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

Yuwei Wang and Burak Kantarci
School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada
E-mails: {ywang786,burak.kantarci}@uottawa.ca

***Abstract*— This paper studies the problem of training federated deep learning models over a mobile environment. Stemming from the federated learning (FL) concept, deep learning models on mobile devices can be trained for various use cases including but not limited to image sorting and prediction of upcoming 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 behaviometric features. However, utilizing the data through conventional approaches can potentially lead to privacy leakages. In this paper, 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 study 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) and under imbalanced distribution. Experimental results show that the proposed reputation-aware FL scheme can achieve improvements in the test accuracy up to 9.30% under different data sets.**

***Index Terms*— Federated learning, deep learning models, data sharing, client selection, mobile networks**

I. Introduction

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 [1]. Traditionally, mobile cloud computing facilitates transmission of data at the edge of the network to cloud servers for processing. As the amount of data increases exponentially, mobile cloud computing experiences bottleneck in the communication network, long end-to-end latency, and user privacy concerns [2]. Among these challenges, user privacy concerns occur as data are collected from users by the service provider, i.e., cloud platform. On the server side, Cloud services need to build and train machine learning models to provide decision support or recommendation services, 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 [3].

With the widespread use of mobile phones, wearable devices and autonomous vehicles, devices that are virtually interconnected, they generate large amounts of data such as images, texts and locations in a ubiquitous manner. 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 aggregated model can be distributed to the users for further local training [4]. 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 training model. 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 [5]. By leveraging federated learning, it becomes possible to use mobile user data without compromising their privacy needs. Meanwhile, increasing computing power of mobile devices makes it possible to keep data locally and complete calculations locally on the device. Unlike traditional machine learning solutions which are based on sharing of aggregated data, centralized storage, and centralized processing, federated learning concept that builds on a distributed machine learning framework which uses local training can ensure user privacy and data security.

Federated learning has inherent privacy benefits compared to the traditional method that aggregates and trains data over a centralized server as it maintains user data locally, and uses machine learning to build a joint model of multiple users into a virtually shared model so to solve the problem of –so called– data islands without violating user privacy or any regulations [6]. After applying federated learning, a constantly updated model in the user's mobile phone is learned and updated based on the data generated by the user, and the updated weight is uploaded to the server in an encrypted form. After receiving a large number of user models, the server aggregates the received models and distributes the aggregated model back to the user. It is worth to note that, the model on user equipment is compressed whereas the model hosted on a cloud server entails a large neural network model, and therefore the energy consumption of model training is significantly lower and almost insignificant [2]. Federated learning is performed iteratively as follows: 1) In each iteration, the edge server

978-1-7281-6339-0/20/$31.00 ©2020 IEEE

sends the current global machine learning model parameters to all edge nodes participating in the federation (of learning); 2) 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; 3) Each edge node uploads the updated model parameters to the edge server; 4) 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 [7]. These steps keep being iterated on until the trained model converges. Various applications can benefit from federated learning such as Gboard, vehicular sensing computational photography, city management, enterprise risk management, defense, telecommunications, IoT, or pharmaceutics [8] . 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) [9]. Consequently, poor data reduces the performance of the generated model and thus prolongs training times. The malicious user is a kind of attacker that may steal privacy information from model parameters or may provide low quality updates to obtain the global models. Since the models trained by users is distributed after each global model aggregation, it might be possible to learn the information regarding malicious model, and valid users may still receive the malicious models from base station. For instance, by analyzing the valid updates, malicious participants can extract personal information of mobile users. Therefore, an alternative scheme to address the potential privacy leak is needed.

With these in mind, this paper introduces a reputation score to assess the performance of each local model based on a variety of performance metrics. Furthermore, the proposed scheme improves the final aggregated model's performance through user selection based on the proposed reputation score. Moreover, the training time is improved as a result of phasing out malicious and poor performing users. With addition of the proposed methodologies, we improve the original implementation by up to 10% for non-IID datasets. We also reduce the training time by roughly 27.7% with our selection method compared to the baseline implementation.

The rest of the paper is structured as follows. In Section II, related works are presented. Section III presents the methodology. Numerical results are presented in Section IV along with discussions. Finally, the paper is concluded in Section V, and future directions are given.

II. Related Work

Lately, a number of existing studies have proposed to leverage federated learning in various environments. Just to name a few, the authors in [10] 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 [11] 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 MNIST dataset to achieve highest classification accuracy after a certain volume of communication traffic. A decentralized FL algorithm is developed in [12] to address challenges concerning transmit power and resource allocation to enable ultra-reliable and low-latency vehicular communications (URLLC). In [13], 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 [14] 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 [15] propose a new multiple access method to achieve rapid aggregation of global model parameters. In other studies such as [16], [17], reduction of the communication overhead is tackled for joint learning. More specifically, the authors in [16] propose a ternary gradient method to reduce the communication latency when federated learning is pursued whereas the authors in [17] 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 [18] to study efficient use of limited resources to implement adaptive joint learning.

Kang et al [19] combined reputation and contract theory to optimize 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 the 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 that study as we aim to directly formulate a reputation score through the test performance of the users (i.e., direct reputation), and focusing on the improvements in the test accuracy of a federated learning model.

III. Methodology

The proposed methodology aims to improve the use of federated learning in a mobile training environment particularly

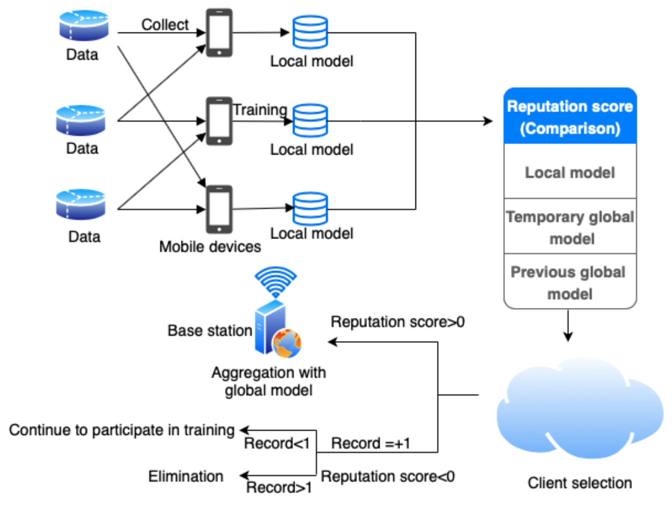

Fig. 1: Federated learning in mobile setting

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. 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 a 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. A client to participate in training in the current epoch is chosen 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
        $L = L \cup \{i\}$;
    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
```

A. 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 the 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 the reputation score calculation. The reputation score is calculated through 3 parameters:

$$score = w_1(acc-avg)+w_2(acc-test_{acc})+w_3(acc-old_{acc}) \quad (1)$$

The training stops when the learning curve converges or the system runs out of local users. In Eq. 1 test accuracy of each local model needs to be compared to 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 the 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 into 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 the previous global model.

Algorithm 2 Computation of reputation score in the local model in epoch e

```
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
```

B. 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.

IV. Experiment Results

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 [20]: MNIST is a handwritten digital dataset, where the training dataset contains 60,000 samples 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 [21]: 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.

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:

- non- Independent and Identically Distributed (non-IID): 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.
- Independent and Identically Distributed (IID): the shuffled data is partitioned to 100 users so that each user acquires 600 images.

Table I 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 I, it can be seen that, as the number of of given chances increases, the value of the test accuracy improves. (Moreover, as the number of eliminated models increases, the value of time consumption increases.) This is due to the fact that an increase in r leads to more calculation procedure, hence, improving test accuracy as well as training accuracy. The record r = 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 I: Selection of the best number of chances (r) given to the poor performing local models before eliminating them

	Base	$r=0$	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$
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

TABLE II: Comparison of users for 10 epochs

	f=0.1	f=0.2	f=0.3	f=0.4	f=0.5
Train Acc.	95.655	95.125	93.09	93.07	91.81
Test Acc.	92.752	92.705	91.015	90.69	91.81

TABLE III: 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	$w_1 = w_2 = w_3$

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 epochs; (4) L, a hyperparameter used in the training of federated learning; controls the speed of the local model adapted to the problem; (5) w, the weights in the reputation calculation.Table II presents the performance under varying fractions of users (f).

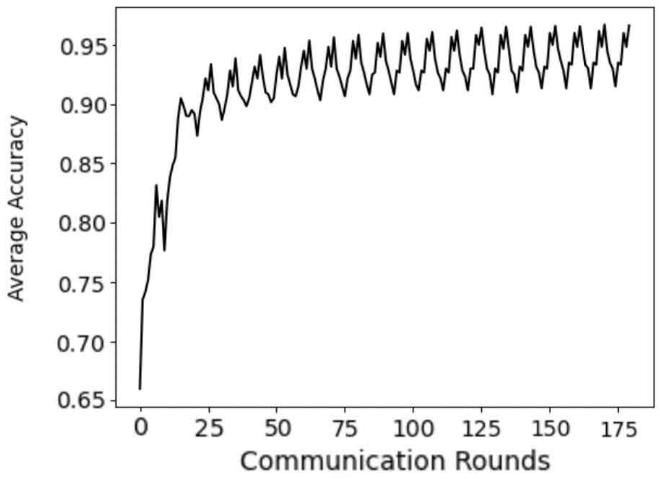

Fig. 2: Average training accuracy vs. Communication rounds

The results of empirical selection of the experimental parameters are presented in Table III. The fraction of users stands for the fraction of selected population for aggregation 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 into the global model. In each epoch, 100 local models are trained and 10% is used to aggregate with the global model. A communication round 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 2 shows the learning curve of the proposed model. Based on this, the model is trained up to 100 communication rounds since a clear convergence is shown in the figure.

Table IV presents the 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 (F-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

TABLE IV: 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

the proposed federated learning methodology is applied. Both the non-IID MNIST and F-MNIST datasets are 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 the test accuracy by 3.335% for MLP and 2.65% for CNN.

V. Conclusions and Future Work

We have developed a novel methodology that improves federated learning on a mobile environment by introducing a reputation-aware user selection scheme so to eliminate the poorly performing local model contributions. 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. Through numerical results, it has been shown that the test accuracy can be improved by at least 1.733% and up to 9.306%.

Our ongoing work includes a new aggregation algorithm to further improve the performance of the global model by formulating an advanced reputation function for the clients.

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] 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.

[2] 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.

[3] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[4] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016.

[5] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," 2019.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," pp. 1273–1282, 2017.

[7] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[8] 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 Comm. Surv. & Tutorials*, 2020.

[9] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, 2019.

[10] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "Performance optimization of federated learning over wireless networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[11] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Workshop on Distributed Infrastructures for Deep Learning*, 2018, pp. 1–8.

[12] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, 2019.

[13] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing- supported internet of things," *IEEE Access*, vol. 7, pp. 69 194–69 201, 2019.

[14] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[15] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—a key technology towards 5g," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[16] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in neural information processing sys.*, 2017, pp. 1509–1519.

[17] X. Li, G. Zhu, Y. Gong, and K. Huang, "Wirelessly powered data aggregation for iot via over-the-air function computation: Beamforming and power control," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3437–3452, 2019.

[18] 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.

[19] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[20] 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.

[21] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.