Reputation-enabled Federated Learning Model Aggregation in Mobile Platforms

Yuwei Wang and Burak Kantarci, *Senior Member, IEEE*

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500928

***Abstract*—Federated Learning (FL) builds on a mobile network of participating nodes that train local models and contribute to the learning model parameters at a central server without being obliged to share their raw data. 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 paper 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. Numerical comparison of the proposed aggregation methodology to a baseline that utilizes standard averaging as well as a second baseline that is scoped to a reputation-based client selection shows an improvement of 17.175% over the standard baseline for not independent and identically distributed (non-IID) scenarios for an FL network of 100 participants. Consistent improvements over the first and second baselines under smaller FL networks with users ranging from 20 to 100 are also shown.**

***Index Terms*—Distributed Learning, Mobile Networks, Federated Learning, Deep Neural Networks, Deep Learning, Reputation systems.**

I. Introduction

The increasing use of smart devices enable the Social Internet of Things (SIoT) phenomenon where smart devices that are equipped with sensors can interact with each other on various tasks by not requiring human intervention [1]. This propels the use of machine learning (ML)-based methods in the applications within smart cities such as smart surveillance and traffic control although applications are not limited to these areas [2].

Mobile devices contain quality data that can be beneficial for many ML models. To enable distributed learning while preserving the privacy of the users Federated Learning (FL) has emerged as a viable concept [3]. By utilizing local data from participant devices, FL is a distributed machine learning methodology to train individual models and share them with a central computation unit for aggregation and redistribution until experiencing an eventual convergence in the performance indicators of the model [2].

The motivation of FL is to build a distributed ML framework that prevents mobile devices from transmitting raw data to the base stations [4]. By doing so, 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 to achieve 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 [5]. It is worth to note 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 with increased transmission efficiency [6]. FL has shown to be promising in various applications such as mobile keyboard prediction,visual object detection and so on [7].

Challenges of FL that are being addressed by the researchers have been centered around improving the privacy capabilities, as well as client selection and user incentives. 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 [8] with the consideration that each local model in the aggregation pipeline contributes equally [9]. In this work, we consider that the participating devices form a mobile network of distributed computing nodes where the performance and reputation of individuals (i.e., nodes) vary across the network. With this in mind, we propose a reputation-enabled weighted aggregation of the local models for distributed learning, and hypothesize that such aggregation would lead to faster convergence. Thus, the contribution of a local model is evaluated by its reputation score, which is formulated based upon three performance metrics (comparison with the performance of 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 last iteration as initially proposed in our previous work [10] for client selection). Particular contribution of this paper 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 convergence at about 100 communication rounds.

The rest of the paper is structured as follows. Section II presents the related work whereas Section III presents the methodology. Experimental settings and numerical results are presented in Section IV along with further discussions. Finally,

Y. Wang and B. Kantarci are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. E-mails: {ywang786, burak.kantarci}@uottawa.ca.

the paper is concluded in Section V alongside future directions and our short term agenda in this field.

II. Related Work

Heterogeneous nature of the mobile devices in a distributed learning environment require optimization strategies subject to the computing power, bandwidth constraints and quality of sensed data [11] [12]. To ensure that the ML models can be trained within a predetermined delay bound, the authors in [13] propose the FedCS scheme that builds on the server's awareness of the resource availability at the participating devices so to choose the clients (i.e, devices) accordingly. It should be noted that it may not be always possible to obtain the needed training time for complex models. The authors in [14] introduce a Hybrid-FL scheme which aims to select the participants with IID (independent and identically distributed) datasets. On 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 [15] leverage Deep Reinforcement Learning (DRL) to filter the clients that are out of the server' 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 [16] to calculate each participant's test accuracy variance so to integrate them into the FedAvg algorithm as aggregation weights aim at fairness.

The authors in [17] improve aggregation by adopting a difference-of-convex (DC) functions algorithm. In [18], 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 [19] tackles the frequency of aggregated updates so 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 [20], 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 [21]. In [22], an analog gradient aggregation method is proposed aiming at fast convergence in the FL network. The study in [23] introduces a communication-efficient secure aggregation to decrease the energy consumption of communication. The study in [24] proposes an asynchronous learning strategy in which the deep layers update at a slower frequency and the shallow layers update at a faster frequency.

III. 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.

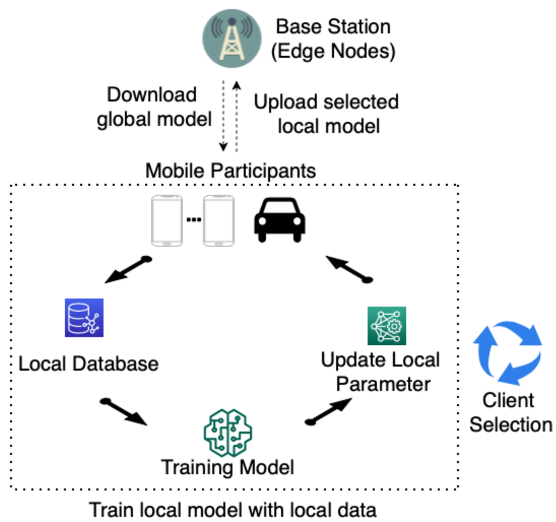

Fig. 1. Local model training

A. Local model training process

As shown in Fig. 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 is utilized in this work, and negative log-likelihood is chosen as the loss function.

B. Client Selection

We adopt the client selection algorithm from our previous work in [10], and describe below briefly.

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 (Rep_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. 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

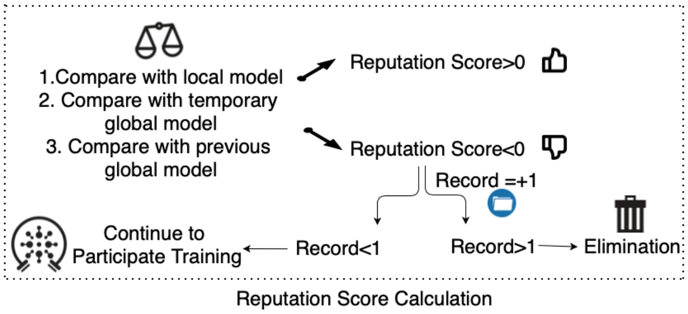

Fig. 2. Reputation calculation process

of the last communication round, respectively.

$$Rep_i = w_1(A_i - \sum_j A_j/n) + w_2(A_i - A_{g_{temp}}) + w_3(A_i - A_{g_{old}}) \quad (1)$$

As shown in Fig. 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 [10].

C. Global model aggregation

As shown in Fig. 3, the server leverages the reputation scores that meet the selection requirement to calculate a normal distribution 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.

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 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. 2 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.

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

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 (3):

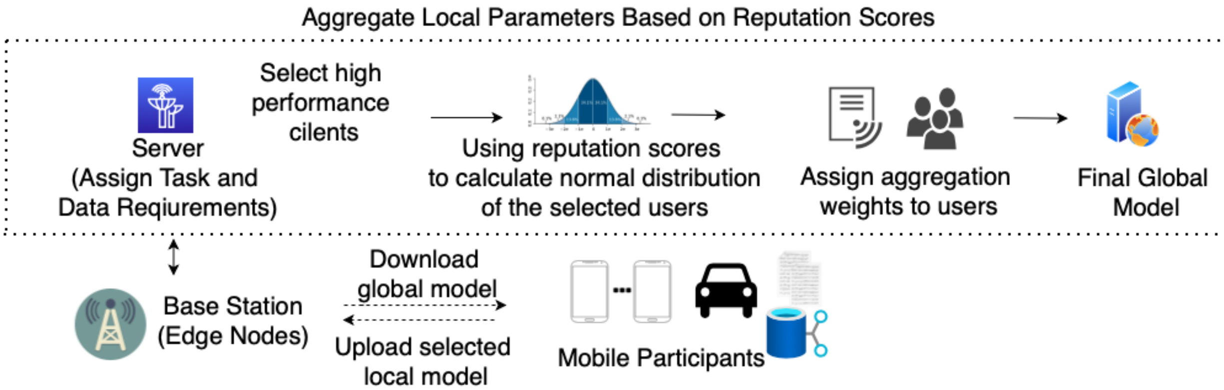

Fig. 3. Minimalist illustration of global model aggregation

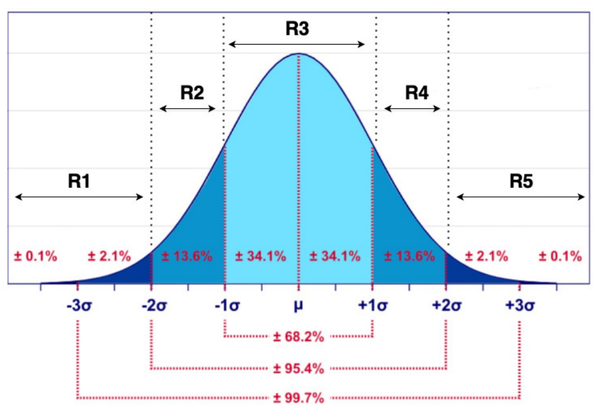

Fig. 4. Five regions with respect to the reputation scores under the assumption that the scores follow a normal distribution.

$$P_{aggr} = \sum_{i=1}^{n} P_i \omega_i^* \tag{3}$$

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.

IV. Experimental Results

Two image-based datasets are utilized to evaluate the proposed methodology. Details of each dataset are as follows:

- MNIST [25]: Handwritten digital data-set includes 60,000 image samples in total, and 10,000 samples are for the test validation. A grey value describes a pixel, 28 x 28 pixels constitute each image of the MNIST data-set.
- Fashion-MNIST [26]: Fashion-MNIST is a substituted image data-set for the MNIST, which contains 70,000 disparate positive image samples in ten categories.

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.

A. *Performance under varying number of users*

Fig. 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 [10] 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 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

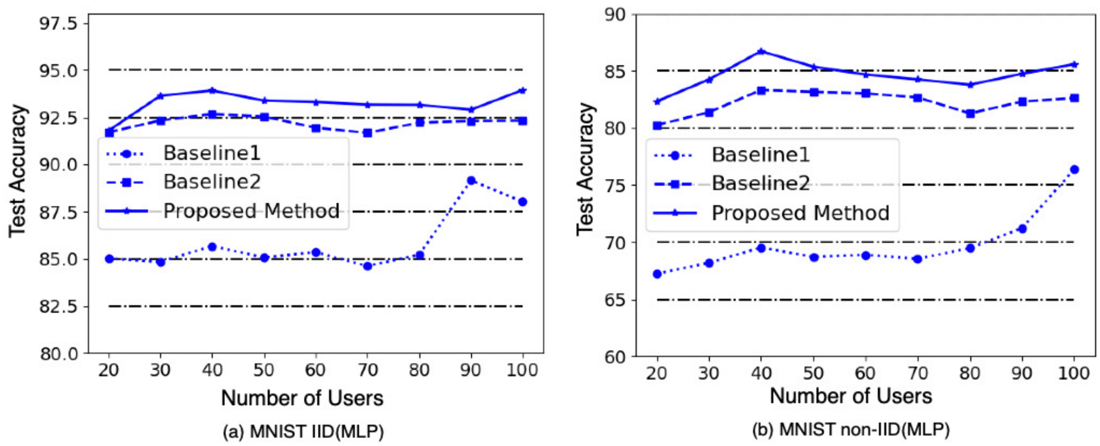

Fig. 5. Average test accuracy comparison under varying number of users. MLP is used as the ML model.

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.

B. Various combinations of ML models and datasets

Table I 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. 5) by utilizing our proposed method since larger variances among the participants occur in a non-IID scenario, which is often the 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.

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.

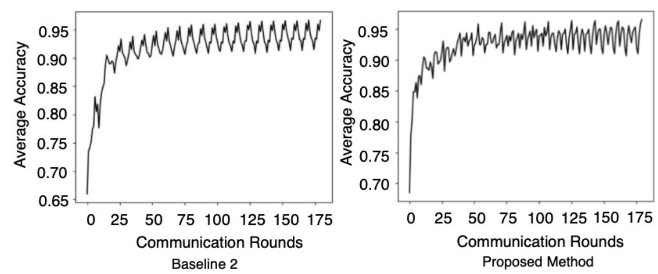

Fig. 6. Comparison of convergence under MNIST IID with MLP as the ML model

C. Convergence performance

Fig. 6 depicts the convergence rate for Baseline 2 and our proposed methodology for FL. Figure 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 I. 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.

V. Conclusions and Future Work

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 paper, 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

TABLE I
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

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.

Our ongoing study addresses dynamically optimizing the reputation threshold for client selection and aggregation based on the specified applications. Furthermore, a contract theory-based approach to incentivize users integrated with the proposed reputation-enabled aggregation is also included in our short term agenda.

ACKNOWLEDGEMENT

This work was supported in part by the the Natural Sciences and Engineering Research Council of Canada (NSERC) under the DISCOVERY Program.

REFERENCES

[1] 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*, pp. 1–1, 2020.
[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.
[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.
[4] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, “Federated learning for mobile keyboard prediction,” *arXiv preprint arXiv:1811.03604*, 2018.
[5] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, “Distributed federated learning for ultra-reliable low-latency vehicular communications,” *IEEE Trans. on Communications*, vol. 68, no. 2, pp. 1146–1159, 2019.
[6] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, “Detailed comparison of communication efficiency of split learning and federated learning,” *arXiv preprint arXiv:1909.09145*, 2019.
[7] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, “Federated learning: A survey on enabling technologies, protocols, and applications,” *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.
[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial Intelligence and Stat., PMLR*, 2017, pp. 1273–1282.
[9] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato, and C. Miao, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.
[10] Y. Wang and B. Kantarci, “A novel reputation-aware client selection scheme for federated learning within mobile environments,” in *IEEE 25th Intl. Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020, pp. 1–6.
[11] J. Serra, L. Sanabria-Russo, D. Pubill, and C. 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)*. IEEE, 2018, pp. 1–6.
[12] H. Chergui and C. Verikoukis, “Offline sla-constrained deep learning for 5g networks reliable and dynamic end-to-end slicing,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 350–360, 2019.
[13] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *IEEE Intl Conf. on Communications (ICC)*. IEEE, 2019, pp. 1–7.
[14] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, “Hybrid-fl: Cooperative learning mechanism using non-iid data in wireless networks,” *arXiv preprint arXiv:1905.07210*, 2019.
[15] H. T. Nguyen, N. C. Luong, J. Zhao, C. Yuen, and D. 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)*. IEEE, 2020, pp. 1–6.
[16] T. Li, M. Sanjabi, A. Beirami, and V. Smith, “Fair resource allocation in federated learning,” *arXiv preprint arXiv:1905.10497*, 2019.
[17] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.
[18] M. R. Sprague, A. Jalalirad, M. Scavuzzo, C. Capota, M. Neun, L. Do, and M. Kopp, “Asynchronous federated learning for geospatial applications,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 21–28.
[19] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, “Adaptive federated learning in resource constrained edge computing systems,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.
[20] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, “A learning-based incentive mechanism for federated learning,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.
[21] Y. Sun, S. Zhou, and D. Gündüz, “Energy-aware analog aggregation for federated learning with redundant data,” in *IEEE Intl. Conf. on Communications (ICC)*, 2020, pp. 1–7.
[22] 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*, pp. 1–1, 2020.
[23] K. Bonawitz, F. Salehi, J. Konečnỳ, B. McMahan, and M. Gruteser, “Federated learning with autotuned communication-efficient secure aggregation,” in *Asilomar Conf. on Signals, Systems, and Computers*, 2019, pp. 1222–1226.
[24] 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*, vol. 31, no. 10, pp. 4229–4238, 2020.
[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[26] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.