent to the cloud. The data used for forming a local model in any participant's device is kept separate and not used for future local model formations. Additionally, Table 2 provides a list of symbols used in the mathematical model along with their respective meanings.

In the first phase, participants are rewarded solely based on the quantity of data they contribute, taking into account

TABLE 2. Symbols and their meanings

Symbol	Meanings
$l_{ij}$	reward received by participant $i$ in iteration $j$
$\alpha_{1ij}$	data contributed by participant $i$ relative to other participants in iteration $j$
$\alpha_{2ij}$	data contributed by participant $i$ given system resources relative to other participants in iteration $j$
$\alpha_{3ij}$	impact of local model contributed by participant $i$ on the global model in iteration $j$
$\lambda_1$	reward received per unit of $\alpha_{1ij}$
$\lambda_2$	reward received per unit of $\alpha_{2ij}$
$\lambda_3$	reward received per unit of $\alpha_{3ij}$
$\eta_i$	system resources of participant $i$
$\beta_{ij}$	amount of data contributed by participant $i$ in iteration $j$
$\phi_i$	number of times participant $i$ responded to the request made by the model owner for an update
$\chi_i$	number of times the contribution made by the participant $i$ had a positive impact on the global model
$\psi_i$	number of times the contribution made by the participant $i$ had a negative impact on the global model
$\Delta\epsilon_{ij}$	the impact that local model's weights of participant $i$ had on the global model in iteration $j$
$\mu_j$	the weights of the global model in iteration $j$
$\nu_{ij}$	the weights of the local model, in the first phase, contributed by participant $i$ in iteration $j$
$\mu_{j-1}\nu_{ij}$	the weights of the updated local model, in the second phase, for participant $i$ in iteration $j$ formed by fitting the local training data on the global model's weights formed in iteration $j - 1$
$\kappa$	threshold value
$\xi_{ij}$	weightage applied to $\mu_{j-1}\nu_{ij}$ for participant $i$ in iteration $j$
$T_{ij}$	linear transformation of the $\epsilon_{ij}$
$\omega(\gamma)$	accuracy of model with weights $\gamma$ on the test dataset

their system resources. In the second phase, when an update request is made by the model owner, all participants receive the current weights of the existing global model, which are then updated by training on their local data to create new local models. The accuracies of the existing global model and the contributed new local models on the test dataset are calculated.



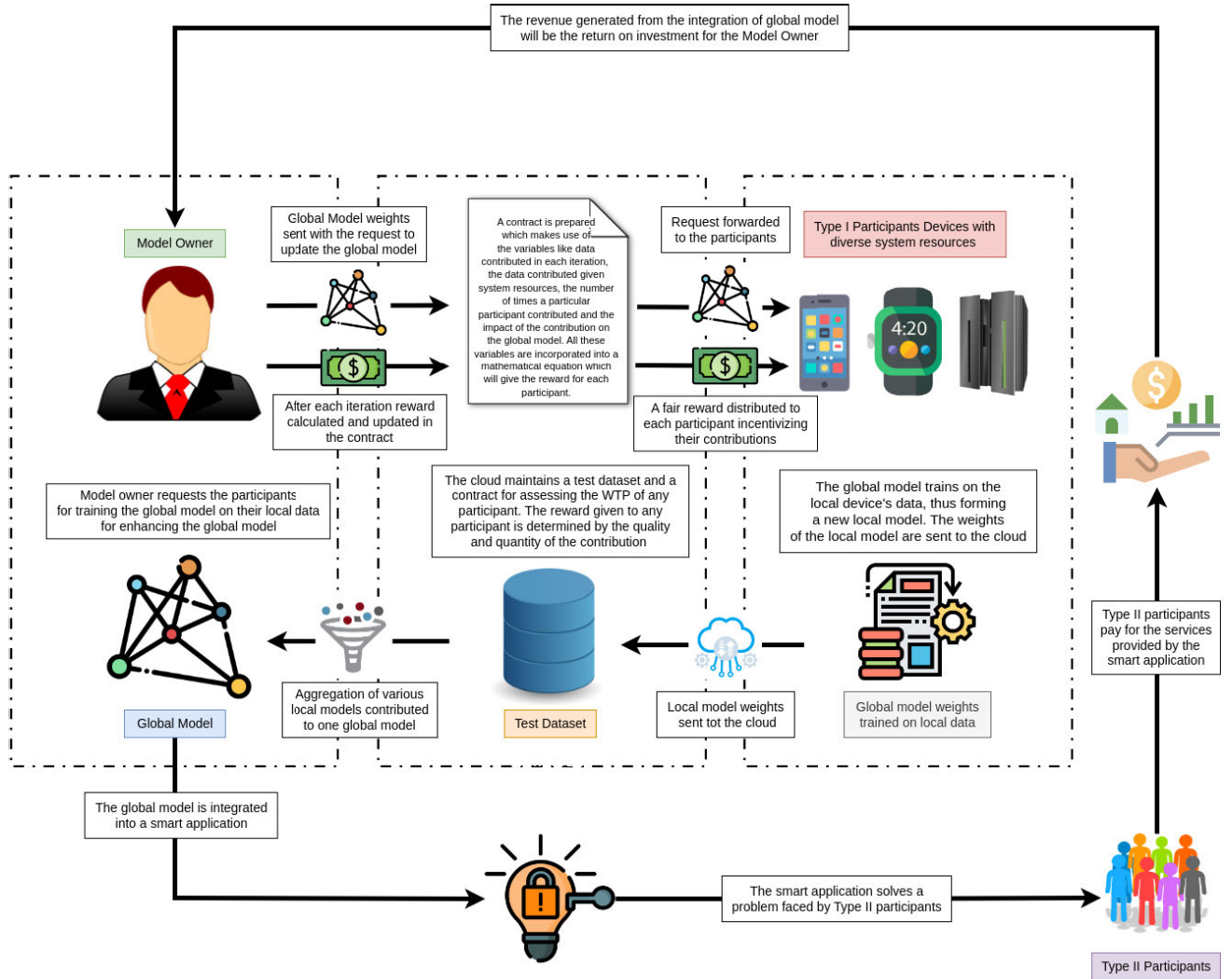


FIGURE 1. Workflow diagram for proposed methodology in practical environments.

The incentivization model considers that participants should respond to the model owner's request within a specified time limit. Contributions made after the deadline will not be rewarded. The incentivization scheme takes various parameters into account, including the amount of data contributed, the system resources of the device, the consistency in responding to requests, and the impact of the participant's contribution on the global model. The final reward after any iteration can be calculated as follows:

$$t_{ij} = \lambda_1 \alpha_{1ij} + \lambda_2 \alpha_{2ij} + \lambda_3 \alpha_{3ij}. \quad (1)$$

where for participant  $i$  in iteration  $j$ ,  $t_{ij}$  represents the reward and  $\alpha_{1ij}$  represents data contributed relative to other participants,  $\alpha_{2ij}$  represents data contributed given system resources relative to other participants, and  $\alpha_{3ij}$  represents the impact the local model had on the global model.  $\lambda_1$  is the reward received per unit of  $\alpha_{1ij}$ ,  $\lambda_2$  is the reward received per unit of  $\alpha_{2ij}$ , and  $\lambda_3$  is the reward received per unit of  $\alpha_{3ij}$ .

Let us consider that we have  $n$  different devices with system resources,  $\eta_1, \eta_2, \dots, \eta_n$ , and participant  $i$  contributed  $\beta_{ij}$  amount of data in iteration  $j$ . We can calculate the total data contributed and total data contributed given system resources as,

$$\beta_{1j} + \beta_{2j} + \dots + \beta_{nj} = \sum_{i=1}^n \beta_{ij}, \quad (2)$$

$$(\beta_{1j}/\eta_1) + (\beta_{2j}/\eta_2) + \dots + (\beta_{nj}/\eta_n) = \sum_{i=1}^n (\beta_{ij}/\eta_i). \quad (3)$$

Let  $\phi_i$  be the number of times participant  $i$  responded to the request made by the model owner for an update. Let  $\chi_i$  and  $\psi_i$  be the number of times the contribution made by participant  $i$  had a positive impact and a negative impact on the global model, respectively.

Assuming  $\beta_{ij}$  is not equal to zero we have,

$$\phi_i = \phi_i + 1. \quad (4)$$

At  $j = 0$  we are in the first phase, otherwise, we are in the second phase.

In the first phase, we do not have a global model prepared. Therefore, we cannot calculate  $\Delta\epsilon_{ij}$ , which is the impact that local model's weights of participant  $i$  had on the global model in iteration  $j$ , hence  $\alpha_{3ij} = 1$  at  $j = 0$ .

In the first phase, the reward of every participant can be calculated using Equation (1) where,

$$\alpha_{1ij} = \frac{\beta_{ij}}{\sum_{i=1}^n \beta_{ij}}, \quad (5)$$

$$\alpha_{2ij} = \frac{(\beta_{ij}/\eta_i)}{\sum_{i=1}^n (\beta_{ij}/\eta_i)}, \quad (6)$$

$$\alpha_{3ij} = 1. \quad (7)$$

For the second phase, the global model's weights are sent to all the participants with the request to update them on their local data.

The impact of the local training on the global model is calculated as,

$$\Delta\epsilon_{ij} = \omega(\mu_{j-1}v_{ij}) - \omega(\mu_{j-1}), \quad \forall j > 0. \quad (8)$$

where  $\omega(\mu_{j-1}v_{ij})$  is the accuracy of the model, formed by fitting the local training data on the global model's weights, on the test dataset, and  $\omega(\mu_{j-1})$  is the accuracy of the previous global model on the test dataset. Note that

$$\Delta\epsilon_{ij} \in [-1, 1]. \quad (9)$$

The participant's contribution to the formation of a new global model can be classified into a positive contribution and a negative contribution based on the value of  $\Delta\epsilon_{ij}$ . A positive contribution should be rewarded, while a negative contribution should be penalized. Erratic participants are those whose contributions negatively impact the global model. However, since negative contributions can sometimes be unintentional, it is important to consider human error and not be too harsh on the participants. To address this, participants who make a negative contribution more than a threshold  $\kappa$  times will not receive any reward for another negative contribution, and the weights contributed by that participant will be discarded in the aggregation process. The threshold ensures that participants are not discouraged by the fear of failure, while also preventing participants from taking the model owner's rewards for granted. For positive contributions, participants are rewarded in a similar way as any other participant with a positive contribution. This approach ensures that participants are appropriately rewarded for their positive contributions while considering the impact of negative contributions and encouraging improvement.

Given  $\beta_{ij}$  is not equal to 0, if  $\Delta\epsilon_{ij} \geq 0$ , we increment  $\chi_i$  as,

$$\chi_i = \chi_i + 1. \quad (10)$$

Otherwise we increment  $\psi_i$  as,

$$\psi_i = \psi_i + 1. \quad (11)$$

We must choose the function such that the amount of reward and penalty increases for higher values of  $\Delta\epsilon_{ij}$ . Therefore, we calculate  $\tan(\Delta\epsilon_{ij})$  due to its property of tending to  $\infty$  and  $-\infty$  when  $\Delta\epsilon_{ij}$  tends to 1 and  $-1$  respectively.

In the second phase, the reward of each participant can be calculated using (1), where

$$\alpha_{1ij} = \frac{\beta_{ij}}{\sum_{i=1}^n \beta_{ij}} \times \sqrt{\phi_i}, \quad (12)$$

$$\alpha_{2ij} = \frac{(\beta_{ij}/\eta_i)}{\sum_{i=1}^n (\beta_{ij}/\eta_i)} \times \sqrt{\phi_i}, \quad (13)$$

$$\alpha_{3ij} = \frac{\chi_i}{\sqrt{\phi_i}} \times \tan(\Delta\epsilon_{ij}). \quad (14)$$

Here, the term  $\tan(\Delta\epsilon_{ij})$  is the reward for improving the accuracy of the global model.

Given  $\psi_i < \kappa$ , we calculate the reward by adding  $\tan(\Delta\epsilon_{ij})$ . In case of a negative contribution  $\Delta\epsilon_{ij} < 0$ , a negative value will be added to the reward. Hence, a penalty will be added which reduces the reward of the participant which he received for contributing data and spending system resources. The value of  $\alpha_{1ij}$  and  $\alpha_{2ij}$  will be calculated similarly as given in equations (12) and (13), and the value of  $\alpha_{3ij}$  will be given by:

$$\alpha_{3ij} = \frac{\psi_i}{\sqrt{\phi_i}} \times \tan(\Delta\epsilon_{ij}). \quad (15)$$

In case  $\Delta\epsilon_{ij} = 0$ ,  $\tan(\Delta\epsilon_{ij})$  will be equal to 0. In our equation  $\alpha_{3ij}$  will be equal to 0, hence, the reward will depend on the values of  $\alpha_{1ij}$  and  $\alpha_{2ij}$ .

In case  $\psi_i \geq \kappa$  and  $\Delta\epsilon_{ij} < 0$ ,  $\alpha_{3ij} = 0$ .

Algorithm 1 outlines the process of creating a local model. In the beginning, a local model for participant  $i$  in iteration  $j$ , denoted as *localModel* <sub>$ij$</sub> , is initialized using the baseline model configuration referred to as *baselineModel* in the algorithm. If we are in the first phase, the local model is trained on the participant's local data using the default initial weights. These default initial weights are determined through Xavier Glorot Initialization. Once trained, the local model is sent to the cloud. If we are in the second phase, the initial weights of the local model are set equal to the weights of the existing global model. The local model is then trained on the participant's local data using these initial weights. The updated local model is sent to the cloud. The algorithm provides a step-by-step guide for initializing and training the local model, ensuring that each participant's contribution is based on their own data while maintaining consistency with the global model.

Algorithm 2 shows the process of rewarding each participant. The meaning of all the symbols used in this algorithm are explained in Table 2.

## B. AGGREGATION OF LOCAL MODEL

To create a robust global model, we need to aggregate the local models received from participants. In the first phase, the global model is selected as the best-performing local model based on its performance on the test dataset. In the

---

**Algorithm 1** Local Model Generation Algorithm.
 

---

```

1:  $localModel_{ij} \leftarrow baselineModel$ 
2: if  $j = 0$  then
3:    $v_{ij} \leftarrow defaultWeights$ 
4:    $localModel_{ij}$  with weights  $v_{ij}$  fits local training data
   and we get updated  $v_{ij}$ 
5: else
6:    $v_{ij} \leftarrow \mu_{j-1}$ 
7:    $localModel_{ij}$  with weights  $\mu_{j-1}v_{ij}$  fits local training
   data and we get updated  $\mu_{j-1}v_{ij}$ 
8: end if
9: Weights  $\mu_{j-1}v_{ij}$  are sent to the cloud along with the
   amount of data trained upon in bytes and system
   resources of the participant's device
    
```

---



---

**Algorithm 2** Reward Calculation Algorithm for Each Participant.
 

---

```

1:  $\lambda_1, \lambda_2, \lambda_3 \leftarrow constants$ 
2: We have the system resources  $\eta_i$  of the device and the
   amount of data in bytes  $\beta_{ij}$  contributed by participant  $i$ 
   in iteration  $j$ . Using these we calculate  $\beta_{ij} / \sum_{i=1}^n \beta_{ij}$  and
    $(\beta_{ij}/\eta_i) / \sum_{i=1}^n (\beta_{ij}/\eta_i)$ .
3: if  $j = 0$  then
4:    $\phi_i \leftarrow 0$ 
5:    $\chi_i \leftarrow 0$ 
6:    $\psi_i \leftarrow 0$ 
7:   if  $\beta_{ij} \neq 0$  then
8:      $\phi_i \leftarrow \phi_i + 1$ 
9:   end if
10:   $\iota_{ij} = \sqrt{\phi_i} * (\lambda_1 * \beta_{ij} / \sum_{i=1}^n \beta_{ij} + \lambda_2 * (\beta_{ij}/\eta_i) / \sum_{i=1}^n (\beta_{ij}/\eta_i) + \lambda_3)$ 
11: else
12:   if  $\beta_{ij} \neq 0$  then
13:      $\phi_i \leftarrow \phi_i + 1$ 
14:      $\Delta\epsilon_{ij} \leftarrow \omega(\mu_{j-1}v_{ij}) - \omega(\mu_{j-1})$ 
15:     if  $\Delta\epsilon_{ij} > 0$  then
16:        $\chi_i \leftarrow \chi_i + 1$ 
17:        $\iota_{ij} = \sqrt{\phi_i} * (\lambda_1 * \beta_{ij} / \sum_{i=1}^n \beta_{ij} + \lambda_2 * (\beta_{ij}/\eta_i) / \sum_{i=1}^n (\beta_{ij}/\eta_i) + \lambda_3 * \chi_i / \phi_i * \tan(\Delta\epsilon_{ij}))$ 
18:     else
19:        $\psi_i \leftarrow \psi_i + 1$ 
20:       if  $\psi_i < \kappa$  then
21:          $\iota_{ij} = \sqrt{\phi_i} * (\lambda_1 * \beta_{ij} / \sum_{i=1}^n \beta_{ij} + \lambda_2 * (\beta_{ij}/\eta_i) / \sum_{i=1}^n (\beta_{ij}/\eta_i) + \lambda_3 * \psi_i / \phi_i * \tan(\Delta\epsilon_{ij}))$ 
22:       else
23:          $\iota_{ij} \leftarrow 0$ 
24:       end if
25:     end if
26:   end if
27: end if
    
```

---

second phase, Polyak averaging is applied to aggregate the weights of multiple local models. Polyak averaging calculates a weighted average of these local models, with different

weightage given to each model in each phase. In the first phase, the weight of the best-performing local model is set to 1, while the weights of other local models are set to 0. Thus, at  $j = 0$ ,

$$\mu_j = \sum_{i=1}^n (v_{ij} * \xi_{ij}), \quad (16)$$

where  $v_{ij}$  represents the weights of the model contributed by participant  $i$  in iteration  $j$ , and  $\xi_{ij}$  is the weightage applied to this model. The value of  $\xi_{ij}$  is given by:

$$\xi_{ij} = 1, \quad (17)$$

for the best-performing local model, and

$$\xi_{ij} = 0, \quad (18)$$

for other local models. In the second phase, the weightage of each local model is proportional to the accuracy of the local model formed by training the global model's weights on the participant's local data. Thus, the global model in the second phase is determined as:

$$\mu_j = \sum_{i=1}^n (\mu_{j-1}v_{ij} * \xi_{ij}), \quad (19)$$

where  $\xi_{ij}$  represents the weightage applied to  $\mu_{j-1}v_{ij}$ , the weights of the model formed by training the global model on the participant's local data. The weightage assigned to each local model is calculated as:

$$\xi_{ij} = \frac{\tau_{ij}}{\sum_{k=1}^n \tau_{ik}}, \quad (20)$$

where  $\tau_{ij}$  is a linear transformation of  $\omega(\mu_{j-1}v_{ij})$ , and is given by:

$$\tau_i = \frac{1 + \omega(\mu_{j-1}v_{ij})}{2}. \quad (21)$$

Algorithm 3 demonstrates the process of aggregating local models into a global model. It starts by initializing the global model  $globalModel_j$  in iteration  $j$  with the baseline model configuration denoted as  $baselineModel$ , where all the weights are set to 0. The weights of the global model are then aggregated based on the proposed aggregation model. Once the weights are finalized, the global model is sent to the model owner.

## IV. RESULTS

In this section, we describe the experiments conducted to evaluate the performance of the proposed incentivization and aggregation methodologies.

### A. EXPERIMENTAL SETUP

A simulation of the proposed methodologies was performed using the MNIST handwritten digits and Extended MNIST (EMNIST) letters datasets. Both datasets are derived from the NIST Special Database 19 [33] and converted to a  $28 \times 28$  pixel image format. MNIST, which stands for Modified

**Algorithm 3** Algorithm for Aggregation of Local Models.

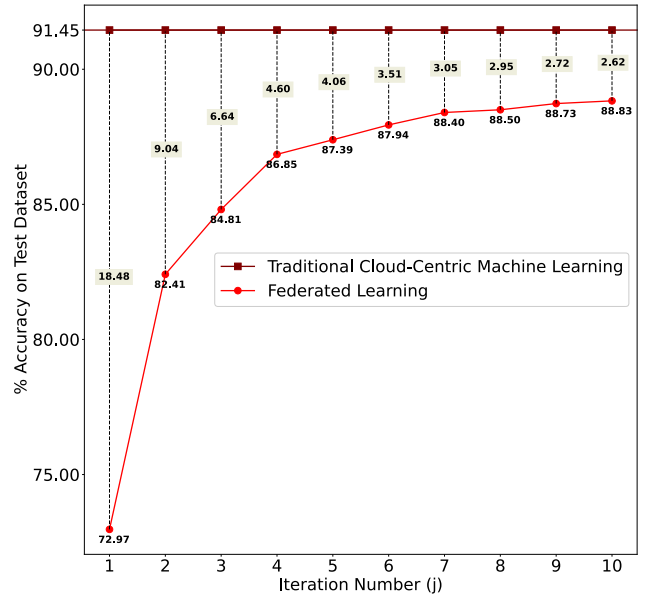
```

1:  $globalModel_j \leftarrow baselineModel$ 
2:  $\mu_j \leftarrow [0, 0, \dots, 0]$ 
3: for  $i = 0, \dots, n$  do
4:   if  $\psi_i < \kappa$  then
5:     if  $j = 0$  then
6:        $\mu_j \leftarrow \mu_j + v_{ij} * \xi_{ij}$ 
7:     else
8:        $\mu_j \leftarrow \mu_j + \mu_{j-1} v_{ij} * \xi_{ij}$ 
9:     end if
10:  end if
11: end for
12: The weights of the  $globalModel_j$  are set to  $\mu_j$  and it is sent to the model owner

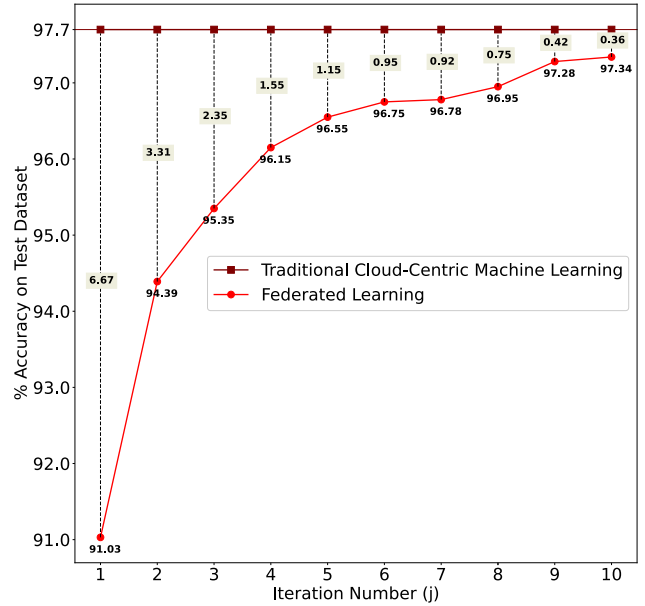
```

National Institute of Standards and Technology, is considered one of the standardized datasets used for learning, classification, and computer vision systems [34]. It has also been used for performance analysis of recently proposed FL techniques [35], [36]. The MNIST handwritten digits dataset consists of 60,000 images in the training dataset, with 10 classes covering digits from 0 to 9, and 10,000 images in the test dataset. The EMNIST letters dataset merges all the uppercase and lowercase classes of the English alphabet to form a balanced 26-class classification task. The EMNIST letters dataset has 145,600 images in the training dataset and 20,800 images in the test dataset [37]. The FL process was conducted with four hypothetical Type I participants who contributed a random amount of data across 10 iterations. The images and their labels from the aforementioned datasets were distributed randomly to ensure that the setup reflects a practical environment. The baseline model used consists of a simple neural network with one hidden layer, with the same number of neurons as the total number of pixels in each image of the dataset. The Rectified Linear Unit (ReLU) is used as the activation function for neurons in the hidden layer. The softmax activation function is used for the neurons in the output layer to obtain a probability for each of the 10 classes for each input. The Adam gradient descent algorithm and logarithmic loss function are used for learning the weights. The number of epochs was fixed at 10, and the batch size was set to 200. Default values were used for the learning rate and other important hyperparameters to keep the baseline model as simple as possible. The purpose here is not to increase the performance of the classification algorithm, but to compare the performance of the Machine Learning and FL models and to reward each participant based on the quality of their contribution and their WTP.

In the incentivization model, the hyperparameters are set as  $\lambda_1 = 1$ ,  $\lambda_2 = 2$ , and  $\lambda_3 = 4$ . These values indicate that, for the hypothetical model owner, the impact created by any local model on the global model is twice as important as the quantity of data contributed by any participant, given their system resources, and four times as important as the



(a) EMNIST letters dataset



(b) MNIST handwritten digits dataset

**FIGURE 2. Comparing accuracies of federated learning and machine learning across iterations on the test dataset for MNIST handwritten and EMNIST letters dataset.**

quantity of data contributed by any participant. The value of  $\kappa$  is set to 2, which means that the model owner will not reward participants solely based on the quantity of data contributed if the participant has contributed poor quality data more than twice. It's important to note that these values may vary from one model owner to another.

**B. PERFORMANCE EVALUATION**

Figure 2 shows the accuracy of the model prepared by applying FL and the baseline Artificial Neural Network (ANN) model to the total number of images contributed in each

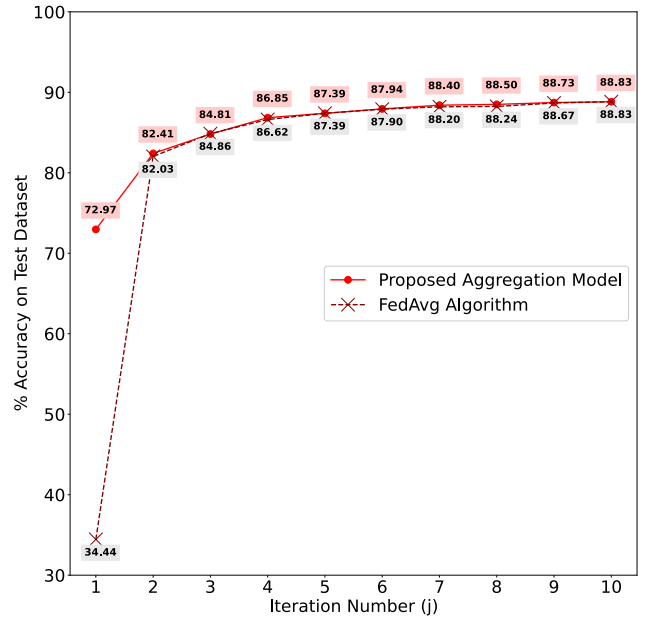


iteration. The x-axis represents the iteration number, and the y-axis shows the percentage accuracy achieved. In the early iterations, the performance of FL is worse compared to Machine Learning, as in the latter, all the images are trained as a whole, whereas in FL, only the weights are aggregated. The comparison shows that our proposed FL model as any other existing FL model follows the trend of performing comparable to the ML model as we increase the number of iterations.

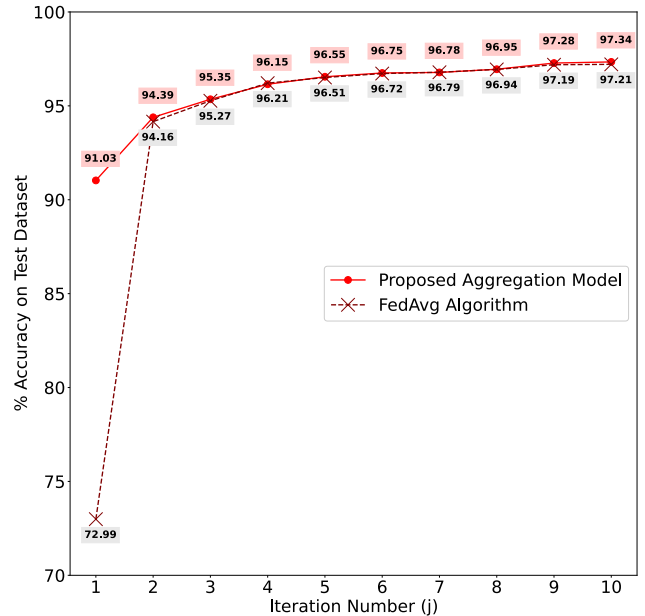
In the MNIST handwritten letters dataset, the difference between the accuracies of Machine Learning and FL techniques on the test dataset reduced to 0.36%, making the performance of both approaches almost similar. In the EMNIST Letters dataset, after 10 iterations, the difference in performance was around 2.62%, which makes the performance of both approaches moderately comparable. The baseline model in both datasets can be altered to enhance the performance of Machine Learning and FL.

Figure 3 compares the performance of the FedAvg algorithm with the proposed aggregation model. The FedAvg algorithm [38] is the most commonly used aggregation scheme for Federated Learning. Our aggregation scheme takes inspiration from the FedAvg algorithm and performs Polyak averaging of the local models' weights, with more weightage given to higher accuracy models. The FedAvg algorithm serves as an important benchmark for comparing our aggregation scheme with existing FL aggregation algorithms. From the figure, we can see that the proposed model shows greater accuracy on the test dataset for the first iteration compared to the FedAvg algorithm for both datasets. As we progress in the number of iterations, both schemes become comparable. The reason for the comparable performance of FedAvg and our proposed aggregation model is the use of high-quality standard datasets for the experiments. Even when dividing the data randomly among participants, each participant's local model is trained on unbiased high-caliber data, resulting in almost equal weightage being applied to every contributed local model's weights for aggregation using Polyak averaging. In a practical environment, we cannot expect such a scenario since we cannot assume good quality data from every participant. In the real-world scenario, our proposed aggregation model is expected to have an edge, as more accurate models will have a higher contribution to make a global model.

Figure 4 compares each attribute mentioned earlier for each contribution across iterations of two participants, along with the reward they earned relative to other participants. The x-axis denotes the iterations, and the y-axis denotes the normalized reward received and the normalized value of the attributes. Since both the reward and attributes are normalized, their values range from 0 to 1. In the performed simulations, Participant A contributed good quality data across all the iterations, while Participant B was an erratic contributor, as it did not contribute in iteration number 5 and 10. Therefore, no reward was provided to Participant B in those iterations. We can observe that the reward is a function of



(a) EMNIST letters dataset

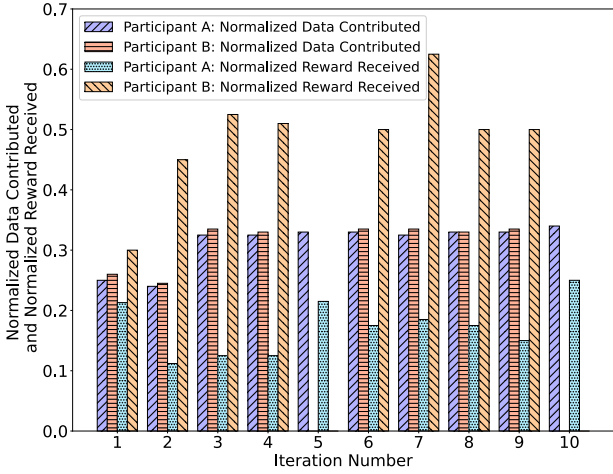


(b) MNIST handwritten digits dataset

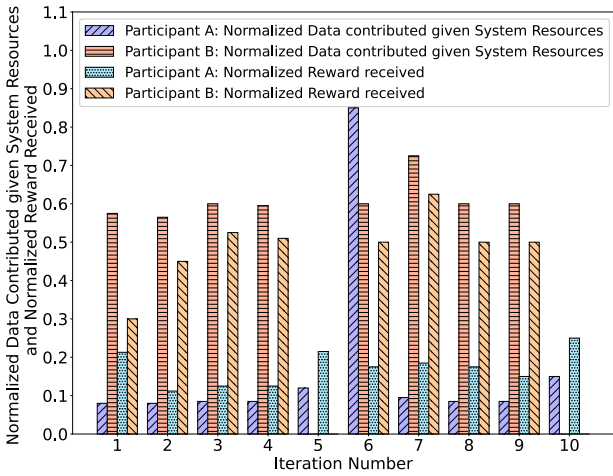
**FIGURE 3. Comparing accuracies of proposed Federated learning aggregation model with FedAvg algorithm across iterations on the test dataset for MNIST handwritten and EMNIST letters dataset.**

all the attributes combined and does not depend solely on a single attribute.

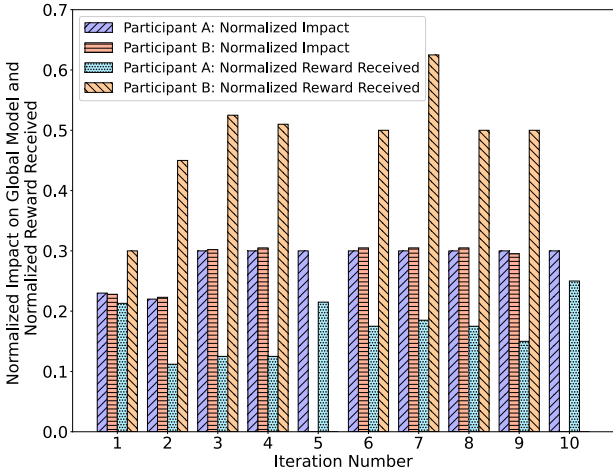
Figure 5 compares our incentivization model with simpler profit-sharing schemes, namely, Egalitarian and Marginal Gain Profit Sharing Schemes. The Egalitarian Profit Sharing Scheme [39] distributes the total reward equally to each participant in each iteration, regardless of their contribution. The Marginal Gain Profit Sharing scheme [39] rewards a participant based on the value added when that participant



(a) Comparing the normalized reward received ( $l_{ij} / \sum l_{ij}$ ) with the normalized amount of data contributed ( $\alpha_{1ij} / \sum \alpha_{1ij}$ )

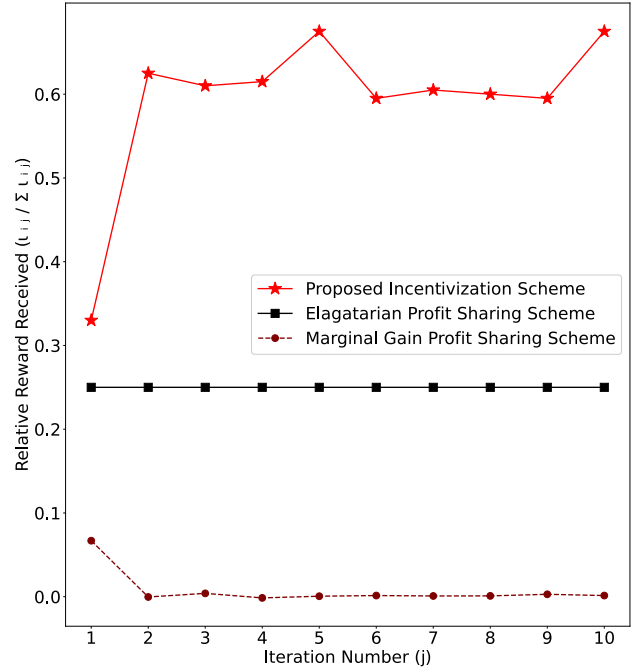


(b) Comparing the normalized reward received ( $l_{ij} / \sum l_{ij}$ ) with the normalized amount of data contributed given system resources ( $\alpha_{2ij} / \sum \alpha_{2ij}$ )

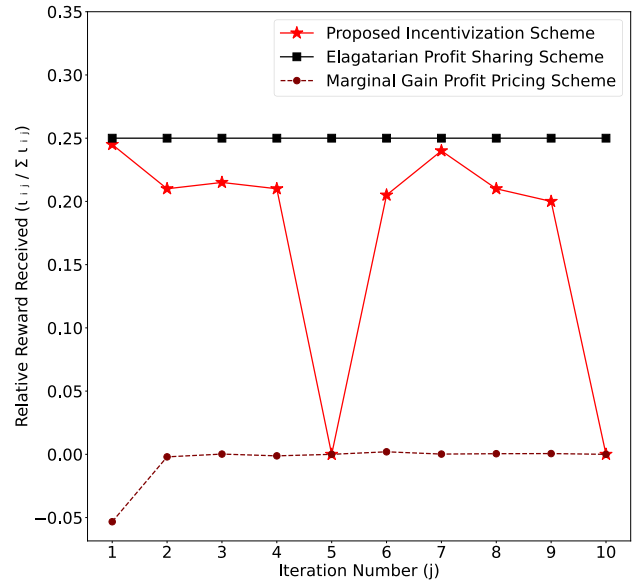


(c) Comparing the normalized reward received ( $l_{ij} / \sum l_{ij}$ ) with the normalized impact of local model contributed on the global model ( $\mu_{j-1}l_{ij} / \sum \mu_{j-1}l_{ij}$ )

**FIGURE 4. Comparing normalized rewards earned by 2 participants with different normalized attributes of their contributions.**



(a) For a Consistent Participant



(b) For an Erratic Participant

**FIGURE 5. Comparing normalized rewards received provided by different incentivization models.**

joins the process. We plot the normalized reward received by a participant using our proposed model and the reward they would have received in other schemes against the iteration number. The Consistent Participant symbolizes a participant who regularly contributed good quality data, whereas the Erratic Participant depicts a participant who did not respond to each request made by the model owner and contributed inferior quality data. Since we considered 4 participants in this simulation, each participant should receive 25% of the

total reward allocated according to the Egalitarian scheme. For the Marginal Gain scheme, we considered that the relative reward received by the participant is equal to the change in accuracy of the global model with and without that participant in a particular iteration. We can observe that for a Consistent Participant, the proposed incentivization model rewards more than both the traditional schemes. In the case of an Erratic Participant who did not contribute in iteration number 5 and 10, the Egalitarian Scheme gives undue rewards in case of no contribution, whereas the rewards from the Marginal Gain scheme hardly change after the first few iterations. The reason is that as the number of iterations increases, the global model's accuracy tends to stabilize, hence the marginal gain of the whole process does not change much. We could have included the amount of data contributed by a participant in a particular iteration along with the change in accuracy as a part of the gain in the Marginal Gain scheme, but that would have made this scheme highly dependent on the data contributed as the number of iterations increased. Our proposed incentivization methodology improves upon these existing methodologies by rewarding participants according to the quality and quantity of their contributions.

## V. CONCLUSION

In this paper, we proposed an incentivization and aggregation methodology that is robust against the variable WTP of participants in a practical environment. The proposed incentivization model utilizes the data contributed by each participant and system resources for each participant's device, as well as the impact that any local model's weights have on the global model. The impact on the global model can be calculated by measuring the change in accuracy of the existing global model on the test dataset before and after training the existing global model's weights on the local data. This change in accuracy is also utilized as the weightage assigned to each local model during the Polyak averaging in the proposed aggregation model. We conducted simulations on the MNIST and EMNIST datasets to demonstrate the applicability of our proposed model. We can apply our proposed incentivization and aggregation methodologies to any specific use-case by using a targeted test dataset and a carefully tuned baseline model. Having a test dataset tailored to the particular use-case is crucial for evaluating the quality of each contribution and determining appropriate rewards. The test dataset is considerably smaller than the training dataset, therefore it can be prepared using the model owner's resources based on the given problem statement. This way, we can effectively utilize the proposed federated learning incentivization and aggregation methodologies. Our performance evaluation shows that FL with the proposed aggregation model performs similarly to machine learning and slightly better than the FedAvg algorithm as the number of iterations increases. Moreover, the proposed incentivization model provides more rewards to consistent participants compared to the Egalitarian and Marginal Gain Profit Sharing schemes.

## REFERENCES

- [1] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, 2020.
- [2] Z. Wan, Z. Zhang, R. Yin, and G. Yu, "KFIML: Kubernetes-based fog computing IoT platform for online machine learning," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19463–19476, Oct. 2022.
- [3] V. Hassija, V. Chamola, and S. Zeadally, "BitFund: A blockchain-based crowd funding platform for future smart and connected nation," *Sustain. Cities Soc.*, vol. 60, Sep. 2020, Art. no. 102145.
- [4] G. S. Aujla, R. Chaudhary, K. Kaur, S. Garg, N. Kumar, and R. Ranjan, "SAFE: SDN-assisted framework for edge-cloud interplay in secure healthcare ecosystem," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 469–480, Jan. 2019.
- [5] Z. Su et al., "Secure and efficient federated learning for smart grid with edge-cloud collaboration," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1333–1344, Feb. 2022.
- [6] Q. Hu, Y. Cai, K. Kang, G. Yu, J. Hoydis, and Y. C. Eldar, "Two-timescale end-to-end learning for channel acquisition and hybrid precoding," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 163–181, Jan. 2022.
- [7] S. Singh, R. Sulthana, T. Shewale, V. Chamola, A. Benslimane, and B. Sikdar, "Machine-learning-assisted security and privacy provisioning for edge computing: A survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 236–260, Jan. 2022.
- [8] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.
- [9] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.
- [10] Z. Zheng, Y. Chi, G. Ding, and G. Yu, "Deep-learning-based resource allocation for time-sensitive device-to-device networks," *Sensors*, vol. 22, no. 4, p. 1551, Feb. 2022.
- [11] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. Data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.
- [12] J. Zhao, H. Zhu, F. Wang, R. Lu, Z. Liu, and H. Li, "PVD-FL: A privacy-preserving and verifiable decentralized federated learning framework," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2059–2073, 2022.
- [13] G. Bansal, V. Hassija, V. Chamola, N. Kumar, and M. Guizani, "Smart stock exchange market: A secure predictive decentralized model," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.
- [14] T. Zhang, A. Song, X. Dong, Y. Shen, and J. Ma, "Privacy-preserving asynchronous grouped federated learning for IoT," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5511–5523, Apr. 2022.
- [15] V. Hassija, G. Bansal, V. Chamola, V. Saxena, and B. Sikdar, "BlockCom: A blockchain based commerce model for smart communities using auction mechanism," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.
- [16] X. Cao, Z. Zhang, J. Jia, and N. Z. Gong, "FLCert: Provably secure federated learning against poisoning attacks," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3691–3705, 2022.
- [17] S. S. G. Chalapathi, V. Chamola, C.-K. Tham, S. Gurunayanan, and N. Ansari, "An optimal delay aware task assignment scheme for wireless SDN networked edge cloudlets," *Future Gener. Comput. Syst.*, vol. 102, pp. 862–875, Jan. 2020.
- [18] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu, "EdgeFed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209191–209198, 2020.
- [19] H.-Y. Tran, J. Hu, X. Yin, and H. R. Pota, "An efficient privacy-enhancing cross-silo federated learning and applications for false data injection attack detection in smart grids," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2538–2552, 2023.
- [20] J. Mills, J. Hu, and G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 630–641, Mar. 2022.
- [21] H. Wu and P. Wang, "Node selection toward faster convergence for federated learning on non-IID data," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3099–3111, Sep. 2022.
- [22] H. Rohmetra, N. Raghunath, P. Narang, V. Chamola, M. Guizani, and N. R. Lakkani, "AI-enabled remote monitoring of vital signs for COVID-19: Methods, prospects and challenges," *Computing*, vol. 105, pp. 783–809, Mar. 2023. [Online]. Available: <https://link.springer.com/article/10.1007/s00607-021-00937-7#citeas>



[23] X. Ding et al., "Privacy-preserving task allocation for edge computing-based mobile crowdsensing," *Comput. Electr. Eng.*, vol. 97, Jan. 2022, Art. no. 107528. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790621004730>

[24] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: Issues and challenges," *Appl. Comput. Informat.*, vol. 14, no. 1, pp. 1–16, Jan. 2018. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01767212>

[25] T. U. Islam, R. Ghasemi, and N. Mohammed, "Privacy-preserving federated learning model for healthcare data," in *Proc. IEEE 12th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2022, pp. 281–287.

[26] M. Shayan, C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Biscotti: A ledger for private and secure peer-to-peer machine learning," 2018, *arXiv:1811.09904*.

[27] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

[28] M. Irfan and A. Farooq, "Auction-based task allocation scheme for dynamic coalition formations in limited robotic swarms with heterogeneous capabilities," in *Proc. Int. Conf. Intell. Syst. Eng. (ICISE)*, Jan. 2016, pp. 210–215.

[29] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symp. (APWCS)*, Aug. 2019, pp. 1–5.

[30] Y. Sarikaya and O. Erceetin, "Motivating workers in federated learning: A Stackelberg game perspective," *IEEE Netw. Lett.*, vol. 2, no. 1, pp. 23–27, Mar. 2020.

[31] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[32] M. S. Siraj, M. S. Hossain, R. Brown, E. E. Tsiropoulou, and S. Papavassiliou, "Incentives to learn: A location-based federated learning model," in *Proc. Global Inf. Infrastruct. Netw. Symp. (GIIS)*, Sep. 2022, pp. 40–45.

[33] P. J. Grother, "NIST special database 19-hand-printed forms and characters database," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 10010178392, 1995.

[34] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2921–2926.

[35] L. Wu, S. Guo, Y. Liu, Z. Hong, Y. Zhan, and W. Xu, "Sustainable federated learning with long-term online VCG auction mechanism," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2022, pp. 895–905.

[36] J. Han et al., "TIFF: Tokenized incentive for federated learning," in *Proc. IEEE 15th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2022, pp. 407–416.

[37] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: An extension of MNIST to handwritten letters," 2017, *arXiv:1702.05373*.

[38] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016, *arXiv:1602.05629*.

[39] S. Gollapudi, K. Kollias, D. Panigrahi, and V. Pliatsika, "Profit sharing and efficiency in utility games," in *Proc. ESA*, 2017, pp. 1–14.



**VIKAS HASSIJA** received the B.Tech. degree from M. D. U. University, Rohtak, India, in 2010, the M.S. degree in telecommunications and software engineering from the Birla Institute of Technology and Science (BITS), Pilani, India, in 2014, and the Ph.D. degree in the IoT security and blockchain from the Jaypee Institute of Information and Technology (JIIT), Noida. He was an Assistant Professor with JIIT for four years. He has eight years of industry experience and has worked

with various telecommunication companies, such as Tech Mahindra and Accenture. He is currently a Post-Doctoral Researcher with the National University of Singapore, Singapore. His research interests include the IoT security, network security, blockchain, and distributed computing.



**VAIBHAV CHAWLA** received the B.Tech. degree in computer science from the Jaypee Institute of Information Technology, Noida. He is currently a Software Engineer with Avalara Technologies Pvt Ltd., a U.S.-based MNC providing tax software solutions. He enjoys solving real-world problems. His research interests include machine learning and blockchain. He has won national-level hackathons.



**VINAY CHAMOLA** (Senior Member, IEEE) received the B.E. degree in electrical and electronics engineering and the master's degree in communication engineering from the Birla Institute of Technology and Science, Pilani (BITS-Pilani), Pilani, India, in 2010 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2016. In 2015, he was a Visiting Researcher with the Autonomous Networks

Research Group (ANRG), University of Southern California, Los Angeles, CA, USA. He was a Post-Doctoral Research Fellow with the National University of Singapore, Singapore. He is currently an Associate Professor with the Department of Electrical and Electronics Engineering, BITS-Pilani, where he heads the Internet of Things Research Group/Laboratory. He is also the Co-Founder and the President of a healthcare startup Medsupervision Pvt. Ltd. His research interests include the IoT security, blockchain, UAVs, VANETs, 5G, and healthcare. He is a fellow of IET. He serves as the Co-Chair for various reputed workshops, such as IEEE GLOBECOM Workshop 2021, IEEE INFOCOM 2022 Workshop, IEEE ANTS 2021, and IEEE ICIAfS 2021, to name a few. He serves as an Area Editor for the *Ad Hoc Networks* (Elsevier) and the *IEEE Internet of Things Magazine*. He also serves as an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE NETWORKING LETTERS, *IEEE Consumer Electronics Magazine*, *IET Quantum Communications*, *IET Networks*, and several other journals. He is listed among the World's Top 2% Scientists identified by Stanford University.



**BIPLAB SIKDAR** (Senior Member, IEEE) received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He was a Faculty Member with the Rensselaer Polytechnic Institute

from 2001 to 2013, where he was an Assistant Professor and an Associate Professor. He is currently a Professor and the Head of the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His current research interests include wireless networks and security for the Internet of Things and cyber-physical systems. He has served as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE INTERNET OF THINGS JOURNAL, and IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY.

...