Performance Enhancement Schemes and Effective Incentives for Federated Learning

by

Yuwei Wang

Thesis Supervisor : Dr. Burak Kantarci

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The advent of artificial intelligence applications demands for massive amount of data to supplement the training of machine learning models. Traditional machine learning schemes require central processing of large volumes of data that may contain sensitive patterns such as user location, personal information, or transactions history. Federated learning (FL) has been proposed to complement the traditional centralized methods where multiple local models are trained and aggregated over a centralized cloud server. However, the performance of FL needs to be further improved, since its accuracy is not on par with traditional centralized machine learning approaches. Furthermore, due to the possibility of privacy information leakage, there are not enough clients willing to participate in FL training process. Common practice for the uploaded local models is an evenly weighted aggregation, assuming that each node of the network contributes to advancing the global model equally, which is unfair with higher contribution model owners.

This thesis focuses on three aspects of improving a whole federated learning pipeline: client selection; reputation enabled weight aggregation; and incentive mechanism. For client selection, a reputation score consists of evaluation metrics is introduced to eliminate poor performing model contributions. This scheme enhances the original implementation by up to 10% for non-IID datasets. We also reduce the training time of selection scheme by roughly 27.7% compared to the baseline implementation.

Then, a reputation-enabled weighted aggregation of the local models for distributed learning is proposed. Thus, the contribution of a local model and its aggregation weight is evaluated and determined by its reputation score, which is formulated as same above. Numerical comparison of the proposed methodology that assigns different aggregation weights based on the accuracy of each model to a baseline that utilizes standard average aggregation weight shows an accuracy improvement of 17.175% over the standard baseline for not independent and identically distributed (non-IID) scenarios for an FL network of 100 participants.

Last but not least, for incentive mechanism, we can reward participants based on data quality, data quantity, reputation and resource allocation of participants.In this thesis, we adopt a reputation-aware reverse auction that was earlier proposed to recruit dependable participants for mobile crowdsensing campaigns, and modify that incentive to adapt it to a FL setting where user utility is defined as a function of the assigned payment from the central server and the user's service cost, such as battery and processor usage. Through numerical results, we show that: 1) the proposed incentive can improve the user utilities when compared to the baseline approaches, 2) platform utility can be maintained at a

close value to that under the baselines, 3) the overall test accuracy of the aggregated global model can even slightly improve.

Acknowledgements

I would like to offer my deepest gratitude to my supervisor Dr. Burak Kantarci for the guidance he provided me to complete this journey. Without the help that he gave me in developing my methodology and the emotional support he gave me, I wouldn't have made it. Your suggestions brought in threads of thought that made my research so much richer. Thanks for giving me an opportunity in this field of study. I will use your selfless support as a model as I move forward.

I would also like to thank JiChu Jiang for his ideas and constant support academically as well as throughout my daily life.

Finally, I want to express my sincere gratitude to my parents for supporting both financially and emotionally all of these years. Saying thank you just doesn't seem to be enough for their continuous encouragement in every critical stage of my life. The completion of my dissertation would not have been possible without their nurturing.

Table of Contents

List of Tables

List of Figures

Abbreviations

DL: Deep learning

$F-MNIST$: Fashion-MNIST is a replacement handwritten digits datasets

FL: Federated Learning

$FMTL$: Federated Multitask Learning

$FedAvg$: Federated averaging: SGD-based averaging local aggregation method

IID: Independent and Identically Distributed

IoT: Internet of things

MEC: Mobile Edge Computing

ML: Machine Learning

MLP: Multilayer-perceptron

$MNIST$: Handwritten digits datasets

QoI: Quality of Information

QoS: Quality of Service

SGD: Stochastic Gradient Descent

$TSCM$: Trustworthy Sensing forCrowd Management

Nomenclature

ε_i	Marginal contribution brought by participant i
R_i	Reputation score of participant i
Δ	A temporary winner set constructed during rewarding stage
A_i	Local model accuracy of user i
B_χ	Training cost for user χ at specific communication round
χ	A selected participant for the temporary winner set in the payment determination period
$\mathfrak{B}^\tau$	Platform budget at round τ
$A_{g_{temp}}$	the accuracy of the temporary global model
$A_{g_{old}}$	the accuracy of the he global model of the last communication round
α	aggregation weight of each user i
ω_i	The aggregation weight for each user i
P_{aggr}	Aggregated model parameters i
P_i	The local model parameter of user i
f	Fraction of users i
B	Local Batch Size i
I	The parameters are updated every time a batch is trained. This process is called an iteration. i
E	Local Epochs i
L	Learning Rate i
$U_{platform}$	Platform utility i
U_{user}	User utility i
R_x	This denotes the region that the users reputation weight falls under on a Gaussian distribution in which is used to calculate the aggregation weight given to the user.
P_{aggr}	stands for the aggregated model parameters

Publications of the Candidate During MCS Studies

Publications that are the direct outcomes of the thesis:

- Wang, Yuwei, B. Kantarci, and Wail Mardini. "Aggregation of Incentivized Learning Models in Mobile Federated Learning Environments." In IEEE Networking Letters, DOI: 10.1109/LNET.2021.3108673, 2021. (**In Press**)
- Wang, Yuwei, and B. Kantarci. "A Novel Reputation-Aware Client Selection Scheme for Federated Learning within Mobile Environments." 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). IEEE, 2020.
- Wang, Yuwei, and B. Kantarci. "On the Aggregation of Incentivized Learning Models in Mobile Federated Learning Environment." In 2021 IEEE International Conference on Communications (ICC).

Chapter 1

Introduction

The increasing use of smart devices enable the Social Internet of Things (SIoT) phenomenon where smart devices equipped with sensors can interact with each other on various tasks without requiring human intervention [87]. This propels the use of machine learning (ML)-based methods in applications within smart cities such as smart surveillance and traffic control, although possible applications are not limited to these areas [95].

With the rise of Internet of Things(IoT), mobile devices are equipped with powerful sensors that can report highly accurate readings, which provides various opportunities for artificial intelligence (AI)-backed applications and services [39]. On the other hand, ubiquitous AI-backed services lead to data security and client confidentiality implications. Traditional machine learning schemes require central processing of large volumes of data that may contain sensitive patterns such as user location, personal information, or transactions history [95]. However, new legislation such as General Data Protection Regulation (GDPR) limits the feasibility of data collection [11]. Meanwhile, the exponential growth of data makes centralized cloud computing become not sufficiently compliant with data privacy legislation, as well as unacceptable end-to-end latency and messy data. As a result, user privacy concerns limit the range of data collection to what is consented to by participants and regulations. Large latency results in prolonged time consumption for training machine learning models to support real-time decision making, such as autonomous vehicular systems; high complexity of data increases the burden on the backbone network and reduces the speed of decision-making services. Unavoidable service delay caused by the long real-time response and limited network bandwidth impedes the progress of IoT services [55],[33],[59],[29].

Mobile devices contain quality data that can be beneficial for many ML models. Fed-

erated Learning (FL) has emerged as a viable concept to enable distributed learning while preserving the privacy of the users [39]. Different from transitional machine learning, which requires transmitting data from a local to a central server, FL only needs to update the parameters of a trained model. By utilizing local data from participant devices, FL becomes a distributed machine learning methodology that facilitates the training of individual models, which is then shared with a central computation unit for aggregation and redistribution until experiencing an eventual convergence in the performance indicators of the model [95]. Federated learning is performed iteratively as follows:

- In each iteration, the edge server sends the current global machine learning model parameters to all edge nodes participating in the federation (of learning);
- According to the received model parameters, each edge node uses the data samples stored locally to update the local model, such as calculating the gradient according to the loss function and updating the parameters;
- Each edge node uploads the updated model parameters to the edge server;
- The edge server performs a global aggregation operation and weights the local model parameters sent by each edge node to obtain a new set of global model parameters[82].

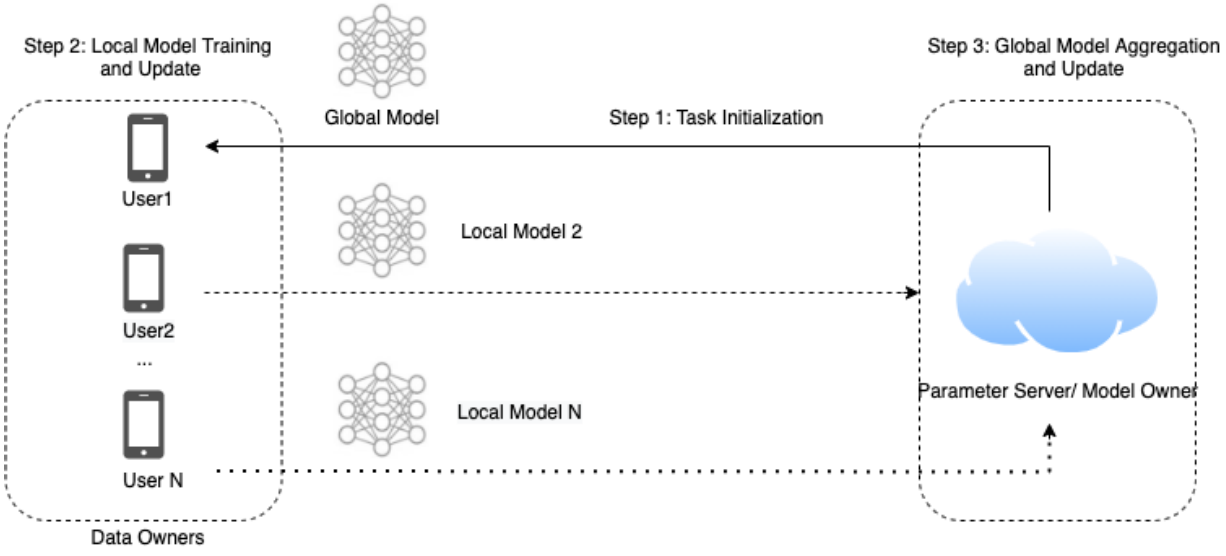

Figure 1.1: General FL training process involving N users.

Challenges of FL that are being addressed by the researchers have been centered around improving the cyberattack detection, as well as model compression and vehicular networks. However, aggregation of a global and distributed learning model remains a challenge. Meanwhile, motivating and selecting quality participants consistently in the FL framework need to be explored as well. The FedAvg algorithm, which stands for an averaging of models parameters, is often used as the standard aggregation method [53] with the consideration that each local model in the aggregation pipeline contributes equally [43]. This is unfair when higher quality model owners are rewarded equal payments and equal aggregation weights in comparison to worse-performing models. Moreover, without proper user selection, malicious users or users with insufficient, poor data or unreliable devices cannot be identified and phased out. Due to the non-direct ratio between users' payments and contributions, the platform utility and user utility cannot be maximized, which means the quality of the generalized final model can still be improved. This thesis tackles the federated learning framework in a more reliable and efficient way. Deep learning approaches such as Convolutional Neural Networks (CNNs) and A multi-layer-perceptron (MLP) are used for the machine learning model that will be trained within the FL scheme. However, it does not matter which models we use in our framework. They are just the deep learning models that we utilize to prove the improvement of our proposed methodology. We focus on three major aspects within Federated Learning to improve the scheme as a whole. Specifically, incentive mechanism, client selection, and model aggregation. Incentive methods are important at the front end of a scheme in order to attract users for participation. Afterward, client selection is required to phase out malicious or users with poor-quality models. Model aggregation is critical for assigning higher aggregation weights to the higher quality local model for more accurate and efficient model generation. By combining these three areas in a scheme, we are able to achieve faster convergence, higher accuracy, and a higher participation rate. It is worth noting that the FL framework is built by Pytorch and is an open resource from GitHub AshwinRJ. The purpose of using the baselines to compare with the proposed method is that 1) models from state of the art have different environment settings are completely different; 2) baseline models from Chapters 4 and 5 are the proposed method from the previous chapter. The proposed method of Chapters 4 and 5 are built upon the previous chapter's proposed method. By doing so, the improvement of the designed methodology can be clearly shown and compared.

1.1 Motivation

Various applications that leverage artificial intelligence (AI)tools and methodologies rely on locally acquired data samples for training machine learning models to make AI-backed deductions. For applications and services running on mobile devices, the common mode is as follows. The data generated by the user on the device is uploaded to the server and followed by a machine learning model (e.g., a neural network model deployed on the server) to be trained based on the collected large amount of data. The trained model can be distributed to the users. As the data on the user equipment continuously gets updated and uploaded to the server, the server will also update the global model based upon the updated weights, which results in an inevitably centralized trained model. As the amount of data increases exponentially, mobile cloud computing experiences bottlenecks in the communication network, long end-to-end latency, and user privacy concerns. Indeed, this protocol facilitates user experience by improving mobile applications and services, whereas user privacy remains an open issue as mobile device data needs to be transmitted to a centralized cloud server. This data transmission process can be attacked easily and lead to user privacy information leakage. On the server-side, cloud services need to build and train machine learning models to provide decision support or recommendation services; the training time of machine learning models can be prolonged by the size of data and complexity of the model. As a result, service delay is inevitable not only due to the limited network bandwidth but also the real-time response requirement by the services. Meanwhile, newly published legislation limits the amount and arrange of data usage, which can restrict the development of AI applications as well as their performance. Therefore, a distributed learning called federated learning is desired to address the issues mentioned above. The motivation of FL is to build a distributed ML framework that prevents mobile devices from transmitting raw data to the base stations [17]. By doing so, the privacy requirements of the users can be fulfilled since the data shared with the central server is limited to the model parameters. Thus, more clients can participate in achieving a more generalized model. Besides, due to the bandwidth constraints over wireless links, only the compressed machine learning models are downloaded and uploaded by the users, and this allows for real-time decisions made by the central server [68]. It is worth noting that low latency is a critical objective for time-sensitive applications such as autonomous vehicles. Last but not least, since the amount of transferred information is limited, the energy consumption of the network is significantly reduced, resulting in an overall higher transmission efficiency [72]. FL has shown to be promising in various applications such as mobile keyboard prediction, visual object detection, and so on [1].

Challenges of FL that are being addressed by the researchers have been centered around

improving privacy capabilities. However, aggregation of a global and distributed learning model still remains a challenge. The FedAvg algorithm, which basically stands for an averaging of models, is often used as the standard aggregation method with the consideration that each local model in the aggregation pipeline contributes equally. Assigning each recruiter the same aggregation weight is unfair for a high-performance local model, which cannot maximize the accuracy of a final global model. Moreover, when gathering data from users, it is possible that the acquired data is of poor quality. The reason can be either of the following: 1) malicious users, 2) unreliable device or 3) difficulty of obtaining labeled data sufficient to support the machine learning-backed applications in some highly specialized subdivisions (such as medical diagnosis). Consequently, poor data reduces the performance of the generated model and thus prolongs training times. A malicious user may steal privacy information from model parameters or may provide low-quality updates to obtain the global models. Malicious users may also leverage fake or poor data to obtain payment. Last but not least, some researchers focus on motivating clients with quality raw data to consistently join in FL training while ignoring the relationship between the payment, contribution, and its combined impact on the global output model.

1.2 Contributions

After studying the state-of-the-art federated learning approaches, an FL scheme for improving its efficiency and accuracy is proposed. Three methodologies were developed to support this scheme; one for reputation score-based client selection, one for assigning different aggregation weights according to the evaluation of its performance, and one for motivating clients with quality data and devices to continually join in FL training tasks.

Our main contributions can be summarized as follows;

- The first contribution introduces an optimal user selection method for federated learning based on reputation scores. In particular, a reputation score to assess the performance of each local model based on a variety of performance metrics is formulated (comparison with the performance of the local model at the current iteration; comparison with the performance of temporary global model generated from current iteration; comparison with the performance of global model from the last iteration). Experimental results show that the proposed reputation-aware FL scheme can achieve improvements in test accuracy varying between 1.73% to 9.30% under different data sets. We also reduced the training time by roughly 27.7% with our selection method compared to the baseline implementation.

- A reputation-enabled aggregation methodology that scales the aggregation weights of users by their reputation scores is proposed. The reputation score of a user is computed according to the performance metrics of their trained local models during each training round. Therefore it can be a metric to evaluate the direct contributions of their trained local model. The particular contribution of this methodology is an aggregation algorithm for FL in a mobile environment to ensure a high accuracy level. Simulation results show that the proposed aggregation methodology improves the accuracy of FedAvg by up to 17.175% for non-IID datasets. Meanwhile, the proposed scheme can converge faster at about 40 communication rounds, whereas the baseline model can achieve convergence at about 100 communication rounds.

- A reputation score-based incentive model for FL aggregation has been derived. Participants with higher quality data and high-performance local models will be assigned a higher payment by the central server. Our proposed incentive mechanism can dynamically adjust the users' compensation to distribute the benefits more fairly. Numerical evaluation has shown that the proposed scheme can improve user rewards but not compromise the platform utility and can even lead to a slight improvement in the test accuracy of the FL system.

.

1.3 Structure of the Thesis

The organization of this thesis is as follows:

This thesis is comprised of six chapters. In Chapter 2 we introduce the literature review of state of the art on federated learning, with specialization in incentive mechanism, reputation score, and aggregation. In section 2.1 we present the reputation score-based selection that has been used to choose quality local models. In Section 2.2, we present aggregation methodologies that have been proposed by researchers for federated learning. In section 2.3, we present the incentive mechanisms that have been adapted for federated learning.

In Chapter 3 we propose a reputation score-assisted client selection of filtering low-quality local models in a federated learning framework. Section 3.3 introduces calculation of reputation score in details. Section 3.4 explains the procedure of eliminating local models. Section 3.6 covers simulation settings and two schemes' performance under a

varied combination of machine learning and data-set. Section 3.7 presents an extended work on federated learning aggregation.

In Chapter 4 we design a reputation-enabled federated learning aggregation methodology in mobile platforms: Section 4.2 contains our proposed methodology to improve the performance of the final generated global model with a focus on assigning corresponding aggregation weights to each local model based on its evaluation. Section 4.4 explains the training details and evaluation metric include: three schemes' performance under the varied amount of participation, varied combinations of data-sets and machine learning models, and their convergence performance. Section 4.5 introduces the direction of the rest of the federated learning pipeline.

In Chapter 5 we design an incentive mechanism integrated into federated learning framework to optimize both platform and user utility: Section 5.1 gives an overview of the proposed methodology. In section 5.2, we present incentive solutions for higher quality local models. Section 5.3 covers experimental settings and performance of incentive mechanisms under different scenarios.

Finally, Chapter 6 summarizes methodologies introduced in this thesis and discusses the research direction that need to be addressed in the FL framework.

Chapter 2

Related Works

FL emerges as a solution for preserving privacy and data security; it is a distributed machine learning method that utilizes multiple mobile devices without uploading data. The mobile devices only need to update their model parameters to the server, in which the raw data is never exchanged. Contrast to traditional centralized machine learning approaches, where the local datasets are required to be transmitted to the server. FL allows multiple participants to collaborate to build robust, reliable machine learning models without accessing the data, thus enabling them to fulfill urgent demands such as obeying strict regulations regarding security and privacy while enlarging usable datasets.

The applications of FL are now widespread over various fields; these applications demand tons of data such as pharmaceutics and telecommunications. With its fast development, research communities have indicated the challenges and opportunities FL faces. Untrustworthy participants constitute one of the main concerns of applying FL on a larger scale. The widely used FedAvg method for global model aggregation also limits the final generated model's performance. Due to the prejudice of users that they firmly believe that their provided data for local training can cause individual information leakage, users may be reluctant to join in model training. Therefore, exploring suitable incentive mechanisms in order to motivate users for training is vital to a successful FL campaign.

2.1 Reputation Schemes and User Selection in Federated Learning

Lately, a number of existing studies have proposed to leverage federated learning in various environments. Just to name a few, the authors in [7] developed a new federated learning (FL) model to minimize the training errors and FL loss function while deploying FL over wireless networks to cope with the constraints of the wireless medium, including packet error and resource availability. To minimize communication traffic between all parties, the authors in [58] evaluate the performances of three federated learning algorithms and compare their performance against a centralized approach by using both IID (Independent and Identically Distributed) and non-IID partitioning of data from the MNIST dataset to achieve highest classification accuracy after a certain volume of communication traffic. A decentralized FL algorithm is developed in [68] to address challenges concerning transmit power and resource allocation to enable ultra-reliable and low-latency vehicular communications (URLLC). In [65], the authors combine multiple deep reinforcement learning (DRL) techniques with FL so to deploy the combined frameworks on Internet of Things (IoT) devices. The ultimate goal of the study is to instruct decisions in real-time to conserve more energy and maintain the quality of service (QoS) while computing the offloading decisions. The study in cite niknam2019federated introduced a decentralized scheme for Federated Learning (FL) to meet the needs of the training process for large-scale data and reduce the energy and network bandwidth consumption while ensuring privacy preservation.

Several researchers have explored the integration of FL with Mobile Edge Computing (MEC). Thus, multiple edge nodes that distribute the training process of a machine learning model in a MEC environment can directly use the data stored locally for model training without sharing their user data. The authors in [18] propose a new multiple access method to achieve rapid aggregation of global model parameters. In other studies such as [88, 41], reduction of the communication overhead is tackled for joint learning. More specifically, the authors in [88] propose a ternary gradient method to reduce the communication latency when federated learning is pursued, whereas the authors in [41] aim at improved communication efficiency of federated learning and propose two methods to reduce the communication cost of the model parameter upload link. Another study that tackled resource-constrained MEC environments and FL is presented in [81] to study the efficient use of limited resources to implement adaptive joint learning.

Kang et al. [26] combined reputation and contract theory of optimizing the reliability of federated learning. The authors developed incentive mechanisms to reward participating users depending on contribution. In that work, reputation was introduced to quantify

the reliability and trustworthiness of users and then use reputation as a selection method for federated learning. Although that work leverages user reputation, it focuses more on the selection of users based on contract theory while using a multi-weighted subjective logic model for reputation. Our work differs from theirs as we are directly formulating our reputation score through the test performance of the users (i.e., direct reputation) and particularly focusing on the improvements in the test accuracy of a federated learning model.

Unreliable and low-quality data shared by local clients can generate low-quality and untruthful trained local models. This can be caused by local clients unintentionally, high mobility of clients or energy and computation power constraints, or poisoning attacks, the malicious users that are trying to use fake data to fraud federated learning system and obtain information of the global model. To choose reliable and high-resources mobile clients, Kang et al.[29] introduced the concept of reputation with a reliable mobile clients selection approach to find out trusted mobile clients for the federated learning task. In this approach, a multiple weight subjective local model is developed to conduct the calculation of reputation metric efficiently, which is based on the local clients' interaction with parameter server and suggested reputation opinions. Furthermore, consortium blockchain is introduced in a decentralized way to manage mobile clients with non-tampering and non-repudiation.

To yield a desired performance global model, a sufficiently large amount of recruiters is usually needed. However, the limited bandwidth makes a federated learning system can only allow part of mobile volunteers to join in training task distribution and re-upload as well as re-download. To guarantee training efficiency, fairness, and quality of the final global model, Huang et al.[20] introduced client selection to reduce the time of model exchange when subjected to some rigid system constraints. Meanwhile, the stochastic and unknown time consumption of training, the status of interaction between task publishers and local clients, and the availability of mobile clients are considered. Then this offline problem is transformed into the problem of online Lyapunov optimization by leveraging dynamic queues to quantify mobile clients joining rate's long-term guarantee. To evaluate each mobile clients' model exchange time, a Contextual Combinatorial Multi-Arm Bandit model is designed, which corresponds to its reputation or historical performance and properties. Finally, a fairness guaranteed selection scheme is developed to combine the optimal problem addressed above with a federated learning framework efficiently.

Rehman et al.[80] introduced a novel definition called fine-grained federated learning, which can train shared models on edge servers in a decentralized way. Meanwhile, a formal extended concept called fine-grained federated learning process for the mobile edge computing framework so as to define the fine-grained federated learning framework's requirements

such as incentive mechanism, bandwidth efficiency, activity monitoring, communication, heterogeneity, and model synchronization. Furthermore, the definition of reputation-aware blockchain-based fine-grained federated learning is introduced to guarantee reliable collaboration training in the mobile edge computing framework.

2.2 Model Aggregation in Federated Learning

Sun et al.[76] utilized digital twins to locate real-time devices' performance and operating state in a digital world. To quantify clients' contributions for the global model, a strategy that can use the true value of digital twins' deviation is employed to improve the performance and reliability of trained models. To minimize the loss function of federated learning, and adaptive calibration of a global aggregation frequency scheme based on a deep Q network is designed to trade-off dynamically between interaction and computation energy consumption under a limited resource budget. To assist federated learning schemes to adapt to the heterogeneous internet of things, an asynchronous federated learning scheme is proposed to enhance the training and interaction efficiency and decrease edge nodes' straggler effect with inter-cluster time-weighted aggregation mechanism. The deep Q network based on frequency calibration is then leveraged to decide different clusters' aggregation frequency.

The heterogeneous nature of the mobile devices in a distributed learning environment requires optimization strategies subject to the computing power, bandwidth constraints, and quality of sensed data [70][9]. To ensure that the ML models can be trained within a predetermined delay bound, the authors in [59] propose the FedCS scheme that builds on the server's awareness of the resource availability at the participating devices so as to choose the clients (i.e., devices) accordingly. It should be noted that it may not always be possible to obtain the needed training time for complex models. The authors in [97] introduce a Hybrid-FL scheme that aims to select the participants with IID (independent and identically distributed) datasets. On the one hand, enforcing IID datasets can improve the performance of the trained FL model. However, this may lead to privacy concerns as some data must be shared to ensure that all recruited participants have IID datasets. The authors in [56] leverage Deep Reinforcement Learning (DRL) to filter the clients that are out of the server's coverage. A possible extension to that study could be testing the boundaries of the participant pool, i.e., how big should participant pool be in order for the DRL-based model selection to work efficiently. A q-Fair FL algorithm is proposed in [40] to calculate each participant's test accuracy variance so as to integrate them into the FedAvg algorithm as aggregation weights aim at fairness.

Table 2.1: Overview of Related Works for Reputation Schemes and User Selection

Approaches	Ref.	key Ideas
Reputation Schemes and User Selection	[29]	a multiple weight subjective local model is developed to conduct the calculation of reputation metric efficiently
	[20]	select client to reduce the time of model exchange
	[80]	reputation aware block-chain to guarantee the reliable collaboration training in FL
	[50]	select data provider and authenticate the truthfulness of shared data
	[102]	replace traditional aggregator with blockchain to filter out unreliable users
	[67]	enabling clients make auction for highly reputed clients to choose truthful clients
	[92]	leverage reputation scores to eliminate low contribution participants
	[64]	reputation score to detect malicious attackers.
	[19]	E3CS aggregation scheme to select clients based on its update performance
	[63]	Blockchian combined scheme to select local clients as well as verify clients in a fast convergence way

The authors in [94] improve aggregation by adopting a difference-of-convex (DC) functions algorithm. In [74], the authors employ an asynchronous FL strategy to ensure the uploaded parameters by the local mobile devices can proceed to the aggregation procedure immediately. It might be possible that a non-IID dataset incurs high latency for convergence. The study in [81] tackles the frequency of aggregated updates so as to scale them in accordance with the restrictions of wireless resources with a fundamental assumption that a guaranteed convergence is possible for every model.

A model selection aggregation method is proposed in [100], local computation ability, as well as the image quality, are used to identify the good quality local deep neural network (DNN) models. To optimize the number of workers scheduled at each epoch, an online energy-limited dynamic worker arrangement policy is developed in [77]. In [16], an analog gradient aggregation method is proposed aiming at fast convergence in the FL network. The study in [6] introduces a communication-efficient secure aggregation to decrease the energy consumption of communication. The study in [8] proposes an asynchronous learning strategy in which the deep layers update at a slower frequency and the shallow layers update at a faster frequency.

The original intention of federated learning was to preserve the confidentiality of users' personal information. Thus, the central server cannot inspect how these local models are generated, which results in a much more powerful data poisoning attack in federated learning. To avoid malicious participants sneaking into the glocal model, [4] proposed a model-replacement scheme to enable a word predictor to recognize certain texts that contain some attacker-chosen word or label.

Due to ever-increasing artificial intelligence applications in vehicle edge computing, the computation ability of vehicle users and the quality of image they collected is vita to the final global model's performance such as efficiency and accuracy.[100] developed a selective model aggregation scheme to evaluate clients' computation capability as well as trained local deep neural network models before sending it to central server participant on aggregation process. Then the authors employ two-dimension contract theory as a distributed frame to solve the problem of central server's information asymmetry. During the interactions between vehicle users and the central server, this problem is transformed into a tractable problem that continually simplifies and relaxes constraints and then employs a greedy algorithm to solve it eventually.

There are some bottlenecks for federated learning, such as non-independent and identically distributed data and high transmission energy consumption. In [77], the authors propose an energy-aware dynamic scheme to scale down the transmission cost corresponding to the number of participants so as to maximize the amounts of updates. Data redundancy

is introduced to solve the problem caused by non-independent and identically distributed data.

To scaling federated learning, the overhead of local model aggregation for many workers arises a secure problem. So et al.[73] designed a secure aggregation scheme called Turbo-Aggregate to achieve the purpose of secure aggregation under the guarantee of users' dropout rate less than 50%. This scheme utilizes coding techniques and secret sharing to inject insecure model aggregation for the privacy of users while recruiting a bunch group of circular strategies to guarantee the efficiency of local model aggregation.

[6] proposed an auto-tuning efficient, secure aggregation framework that employs random rotation to enable the workers' updates to be quantized more efficiently while guaranteeing the central server learns a certain amount of workers' model contribution without leaking individual's contribution in an unaggregated form.

Through Yap et al.'s theoretical analysis [96], there are arguments for federated learning: the overhead of local model aggregation will lead to gradient biases; the optimization objectives and expected target contribution is inconsistency in FedAvg training instances. To tackle the problem mentioned above, they developed an unbiased gradient aggregation solution with gradient evaluation and gradient descent algorithm. To provide a consistent objective and prove the expected distribution, the authors also formulated a controllable meta updating algorithm.

The limited transmission bandwidth becomes a big obstacle for updating the federated learning's local models. Yang et al. [94] proposed a fast aggregation framework based on the principle of over-the-air computation through exploiting wireless multiple channel's superposition properties. Beam-forming design and joint device selection consist of this framework, which is transformed as a low rank and sparse problem. To solve this problem, a difference of convex functions is designed to decrease the low rank and sparse problem. A difference of convex algorithm is further designed through continually convex relaxation to guarantee the convergence rate.

One of the federated learning's challenges is to decrease the communication between users and servers since local devices usually have limited transmission bandwidth. Chen et al. [8] designed an asynchronous learning approach over a temporally weighted aggregated model on the central server and the users. In this asynchronous learning approach, deep neural networks' different layers are grouped into deep layers and shallow layers and the parameters of shallow layers are aggregated more frequently than the deep layers. Then, a temporally weighted aggregated model on the central server by utilizing the trained local models is leveraged to improve the global model's performance, such as test accuracy and convergence speed.

The limited resource environment and unstable local gradients have become the main bottleneck for federated learning. Previous work is only focused on exploring the channel state information; Guo et al. [16] present an analog gradient aggregation approach that can improve traditional channel state information transceiver adaptation. In this approach, the transceiver's parameters are optimized with non-stationary local parameters based on variable feedback. To solve aggregation errors caused by noisy communication, the authors also propose a novel learning rate stochastic gradient descent, which corresponds to the quality estimation of local gradients in each iteration. The proposed approaches greatly improve the convergence speed during federated learning iterations.

The requirement of the state-of-the-art federated learning protocols needs each device to quantify its trained local model into the same level of quantization, which greatly decreases the model updates' performance due to the lack of different bandwidths at different local devices. Elkordy et al. [14] design a secure aggregation scheme that allows local devices to update privately while can utilize different quantization. This scheme enables devices to consistently adjust the level of their quantization to the bandwidth that is available to them, which can significantly improve training accuracy as well as time-consuming for transmission. In particular, the network of devices is partitioned into several groups and divides different devices' local updates into segments. The aggregation protocols are then applied to segments that have specific communication between devices.

Most of the existing research' simulation is based on unlimited resource assumptions and cannot monitor federated learning systems dynamically. Therefore, Lu et al. [51] present digital twin edge networks, which is developed by adopting digital twins with local edge network to fill up the digital system and physical edge network's gap. A block-chain empowered federated learning mechanism is then developed to enhance the protection of data privacy and security of interaction for digital twin edge networks, which is developed by adopting digital twins with local edge networks to fill up the digital system and physical edge network's gap. A blockchain empowered. Meanwhile, reinforcement learning empowered by digital twin and asynchronous aggregation algorithm is developed to strengthen the proposed scheme's efficiency.

2.3 Incentives for Federated Learning

Incentive mechanisms are an important research direction for federated learning. [21] propose a FedAR algorithm to give a punishment or reward to participants based on their models' performance as well as accelerate the training process under limited local computing resources. In [101], Zhan et al. combined a Stackelberg game with FL and proposed

Table 2.2: Overview of Related Works for Model Aggregation

Approaches	**Ref.**	**key Ideas**
Model Aggregation	[76]	time weighted aggregation mechanism to decide different clusters' aggregation frequency
	[4]	word predictor to recognize attacker-chosen word or label before aggregation
	[100]	selective model aggregation scheme to evaluate clients' computation capability before aggregation
	[77]	energy-aware dynamic scheme to maximize the amounts of updates
	[73]	Turbo-Aggregate to enable security of aggregation
	[6]	auto-tuning efficient secure aggregation framework to make sure updates efficiently
	[96]	unbiased gradient aggregation to make target contribution consistency in FedAvg training instances
	[94]	fast aggregation framework to solve the limited transmission bandwidth
	[8]	asynchronous learning approach over a temporally aggregated model to decrease the communication between users and server
	[16]	analog gradient aggregation approach to improve traditional channel state information transceiver adaptation
	[14]	ecure aggregation scheme to allow local devices update privately
	[51]	asynchronous aggregation algorithm to strengthen FL system efficiency

Nash equilibrium to evaluate clients' performance. A fairness-aware incentive scheme is formulated by Yu et al. [98] to motivate users with high-quality data and meet the requirements of fairness criteria through a real-time algorithm. Liu et al. [48] derived a FedCoin scheme, which is a blockchain-based payment system to assess clients' contribution fairly and assign a feasible profit distribution. Kang et al. [26] design a theory-based contract incentive mechanism to select high-reputation workers and stimulate them to participate in FL to prevent a malicious attack. Khan et al. [31] present a Stackelberg-game-based scheme for FL incentive mechanism to enable FL participants and base station to strategically maximize their utility. Bao et al. [5] developed FLChain to assess the contribution and reliability of FL participants as well as a federation establishment algorithm to maximize overall participant profit. Toyoda et al. [79] derived a competitive incentive mechanism combined with a full-fledged protocol for blockchain. However, none of the above works have a double selection(reputation score selection and platform utility selection) for participants to guarantee a virtuous circle for the FL platform.

Mobile Crowdsensing is a well-developed field where its application is similar to federated learning in as users are recruited to participate and compensated with payment for their contribution. A game theory such as contract theory, auction theory, and the Stackelberg game have been applied to mobile crowd-sensing(MCS) systems to improve its profitability and reliability. In [23], Sun et al. proposed an incentive scheme based on the heterogeneous belief values in real-time to maximize the total sensing profit for consecutive horizontal crowd-sensing. Payment to the workers is based on the sensing data quality they provided. Wenet al [89] proposed a quality-driven auction(QDA) based incentive scheme. A deferentially private incentive scheme that is dependent on singleminded reverse combinatorial auction has been proposed by Jin et al. [25]; it is used to preserve workers' personal information such as bid price and minimize the platform' total cost. Thepvilojanapong et al. [78] developed a SenseUtil scheme which is based on the concept of microeconomics to attract more participants to collect sensing data while guaranteeing the platform payment at a moderate ratio. G. Jaimes et al. [22] present a cooperative incentive mechanism that is based on the user's behavioral model in reverse auctions to motivate participants' cooperation instead of participants' competition to guarantee more participation. Liu et al. [46] proposed a novel incentive method that applies the reverse auction for location-aware sensing(IMRAL), it can assure higher assignment coverage ratio and higher user utility; a user interaction incentive model(UIBIM) is also designed to prevent situations where a user may retreat from MCS system.

Due to the fact that the incentive mechanism cannot be directly applied to federated learning, since the unshared information as well as hard to evaluate participates' contribution at federated learning. Zhan et al. [101] developed a deep reinforcement learning-based

incentive scheme to motivate the edge devices to participate in federated learning training. In particular, this scheme focus on determining the optimal strategies for local devices as well as the optimal payment strategies for server. The Stackelberg game is integrated into federated learning, and the Nash equilibrium is leveraged for the server to evaluate trained models' contributions with respect to their training accuracy. Finally, a deep reinforcement learning-based incentive algorithm is employed dynamically.

The selection for workers who participate in federated learning training has not been studied and adopted in the incentive mechanism yet. Thus, Kang et al. [26] first designed a reputation metric to evaluate the performance of local devices. A multi-weight logic model is then developed to help select the reliable local trained model based on reputation metrics. A decentralized blockchain is leveraged to make sure the security of workers' reputation management. Meanwhile, contract theory combined with reputation consists of an incentive mechanism that is proposed to motivate high-performance local devices with high-quality local data to participate in training tasks.

Consistent long-term participation and motivating high-performance edge nodes are critical to the fairness of federated learning. Yu et al. [98] proposed an incentive scheme, which can dynamically adjust payment. This scheme is a polynomial algorithm that achieves fairness by installment among participants. It divides the platform budget among participants in real-time by minimizing the unfairness of participants in terms of the whole time to receive the full payment and amount of payment while maximizing the total platform budget. Even the expectation payment is achieved and paid to participants, the server will continue to pay the participants based on its further contribution.

In [31], Khan et al. designed a Stackelberg game-based mechanism to attract mobile devices to join in federated training tasks. In this incentive mechanism, mobile devices can decide the number of local training communication rounds to maximize the user utility. Moreover, the central server can choose the best performance of the locally trained model to optimize the accuracy of a global model. Meanwhile, the central server's utility is calculated through key performance metrics' functions such as the global model's accuracy and the number of global communication rounds.

There are some challenges for federated learning, such as mobile devices abort training or attackers sending poison information to participants. This situation may cause miscalculation of parameter servers and no sufficient motivation to attract mobile devices with high-quality data and high computation resources. Bao et al.[5] developed a federated learning chain to build an incentive, trustworthiness, and reliable Fl environment. In this chain, users' contributions and reliability are evaluated to distribute profit fairly, and a federation establishment algorithm is designed to maximize the total users' profit. Finally,

a method called DDCBF is present to decrease the time consumption of chain queries for unreliable behavior detection and assign profit.

Evaluating workers' efforts such as Shapley's value is important to yield a fair payoff. However, naively adopting Shapley value into federated learning will have high time consumption as well as take too much computation resources. Therefore, Liu et al. [48] proposed a FedCoin system that is based on a blockchain. In particular, the Shapley value of each local device represents its contribution to the global model, which is calculated by the proof of Shapley value consensus protocol that replaces transitional hash-based protocol. All the payments are then determined and recorded in the blockchain in an immutable way. Under the FedCoin system, the payoff is assigned in a decentralized way, which does not need to rely on a central server so as to provide fairness to the incentive system through a third party.

Under training assigned FL task, there will be no monitoring, and users' honesty is not guaranteed. Furthermore, a rigorous reward policy is needed to maximize both server and users' utility. Toyoda et al. [79] designed a novel methodology by using mechanism design to realize the desired purpose when users act rationally. The intrinsic idea is to motivate users to compete with each other and follow the stated protocol to achieve a win-win. A full-fledged generic protocol is designed by mechanism design, which can be adopted to the existing public chain. The incentive competition is achieved by an auction-based game theory which is called contest theory in economics.

In federated learning frameworks, mobile devices usually prefer not to participate in training tasks for free, and they may provide different dimension resources, which will result in lower performance of the FL framework. To address the challenges mentioned above, Zeng et al. [99]propose an incentive mechanism called FMore that can procure an auction of multiple winners at multi-dimension. This scheme is incentive compatible and lightweight, which covers scoring functions and employ game theory to optimize local participants' strategies. Furthermore, the utility theory is introduced to assist global aggregators in recruiting high-quality resources with low cost to achieve fast convergence.

The convergence rate of distributed learning is an important evaluation metric. However, due to the heterogeneous of mobile edge nodes, high latency is an obstacle for federated learning fast convergence. To tackle this problem, Sarikaya et al. [69] introduce the Stackelberg game into federated learning protocol to attract local edges to mitigate the latency in each batch. In the parameter update step, the local edges will decide computation resources assigned to calculate the parameter from privacy data based on incentive profit offered by the model owner. The model owner is the buyer that distributes its finite budget unevenly among local edges based on their contribution. The equilibrium solution

is derived by quantifying the average time consumption of a single iteration.

Pandey et al. [61] developed a utility maximization problem to motivate participants to build a high-quality global model efficiency. A crowdsourcing framework is developed to increase the efficiency of communication in the federated learning system. This framework helps local devices determine learning problems based on offered incentives. The authors then designed an incentive mechanism to reduce the unnecessary communication between clients and parameter servers. A two-stage Stackelberg game is formulated, the first stage is for local devices to select the best strategy to minimize their training cost with offered incentive, and the second stagy is for parameter server to build a global model by utility function to offer a reward with local devices. Then this two-stage Stackelberg game is transformed into an equivalent optimization problem, which is solved by exponential complexity effort.

In mobile crowd-sensing, a parameter server may not have sufficient data to build a high-performance model. Thus, Niyato et al. [45] present privacy preserved incentive approach to utilize collaborative machine learning in local devices. Under information asymmetry, the authors facilitate a self-revealing mechanism with contract theory to solve the mismatched between parameter servers and local devices. To prevent free-riding attacks while keeping the stability of the federated learning system, the coalitional game theory is employed, which rewards parameter servers corresponding to its marginal contribution to solving the mismatches between parameter servers. Finally, a hierarchical incentive approach is proposed to solve the contract algorithm and then solve the coalitional game by leveraging the split and merge algorithm with the inherent hierarchical structure of the whole federation system.

To solve the problem that is unrealistic to assume every local client is willing to share their computation and information resources to join in federated learning training task, Thi Le et al. [36] proposed an auction scheme. In this scheme, a base station with multiple local clients is considered. They assume that the local clients are sellers, while the base station is auctioneer, and an incentive mechanism is developed between them. During the auction process, the local clients update their prices based on their minimal energy consumption under federated learning task training. The minimal energy consumption is based on selecting the optimal strategy for its local resources and accuracy. Then a primal-dual greedy auction mechanism is leveraged to determine winners corresponding to maximize base station profit.

Ding et al. [12] present research on multiple dimensional federated learning incentive mechanisms under different information deficiencies. They study the balance of base station between payoff to users and global model's accuracy, and the results showed that

such balance would transfer into balance on multiple dimensional of local clients' privacy information. The base station's balanced problem is solved by formulating multiple dimensional contracts. The two-dimensional privacy information is converted into one-dimensional, which help the base station to characterize and evaluate different payments to local clients' preference.

Table 2.3: Overview of Related Works for Incentives

Approach	Ref.	key Ideas
Incentive Mechanism	[52]	reputation mechanism to reward local clients to guanrrantee fairness
	[101]	deep reinforcement learning-based incentive scheme to motivate the edge devices to participate training
	[26]	incentive mechanism to motivate high performance local devices with high quality local data to participate in training
	[98]	incentive scheme to dynamic adjust payment
	[31]	Stackelberg game based mechanism to attract mobile devices
	[5]	FL chain to build a incentive, trustworthiness and reliable Fl environment
	[48]	FedCoin system to determine and assign payoff fairly
	[79]	incentive mechanism to motivate users compete with each other and follow the stated protocol to achieve win-win
	[99]	FMore to procure auction of multiple winners at multi dimension
	[69]	stackelberg game to attract local edges to mitigate the latency in each batch
	[61]	utility maximization method to motivate participants to build a high quality global model efficiency
	[45]	privacy preserved incentive approach to solve the mismatched between parameter server and local devices
	[36]	primal dual greedy auction mechanism to determine winners corresponding to maximize base station profit
	[60]	utility maximization problem with a novel crowd-sourcing framework to enable a certain amount of local users to train models
	[10]	FVCG to motivate local participants join in training consistently and submit their minimal computation costs honestly
	[44]	contract theoretic task aware incentive mechanism to adjust payment dynamically depend on service latency and age of information
	[102]	incentive smart contract to attract local users share their computation resources and privacy data
	[83]	two tier incentive mechanism to motivate unmanned aerial vehicles' high quality trained local model
	[66]	budget bounded incentive mechanism to guarantee bounded budget is assigned fairly to local devices
	[47]	block-chain FL framework to motivate and benefit local clients from their useful resources and detect malicious users

Chapter 3

Reputation-aware Client Selection Scheme for Federated Learning within Mobile Environments

3.1 Introduction

This chapter studies the problem of training federated deep learning models over a mobile environment. Stemming from the federated learning concept, deep learning models on mobile devices can be trained for various use cases including but not limited to image classification and prediction of keyboard typing words. Mobile devices have access to rich data sets through embedded sensors and as well as installed software, and these feature rich data can facilitate solid training models, including personal images and other behavior metric features. However, utilizing the data through conventional approaches can potentially lead to privacy leakages. In this chapter, we propose an alternate strategy that builds on the Federated Learning (FL) concept, to keep the training data on distributed mobile devices, and train a shared model by aggregating updated local models. The contribution of this chapter is an optimal user selection method for the federated learning environment based on reputation scores. Through extensive validation experiments considering two different model architectures and three datasets, our experiments show that the proposed approach is stable over data that is not independent nor identically distributed (i.e., non-IID).

3.2 Methodology

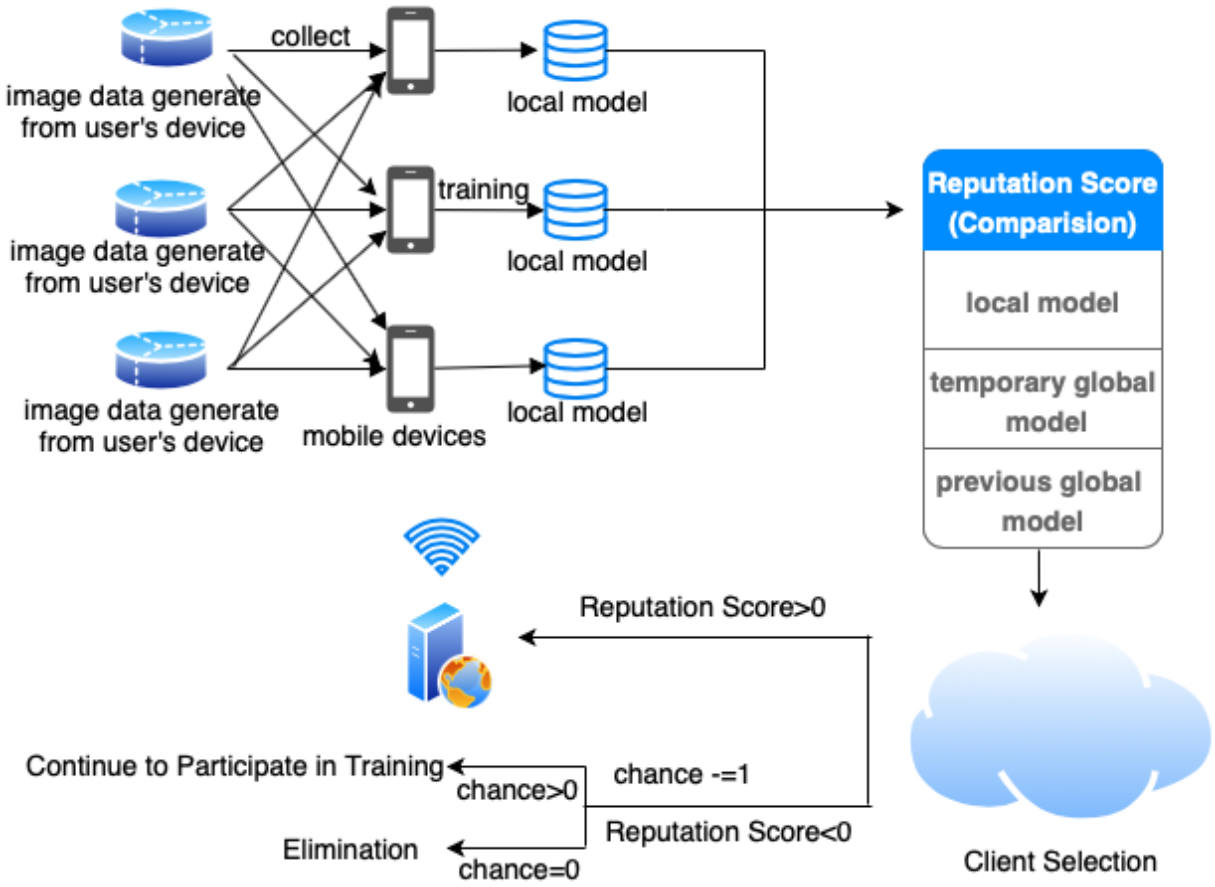

Figure 3.1: Federated learning in mobile setting

The proposed methodology aims to improve the use of federated learning in a mobile training environment particularly to help address the privacy concerns. To this end, its objective is to obtain an aggregated global model (i.e., final model) with higher accuracy by eliminating poorly performing models or models that do not contribute to the improvement of the global model. The eliminated models could be provided by either of the following sets of users: 1) malicious users that also pursue data poisoning attacks, 2) cooperative users that could contribute with data of poor quality. The proposed framework to eliminate the contributions of these sets of users is illustrated in Fig. 3.1. As shown in the figure, elimination of the poor contributions to the aggregated model is performed through the

calculation of a reputation score for each user. Reputation score of a user denotes their contributions and performance.

Given a base station that cooperates with the user devices to execute an federated learning scheme with a set of C clients. Being committed to keeping all datasets in user devices, federated learning ensures that the base station and users learn a shared learning model collaboratively. In federated learning, each client trains their local model by using the training data generated locally at their mobile device. Global model is aggregated at the base station and distributed as a shared learning model. The shared federated learning model is used to improve each user's local model so to enable users to perform a federated learning task without transmitting their data to a central unit. Thus, the global federated learning model is generated at the base station by using local model parameters of each client. The parameters of a local model are uploaded from the client to the base station while the global federated learning model is downloaded by each user device.

In a mobile environment with N users (i.e., clients), let e and i represent the number of epoch and the id of a local client that has participated in training, respectively. $Local[i]$ stores the accuracy, training loss and accuracy score of user i during the training process. The training results of a specific client in $e-th$ epoch are saved in $Local[i][e]$. The global model features are stored in dictionary, $Global[]$.

Algorithm 1 illustrates the overall process of our proposed user selection scheme. Algorithm 1 chooses the client to participate in training in current epoch based on the reputation score of the models updated by selected client in previous epochs

Algorithm 1 Client selection procedure

Data: Data of local model, dictionary Local[]
Result: L: list of client IDs
for i *in* N *users* **do**
  if *Local[*$client_i$*] is empty dictionary*
  //first epoch or first time this client being selected **then**
    $L = L + i$;
  else
    e = number of epoch
    loop over e
    let n_i be the total of (Local[$client_i$][e]['score'] <0)
    if n_i >2 **then**
      delete i from L
    end
  end
end

Figure 3.2: L12:Algorithm 1

3.3 Reputation score

Reputation score is an evaluation for each local model to determine whether the parameters of local models should be chosen for aggregation into the global model. Since there are many under performing models as well as malicious users to reduce the accuracy of global model, reputation score helps to make a judgement so to eliminate those under performing models. Each user has a record and a reputation score which will be initialized at the beginning of each epoch.

Algorithm 2 describes the complete process of reputation score calculation. The reputation score is calculated from 3 parameters:

$$R_i = w_1(A_i - \sum_j A_j/n) + w_2(A_i - A_{g_{temp}}) + w_3(A_i - A_{g_{old}}) \tag{3.1}$$

Data: Data of local model Local[], and data of previous global model Global[]
Result: Local[i][e][score]
for *all user i* **do**
  total += Local[i][e][accuracy]
end
avg = total / n
for *all user i* **do**
  if *Global[e-1] has no value* **then**
    Local[$client_i$][e][score] = (Local[$client_i$][e][accuracy] - avg)*w_1 + (Local[$client_i$][e][accuracy] - Global[e][test accuracy])*w_2;
  else
    Local[$client_i$][e][score] = (Local[$client_i$][e][accuracy] - avg)*w_1 + (Local[$client_i$][e][accuracy] - Global[e][test accuracy])*w_2 + (Local[$client_i$][e][accuracy] - Global[e-1][accuracy])*w_3;
  end
end

Figure 3.3: Algorithm 2

The training stops when the learning curve converges or the system run out of local users. The accuracy is yield through calculating the ratio between the amount of correct predictions to the total amount of predictions, the range of accuracy range from 0 to 100%. In Eq. 3.1 test accuracy of each local model needs to be compared with the three metrics: (1) Comparison with the test accuracy of local models in each epoch: the trained local models of each epoch are used to compute an average test accuracy. The models that perform worse than average will be less favored whereas those outperform the average will be favoured. (2) Comparison with a temporary global model: the local models that are trained further but have not been aggregated with the global model are used to generate a temporary global model. The temporary global model should perform better than each local model. Therefore comparison with the temporary global model will often result in a negative contribution to the reputation. By utilizing a temporary global model we asses the possible aggregated performance of all local models for a specific iteration. The purpose of this metric is to choose the best models at each epoch. The poor models with negative reputation get eliminated before aggregation into the global model. (3) Comparison with the previous global model of the last epoch: each local model's test accuracy is compared to the test accuracy of the previous global model from the previous epoch to assess the improvement. Through comparison with each metric, if the reputation score is positive, it indicates a positive contribution for the local model. This is due to the three comparisons based on the local model test accuracy. Poorer quality of local models with respect to the temporary global model would result in negative impact by this comparison metric. As the local models are trained further after each epoch, their performance is likely to improve, therefore when compared to the previous global model, it will often provide positive contribution to the reputation score of the local model, signifying an improvement over previous global model.

3.4 Elimination of local models

The usefulness of local models is calculated through their reputation score. The reputation score is a weighted function of various test accuracy comparisons. If the local model is determined to be under-performing, its parameters are not aggregated into the global model in that epoch. A record is kept for each local model to keep track of the number of times it has been declined for global model aggregation. The local model is eliminated from training when it is declined for a certain amount of times. A local model that is declined multiple times due to poor performance results would often continue to perform poorly relative to the other local models. Therefore, by eliminating the constantly poorly

performing models, it becomes possible to decrease the total running time while improving performance. This is shown in our experimental results.

3.5 Simulation Settings

We evaluate of our proposed methodology on three real image datasets: MNIST, Fashion MNIST, and CIFAR-10. For the sake of credibility of our proposed scheme, the data is chosen to be similar to the images generated by real mobile devices. Datasets are explained in detail below:

- MNIST[37]: MNIST is a handwritten digital dataset, where the training dataset contains 60,000 images and the test dataset contains 10,000 samples. Each image in MNIST dataset consists of 28 x 28 pixels, each pixel is represented by a gray value .
- Fashion-MNIST[90]: Fashion-MNIST is an image dataset that replaces the MNIST handwritten digit set. It covers a total of 70,000 different positive images from ten categories. The size, format, and training / test set division of Fashion-MNIST are identical to those of MNIST.
- CIFAR-10[35]: The CIFAR-10 data set consists of 10 types of 32x32 color pictures that are evenly distributed over 60,000 images. Among these, 50,000 pictures are used for training whereas 10,000 are used for testing. The CIFAR-10 dataset is divided into 5 training batches and 1 test batch, each training batch contains 10000 images where 1000 images are randomly selected from each category. The training set batch contains the remaining 50,000 images in random order.

Our proposed solution builds on two model families on the three datasets listed above. The following two models are used to address handwriting digit recognition: (1) A multi layer-perceptron with 1-hidden layer with 64 units each using ReLu activation. (2) a Convolutional Neural Network (CNN) with two convolution layers of 5×5 (the first layer with 10 channels, the second with 20 channels), which contains a full connected layer of 320 units.

Different types of distributed data from clients influence the federated learning optimization. We investigate two ways of distributing the MNIST datasets over participants (i.e. mobile device users) as follows:

Table 3.1: Selection of the best number of chances given to the poor performing local models before eliminating them

	Base	0	1	2	3	4	5	6	7
Train Accuracy	93.16	92.54	92.69	95.37	95.17	95.23	94.32	94.72	950
Test Accuracy	90.56	89.68	89.65	92.49	92.33	92.48	91.56	92.21	92.62
Loss Value	-0.612	-0.802	-0.831	-0.824	-0.809	-0.826	-0.812	-0.785	-0.822

- non-IID (Independent and Identically Distributed): the digit label is added to sort data, and partitioned into 1200 groups of 50 samples per group. Then, each user is assigned a minimum of one and up to 30 group data randomly.
- IID: the shuffled data is partitioned to 100 users so that each users acquires 600 images.

The reputation score is calculated in every epoch after the local models' training has been completed. There are three parameters that control the computation process: (1) f: the fraction of users that are selected to perform computation on each iteration; (2) B, the batch size of local updates in each user;(3)E, the number of local epoch; (4)L, a hyperparameter used in the training of federated learning, controls the speed of local model adapted to the problem; (5)w, the weights of reputation score, which is determined empirically. Choosing 100 users is based on the number of users selected in the state of art

3.6 Experimental Results

Table 3.1 uses the MNIST database to find the optimal number of chances given to the users before they are eliminated, the Base model means all local models be used regardless of their performance. 0-7 record represents elimination after reputation falls below threshold which the value is 0. For instances, 0 means the local model is eliminate after not reaching the threshold once; 1 means local model is eliminated after not reaching the threshold twice. From Table 3.1, we see that, as the number of eliminated models increases, the value of test accuracy has been improved. Moreover, as the number of eliminated models

Table 3.2: Comparison of users for 10 epochs

Fraction of users	0.1	0.2	0.3	0.4	0.5
Train Accuracy	95.655	95.125	93.09	93.07	91.81
Test Accuracy	92.752	92.705	91.015	90.69	91.81
Loss Value	-0.74	-0.824	-0.785	-0.782	-0.782

Table 3.3: Parameter settings in the simulations

Parameter	Setting
Fraction of users	f = 0.1
Local Batch Size	B = 10
Local Epochs	E = 10
Learning Rate	L = 0.01
Weights of Reputation Score	$w_1 = w_2 = w_3$

Table 3.4: Test accuracy improvement under different datasets

Test Accuracy	MLP IID MNIST	MLP Non-IID MNIST	MLP IID F-MNIST	MLP Non-IID F-MNIST	CNN IID MNIST	CNN Non-IID MNIST	CNN IID CIFAR	MLP IID CIFAR
Proposed method	92.324	82.627	92.832	81.473	98.25	90.381	47.651	38.927
Baseline method	88.023	76.389	89.44	76.854	96.517	81.075	45.001	35.592

increases, the time consumption increases. This is due to the fact that an increase number in the number of eliminated models leads to more calculation procedure, hence, improving test accuracy as well as training accuracy. The record = 2 is selected due to the fact that the proposed methodology jointly considers the improvement of test accuracy and time consumption, hence, it can optimize the user selection to reduce the loss value as well as increase the test accuracy.

Table 3.2 presents the performance under varying fractions of users (f).

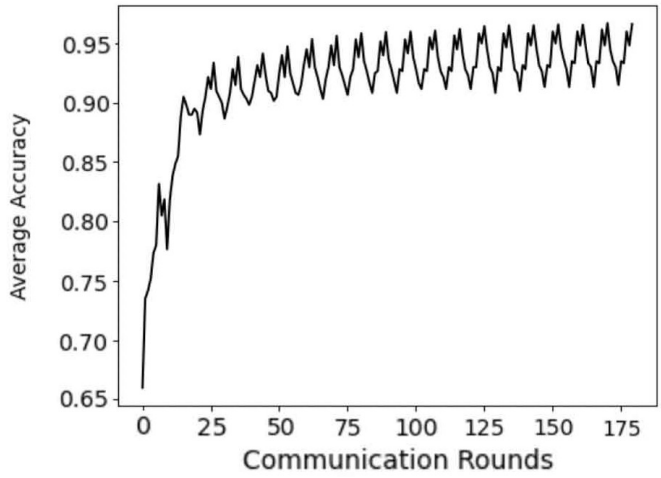

Figure 3.4: Average training accuracy vs. Communication rounds

Experimental settings and selection of experimental parameters are presented in Table 3.3. The fraction of users stands for the fraction of selected population for aggregation

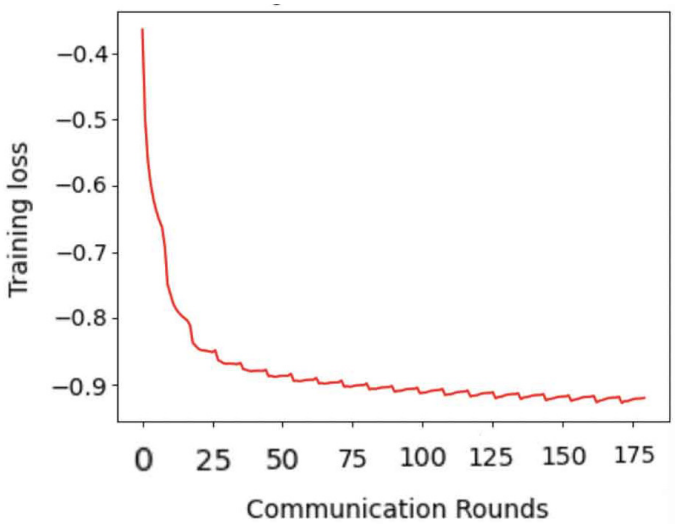

Figure 3.5: Training loss vs. Communication rounds

in each epoch. This is a constant value for each epoch. Batch size of local updates for each user is set at 0.1 whereas the number of local training epochs for each user is set at 10. The three comparison methods used to derive the reputation score are weighted based on their contribution, i.e., $w1, w2, w3$, and this study considers equal weights. A communication round denotes the aggregation of a local model to the global model. In each epoch, 100 local models are trained and 10% is used to aggregate with the global model. A communication round is stands for the time when a local model is sent to the global model for aggregation. The learning curve converges at 100 communication rounds which translates into 10 epochs. The training approach runs on a Ryzen 3900x and Nvidia 2080 Super. Figure 3.4 and 3.5 show the learning curves of the proposed model. Based on these, curves the model was trained up to 100 communication rounds since a clear convergence is shown in both figures.

Table 3.4 presents test accuracy under the three datasets. As seen in the table, the proposed method improves the original (baseline) federated learning methodology in all cases. For all IID datasets, there exists an incremental improvement whereas for the IID MNIST dataset, improvements of 1.733% and 4.301% are achieved under the MLP and CNN models, respectively. Under the Fashion MNIST IID data set the improvement is at the order of 3.392%. The IID datasets originally have equal distribution of data for each user, this causes each user to have a similarly trained local model. This is also why the original test accuracy levels for the IID datasets are much higher than the non-IID counterparts. However, even with each local model containing equal data distribution, the

proposed reputation-based local model selection methodology is still able to improve the final test accuracy by eliminating the poorly performing local models. When the shuffled data is distributed across all users under IID partitioning, users with redundant data still remain, and those users cannot contribute to improving the global model. For all non-IID-partitioned datasets, significant improvement can be achieved. Under the MNIST non-IID dataset, the test accuracy is improved by 6.238% and 9.306% for the MLP and CNN model, respectively while the Fashion-MNIST non-IID dataset experiences an accuracy improvement of 4.619% improvement with the implementation of the proposed reputation-aware local model selection methodology. The nature of the non-IID datasets causes the original test accuracy to be lower than that of the IDD datasets. Thus, since the data distribution is uneven, the trained local model of each user results in significant difference in the accuracy performance. It is worth to note that this aligns better with a real world scenario since recruited users often possess heterogeneous data due to their sensor reliability. More under-performing models are present in the non-IID scenario and this results in a larger increase in performance when the proposed federated learning methodology is applied. The non-IID MNIST and F-MNIST datasets both were improved to a test accuracy of 82.627% and 81.473%, respectively. It is possible that the MLP model reaches its learning limit on these datasets, however, the proposed methodology shows to have significantly improved the performance when compared to the original (baseline) methodology. This is further demonstrated under the CIFAR non-IID dataset at an improvement of 9.3% in terms of test accuracy. The unevenly distributed data leads each client to receive different number of data samples, and generate more models with poor quality, which leads to lower test accuracy. Thus, after eliminating the models of poor performance through the proposed scheme, the test accuracy is significantly improved. Due to the high complexity of the CIFAR data set, the original test accuracy is low, however, filtering out the low-quality local models result in improving test accuracy by 3.335% for MLP and 2.65% for CNN. The 95% confidence interval of proposed method is 0.0711.

3.7 Conclusion

In this chapter, we have developed a methodology that improves federated learning in a mobile environment by introducing a reputation-aware user selection scheme so to eliminate poor performing local model contributions. We focus on quantifying the performance of trained local models and filtering out unqualified ones so that the FL system can yield a high performance final global model and avoid needlessly expending computational energy. In our proposed methodology, calculation of each models' reputation score is based on com-

parison with dynamic metrics so as to evaluate models with gradual increasing standards. Experimental results have shown that the proposed reputation-aware federated learning methodology results in significant improvements in the accuracy of the aggregated model when compared to the original implementation of the federated learning-based solution. More specifically, the test accuracy can be improved by at least 1.733% and up to 9.306%.

Chapter 4

Reputation-enabled Federated Learning Model Aggregation in Mobile Platforms

4.1 Introduction

FL is still a developing field, current aggregation methodologies are primitive in the sense of considering all updated models are equal in weight, this does not guarantee the most optimized approach in terms of final model performance as well as training iterations required. We introduce a reputation-based aggregation method in FL mobile environment that aims at maximizing the final generated model's performance by assigning aggregation weights based on their dynamic performance during each communication round. The server aggregates the uploaded model parameters to generate a global model. Common practice for the uploaded local models is an evenly weighted aggregation, assuming that each node of the network contributes to advancing the global model equally. Due to the heterogeneous nature of the devices and collected data, it is inevitable to have variations between the contributions of the users to the global model. Therefore, users (i.e., devices) with higher contributions should be weighted higher during aggregation. With this in mind, this chapter proposes a reputation-enabled aggregation methodology that scales the aggregation weights of users by their reputation scores. Reputation score of a user is computed according to the performance metrics of their trained local models during each training round, therefore it can be a metric to evaluate the direct contributions of their trained local model.

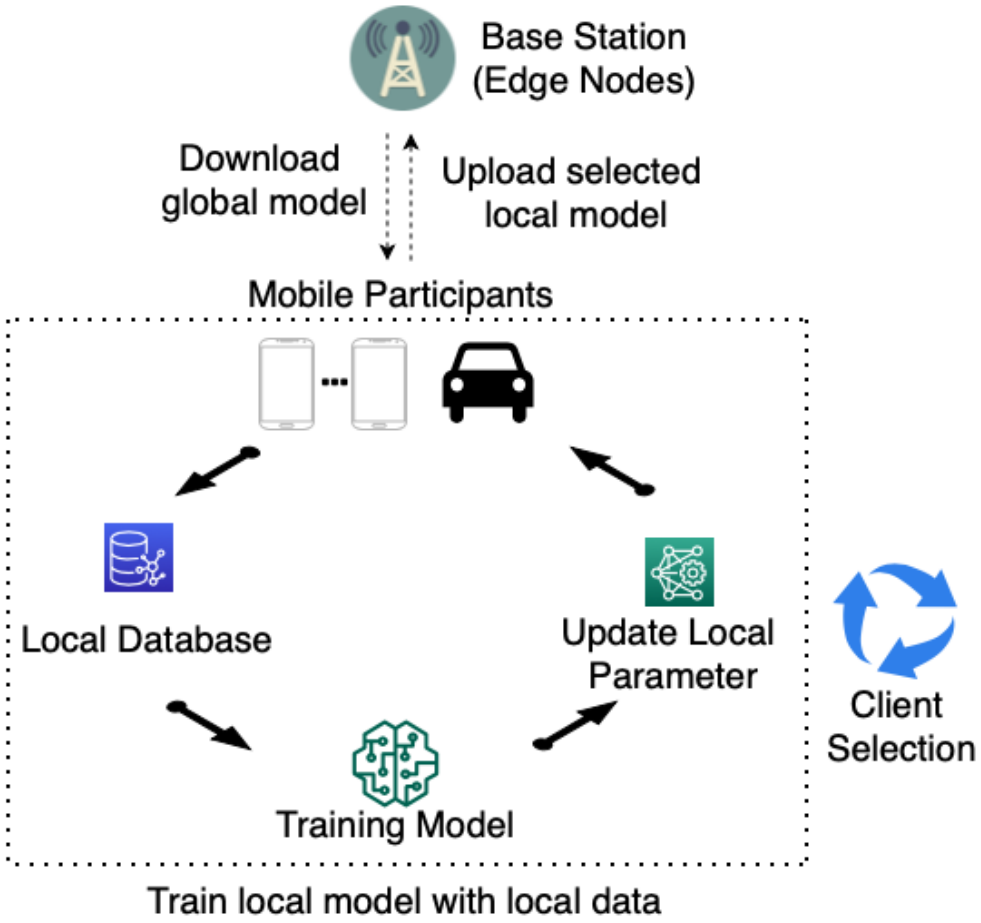

Figure 4.1: Local model training

4.2 Methodology

The goal of the proposed methodology is to improve fairness in the aggregation process to attain higher accuracy as well as to achieve a faster convergence speed in an FL network setting, which builds on a network of smart mobile devices.

4.2.1 Local model training process

As shown in Fig. 4.1, the base station initially assigns training tasks to each participating local device and the requirements of the training data (e.g., data type, size and resources).

The local clients (i.e., participants of the FL network) who meet the data requirement as well who are as willing to join the training task are recruited for training. The initial global model, hyper parameters and details of training process are given to the recruits. Each recruit uses their local mobile device and dataset to train a local model. Instead of transmitting its local data to the server, each participant respectively updates the weights of their local model. The goal of each client is to minimize the value of the loss function which means they need to find the optimal local weights. In a series of iterations and epochs, the local weights are continually transmitted by the participants. The base station sends parameters of the aggregated global model back to the local data owners. The steps of updating local weights and transmitting the updated global model keep repeating until an acceptable training and test accuracy is achieved. It is worth to note that SGD optimizer[13] is utilized in this work, and negative log-likelihood is chosen as the loss function.

4.2.2 Client Selection

We adopt the client selection algorithm from our previous work in [84], and describe below briefly.

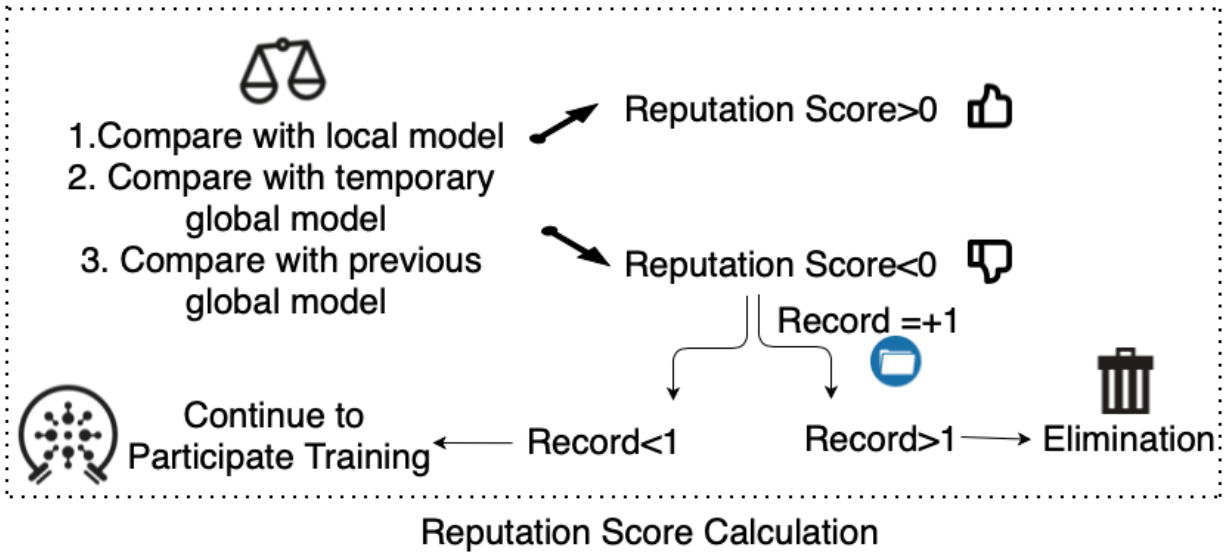

Figure 4.2: Reputation calculation process

Before local parameters are aggregated into the global model, only the participants who pass the client selection process are chosen to update their local weights. Reputation

score of user i (R_i) is calculated during this process to evaluate the performance of the local models or whether a particular local model should be selected to participate in the updating process by using Eq.4.1. In the equation A_i stands for the accuracy of the local model of user i whereas $A_{g_{temp}}$ and $A_{g_{old}}$ denote the accuracy of the temporary global model and the global model of the last communication round, respectively.

$$R_i = w_1(A_i - \sum_j A_j/n) + w_2(A_i - A_{g_{temp}}) + w_3(A_i - A_{g_{old}}) \tag{4.1}$$

As shown in Fig. 4.2, at the beginning of the client selection process, each participant has an initial reputation score. The reputation score consists of three metrics:

- (1) Comparison with average test accuracy of local models in the current round such that the local models that outperform the average will obtain positive scores whereas those perform worse will get negative scores.
- (2) Comparison with the temporary global model which is generated from further trained local models in this specific communication round. A negative contribution denotes the temporary global model's outperforming the local model. The purpose of this metric is to eliminate poor models before aggregating into global models as well as choosing the best performance models in each iteration.
- (3) Comparison with the global model of the last communication round. This metric will often result in a positive contribution due to the high possibility of further trained local model's better performance in comparison to the global model of last communication round.

Reputation score is used to prevent poor performing local models from participating in the aggregation process, in order to be accepted for aggregation, the reputation score has to be above a predetermined threshold, a record is also kept for the number of times the local model's reputation score is below the threshold. It is worth to note that the threshold for the reputation score is selected empirically. Each local model's record is set to an initial value, and the value increases every time when the reputation score is below the threshold. The local model is eliminated after it is declined for a certain number of times. By eliminating the poorly performing models, we can consistently filter out the participants that yield poor contribution while improving the model's performance as well as possibly decreasing the total running time as shown in the previous work [84].

4.2.3 Global model aggregation

As shown in Fig. 4.3, the server leverages the reputation scores that meet the selection requirement to calculate a normal distributionnormal distribution checked by utilizing QQ plot) that would then be used to assign aggregation weights. The normal distribution is calculated after each specific communication round to enable our proposed weighted aggregation methodology. The rationale for calculating a normal distribution is that its adding the least amount of prior knowledge to the model. Therefore, a normal distribution is built from the reputation scores of users that are accepted for aggregation. There are 5 regions that are present in a normal distribution they are defined as illustrated in Fig. 4.4.

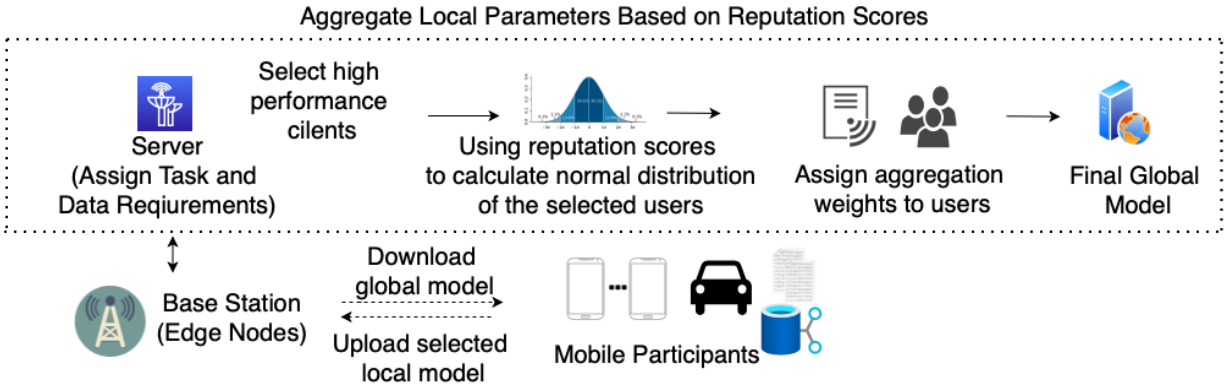

Figure 4.3: Minimalist illustration of global model aggregation

Aggregation weights are assigned to a user depending on where their reputation score is present on the normal distribution. In the standard FedAvg calculation for $N = \{1, 2, ..., n\}$ users, the weight for each user during the aggregation would be equally split, thus if we denote aggregation weight of each user as α, then $\alpha = 1/n$ which means each user has the same aggregation weight. During a specific aggregation round, if all users fall into the central region of $\mu \pm \sigma$ then our method would be equivalent to the FedAvg algorithm; hence the proposed method aims to address the scenarios where there exist users that have reputation scores outside of the $\mu \pm \sigma$ quadrant (i.e., R3) of the distribution curve.

The aggregation weight for each user within a specific region is denoted as ω_i. If the reputation score of user i falls into R3, we denote the aggregation weight for that specific user as $\omega_{\mu\pm\sigma}$. We scale the average aggregation weight assigned to each user by the coverage percentage of the distribution quadrant that their reputation score resides in. Thus, since the $\mu \pm \sigma$ region covers 0.682 of the distribution, we set $\omega_{\mu\pm\sigma}$=0.682α as the aggregation weight. We multiply the original averaged weight of each user α by the region coverage to

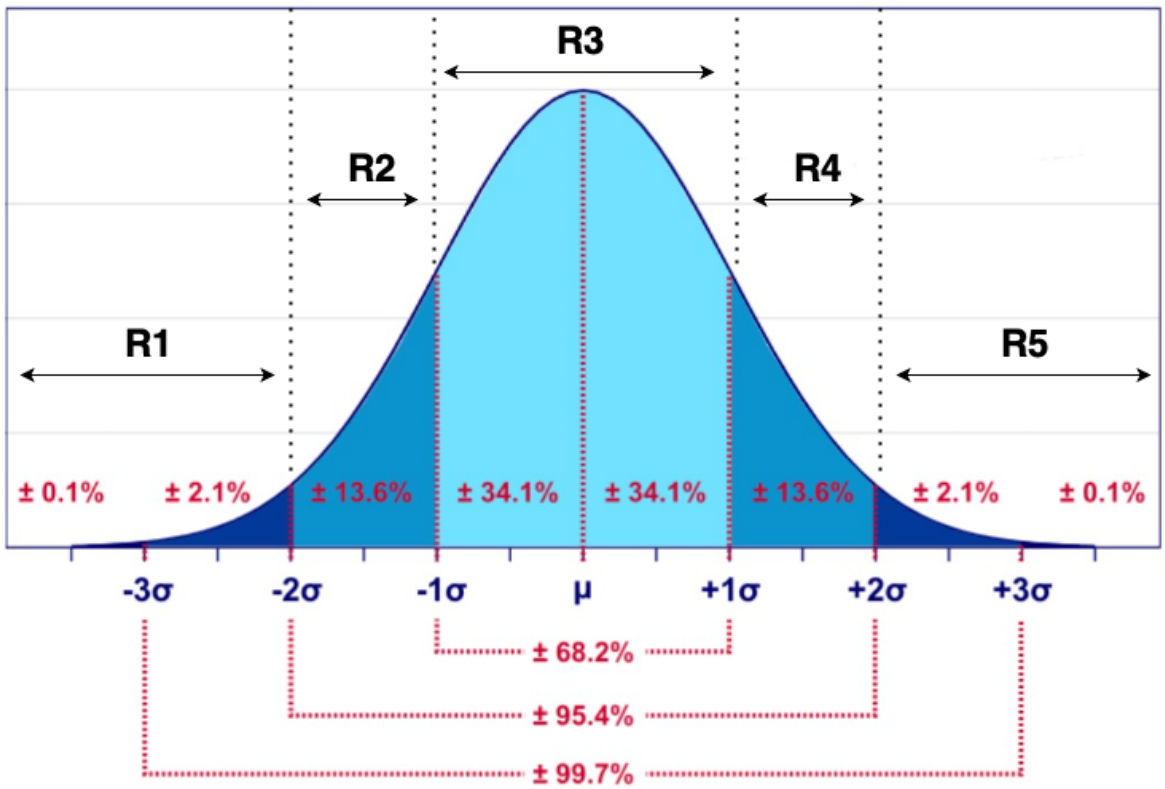

Figure 4.4: Five regions with respect to the reputation scores under the assumption that the scores follow a normal distribution.

assign new weights.

The R2 and R4 quadrants (i.e., $\mu \pm 2\sigma$) cover 27.2% of the distribution. Thus, the user would obtain 27.2% of the averaged aggregation weight, α. This value is then added/subtracted to the value obtained from R3 region aggregation weight depending on whether the reputation score resides in $[\omega_{\mu-2\sigma}, \omega_{\mu-\sigma}]$ or $[\omega_{\mu+\sigma}, \omega_{\mu+2\sigma}]$. As a result users with reputation in R2 will be assigned lower aggregation weights when compared to those in R3 as shown in Eq. 5.1 whereas the users in R4 will be assigned higher aggregation weights compared to those under R3. As seen in the same equation, the same weight assignment runs for the R1 and R5 regions given that these regions cover 4.2% of all reputation values.

$$\omega_i = \begin{cases} \omega_{\mu-3\sigma} = \omega_{\mu-2\sigma} - 0.042\alpha, & Rep_i \in \{R1\}. \\ \omega_{\mu-2\sigma} = \omega_{\mu\pm\sigma} - 0.272\alpha, & Rep_i \in \{R2\}. \\ \omega_{\mu\pm\sigma} = 0.682\alpha, & Rep_i \in \{R3\}. \\ \omega_{\mu+2\sigma} = \omega_{\mu\pm\sigma} + 0.272\alpha, & Rep_i \in \{R4\}. \\ \omega_{\mu+3\sigma} = \omega_{\mu+2\sigma} + 0.042\alpha, & Rep_i \in \{R5\}. \end{cases} \tag{4.2}$$

Each weight factor, that is calculated for a user (ω_i) is normalized with respect to the total of these calculated weights. The normalized weight assigned to a user is denoted by ω_i^*, where $\omega_i^* = \frac{\omega_i}{S}$ where $S = \sum_{i=1}^{n} \omega_i$. In the aggregation process, n users are selected each time. Each user has their model parameters that need to be aggregated as well as their assigned aggregation weight. Final aggregation is the weighted sum of all contributions by each user as formulated in (4.3):

$$P_{aggr} = \sum_{i=1}^{n} P_i \omega_i^* \tag{4.3}$$

In the equation above, P_{aggr} stands for the aggregated model parameters whereas the local model parameter of user i (out of the n users) is denoted by P_i. It is worth to note that in each communication round, the normal distribution of reputations is modeled upon the accumulation of user reputation scores continuously as communication rounds elapse. Thus, the instantaneous reputation of a user in a communication round is rolled onto the reputation of that user calculated in the previous communication round.

4.3 Simulation Settings

Our proposed methodology is evaluated with two machine learning models combined with the two datasets, details of the two model families are below:

- Multi layer-perceptron (MLP): each unit leverages ReLu activation and has 1-hidden layer with 64 units in total.
- Convolutional Neural Network (CNN): includes a fully connected layer of 320 units, two convolutional layers in which the first layer has ten channels and its second layer has twenty channels.

Due to collecting data from heterogeneous devices, the performance of the FL framework can be impacted. Two ways of distributing MNIST data-set is considered to address the problem as described below:

- Independent and Identically Distributed (IID): The shuffled data-set is evenly partitioned 100 times with each partition containing 600 image samples.
- non-IID: The data-set is divided into 1200 groups with 50 image samples in each group by the digital label of each sample. The image samples are assigned to each participant randomly and varying between 1 and 30.

4.4 Experimental Results

4.4.1 Performance under varying number of users

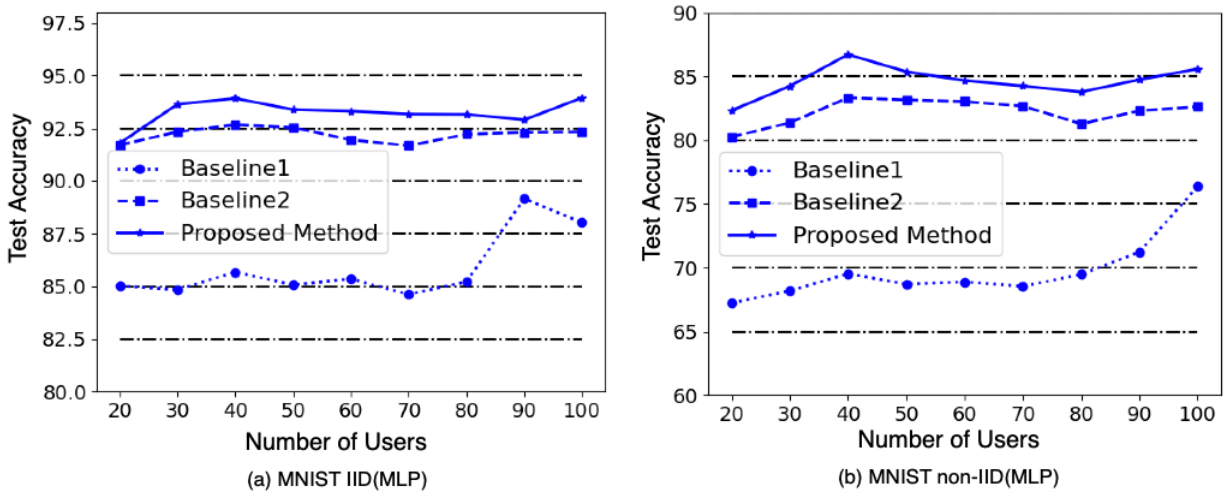

Figure 4.5: Average test accuracy comparison under varying number of users. MLP is used as the ML model.

Fig. 4.5 illustrates the performance of our proposed method under varying number of participating users. Our proposed method includes reputation-based client selection and reputation-scaled aggregation, while Baseline 2 is from the previous work in [84] that employs reputation-based client selection, and Baseline 1 builds upon the widely known FedAVG approach. We show that the addition of the reputation-scaled aggregation method

Table 4.1: Test accuracy improvement under different datasets with 100 users

Test Accuracy	MLP IID MNIST	MLP Non-IID MNIST	MLP IID F-MNIST	MLP Non-IID F-MNIST	CNN IID MNIST	CNN Non-IID MNIST
Baseline 1	88.023	76.389	89.44	76.854	96.517	81.075
Baseline 2	92.324	82.627	92.832	81.473	98.25	90.381
Proposed method	93.945	85.59	94.269	92.83	98.68	98.25

consistently improves the results obtained under Baseline 2. All of the clients that are selected for aggregation have positive contributions to the global model, which translate into their reputation scores. The reputation score is generated through the local model performance metrics, therefore it is possible to use it as a measure of the local model contribution. The performance increase is backed by the additional influence of better performing local models during the aggregation. If there are not any local models that significantly outperform the rest, the aggregation methodology is expected to coincide with the FedAvg algorithm. However, in the case of a heterogeneous environment, the proposed method in this paper ensures that users in this mobile FL network environment who contribute with local models that have higher computational capabilities and richer data are given higher reputation scores and weighted more during the aggregation.

4.4.2 Various combinations of ML models and datasets

Table 4.1 presents the performance results of our proposed model in comparison to the two baseline approaches. Test accuracy is used as the performance metric for comparison, each value represents an average of ten runs. Two ML models are used under the FL-based framework: MLP and CNN. For MLP models, we use MNIST and FMNIST datasets. For the MNIST IID dataset using MLP, our proposed model leads to an improvement of 5.92% and 1.62% in comparison to Baseline 1 and Baseline 2, respectively. Furthermore, the accuracy is further improved by 9.20% and 2.96% over Baseline 1 and Baseline 2, respectively under the non-IID MLP MNIST dataset. Higher improvements are seen in the non-IID datasets (as also observed in Fig. 4.5) by utilizing our proposed method since larger variances among the participants occur in a non-IID scenario, which is often the

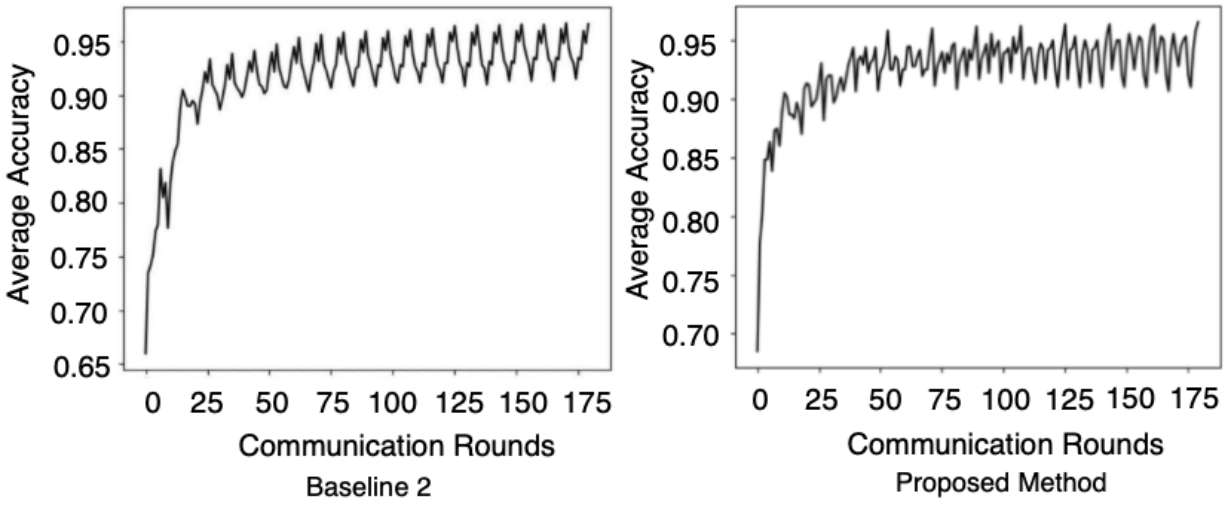

Figure 4.6: Comparison of convergence under MNIST IID with MLP as the ML model

case in Federated Learning. This trend is repeatedly shown in every non-IID versus IID partitioning scenarios of each dataset. A remarkable improvement is observed when MLP is used with the F-MNIST non-IID dataset. An improvement of 15.976% and 11.375% is achieved over Baseline 1 and Baseline 2, respectively.

For CNN based models, the MNIST dataset is used. Although the CNN-based model achieved a high accuracy of 96.517% and 98.25% for Baseline 1 and Baseline 2, respectively, our proposed methodology is still able to improve it further to 98.68% under the IID scenario. Baseline 1 is only able to achieve 81.075% accuracy under the non-IID MNIST dataset using the CNN model. However, our proposed methodology is able to improve the performance by 17.175% and boost the accuracy up to 98.25%, which is almost as high as the IID dataset's test accuracy. The 95% confidence interval for proposed method is 0.062.

Overall, our proposed FL approach shows a consistently improving trend over Baseline 1 and Baseline 2 models across different datasets and ML models. Larger improvements are seen under non-IID datasets where the contribution of each user is varied more due to their distribution of local data.

4.4.3 Convergence performance

Fig. 4.6 depicts the convergence rate for Baseline 2 and our proposed methodology for FL. Figure 4.6.a shows that the convergence of Baseline 2 is at approximately 100 communication rounds whereas the proposed methodology converges at around 40 communication

rounds. This confirms that by applying the weighted aggregation based on user contributions/reputations enables higher efficiency in addition to the improved accuracy metrics, also seen in Table 5.2. The FedAvg aggregation algorithm inhibits the contributions of the outperforming local models during aggregation and assigns the same weight to all users in the aggregation process. These two points slow down the convergence of the FL model. Thus, our proposed reputation-enabled aggregation methodology addresses these points by introducing a relative measure of contribution from the standpoint of continuously assessed reputation scores of the users based on local model performances, as well as a probability distribution-driven weight determination.

4.5 Conclusion

Federated Learning (FL) recruits users to train ML models on their own devices with their own local data, and the parameters of the local models are aggregated into a global model. In this chapter, we have proposed a new global model aggregation methodology to improve the efficiency and accuracy performance of the aggregated models in FL. To do so, the proposed model assigns different aggregation weights to the participating users, who are considered to be a part of a mobile network of distributed computing nodes, according their reputation standing within the FL network. The proposed method has been compared to the standard FedAvg algorithm and the previously proposed reputation-based client selection algorithm in terms of convergence and accuracy performance under various experimental settings. Our experimental results have shown improvements over these two baseline approaches, and particularly under the non-IID scenarios, up to more than 17% and 8% accuracy improvements have been achieved when compared to these two baseline approaches. Furthermore, the convergence speed has been reduced by approximately 60% when compared to the previously proposed reputation-based client selection scheme in an FL network.

Chapter 5

Aggregation of Incentivized Learning Models in Mobile Federated Learning Environments

5.1 Introduction

Training local models for FL is a computation intensive task that consumes the user's battery and takes up ram and CPU usage. If a user cannot obtain a favourable payout for their efforts then it is very likely that they would be reluctant to participate. Current state-of-the-art does not consider global model performance along with user utility in conjunction with awareness of the users reputation. This chapter presents a reputation and budget-constrained selection methodology along with an auction-driven incentive scheme for FL in mobile environments. Our aim is to guarantee that the user is paid more than their operating costs while achieving the highest performance results under constraint budgets. The reputation score is built on the performance metrics of the local models, and the incentive aims to ensure all participants are rewarded according to the quality of their contributions. With the dynamic adjustment of the user compensation to distribute the benefits more fairly, the proposed incentive increases user utility with the increasing platform budget. Numerical evaluations have shown that the proposed scheme can improve the participant utility, and do not compromise the platform utility nor the test accuracy of the global FL model.

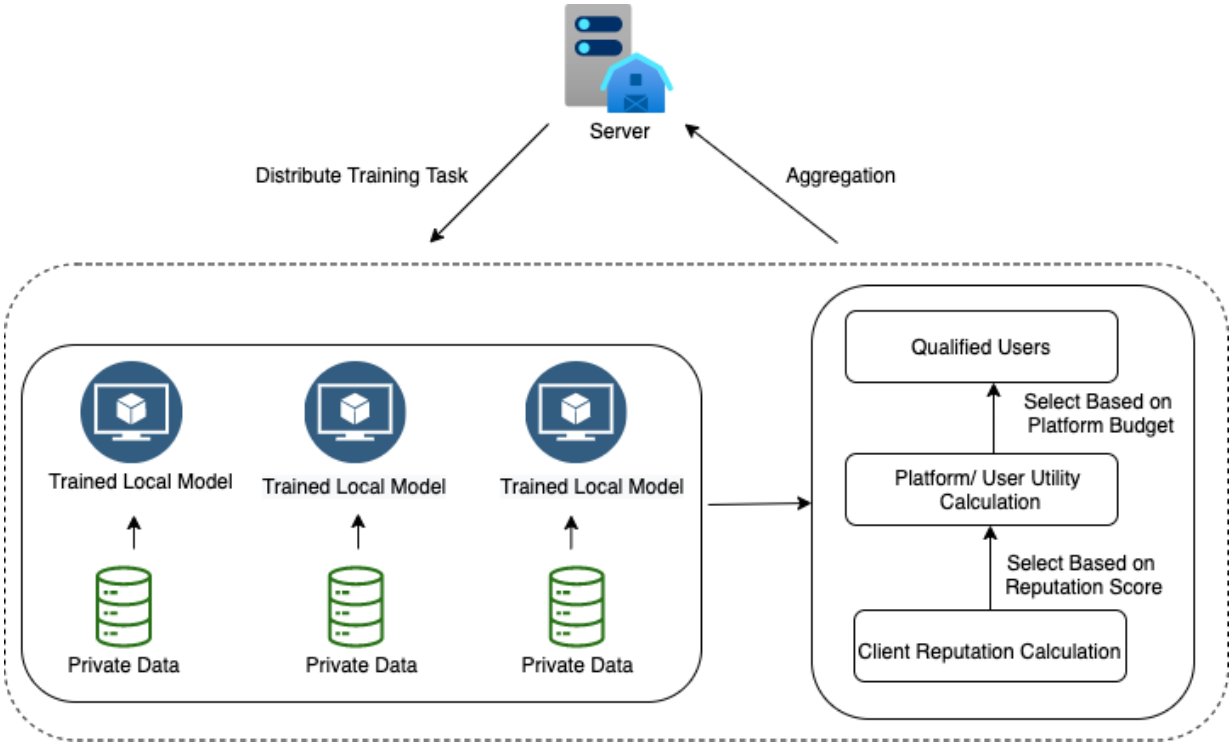

Figure 5.1: Overall FL process with incentives

5.2 System Model and Methodology

We incorporate federated learning and reputation scores to assign rewards to each participant, as shown in Fig.5.1. The goal is to improve participant (i.e., user) utility while maintaining the platform utility and aggregated model performance at the desired level. We adopt and adapt a previously proposed incentive mechanism, Trustworthy Sensing for Crowd Management (TSCM), that was originally tailored for a Mobile Crowdsensing (MCS) environment [30]. The incentive keeps track of the varying reputation of every participant and ensures that each selected user is guaranteed a minimum reward that is equal to their sensing cost, and when users are compared according to their output models, fair distribution of rewards is ensured. We assume a budget-limited FL environment, where only the client who contributes with a "worthy local model" can share the distributed reward. The definition of the "worthiness" is provided later in this section. To understand the budget-limited FL methodology, we focus on monitoring the client activity to assign fair payments while training the FL model.

5.2.1 Reputation Score

Evaluation of a local model performance and deciding whether a local model is good enough to be selected as a participant in the updating process is achieved setting $R_i = R_i^L + R_i^T + R_i^P$ where R_i^L, R_i^T and R_i^P are the outcome of three different comparisons with the accuracy of the local model, and these are formulated in Eq. 5.1, and elaborated below.

$$\begin{cases} R_i^L = w_1(A_i - \sum_j A_j/n). \\ R_i^T = w_2(A_i - A_{g_{temp}}). \\ R_i^P = w_3(A_i - A_{g_{old}}). \end{cases} \tag{5.1}$$

At the beginning of the client selection process, each participant has an initial reputation score R_i. The reputation score consists of three metrics:

- (1) Comparison of the test accuracy of user i (A_i) with the average test accuracy of local models in the current iteration is denoted as R_i^L. Therefore, the users that outperform the average receive a positive score, while those below the average receive a negative score.
- (2) Comparison of A_i with the temporary global model's accuracy $A_{g_{temp}}$ that is generated from the trained local models' aggregation in this specific iteration is denoted as R_i^T. A positive contribution means that the local models outperform the temporary global model.
- (3) Comparison of A_i with the global model accuracy $A_{g_{old}}$ of the last iteration is denoted as R_i^P. This would normally have a positive score because the global model from the last iteration improves with further training.

The purpose of the reputation score is to choose reliable local models to participate in the updating process. A predetermined threshold selected empirically is used to select the local models whose reputation score is above this threshold. Meanwhile, a negative reputation indicator count is maintained for every user i to store the number of times that user-i has a reputation score below the threshold. The users whose records exceed a certain threshold are rejected to participate in the training.

5.2.2 Modified Crowdsensing Incentive into FL

Our proposed incentive scheme builds on the previously proposed Trustworthy Sensing for Crowd Management (TSCM)[30, 62]. The rationale for TSCM is to include trustworthiness

and reputation scores by considering the recent performance. Modified TSCM ensures that the payment assigned to the user will be no less than their training cost. As shown in Eq. 5.2, ε_χ is denoted as a marginal contribution that the individual user brings according to their accuracy improvement over the rest of the users. For the reward of users' accuracy improvement, B_χ represents the training cost for the user at this communication round, χ represent the current users to be rewarded:

$$(\varepsilon_\chi - B_\chi)/R_\chi > (\varepsilon_{\chi+1} - B_{\chi+1})/R_{\chi+1} \tag{5.2}$$

Following upon client selection, the users who have been recruited in this communication round are temporarily removed from the participant set one by one to determine their payments. For instance, once user χ is chosen as a winner to be removed, a temporary winners set (Δ) is constructed out of the rest of the users with a positive reputation score at this iteration. The set Δ is used to select users that aid in determining the reward of participant χ. The user with maximum worthiness over the temporary winner set, denoted by χ_i, (see Eq. 5.3) is added to the temporary set to help determine the reward of user χ.

$$\chi_i = argmax_{i \in P' | \Delta}(\varepsilon_i(\Delta) - B_i/R_i) \tag{5.3}$$

Each member χ_i in the temporary winner set is searched to find the maximum possible reward for user χ by weighing the value and cost of the temporarily removed participant and every other participant in the temporary set Δ. To do so, the system ensures the following two conditions: 1) participant χ will be rewarded the maximum accuracy-to-training cost value over the temporary winners set. To do so, Eq. 5.4 is run iteratively as the temporary winners set Δ gets built. 2) Participant χ is rewarded the highest possible value as the temporary winners set Δ gets formed. To do so, Eq. 5.5 runs iteratively as the temporary winners set Δ gets built.

$$\theta = min\left\{A_\chi/B_\chi, A_{\chi i}/B_{\chi i}\right\} \tag{5.4}$$

$$P_\chi = max\left\{P_\chi, A_\chi/\theta\right\} \tag{5.5}$$

This gradually rebuilt circulation continues until it reaches either of the following two conditions: $B_{\chi_i}/R_{\chi_i} \geq \varepsilon_{\chi_i}$ or $\Delta = P'$, where R_{χ_i} stands for the reputation score of the temporary winners set member. At the end of these iterations, if the condition $B_{\chi_i}/R_{\chi_i} < \varepsilon_{\chi_i}$ is met, the payment assigned to participant χ is re-adjusted as: $P_\chi = max(P_\chi, \varepsilon_\chi(\Delta))$. It is worth noting that only the participants that are assigned rewards can contribute to the updating process of the global FL model.

5.3 Experimental Setup And Numerical Results

The proposed FL incentive mechanism is evaluated based on three metrics: i) the number of recruited users, ii) platform utility, and iii) total user utility. Platform utility($U_{platform}$) is formulated in Eq. 5.6 which denotes the total difference between the platform's gain and the payment awarded to the user in each communication round, τ. The platform's gain from participant i is calculated as the accuracy of the participant's local model scaled to the platform budget $\mathfrak{B}$. All symbols have a superscript τ to denote the communication round.

$$U_{platform} = \sum_{\tau}(A_i^{\tau} \cdot \mathfrak{B}^{\tau} - P_i^{\tau}) \tag{5.6}$$

User utility(U_{user}) denotes the sum of the differences between the reward received by the user and the training cost of the participant's local model as formulated in Eq. 5.7.

$$U_{user} = \sum_{\tau}(P_i^{\tau} - B_i^{\tau}) \tag{5.7}$$

In the simulations, we set the fraction of users f to 0.1, local batch size B = 10, local epochs E = 10, learning rate for each local model $\eta_i = 0.01$, weights of reputation scores $\omega_1 = \omega_2 = \omega_3 = 0.33$. The parameter settings are consistent with the previous work in [84] in order to have a fair comparison between them. The computation cost of users is uniformly assigned from 1 to 5 to comply with the TSCM scheme [62]. We consider two local training models: 1) A multi-layer-perceptron (MLP) which utilizes ReLu activation for each unit as well as one hidden layer included 64 units in total, 2) A Convolutional Neural Network (CNN), which contains 320 units, consists of two convolutional layers in total, in which the first layer includes ten channels whereas the second includes twenty channels. A well-known digital classification data-set called MNIST [**?**] is used, which has 60,000 data samples, 28×28 pixels (grey value) formed each image in MNIST. The data set is shuffled, and uniformly selected equal partitions of 100 groups (i.e., 60000 / 600) are formed, and each mobile device is assigned with one particular group. We deploy I = 100 distributed mobile participants that are willing to follow the server's set of given instructions. Their training costs (i.e., B_i) are uniformly initialized between 1 and 5 at the beginning of each epoch of the experiment. We mainly conduct experiments to inspect the impact of varying platform budgets. All simulation results represent average of ten rounds.

We study the impact of the proposed incentive compared to two baselines techniques under the same simulation settings. The two baseline techniques are explained as follows:

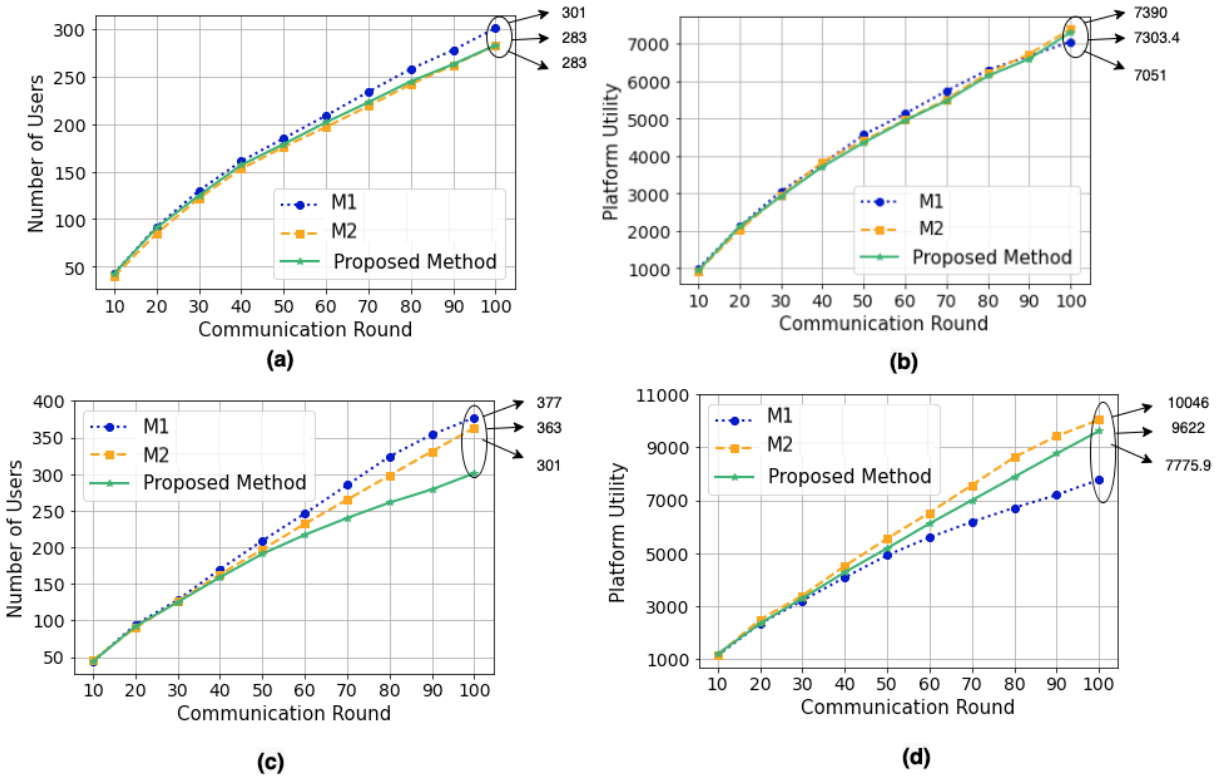

Figure 5.2: Comparison of the three recruitment schemes when platform budget = 30: (a) number of recruits, local model: MLP; (b) platform utility, local model: MLP; (c) number of recruits, local model: CNN (d) platform utility, model: CNN

- **M1 (Baseline1)** is from the previous work in [84] that leverages reputation score to select clients. M1 assumes that a unlimited budget and fulfills the cost of all users that pass the reputation test. The number of users selected would not be constrained by the platform due to insufficient resources to compensate the selected users.
- **M2 (Baseline2)** builds upon M1, however, it utilizes user selection based on budget constraints. The FL server does not select users beyond the initial budget. The users recruited are taken as a first come first serve basis. Here the users are still rewarded their reported participation cost.

Users are not assigned when the server does not have a sufficient budget to compensate each user during each communication round. As depicted in Fig.5.2(a) when each local model is an MLP, it is found that the number of users recruited by the proposed method

Table 5.1: Test accuracy under two datasets with 100 users

Test Accuracy	MLP IID MNIST	CNN IID MNIST
M 1	92.324	98.25
M 2	92.52	98.27
Proposed method	92.52	98.27

Table 5.2: Comparison of the three recruitment schemes with varying platform budget when the local models are MLPs

Budget ($\mathcal{B}$)⇒	10	20	30	40	50	10	20	30	40	50
Utilities ⇓	Local Model: MLP					Local Model: CNN				
M1 - Platform	908.2	3,921.6	7,051	9,129.4	12,801.	71,174.5	4,651.4	7,775.9	12,030.6	15,890.3
M1 - User	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
M2 - Platform	957.5	4,187.7	7,390	9,842.3	13,533.	42,407	5,179.3	10,046	13,179.9	16,955.1
M2 - User	0	0	0	0	0	0	0	0	0	0
Proposed method - Platform	954.7	4,069.3	17,303.4	9,560.2	13,199.	31352.7	5060.6	9622	13027.4	16514.6
Proposed method - User	46.89	108.11	140.32	153.1	241.7	40	128.6	203.5	268.8	284.92

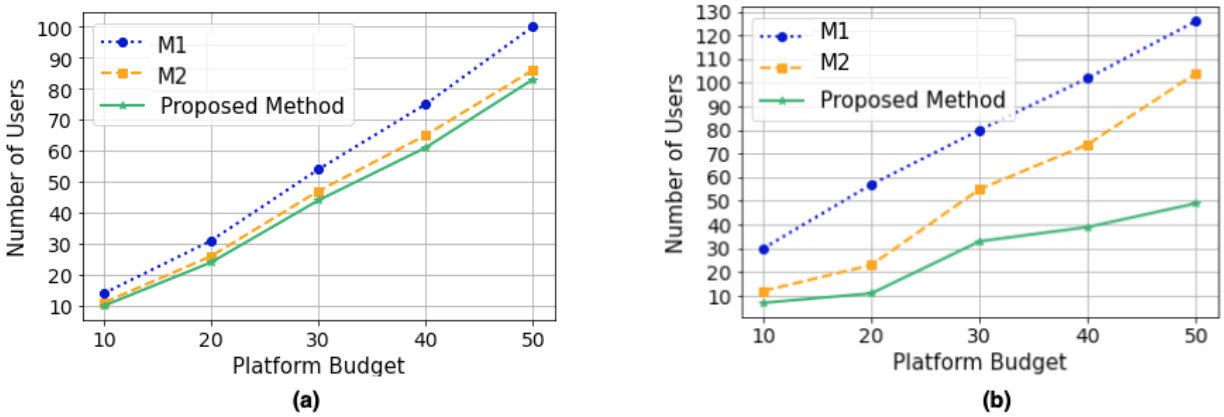

Figure 5.3: Number of users under the three recruitment schemes with varying platform budget at 10th epoch: (a) MLP; (b) CNN

as well as M2 are slightly lower than the M1. This is due to the fact that the M2 and the proposed method both have double selection criteria. The double selection eliminates more users due to the additional factor of having a limited budget. This impacts the system the most at lower budgets. The training accuracy is 96.58%, and the test accuracy is 92.52% for the proposed method compared to the training accuracy of 95.46% and the test accuracy of 92.32% for M1. This proves that double selection is more viable than selection with only a reputation score since the accuracy remains the same, but less rewards are sent to the users. M1 assumes that all users selected by their reputation score are recruited and paid according to their cost. The proposed incentive limits the budget and ends up recruiting fewer users but maintaining the overall global model accuracy even with a slight improvement.

Fig.5.2(b) shows the changes of platform utility over each epoch. M2 results in the highest platform utility due to the double selection criteria, selecting higher quality users, resulting in fewer users recruited. Even though fewer users are selected, a higher average platform utility is achieved since high local models are recruited. The proposed method has a slightly lower platform utility than M2 because the server rewards the clients, resulting in a non-zero user utility while maintaining platform utility. Fig.5.2 (c) and (d) present the results under local models of CNNs. The difference between the baselines and the proposed incentive is further remarkable when compared to (a) and (b) after 50 communication rounds since the accuracy of the CNN local model(98.27%) is higher than the MLP local model. At around 50 communication round, the CNN model starts to converge, the threshold to select users is raised rapidly, and more rewards are split to users based on the

accuracy improvement, which results in a significant difference in the number of users at 100 communication rounds when comparing the proposed incentive and the two baselines. Even though the proposed method recruits fewer users to participate in aggregation, it still achieves higher platform utility than M1. Although the proposed method splits the platform utility to reward users, it still maintains almost the same platform utility as M2. Moreover, the accuracy of the global FL model under M1 and M2 is 98.25% in Table5.1, which is even slightly lower than that of the FL model under the proposed incentive.

Table 5.2 presents the platform utility and user utility under varying platform budgets when the local model of each participant is an MLP. The platform utility under M2 is always higher than that under the proposed incentive. This is due to the fact that the proposed incentive aims to reward the participants by aiming at non-zero user utilities. On the other hand, the platform utility under the proposed incentive is always higher than M1's platform utility under varying platform budgets. Table 5.1 presents the platform utility of the two baselines and the proposed incentive when the local models are CNNs. The results show the same trend as Table 5.2.

Lastly, we evaluate the number of recruited participants (i.e., users) under varying budgets of the two recruitment baselines and the proposed incentive. Fig. 5.3(a) shows the number of recruited users at the 10th epoch under different budgets for the three schemes when the local participant models are MLPs. M2 and the proposed incentive employ a lower number of users. This is because these two schemes do not recruit qualified participants if the platform budget is not sufficient. This proves that our incentive mechanism works well under different sizes of participant pools or scale.In Fig. 5.3(b), when the local models are CNNs, as the platform budget increases, the proposed incentive recruits the least number of users with the added constraints. The rate of increases in the number of users is also less for the proposed method under the local CNN models. This because it prioritizes recruiting high-quality models over quantity of data. Due to no limitation in the platform budget under M1, the number of recruited users is consistently the highest under that baseline recruitment scheme.

5.4 Conclusions

In this chapter, we have proposed a new reputation score-based incentive scheme for global Federated Learning (FL) model aggregation. Data owners with higher quality data and high-performance local models are assigned a higher rewards generated by the central FL platform. Our proposed incentive adopts a previously proposed reverse auction procedure to dynamically adjust the users' compensation so to distribute the benefits more fairly.

Numerical evaluations have shown that the proposed scheme can improve the participant utility, and do not compromise the platform utility nor the test accuracy of the global FL model.

Chapter 6

Conclusion

Artificial intelligence applications are becoming at the forefront of technological advancements. Training complex models require the use of quality data. The advancements in mobile devices make it a hub for collecting and sensing various data. Mobile devices are equipped with an array of sensors and have the increasing processing power. However, leveraging these rich data silos to train machine learning models may infringe upon the ever stringent privacy laws. Federated Learning emerged as a privacy-preserving solution to utilize data from users without transmitting the raw data itself. The users would use the internal microprocessors within their mobile devices to train a machine learning model with local data and upload the model parameters instead of the data itself. After exploring the state-of-the-art, we have improved the Federated Learning scheme on three critical areas to improve the efficiency of model training through user selection and model aggregation, as well as incentivizing user participation through fair payment schemes.

We proposed a reputation-aware client selection scheme for mobile environments. Federated learning is subjected to data heterogeneity as well as device heterogeneity. Therefore user selection is the foremost area that was tackled. Selecting users with valid contributions versus malicious users allowed us to improve training efficiency and raise model performance. A reputation score was generated based on three hand-crafted performance metrics. The user would be able to admit into the scheme based on their reputation score and would be eliminated if it is constantly underperforming. That specific user may have quality data. However, it may not mesh with the needs of the given task. This scheme was proven to improve the test accuracy of the final global model varying from 1.73% to 9.30%. They were achieving higher gains on Non-IID datasets due to a greater difference in model performance from user to user.

We also proposed a reputation-enabled Federated Learning model aggregation scheme. To improve model performance further, we tackled one of the most integral parts of a federated learning scheme; the aggregation of the collected local models. The standard aggregation technique across federated learning schemes is to use the FedAvg algorithm that has an equal aggregation weight for all submitted models. We utilized the reputation score that was generated from our first work and developed an aggregation methodology that leverages Gaussian distribution to assign aggregation weights depending on contribution and model quality. This resulted in a reduction of 60% for the time required for model convergence of the global model as well as accuracy improvements of 17% on Non-IID datasets.

Lastly, we proposed the aggregation of incentivized learning models in mobile Federated Learning Environments. An efficient and optimized scheme in terms of accuracy was proposed with our first two works; however, incentivizing user participation is vital for a successful federated learning scheme. A federated learning scheme cannot function without user participation. Our proposed incentive scheme is a reverse auction focused on fair payment to users, ensuring that platform utility remains the same while providing positive user utility for the participants. We demonstrated that under platform budget constraints, our proposed scheme was able to maintain performance accuracy while constantly providing positive user utility.

6.1 Future Directions

Future directions are given as below;

- In traditional distributed methods, the transmission environment is always assumed that there is no packet loss, the transmission rate is high and stable. However, this assumption is not suitable for heterogeneous mobile devices and wireless networks in the federated learning environment, such as upload speed is usually slower than download speed, the unstable connection with wireless channels may cause some users to drop out during the training process. Therefore, the possible approach to improve the upload speed is considering compressing the size of local model updates during each iteration from users sent to parameter server so as to guarantee only optimized and specific updates are able to be sent to the server. This is one area that we have yet to tackle; in each of our contributions, we assumed a stable transmission environment without any limitations. The networking aspects of FL should be explored in

conjunction with defense strategies to further incentivize user participation through guaranteeing data security.

- Even with the existing resource management approach, scalability is still an issue, in which mobile devices that have long propagation distances from the central server should be dropped in order to increase the signal-to-noise ratio. Thus, our ongoing work involves addressing the issue of signal distortion that can cause reductions in accuracy as well as enabling the federated learning framework to be applicable at a larger scale. User participation is a crucial aspect of any FL environment, the distance between user and server or between user to user could all impact the performance of an FL system. Therefore exploring the impact of these implications would help us develop more efficient recruiting schemes.

- It is very necessary to address the limitation of resource management as to future work. Two constraints are 1) utilizing communication and computation resources efficiently at FL edge, and 2) distinct delay in convergence speed of unbalanced, non-identical, and non-independent datasets. Thus, our future work to address resource limitation involves 1) varying the frequency of aggregation to guarantee the desired final model's performance under available resources efficiently and 2) assigning new received updates with varied weights according to which epoch and communication round it belong to. These additions would be beneficial as extensions of our aggregation methodology.

- As an extension to our user recruitment scheme, we can focus on improving the security of our overall FL scheme. Currently, our recruitment scheme already has properties that can help combat malicious users. How

References

[1] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[2] Mohammad Mohammadi Amiri and Deniz Gündüz. Federated learning over wireless fading channels. *IEEE Transactions on Wireless Communications*, 19(5):3546–3557, 2020.

[3] Tran The Anh, Nguyen Cong Luong, Dusit Niyato, Dong In Kim, and Li-Chun Wang. Efficient training management for mobile crowd-machine learning: A deep reinforcement learning approach. *IEEE Wireless Communications Letters*, 8(5):1345–1348, 2019.

[4] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020.

[5] Xianglin Bao, Cheng Su, Yan Xiong, Wenchao Huang, and Yifei Hu. Flchain: A blockchain for auditable federated learning with trust and incentive. In *2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*, pages 151–159. IEEE, 2019.

[6] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design, 2019.

[7] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H Vincent Poor, and Shuguang Cui. Performance optimization of federated learning over wireless net-

works. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.

[8] Y. Chen, X. Sun, and Y. Jin. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10):4229–4238, 2020.

[9] Hatim Chergui and Christos Verikoukis. Offline sla-constrained deep learning for 5G networks reliable and dynamic end-to-end slicing. *IEEE Journal on Selected Areas in Communications*, 38(2):350–360, 2019.

[10] Mingshu Cong, Han Yu, Xi Weng, Jiabao Qu, Yang Liu, and Siu Ming Yiu. A vcg-based fair incentive mechanism for federated learning. *arXiv preprint arXiv:2008.06680*, 2020.

[11] Adriana Deac et al. Regulation (eu) 2016/679 of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and the free movement of these data. *Perspectives of Law and Public Administration*, 7(2):151–156, 2018.

[12] Ningning Ding, Zhixuan Fang, and Jianwei Huang. Incentive mechanism design for federated learning with multi-dimensional private information. In *18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, pages 1–8. IEEE, 2020.

[13] Jarek Duda. Sgd momentum optimizer with step estimation by online parabola model. *arXiv preprint arXiv:1907.07063*, 2019.

[14] Ahmed Roushdy Elkordy and A Salman Avestimehr. Secure aggregation with heterogeneous quantization in federated learning. *arXiv preprint arXiv:2009.14388*, 2020.

[15] Shaohan Feng, Dusit Niyato, Ping Wang, Dong In Kim, and Ying-Chang Liang. Joint service pricing and cooperative relay communication for federated learning. In *Intl. Conf. on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data*, pages 815–820, 2019.

[16] H. Guo, A. Liu, and V. K. N. Lau. Analog gradient aggregation for federated learning over wireless networks: Customized design and convergence analysis. *IEEE Internet of Things Journal*, pages 1–1, 2020.

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

[18] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. Mobile edge computing—a key technology towards 5G. *ETSI white paper*, 11(11):1–16, 2015.

[19] Tiansheng Huang, Weiwei Lin, Keqin Li, and Albert Y Zomaya. Stochastic client selection for federated learning with volatile clients. *arXiv preprint arXiv:2011.08756*, 2020.

[20] Tiansheng Huang, Weiwei Lin, Wentai Wu, Ligang He, Keqin Li, and Albert Zomaya. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Transactions on Parallel and Distributed Systems*, 2020.

[21] Ahmed Imteaj and M Hadi Amini. Fedar: Activity and resource-aware federated learning model for distributed mobile robots. *arXiv preprint arXiv:2101.03705*, 2021.

[22] L. G. Jaimes, A. Chakeri, J. Lopez, and A. Raij. A cooperative incentive mechanism for recurrent crowd sensing. In *SoutheastCon 2015*, pages 1–5, 2015.

[23] Jiajun Sun. An incentive scheme based on heterogeneous belief values for crowd sensing in mobile social networks. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 1717–1722, 2013.

[24] Ji Chu Jiang, Burak Kantarci, Sema Oktug, and Tolga Soyata. Federated learning in smart city sensing: Challenges and opportunities. *Sensors*, 20(21):6230, 2020.

[25] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov. Enabling privacy-preserving incentives for mobile crowd sensing systems. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pages 344–353, 2016.

[26] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6):10700–10714, 2019.

[27] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6):10700–10714, 2019.

[28] Jiawen Kang, Zehui Xiong, Dusit Niyato, Han Yu, Ying-Chang Liang, and Dong In Kim. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, pages 1–5. IEEE, 2019.

[29] Jiawen Kang, Zehui Xiong, Dusit Niyato, Yuze Zou, Yang Zhang, and Mohsen Guizani. Reliable federated learning for mobile networks. *IEEE Wireless Communications*, 27(2):72–80, 2020.

[30] B. Kantarci and H. T. Mouftah. Trustworthy sensing for public safety in cloud-centric internet of things. *IEEE Internet of Things Journal*, 1(4):360–368, Aug. 2014.

[31] Latif U Khan, Shashi Raj Pandey, Nguyen H Tran, Walid Saad, Zhu Han, Minh NH Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine*, 58(10):88–93, 2020.

[32] Latif U Khan, Nguyen H Tran, Shashi Raj Pandey, Walid Saad, Zhu Han, Minh NH Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. *arXiv preprint arXiv:1911.05642*, 2019.

[33] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. On-device federated learning via blockchain and its latency analysis. *arXiv preprint arXiv:1808.03949*, 2018.

[34] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. 2016.

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

[36] Tra Huong Thi Le, Nguyen H Tran, Yan Kyaw Tun, Minh NH Nguyen, Shashi Raj Pandey, Zhu Han, and Choong Seon Hong. An incentive mechanism for federated learning in wireless cellular network: An auction approach. *arXiv preprint arXiv:2009.10269*, 2020.

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

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

[39] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[40] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[41] Xiaoyang Li, Guangxu Zhu, Yi Gong, and Kaibin Huang. Wirelessly powered data aggregation for iot via over-the-air function computation: Beamforming and power control. *IEEE Transactions on Wireless Communications*, 18(7):3437–3452, 2019.

[42] Ziyuan Li, Jian Liu, Jialu Hao, Huimei Wang, and Ming Xian. Crowdsfl: a secure crowd computing framework based on blockchain and federated learning. *Electronics*, 9(5):773, 2020.

[43] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2020.

[44] Wei Yang Bryan Lim, Zehui Xiong, Jiawen Kang, Dusit Niyato, Cyril S Leung, Chunyan Miao, and Sherman Shen. When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries. *IEEE Transactions on Industrial Informatics*, 2020.

[45] Wei Yang Bryan Lim, Zehui Xiong, Chunyan Miao, Dusit Niyato, Qiang Yang, Cyril Leung, and H Vincent Poor. Hierarchical incentive mechanism design for federated machine learning in mobile networks. *IEEE Internet of Things Journal*, 7(10):9575–9588, 2020.

[46] Y. Liu, X. Xu, J. Pan, J. Zhang, and G. Zhao. A truthful auction mechanism for mobile crowd sensing with budget constraint. *IEEE Access*, 7:43933–43947, 2019.

[47] Yi Liu, Jialiang Peng, Jiawen Kang, Abdullah M Iliyasu, Dusit Niyato, and Ahmed A Abd El-Latif. A secure federated learning framework for 5G networks. *IEEE Wireless Communications*, 27(4):24–31, 2020.

[48] Yuan Liu, Zhengpeng Ai, Shuai Sun, Shuangfeng Zhang, Zelei Liu, and Han Yu. Fedcoin: A peer-to-peer payment system for federated learning. In *Federated Learning*, pages 125–138. Springer, 2020.

[49] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Transactions on Industrial Informatics*, 2019.

[50] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Transactions on Vehicular Technology*, 69(4):4298–4311, 2020.

[51] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. Communication-efficient federated learning and permissioned blockchain for digital twin edge networks. *IEEE Internet of Things Journal*, 2020.

[52] Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. In *Federated Learning*, pages 189–204. Springer, 2020.

[53] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Stat., PMLR*, pages 1273–1282, 2017.

[54] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. pages 1273–1282, 2017.

[55] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[56] Huy T Nguyen, Nguyen Cong Luong, Jun Zhao, Chau Yuen, and Dusit Niyato. Resource allocation in mobility-aware federated learning networks: A deep reinforcement learning approach. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 2020.

[57] Solmaz Niknam, Harpreet S Dhillon, and Jeffrey H Reed. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, 58(6):46–51, 2020.

[58] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*, pages 1–8, 2018.

[59] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[60] Shashi Raj Pandey, Nguyen H Tran, Mehdi Bennis, Yan Kyaw Tun, Zhu Han, and Choong Seon Hong. Incentivize to build: A crowdsourcing framework for federated learning. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[61] Shashi Raj Pandey, Nguyen H Tran, Mehdi Bennis, Yan Kyaw Tun, Aunas Manzoor, and Choong Seon Hong. A crowdsourcing framework for on-device federated learning. *IEEE Transactions on Wireless Communications*, 19(5):3241–3256, 2020.

[62] Maryam Pouryazdan, Burak Kantarci, Tolga Soyata, Luca Foschini, and Houbing Song. Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing. *IEEE Access*, 5:1382–1397, 2017.

[63] Youyang Qu, Shiva Raj Pokhrel, Sahil Garg, Longxiang Gao, and Yong Xiang. A blockchained federated learning framework for cognitive computing in industry 4.0 networks. *IEEE Transactions on Industrial Informatics*, 2020.

[64] Mohamed Abdur Rahman, M Shamim Hossain, Mohammad Saiful Islam, Nabil A Alrajeh, and Ghulam Muhammad. Secure and provenance enhanced internet of health things framework: A blockchain managed federated learning approach. *Ieee Access*, 8:205071–205087, 2020.

[65] Jianji Ren, Haichao Wang, Tingting Hou, Shuai Zheng, and Chaosheng Tang. Federated learning-based computation offloading optimization in edge computing- supported internet of things. *IEEE Access*, 7:69194–69201, 2019.

[66] Adam Richardson, Aris Filos-Ratsikas, and Boi Faltings. Budget-bounded incentives for federated learning. In *Federated Learning*, pages 176–188. Springer, 2020.

[67] Palash Roy, Sujan Sarker, Md Abdur Razzaque, Md Mamun-or Rashid, Mohmmad Mehedi Hassan, and Giancarlo Fortino. Distributed task allocation in mobile device cloud exploiting federated learning and subjective logic. *Journal of Systems Architecture*, 113:101972, 2021.

[68] Sumudu Samarakoon, Mehdi Bennis, Walid Saad, and M&39;erouane Debbah. Distributed federated learning for ultra-reliable low-latency vehicular communications. *IEEE Transactions on Communications*, 2019.

[69] Yunus Sarikaya and Ozgur Ercetin. Motivating workers in federated learning: A stackelberg game perspective. *IEEE Networking Letters*, 2(1):23–27, 2019.

[70] Jordi Serra, Luis Sanabria-Russo, David Pubill, and Christos Verikoukis. Scalable and flexible iot data analytics: when machine learning meets sdn and virtualization. In *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6. IEEE, 2018.

[71] Nir Shlezinger, Mingzhe Chen, Yonina C Eldar, H Vincent Poor, and Shuguang Cui. Uveqfed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing*, 2020.

[72] Abhishek Singh, Praneeth Vepakomma, Otkrist Gupta, and Ramesh Raskar. Detailed comparison of communication efficiency of split learning and federated learning. *arXiv preprint arXiv:1909.09145*, 2019.

[73] Jinhyun So, Başak Güler, and A Salman Avestimehr. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. *IEEE Journal on Selected Areas in Information Theory*, 2021.

[74] Michael R Sprague, Amir Jalalirad, Marco Scavuzzo, Catalin Capota, Moritz Neun, Lyman Do, and Michael Kopp. Asynchronous federated learning for geospatial applications. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 21–28. Springer, 2018.

[75] Lichao Sun, Jianwei Qian, Xun Chen, and Philip S Yu. Ldp-fl: Practical private aggregation in federated learning with local differential privacy. *arXiv preprint arXiv:2007.15789*, 2020.

[76] Wen Sun, Shiyu Lei, Lu Wang, Zhiqiang Liu, and Yan Zhang. Adaptive federated learning and digital twin for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 2020.

[77] Yuxuan Sun, Sheng Zhou, and Deniz Gündüz. Energy-aware analog aggregation for federated learning with redundant data. In *IEEE Intl. Conf. on Communications (ICC)*, pages 1–7, 2020.

[78] N. Thepvilojanapong, K. Zhang, T. Tsujimori, Y. Ohta, Y. Zhao, and Y. Tobe. Participation-aware incentive for active crowd sensing. In *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, pages 2127–2134, 2013.

[79] Kentaroh Toyoda and Allan N Zhang. Mechanism design for an incentive-aware blockchain-enabled federated learning platform. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 395–403. IEEE, 2019.

[80] Muhammad Habib ur Rehman, Khaled Salah, Ernesto Damiani, and Davor Svetinovic. Towards blockchain-based reputation-aware federated learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 183–188. IEEE, 2020.

[81] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

[82] Xiaofei Wang, Yiwen Han, Chenyang Wang, Qiyang Zhao, Xu Chen, and Min Chen. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network*, 33(5):156–165, 2019.

[83] Yuntao Wang, Zhou Su, Ning Zhang, and Abderrahim Benslimane. Learning in the air: secure federated learning for uav-assisted crowdsensing. *IEEE Transactions on Network Science and Engineering*, 2020.

[84] Yuwei Wang and Burak Kantarci. A novel reputation-aware client selection scheme for federated learning within mobile environments. In *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6. IEEE, 2020.

[85] Yuwei Wang and Burak Kantarci. A novel reputation-aware client selection scheme for federated learning within mobile environments. In *2021 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2021.

[86] Yuwei Wang and Burak Kantarci. Reputation-enabled federated learning model aggregation in mobile platforms. In *IEEE Intl. Conf. on Communications (ICC)*, pages 1–6, 2021.

[87] L. Wei, J. Wu, C. Long, and B. Li. On designing context-aware trust model and service delegation for social internet of things. *IEEE Internet of Things Journal*, pages 1–1, 2020.

[88] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in neural information processing systems*, pages 1509–1519, 2017.

[89] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen. Quality-driven auction-based incentive mechanism for mobile crowd sensing. *IEEE Transactions on Vehicular Technology*, 64(9):4203–4214, 2015.

[90] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[91] S. Xu, X. Chen, X. Pi, C. Joe-Wong, P. Zhang, and H. Y. Noh. ilocus: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing. *IEEE Transactions on Mobile Computing*, 19(8):1831–1847, 2020.

[92] Xinyi Xu and Lingjuan Lyu. Towards building a robust and fair federated learning system. *arXiv preprint arXiv:2011.10464*, 2020.

[93] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones. *IEEE/ACM Transactions on Networking*, 24(3):1732–1744, 2016.

[94] Kai Yang, Tao Jiang, Yuanming Shi, and Zhi Ding. Federated learning via over-the-air computation. *IEEE Transactions on Wireless Communications*, 19(3):2022–2035, 2020.

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

[96] Xin Yao, Tianchi Huang, Rui-Xiao Zhang, Ruiyu Li, and Lifeng Sun. Federated learning with unbiased gradient aggregation and controllable meta updating. *arXiv preprint arXiv:1910.08234*, 2019.

[97] Naoya Yoshida, Takayuki Nishio, Masahiro Morikura, Koji Yamamoto, and Ryo Yonetani. Hybrid-fl: Cooperative learning mechanism using non-iid data in wireless networks. *arXiv preprint arXiv:1905.07210*, 2019.

[98] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 393–399, 2020.

[99] Rongfei Zeng, Shixun Zhang, Jiaqi Wang, and Xiaowen Chu. Fmore: An incentive scheme of multi-dimensional auction for federated learning in mec. *arXiv preprint arXiv:2002.09699*, 2020.

[100] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7):6360–6368, 2020.

[101] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7):6360–6368, 2020.

[102] Weishan Zhang, Qinghua Lu, Qiuyu Yu, Zhaotong Li, Yue Liu, Sin Kit Lo, Shiping Chen, Xiwei Xu, and Liming Zhu. Blockchain-based federated learning for device failure detection in industrial iot. *IEEE Internet of Things Journal*, 2020.

[103] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, and Dusit Niyato. Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system. *arXiv preprint arXiv:1906.10893*, 2019.

[104] Guangxu Zhu, Yong Wang, and Kaibin Huang. Low-latency broadband analog aggregation for federated edge learning. *arXiv preprint arXiv:1812.11494*, 2018.

[105] Mingqiang Zhu, Liu Chang, Nan Wang, and Ilsun You. A smart collaborative routing protocol for delay sensitive applications in industrial iot. *IEEE Access*, 8:20413–20427, 2020.

Appendix A

Reputation weights

There are three parameters within the reputation formulation. Each of these parameters are assigned a weight when creating a user's reputation score, however these weights require further analysis to optimize the performance of the system. We first assigned 100% of the weights to each of the three metrics to gauge the performance increase if the reputation score is only compiled through an individual metric. The results show that each individual metric does indeed improve the output accuracy when it is incorporated into the reputation score. In addition we also tried making one of the weights 0.5 and the other two 0.25 so as far as to view the performance of the model by including all three but giving further priority to a single metric. Using this method we can see the same trends in that the third metric w_3 has more impact on the output accuracy, whereas w_1 has the least improvements. Using the results from our first experiment, we then take the improvements of each metric as a percentage of the other metrics to formulate each weight.

We can denote wx as a experimental setup where 100% of the weight is given to one of the three metrics. $accuracy_{wx}$ is the resulting accuracy from that experiment. $accuracy_{baseline}$ is the baseline accuracy where no reputation scheme was used. The difference between these two values is the increment ($increment_{wx}$) shown in equation A.

$$accuracy_{wx} - accuracy_{baseline} = increment_{wx} \tag{A.1}$$

To determine the weight of an individual metric we weigh it as the fraction of total increments, shown in equation A.

$$w_x = \frac{increment_{wx}}{increment_{w1} + increment_{w2} + increment_{w3}} \tag{A.2}$$

Table A.1: Different combination weight for calculation of reputation score

Combination	Test Accuracy	Training Accuracy	Loss Value
$w_2 = w_3 = 0; w_1 = 1$	92.86	95.77	-0.845
$w_1 = w_3 = 0; w_2 = 1$	92.91	96.06	-0.841
$w_1 = w_2 = 0; w_3 = 1$	93.04	96.10	-0.841
$w_3 = w_2 = 0.25; w_1 = 0.5$	93.26	96.18	-0.849
$w_1 = w_3 = 0.25; w_2 = 0.5$	93.19	96.41	-0.839
$w_1 = w_2 = 0.25; w_3 = 0.5$	93.34	96.36	-0.842
$w_1 = w_2 = w_3 = 0.3333$	93.79	96.48	-0.843
$w_1 = 0.3328; w_2 = 0.3293; w_3 = 0.3380$	93.91	96.55	-0.849

Each results is the average between 10 runs as to account for outliers in single runs. The dataset used was an IID version of MNIST with MLP as the base model. Both reputation based client selection and reputation enabled aggregation methodologies were applied. The baseline model without any added methodologies had a baseline result of 88.02% accuracy. By formulating the reputation score based on each metric, we can see that each metric allows for an improvement over the baseline. With the highest improvement seen with the third metric. When including all three metrics but giving priority to an individual one in a 0.5-0.25-0.25 distribution, we can see that the overall results are improved over only utilizing one metric. This is probably due to reducing the bias of a single metric. Using a fair distribution allotted for even higher accuracy gain. However, by using equation A to assign weights allowed us to further optimize our model. Only slight deviations are necessary since all three metrics perform relatively equally.