FedEdge: Communication and Data Efficient Federated Learning for Edge Computing

Abstract—**Federated learning has emerged recently as a technique for training a global model in a decentralized manner. The central sever aggregate the updated parameters from the edge clients enabling the raw data remained at the local clients. Comparing with traditional distributed learning at data center, reducing communication overhead, especially the uplink transmission cost from the edge clients to the central server is one of the most vital goals in federated learning optimization. In this paper, we propose an novel federated learning structure that joint consider the realist communication network configuration and the training accuracy. We implement the proposed system in fedSGD, fedavg and Per-fed algorithms. The model accuracy, communication round, and the total latency are presented for evaluations. The competitive numerical experiments reveal the outperformance and the compatibility of the system.**

Index Terms—**Federated Learning, Edge Computing**

I. Introduction

Massive machine learning models are commonly trained and conducted at the data center. However, with the increasing amount of data, storing and conducting the whole data at a single node is undesirable. Transmitting the raw data from the edge users is neither practical nor effective due to the privacy and communication constrains. Hence, it is increasingly attractive to move computation from the cloud to the edge where the local dataset is accessible, and send the local models back to the server for aggregation. Federated learning consequently has been proposed [1] as an alternative approach for training a high quality global model without ever sending raw data to the cloud.

Communication cost is the largest bottleneck in federated optimization, while uplink is of utmost importance in communication network systems. The typically upload bandwidth is limited by 1 MB/s or less. Uplink communication-efficient structure is in desperate need for federated learning system. On the other hand, any single on-device dataset is small compared to the total dataset size. To achieve personalized federated learning and obtain a sensitive global model that can quickly adapt to test set, the local dataset can be split as support set and query set for better training performance. Data-efficient federated learning system in this scenario is on great importance.

In the context of sixth generation (6G) communications, massive machine nodes, e.g., distributed sensors, operating in the Internet of Things (IoT) are with simple structures and of limited complexity and energy, which might not well support multi-band transmission and sophisticated encoding/decoding [2]. Therefore, simpler multiple access schemes are usually preferred in these application scenarios. Following this rationale, we adopt the dynamic time-division multiple access (TDMA) for enabling multiple access by distributed nodes for uploading their local training models to the data center through a shared wireless medium. In particular, each node intending to upload its local training model will be allocated a specific time slot with a fixed length [3]. Finally, multiple and non-overlapping time slots comprise an uplink transmission frame with a dynamic duration jointly depending on the maximum allowed frame duration and the number of participating nodes. TDMA is a proper workaround for communication scenarios under the complexity and energy consumption constraints but, on the other hand, increases the network latency that is proportional to the number of participating nodes for federated optimization.

This paper makes the following contributions:

- We proposed a novel federated learning framework that provides communication efficient and data efficient federated learning approaches. The framework is theoretically compatible federated averaging and personalized federated algorithms.
- We joint consider the model accuracy and the communication characters. The edge users conduct the transmission to either the nearby user or base station based on the communication latency.
- We compare the algorithms under the proposed system with commonly used federated learning benchmarks. A realistic traffic prediction paradigm is adopted to evaluated the performance of the framework. Extensive experimental results reveal the effectiveness and robustness.

II. Related Works

A variety of different approaches to have been proposed to address the challenges in federated learning. Authors in [4] proposed the federated averaging algorithm (FedAvg) to reduce the communication round by replacing stochastic gradient descent (SGD) with multi-update of the local models. [5], [6] tackled heterogeneity problem in federated learning by adding regular term. [5] theoretically provided the convergence guarantees for non-identical distributions dataset, while [6] utilized Moreau envelopes as edge user's regularized loss functions to obtain personalized model and proved that the convergence rate of the proposed framework achieved quadratic speedup for strongly convex and sublinear speedup of order 2/3 for smooth non-convex objectives. Multiple approaches have been proposed to achieve personalized model in FL. Authors in [7] proposed an algorithm, in terms of L2GD, which combined the optimization of the local and global models. [8] studied a personalized variant of the FedAvg algorithm to obtain an initial shared model that users can easily adapt to their local dataset by performing one or a few steps of gradient descent with respect to their own data. The above mentioned investigations aim to enhance communication efficiency by controlling strength of local model optimization, however, ignored the physical communication dependencies. more literature review about physical layer, Mingzhe Chen

III. Problem Formulation and System Design

A. Federated Learning Model

The objective function of federated learning is:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d}\{f(\boldsymbol{w}):=\sum_{i=1}^{I}p_i f_i(\boldsymbol{w})\} \tag{1}$$

where $\boldsymbol{w}$ represent the global model. The aggregation weight corresponding to the local objective function f_i is represented as $p_i=\frac{n_i}{n}$, where $n=\sum n_i$ is the total number of the data points. $f_i(\boldsymbol{w}):\mathbb{R}^d\rightarrow\mathbb{R}, i=1,\cdots,I$ denotes the objective function over the data distribution D_i of the user i:

$$f_i(\boldsymbol{w})=\mathbb{E}_{x_i\in D_i}[f_i(\boldsymbol{w};x_i)] \tag{2}$$

At each communication round t, the central server transmit the latest global model $\boldsymbol{w}$ to all clients via downlink broadcast. Then, after all clients complete the local updates, the server collect the latest local models from a uniformly sampled subset S_t of clients to perform the model aggregation. The baseline algorithm, FedSGD, provides the update process such as:

$$\boldsymbol{w}^{t+1}=\boldsymbol{w}^t-\eta\sum_{i=1}^{I}\frac{n_i}{n}\Delta f_i(\boldsymbol{w}^t) \tag{3}$$

FedAvg provides a equivalent update $\boldsymbol{w}_i^{t+1}=\boldsymbol{w}_i^t-\eta\Delta f_i(\boldsymbol{w}_i^t)$ and add more computation to each user by iterating the locl update multiple times before the global averaging. In this manner, the parameter E is defined to tune the number of training passes each user makes over the local datset before each communication round.

B. System Model

We consider federated learning process with one base station as the server and across I wireless users, each with its own local dataset. The local models are trained based on the users' own dataset, and can be sent to the server directly via uplink or relaid by other users via the D2D links. The uplink rate of user i is given by

$$c_i^{\mathrm{U}}=B_i^{\mathrm{U}}\mathbb{E}_{h_i}\left(\log_2\left(1+\frac{P_i^{\mathrm{U}}h_i}{I^{\mathrm{U}}+B^{\mathrm{U}}N_0}\right)\right), \tag{4}$$

where B_i^{U} and P_i^{U} are the bandwidth and the transmit power allocated to the uplink channel, respectively. $h_i=o_i d_i^{-2}$ is the channel gain between user i and the central server with d_i representing the distance between user i and the central server and o_i being the Rayleigh fading parameter. I^{U} is the interference caused by other users occupying the same bandwidth as the user we are considering. N_0 is the noise power spectral density.

The data rate of the D2D channels are calculated as

$$c_{i,j}^{\mathrm{D}}=B^{\mathrm{D}}\mathbb{E}_{h_{i,j}}\left(\log_2\left(1+\frac{P^{\mathrm{D}}h_{i,j}}{I^{\mathrm{D}}+B^{\mathrm{D}}N_0}\right)\right), \tag{5}$$

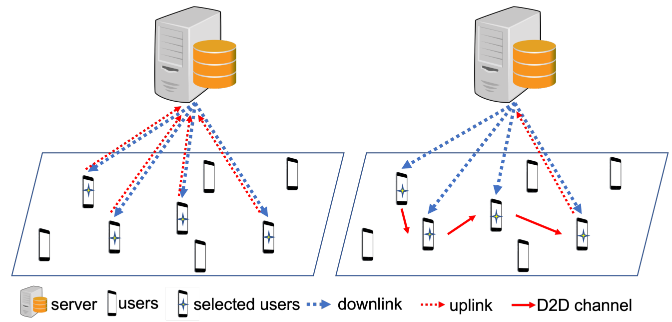

Fig. 1: System of FedEdge for communication efficient FL

where B^{D} and P^{D} are the bandwidth and the transmit power allocated to the D2D channel, respectively. $h_i=o_{i,j}d_{i,j}^{-2}$ is the channel gain between user i and the user j with $d_{i,j}$ representing the distance between user i and user j and $o_{i,j}$ being the Rayleigh fading parameter. I^{D} is the interference caused by other users occupying the same bandwidth as the user we are considering. N_0 is the noise power spectral density.

The transmission delays between user i and the central server over uplink and the transmission delays between user i and use j over the D2D links are respectively specified as

$$l_i^{\mathrm{U}}\left(\boldsymbol{r}_i,P_i\right)=\frac{Z\left(\boldsymbol{w}_i\right)}{c_i^{\mathrm{U}}}, \tag{6}$$

$$l_{i,j}^{\mathrm{D}}=\frac{Z\left(\boldsymbol{w}_i\right)}{c_{i,j}^{\mathrm{D}}}, \tag{7}$$

$Z(\boldsymbol{w})$ in the unit of bits is the size of the model $\boldsymbol{w}$ to be transmitted via wireless links. In particular, $Z(\boldsymbol{w}_i)$ represents the number of bits that each user i requires to transmit local FL model $\boldsymbol{w}_i$ to either the BS or the other users. Here, $Z(\boldsymbol{w})$ is determined by the type of implemented FL algorithm. Generally, $\boldsymbol{w}$ remains the same structure for all the users, hence $Z(\boldsymbol{w}_i)$ is uniform among the transmission between the user to the BS and to other users.

We consider the TDMA techniques while collecting the updated local models, where the models are transmitted sequentially in the time domain for sharing the communication medium. Therefore, the total network latency of one iteration consisting of the transmission phases of all I nodes adopting the TDMA protocol can be expressed as:

$$L^{\mathrm{tx}}=\sum_{i=1}^{I}l_i \tag{8}$$

where $l_i=\min\{l_i^{\mathrm{U}},l_{i,j}^{\mathrm{D}}\}$ is the transmission latency of user i.

The global model is aggregated after completing the transmission of all the local models. Therefore, the updated models with total latency that exceed the threshold will not contribute to the aggregation process.

IV. Communication-Efficient FedEdge

To achieve communication-efficient training, the users suppose to be smart to choose where to transmit their updated

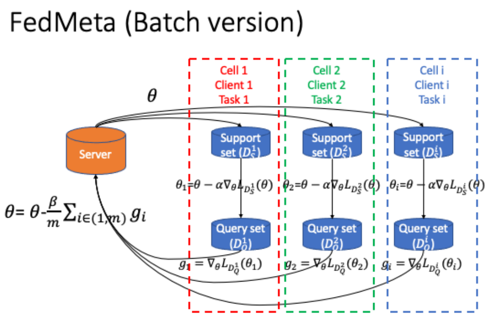

Fig. 2: System of FedMeta

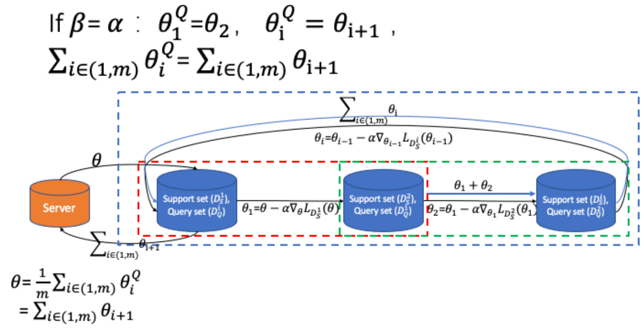

Fig. 3: System of FedEdge

local models. We reformulate problem (1) by jointly consider the model accuracy and the communication-efficiency.

$$\min_{\boldsymbol{w}\in\mathbb{R}^d}\{f(\boldsymbol{w}):=\sum_{i=1}^{I}p_i f_i(\boldsymbol{w})+L^{\text{tx}}\}, \tag{9}$$

$$f_i(\boldsymbol{w})=\mathbb{E}_{x_i\in D_i}[f_i(\boldsymbol{w};x_i)]+\mu L_i^{\text{C}}(\boldsymbol{w}-\boldsymbol{w}_0), \tag{10}$$

where $L_i^{\text{C}}(\boldsymbol{w}-\boldsymbol{w}_0)=K_i\Delta T_i(\boldsymbol{w}-\boldsymbol{w}_0)$ represent the overhead of the local model updating, and μ is a penalty parameter that control the strength of the model caring about the latency. K_i is the optimization steps of user i and $\Delta T(\boldsymbol{w}-\boldsymbol{w}_0)$ as the function of $\boldsymbol{w}$ reveals the interval of each step. The value of ΔT varies by users due to the variable computing device conditions, and $\boldsymbol{w}_0$ is the baseline global mode received from the server at each iteration.

FedEdge provides a intuitive solution to determine the strength of $\boldsymbol{w}$ to the local model. The basic idea is allowing the user to tune the local optimization step by considering the transmission dependencies. The updating step K_i is quantified as:

$$K_i=\min\{E\left(\frac{l_i^U}{l_i}\right),E_{\max}\}, \tag{11}$$

where $E_{\max}$ represent the largest local optimization steps. The intuition is that the lower transmission latency reserves more time intervals for local optimization. Moreover, since the the users are randomly selected at each global iteration, the user should take the advantage of the low transmission latency at currant iteration. In the extreme case, when $l_i=\min\{l_i^{\text{U}},l_{i,j}^{\text{D}}\}=l_i^{\text{D}}\ll l_i^{\text{U}}$, the ratio between l_i^{U} and l_i is infinite large, then the local optimization step K_i should be bounded by the maximum steps $E_{\max}$. On the other hand, when l_i^{D} is larger than l_i^{U} which means that the $l_i=l_i^{\text{U}}$, then the local optimization step K_i equals to the normal updated step E which is the pre-setted hyper-parameter.

V. Data-Efficient and Personalized FedEdge

The goal of personalized federated learning is to train a sensitive global model in a collaborative way while accounting the heterogeneity of data and the partial participation of users. Authors in [9] provide a model-agnostic meta-learning algorithm to train the model which can quickly adapt to a new task with only a few steps of inner update. Formally, the local dataset of the i-th client is divided into support set D_i^{s} and query set D_i^{q}. The local model θ is first updated by training on support set with learning rate η, and then fine tuned by query set and aggregated at the server with learning rate β. The meta-objective is given as:

$$\min_{\theta}\sum_{\mathcal{T}_i\sim p(\mathcal{T})}\mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}\right)=\sum_{\mathcal{T}_i\sim p(\mathcal{T})}\mathcal{L}_{\mathcal{T}_i}\left(f_{\theta-\beta\nabla_\theta\mathcal{L}_{\mathcal{T}}(f_\theta)}\right) \tag{12}$$

where $\mathcal{L}$ is the loss function, and $\mathcal{T}_i$ is the task following task distribution $p(\mathcal{T})$.

For federated learning, each training process at the edge is referred as a task. Herein we can rewrite the local objective function with similar formulation as (2):

$$f_i(\boldsymbol{\theta})=\mathbb{E}_{x_i\in D_i^{\text{q}}}[f_i(\boldsymbol{\theta}-\beta\nabla\mathbb{E}_{x_i\in D_i^{\text{s}}}[f_i(\boldsymbol{\theta};x_j)];x_i)] \tag{13}$$

The global model is updated by the outer gradient and aggregation process:

$$\boldsymbol{\theta}^{t+1}=\boldsymbol{\theta}^t-\beta\sum_{i=1}^{I}\frac{n_i^{\text{q}}}{n}\nabla f_i(\boldsymbol{\theta}^t), \tag{14}$$

where n_i^{q} is the number of samples selected from the query set of the user i. In the realistic federated learning scenario, the samples are usually information recorded by the edge devices, such as mobile-phone, IoT sensors, and wearable devices. Therefore, the number of samples at each edge client is commonly smaller than the number of clients. The limitation of the data resulting the challenge of splitting support set and query set. On the other hand, the edge clients are naturally grouped in terms of geography dependencies. FedEdge opens new options for the case where the number of data distributed among the users is limited to be split into support set and query set. We bound a pair of users as one task: the dataset of user i is referred as support set, and the dataset of user $i+1$ is settled as query set.

$$f_i(\boldsymbol{\theta})=\mathbb{E}_{x_i\in D_{i+1}}[f_i(\boldsymbol{\theta}-\eta\nabla\mathbb{E}_{x_i\in D_i}[f_i(\boldsymbol{\theta};x_j)];x_i)] \tag{15}$$

The local models and are sequentially transferred until the last user in the selected subset at current round. The last user send the fine-tuned model to the first user in the subset and conduct the fast adaption with the first user's local dataset.

TABLE I: System Parameters.

Parameter	Value	Parameter	Value
η	2	N_0	-100dBm/Hz
P	1 W	B^{U}	20MHz
R	5	B^{D}	20MHz
$E_{\max}$	20	P_{B}	1 W

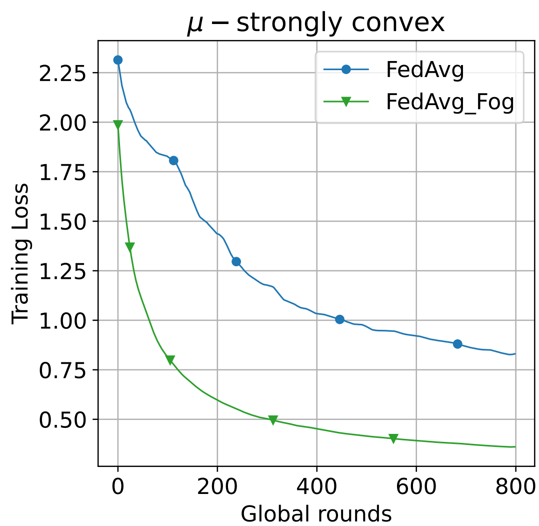

Fig. 4: Caption

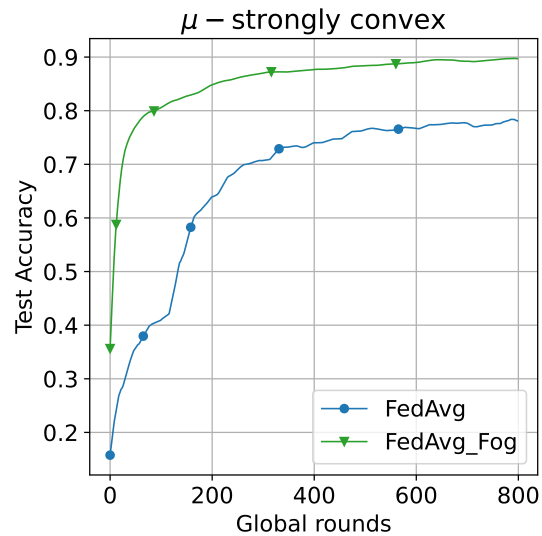

Fig. 5: Caption

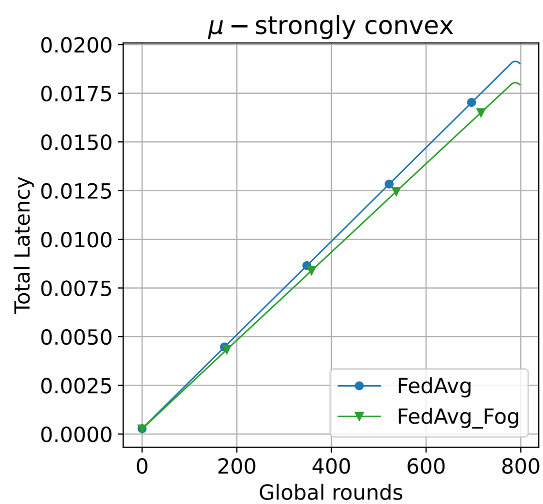

Fig. 6: Caption

VI. Numerical Results

In this section, we implement the framework we proposed in communication-efficient and data-efficient federated learning scenarios. We compare FedEdge algorithms with Fedavg and PerFed, which are the most commonly adopted communication-efficient decentralized training algorithm and personalized federated learning algorithm, respectively.

A. Dataset and Experimental Settings

We adapt the realistic dataset MNIST which is widely considered for investigating the classification tasks. The dataset is uniformly assigned to $I = 20$ users, with each user maintains the sub-dataset contains 2 of the 10 labels.

Traffic prediction dataset, Omnglot dataset @Chuanting

The value of parameters are shown in Table I

B. Numerical Results for Communication-Efficient FedEdge

In order to highlight the improvement of communication efficiency of FedEdge framework. We compare the proposed algorithm with federated learning benchmark, FedAvg. For fair comparison, we set the local update step of FedAvg twice as FedEdge, since by processing FedEdge the local model is updated by support set and query set. The time limitation for one communication round is 1 ms. The packages with latency exceed the threshold will be dropped, herein have no contribution to the global model update. The comparisons for MNIST dataset are shown in Fig. 4 5, 6. Fig. 4 reveals the faster reduction of loss for FedEdge. Fig. 5 evaluate the algorithms by sample accuracy on test set. The test accuracy of FedEdge achieves 90% while the test accuracy of FedAvg is onlky 78%. This is due to the communication-efficient algorithm reduce the communication latency and update the global model frequently.

C. Numerical Results for Data-Efficient and Personalized FedEdge

We compare the proposed algorithm with personalized federated learning benchmarks, Per-FedAvg, pFedMe (GM), and pFedMe (PM). The results show that both FedEdge and pFedMe perform better than Per-FedAvg. The reason is by implementing pFedMe algorithm, the step of the local model is optimized with respect to the Moreau envelopes decorated of local loss function. While the proposed FedEdge framework fully utilizes the available data among the group of users, determines the local optimization steps by considering computational overhead, and takes the advantage of neighborhood transmission to reduce the communication latency. The pFedMe algorithm performs better training loss reduction but

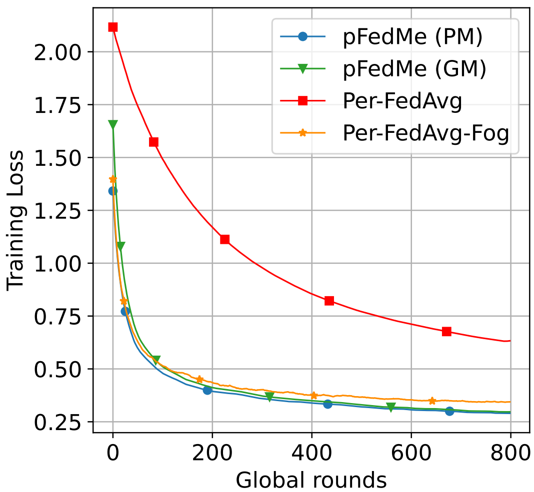

Fig. 7: Caption

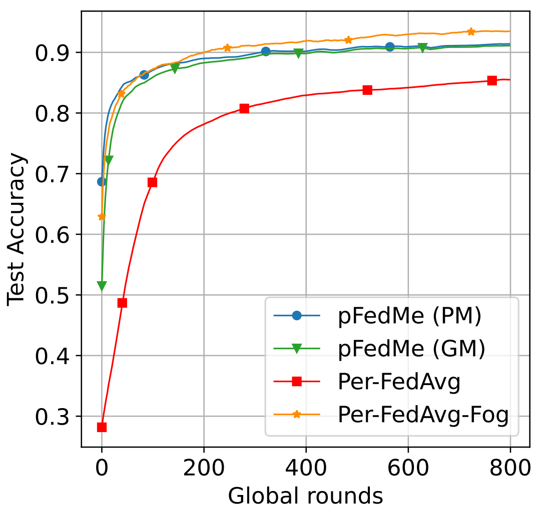

Fig. 8: Caption

FedEdge out-stands in the test accuracy evaluating. The result validates the ability of FedEdge's fast adaption on new tasks.

VII. Conclusion

In this paper, we proposed an innovative decentralized structure that enable communication-efficient and data-efficient federated learning. The proposed framework take into consideration both model accuracy and physical dependency while training for the sensitive global model and the personalized local models. We evaluate the framework by comparing the proposed algorithms with several commonly adopted federated learning benchmarks on two realistic dataset. The extensive empirical results reveal the effectiveness and the compatibility of the framework. In the future work, we will provide the theoretically analysis of the convergence properties.

References

[1] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[2] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6G be?" *Nature Electronics*, vol. 3, no. 1, pp. 20–29, 2020.

[3] A. Kanzaki, T. Uemukai, T. Hara, and S. Nishio, "Dynamic TDMA slot assignment in ad hoc networks," in *Proc. IEEE International Conference on Advanced Information Networking and Applications*, Xi'an, China, 2003, pp. 330–335.

[4] 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 statistics*. PMLR, 2017, pp. 1273–1282.

[5] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[6] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," *arXiv preprint arXiv:2006.08848*, 2020.

[7] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," *arXiv preprint arXiv:2002.05516*, 2020.

[8] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," *arXiv preprint arXiv:2002.07948*, 2020.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.