

D2D-Assisted Federated Learning in Mobile Edge Computing Networks

Xueqing Zhang^{*†}, Yanwei Liu^{*†¶}, Jinxia Liu[‡], Antonios Argyriou[§] and Yanni Han^{*}

^{*}Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

[†]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[‡]Zhejiang Wanli University, Ningbo, China

[§]University of Thessaly, Volos, Greece

Abstract—With the proliferation of edge intelligence and the breakthroughs in machine learning, Federated Learning (FL) is capable of learning a shared model across several edge devices by preserving their private data from being exposed to external adversaries. However, the distributed architecture of FL naturally introduces communication between the central parameter server and the distributed learning nodes. The huge communication cost poses a challenge to practical FL, especially for FL in mobile edge computing (MEC) networks. Existing communication-efficient FL systems predominantly optimize their intrinsic learning process and are not concerned with the implications on the network. In this paper we propose a FL scheme that leverages Device-to-Device (D2D) communication (hence called D2D-FedAvg) and is suitable for mobile edge networks. D2D-FedAvg creates a two-tier learning model where D2D learning groups communicate their results as a single entity to the MEC server leading to traffic reduction. We propose the schemes for D2D grouping, master UE selection, and also D2D exit in the learning process and then form a complete D2D-assisted federated averaging algorithm. Via extensive simulations on the Federated Extended MNIST dataset, the feasibility and convergence of D2D-FedAvg scheme are evaluated. Our results show that D2D-FedAvg lowers the communication cost relative to the typical Federated Averaging (FedAvg) in cellular networks as the number of users is increased (for 100 cellular users 37% traffic reduction), while keeping the same learning accuracy with FedAvg across the board.

Index Terms—Edge Intelligence, Federated Learning, D2D, MEC

I. INTRODUCTION

Nowadays, different types of user equipment (UE), such as mobile phones, autonomous vehicles and wearable devices, generate a huge amount of data that continues to grow exponentially [1]. The applications used in these devices require access to the cloud for real-time processing. However, due to the volume of produced data at the UEs, it is impractical to send all the data to the cloud. In addition, the data privacy requirements for user data require local data processing whenever possible. The newly defined 5G mobile networks embrace this paradigm shift in mobile computing, where the move is from centralized cloud computing towards mobile edge computing (MEC) [2].

To support the required processing at large scale, MEC systems usually require cooperation between the UEs and

edge server. Since this cooperation is instrumental in MEC systems, Edge Intelligence [3][4] has attracted significant research efforts [5]. Traditionally, the MEC framework assumes that all data is transmitted from smart devices to MEC servers via a cellular network to perform their tasks. However, these devices are owned by individuals, and they are generally unwilling to share data due to privacy issues. Thus, in the context of edge intelligence, Federated Learning (FL) [6] plays an important role for exploiting the knowledge that is contained in data scattered across the users.

The FL paradigm aims to distribute the training of a machine learning model in multiple end nodes of MEC networks and execute any machine learning algorithm at the end nodes under the supervision of the edge server [7]. FL has been proposed by McMahan et al. [8] to enable collaborative machine learning over a large number of edge devices without central training. Serving as an enabling technology in mobile edge intelligence [9], FL learns a shared model by aggregating locally-computed updates. The globally shared model is updated by averaging a local stochastic gradient descent (SGD). During model aggregation, a large number of parameters are communicated between edge server (parameter server) and the UEs.

As the number of end nodes increases and the dimension of the artificial intelligence model increases, the updating and conversion process of the learning model parameters will lead to a heavy communication overhead. Moreover, mobile communication resources are limited in mobile edge networks, and the edge services face the problem of high traffic demand which often occurs in some areas. In such cases, it is challenging for edge servers to effectively manage the traffic load, and meet the traffic demand necessary for timely communication in FL. This of course impacts FL efficiency. Hence, reducing the communication overhead for FL is critical for edge intelligent applications [10].

There are two types of approaches for reducing communication overhead in FL. One suggests the reduction of participants in FL. Nishio et al. [11] proposed a FedCS framework that controls the number of participants based on their computational capabilities and consequently reduces the communication overhead. The second class of works compresses the bits of the transmitted parameters [12] or reduces the number of communication rounds [13] between the edge server and the end node. In [14], the authors proposed to add the fog node, a middleware platform between cloud and edge

This work was supported in part by National Natural Science Foundation of China under Grant 61771469 and Ningbo Natural Science Foundation under Grant 2019A610109.

[¶]Corresponding Author.

device, to reduce the backhaul communication to the cloud. Similarly, [15] proposed a client-edge-cloud hierarchical FL system to reduce communication overhead. CMFL [10] provides clients with feedback information regarding the need for global updates of the model so as to reduce unnecessary updates. These approaches can achieve a trade-off between learning robustness and communication efficiency by optimizing an internal parameter of FL. However, these works did not consider communication resource optimization that is dedicated for FL in mobile networks. Recently, FL with a particular focus on wireless communications has been studied in [16]. Nevertheless, to the best of our knowledge, the existing literature [17][18] of merging the device-to-device (D2D) communication into FL not consider reducing the communication overhead of FL. The D2D grouping can be exploited to optimize the communication process in FL.

D2D communication was proposed as a mechanism that enhances spectrum utilization in cellular systems [19]. As the name suggests, a communication link is established directly between the devices for data transmission instead of relaying data through a base station, an approach that greatly reduces the network traffic to the base station. Compared with the other pass-through technologies that do not rely on the network infrastructure, D2D is more flexible, enabling both connection and resource allocation under the control of the base station. In the meanwhile, D2D communication uses the authorized frequency band of telecom operators, with controllable interference environment and high reliability of data transmission.

With the development of MEC in 5G networks, D2D in MEC networks went a step further [19]. In a MEC system combined with D2D, two or more devices in mutual proximity establish direct local short range links and bypass the base station, and these links allow user's processing to be offloaded to a nearby device with more available computing resources. Since FL in MEC networks introduces naturally a communication between multiple UEs and the MEC server (as the parameter server), it is possible to combine D2D with FL to offload a part of FL's communication traffic from the MEC server to the local links between UEs.

Inspired by the D2D paradigm, we propose to adopt D2D communication into FL so as to reduce the communication overhead between the edge server and the end nodes. Specifically in this paper, we propose a D2D-assisted hierarchical FL scheme that is suitable for mobile edge networks. Nevertheless it is far from trivial to decide how this will be accomplished, requiring thus a new system model and a set of algorithms for distributed D2D-based FL. To this aim we introduce first the notion of the *D2D-assisted learning group*, where a master learning UE (MUE) acts as a client that communicates with the edge server for updating the model parameters of the D2D group. The remaining devices in the same D2D group act as the slave learning UEs (SUEs) that participate in collaborative learning with the MUE. Thus, the D2D group can be considered as a new tier in FL. To ensure smooth integration of D2D communication into FL,

novel algorithms are proposed for FL-oriented D2D grouping, MUE selection, and also D2D exit. During the D2D learning group formation, we solve the MUE selection problem by a multiple objective optimization approach with considering the tradeoff between D2D link quality (in terms of effective transmit power) and the expended compensation cost.

The rest of the paper is organized as follows. In Section II, the proposed D2D-assisted FL scheme is described. Section III analyzes the convergence of the proposed FL scheme. Simulation results are provided in Section IV while Section V concludes the paper.

II. D2D-ASSISTED HIERARCHICAL FEDERATED LEARNING SYSTEM

In this section, we present the proposed D2D-assisted hierarchical FL system, called D2D-FedAvg, which integrates D2D with FL. As illustrated in Fig.1, the proposed system consists of a two-tier learning process. The first tier is the D2D learning group that transfers the learning parameters over D2D links. In the D2D learning group, the MUE assumes the role of the parameter server for modeling the aggregation of SUEs in the group. Another tier in the D2D-FedAvg system is the MEC server (edge server) learning group. The MEC server learning group takes charge of the whole learning process. The MEC server is usually deployed near the base station. In this tier, the MEC server is responsible for model aggregation of all the MUEs and also the independent UEs (called CUEs) who do not belong to any D2D groups in the cellular network. During model aggregation, the model parameters are communicated over the cellular network links.

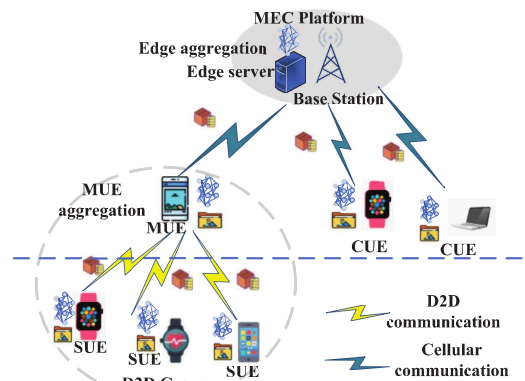


Fig. 1. D2D-assisted hierarchical federated learning system.

A. Problem Formulation for D2D-FedAvg

We focus on supervised FL in D2D-FedAvg. We define the training dataset as $\{x_j, y_j\}_{j=1}^{|D|}$, where the total number of training samples is $|D|$, the vector x_j is the j -th input of the D2D-FedAvg model, and the scalar y_j is the corresponding label. Furthermore, the loss function of the j -th data sample that is defined as $f(x_j, y_j, w)$, is abbreviated further as $f_j(w)$. This loss function captures the error of the model for the j -th data sample. The loss function $f_j(w)$ depends on the FL model which can be convex, e.g., logistic regression, or

non-convex, e.g., convolutional neural networks. Now assume that the D2D-FedAvg system includes N_C CUEs and N_{D2D} D2D groups. The CUEs are with local datasets $\{D_i\}_{i=1}^{N_C}$ and $\sum_{i=1}^{N_C} |D_i| = |D_C|$, where $|D_C|$ denotes the complete dataset for CUEs who participate in FL. Similarly, for the n -th D2D group in the FL system, there are N_D^n UEs with local datasets $\{D_i\}_{i=1}^{N_D^n}$, and $\sum_{i=1}^{N_D^n} |D_i| = |D_{D2D}^n|$, where $|D_{D2D}^n|$ denotes the overall dataset for n -th D2D group in the FL system.

For the i -th CUE in the FL system, the local training process requires the minimizing the empirical loss function $F_i(w)$ based on the training dataset as,

$$F_i(w) = \frac{\sum_{j \in D_i} f_j(w)}{|D_i|} \quad (1)$$

Similarly for the n th D2D group, the model training process is formulated as the minimization of the local loss function $F_{D2D}^n(w)$,

$$F_{D2D}^n(w) = \frac{\sum_{i=1}^{N_D^n} |D_i| F_i(w)}{|D_{D2D}^n|} \quad (2)$$

After local training, global training at the MEC server tier is performed based on the following function

$$F(w) = \frac{\sum_{i=1}^{N_C+N_D} \{|D_C| F_C(w) + |D_{D2D}| F_{D2D}(w)\}}{|D_C| + |D_{D2D}|} \quad (3)$$

$$F(w) = \frac{\sum_{i=1}^{N_C} |D_i| F_i(w) + \sum_{n=1}^{N_{D2D}} |D_{D2D}^n| F_{D2D}^n(w)}{|D_C| + \sum_{n=1}^{N_{D2D}} |D_{D2D}^n|} \quad (4)$$

where $F(w)$ is the global loss function in D2D-FedAvg system. Finally, the learning problem for D2D-FedAvg is to minimize $F(w)$,

$$w^* = \arg \min F(w) \quad (5)$$

Note that there is an inherent complexity of most machine learning. It is easy to see that we cannot find a closed-form solution to (5). Consequently, (5) is solved by gradient descent [20].

B. D2D-assisted Learning Process

For our D2D-FedAvg system, we adopt the adaptive process model [13] to reduce the communication rounds necessary for updating the model. We also consider the CUE as a D2D group that contains only one UE and thus there are totally $L = N_{D2D} + N_C$ D2D groups with the disjoint UE sets $\{U^l\}_{l=1}^L$, where U^l denotes the set of UEs in the l -th group. By including the N_C CUEs, there are totally $N = \sum_{n=1}^{N_{D2D}} N_D^n + N_C$ UEs in the system. The N UEs are indexed by the subscript i and superscript l , with distributed datasets $\{D_i^l\}_{i=1}^N$.

For our learning model $w_i^l(k)$ denotes the local model parameters in the UE after the k -th local update, where k

is the index for the update step. Thus, for the i -th UE in the system, the update rule is

$$w_i^l = \begin{cases} w_i^l(k-1) - \eta \nabla F_i^l(w_i^l(k-1)), & k/k_1 \neq 0 \\ \frac{\sum_{i \in U^l} |D_i^l| [w_i^l(k-1) - \eta \nabla F_i^l(w_i^l(k-1))]}{|D^l|}, & k/k_1 = 0 \\ \frac{\sum_{i=1}^N |D_i^l| [w_i^l(k-1) - \eta \nabla F_i^l(w_i^l(k-1))]}{|D|}, & \text{and } k/(k_1 k_2) \neq 0 \\ \frac{\sum_{i=1}^N |D_i^l| [w_i^l(k-1) - \eta \nabla F_i^l(w_i^l(k-1))]}{|D|}, & k/(k_1 k_2) = 0 \end{cases} \quad (6)$$

where η is the learning rate. Every k_1 local updates at the UEs, each MUE aggregates the models of the SUEs, and every k_2 MUE aggregations, the MEC server aggregates all the models from the MUEs. The number of total rounds of the local updates for each UE in one learning iteration is denoted as T , which is an integer multiple of $k_1 k_2$.

In this paper, we assume that the D2D communication utilizes the frequency division multiple access protocol to avoid mutual interference between different UEs. For the proposed D2D-FedAvg scheme, the key problem is how to coordinate the D2D communication with the model updating procedures of FL and make them work harmoniously. To achieve this, we designed the following procedures: global model configuration, D2D learning group formation, MUE model aggregation, and edge server model aggregation.

Step 1: Global model configuration. During the initialization step of D2D-FedAvg or after the edge server completes the global aggregation operation, the edge server broadcasts the global model weights and training programs to the UEs in the learning system. In particular, each UE has different data resources, device battery, computing capabilities, and wireless channel conditions. If these heterogeneities among devices are not considered, the entire training process will be inefficient. To alleviate the affection of the heterogeneous client devices on D2D-FedAvg performance, we use the FedCS [11] protocol to select client devices that participate in learning.

Step 2: D2D learning group formation. Before initiating D2D-FedAvg, the D2D learning group is formed as follows.

a) D2D grouping. The Proximity Service function is used to determine whether a UE is authorized to act as a D2D-UE. For better characterization, an available discovery list for each UE is created. Each row in the discovery list contains the groupid and the role of the UE with 1 indicating the UE be a MUE, 0 indicating a SUE, and -1 indicating a CUE. Assume that all UEs are initially qualified as D2D-UE, and the maximum distance between MUE and SUE creating one D2D pair is d_{max} . Taking the coordinate of the edge server as the origin in world space, the edge server records and updates the coordinates of the UEs.

Since link quality is mainly affected by the communication distance, the Euclidean distance between two UEs is utilized as the metric for D2D grouping. At the start, each UE forms a group. After that, if the distance between two UEs is $d \leq d_{max}$, they form a new group. In this process, some UEs may belong to more than one group. We compute the distance between two UEs that are furthest apart in the group,

and the middle position of these two UEs is defined as the center position of the group. When the distance d between two groups is larger than $2d_{max}$, the UE is assigned to the group closest to it. When d is no more than $2d_{max}$, the two groups are merged to form a new group. In the new group, for the two UEs whose distance is larger than d_{max} , they are removed from the group and enter the next round of grouping. During this process, the edge server records the number of UEs for each group, and updates the groupid in the discovery list. Once all UEs have formed the groups and each UE has a unique groupid, the edge server takes the UE as a CUE for a group with only one UE. For the l -th group with more than one UE, the edge server also records the distance $d_{i,j}^l$ between i -th UE and j -th UE in the group.

b) MUE selection. Since the MUE plays an important role of linking the edge server and the SUE for the learning process, it has to be selected carefully. The MUE will consume more energy for training the model in D2D group. To encourage a UE who wants to be a MUE, there is an incentive mechanism where the MEC service provider pays the MUE a compensation. The compensation cost is determined in terms of the MUE's contribution to the reduction of the communication resources required for learning in the cellular network. Therefore, it is particularly important to select a MUE with high effective transmit power but little compensation cost in the overall D2D group.

The FL training process involves a large number of parameter interactions. Moreover, when the D2D communication broadcasts messages over wireless channels, D2D communication is vulnerable to diverse attacks due to the broadcast nature of wireless communication [21]. For example, an attacker can be much easier to gain critical or private information by secretly listening to the broadcast communication frequently among devices. Thus, we adopt the way of unicast D2D communication for FL in this paper. For the i -th UE u_i^l in the l -th D2D group, its transmit power towards j -th UE u_j^l is subject to a power control scheme and usually computed as

$$P_d(u_i^l, u_j^l) |dB = \min\{P_{\max} |dB, 10\lg(M_r) + P_0 |dB + \rho \cdot PL(d_{i,j}^l/1000) |dB + \Delta_{TF} + q(k)\} \quad (7)$$

where $P_{\max} |dB$ denotes the maximum power that one UE can transmit, and M_r denotes the number of resource blocks allocated to u_i^l . PL denotes the path loss components of the D2D link channel fading model [22], that is

$$PL(d) = 148.1 + 40\lg(d[km]). \quad (8)$$

where P_0 , ρ , Δ_{TF} and $q(k)$ are user-specific configuration parameters, indicating the spectral power density, the path-loss compensation factor, a UE-specific parameter depending on the applied Modulation Coding Scheme (MCS), and a higher-layer closed-loop command to increase/decrease power level respectively.

Generally there is an attenuation of the MUE transmit power over the link from MUE to SUE. Let us denote the total attenuation experienced by the signal from u_i^l to u_j^l

as $A(u_i^l, u_j^l) = PL(d_{i,j}^l/1000)$. Given the transmit power and the attenuation factor, the effective transmit power at MUE u_i^l (also the useful power received at SUE u_j^l) is $P(u_i^l, u_j^l) = \frac{P_d(u_i^l, u_j^l)}{A(u_i^l, u_j^l)}$. In the l -th D2D group, the minimal effective transmit power among all D2D links from MUE u_i^l to SUE u_j^l ($i \neq j$) is

$$P(u_i^l) = \min_{u_i^l, u_j^l \in U^l, i \neq j} P(u_i^l, u_j^l) \quad (9)$$

Usually, a larger $P(u_i^l)$ indicates that MUE u_i^l can work better for model training over D2D links in the l -th D2D group. However, different MUEs have different service capabilities depending on their positions in the D2D group. Consequently, different MUEs will demand different compensation costs for learning.

Assume the transmission rate for one link from u_i^l to e (e is the j -th UE u_j^l in the l -th D2D group or edge server) is

$$R(u_i^l, e) = w_d \log_2(1 + \gamma(u_i^l, e)), \quad (10)$$

$$\gamma(u_i^l, e) = |h(u_i^l, e)|^2 P(u_i^l, e) / (I(e) + N_d) \quad (11)$$

w_d is the channel bandwidth, N_d is the noise power. The value of $\gamma(u_i^l, e)$ depends on the specific link. $h(u_i^l, e)$ is the channel gain from u_i^l to e , $P(u_i^l, e)$ is the transmit power for u_i^l to e and $I(e)$ is the interference power received from e by affecting the transmission. The time for transmitting z bits via the link from u_i^l to e is $T'(z) = z/R(u_i^l, e)$. The compensation cost of the MUE is defined as a scaled square of the transmission time,

$$C(u_i^l, e) = c_d \cdot \{T'(z)\}^2 = \frac{z^2}{\varphi(u_i^l, e)} \quad (12)$$

where c_d is the compensation adjustment factor for u_i^l , and $\varphi(u_i^l, e) = (R(u_i^l, e))^2 / c_d$ is the type parameter. To estimate the compensation cost when u_i^l is taken as MUE, its contribution that is characterized by z needs to be estimated first.

Every k_1 local updates, the MUE performs the aggregation step. Once u_i^l is selected as a MUE, it does not upload parameters via the D2D link for the MUE aggregation process. Thus, there is no compensation cost during upload. When downloading parameters, if u_i^l is a SUE, it only needs to download the parameters through the D2D link. Similarly, if it is a MUE, it is not necessary to download the model parameters, but the aggregated parameters need to be transmitted to other SUEs in the group, which increases the cost of parameter distribution. Now let us assume that the required data for parameter transfer in a update or a download process between two UEs is b bits. For a group where u_i^l is the MUE, and the model training requires T/k_1 UE aggregations, the compensation cost for MUE aggregation is estimated as $(T/k_1) \sum_{j=1, j \neq i}^{|U^l|} C(u_i^l, u_j^l)$, where $|U^l|$ is the UE number of the l -th D2D group.

Every k_2 MUE aggregations, a global aggregation in the edge server is needed. Regardless of whether u_i^l is a MUE

or a SUE, u_i^l needs to download the parameters from the edge server. There is almost no additional cost during the download process. During the upload process, if u_i^l is a MUE, it needs to upload b bits worth of parameters. In one global aggregation, the model requires $T/(k_1 k_2)$ MUE aggregations and consequently the compensation cost for global aggregation is estimated as $(T/(k_1 k_2))C(u_i^l, s)$, where s is the edge server and $C(u_i^l, s)$ is obtained in terms of Eq.(12) by substituting the parameters related to e with those of edge server s .

Thus, the overall compensation cost for u_i^l as MUE in its group is

$$C(u_i^l) = (T/k_1) \sum_{j=1, j \neq i}^{|U^l|} C(u_i^l, u_j^l) + (T/(k_1 k_2))C(u_i^l, s) \quad (13)$$

From the above analysis, it can be seen that MUE selection is a multi-objective optimization problem that requires the highest effective transmit power for the group but the least compensation cost. For this class of problems, near-optimal solutions can be found within a bounded amount of time. To find a trade-off between the effective transmit power and the compensation cost for selecting the MUE, the multi-objective optimization problem can be converted into a single objective optimization problem by a weighting factor λ ,

$$\min_{u_i^l \in U^l} \left\{ \frac{\lambda}{P(u_i^l)} + (1 - \lambda)C(u_i^l) \right\} \quad (14)$$

where $\lambda \in [0, 1]$ is utilized to weight the effective transmit power and the compensation cost into the decision making at the same time. By solving (14) with a searching algorithm, the appropriate MUE can be selected in each D2D group.

c) D2D session establishment. When the MUE selection is completed, the D2D links among devices will be established. There are two ways for establishing the D2D communication session: one is the centralized way and another is the distributed way. The centralized mode is to establish a D2D communication session under the full control of the base station. Comparably, the distributed mode is that D2D users autonomously control the establishment, maintenance and release of communication. Since the centralized mode can give full play to the advantages of D2D communication, it is convenient to control and manage resources and interference. Therefore, this paper adopts the centralized control scheme. We set up the D2D connection according to the discovery list of each device, that is, the UE with the role 0 in the discovery list needs to be connected to the MUE with the role 1 under the same group.

d) UE exit scheme. The UEs may change their locations dynamically, and consequently the MUE may be replaced. We design a D2D-UE exit scheme. In each group, when the SUE exits, it needs to send an exit request to the MUE to which it belongs. After receiving the request, the MUE disconnects from the SUE and reassigns the role of UE. When the MUE exits, the MUE re-selection process is performed, and then reassign it the UE role by updating the D2D

discovery list. Since FL allows UE to exit the training process without significantly lowering the learning performance [11], MUE re-selection will not introduce additional performance overhead.

Step 3: MUE model aggregation. After receiving the local model parameters of all SUEs, the MUE averages the results. In wireless communication, D2D is susceptible to a number of security attacks so that some devices may become untrusted. Consequently, data sharing between D2D pairs puts the data privacy at risk. To protect the user data privacy, differential privacy protection is used for communication of the gradient.

At each step of the mini-batch SGD, a batch B of random examples is formed and the gradients for a group of examples M ($M = B$) are computed. After that, each gradient $g_k(x_j)$ is clipped by (15), where C is a clipping threshold.

$$\bar{g}_k(x_j) = \frac{g_k(x_j)}{\max(1, \frac{\|g_k(x_j)\|_2}{C})} \quad (15)$$

$$\tilde{g}_k = \frac{1}{M} \left(\sum_j^M \bar{g}_k(x_j) \right) + N(0, \sigma^2 C^2 I) \quad (16)$$

where $N(0, \sigma^2 C^2 I)$ is the normal (Gaussian) distribution with mean 0 and standard deviation σC . We choose $\sigma C = \sqrt{2 \log \frac{1.25}{\delta}} / \epsilon$ for (ϵ, δ) -differentially private by standard arguments [23] with the M samples. Specifically, the noise is added to the gradients of the M samples to realize differential privacy protection of the gradients, and the model parameter w is updated on the opposite direction of the averaged noisy gradient \tilde{g}_k .

Step 4: Edge server model aggregation. After receiving the model parameters uploaded by all MUEs and CUEs for the FL, the edge server creates a weighted average of all the model parameters to obtain the new global model parameters. During the model aggregation, differential privacy protection is also used for processing the gradient.

For easily understanding, the key model aggregation procedures of D2D-FedAvg are summarized in Algorithm 1.

III. CONVERGENCE ANALYSIS OF D2D-FEDAVG

The proposed D2D-FedAvg utilizes D2D networking to partly offload the communication overhead that is generated by FL in cellular networks. The D2D group forms a sub-group for FL. The D2D sub-group is essentially a UE clustering process. The model aggregation process in D2D-FedAvg is divided into two subsequent procedures: MUE aggregation in the D2D subgroup, and edge aggregation in the MEC server. This type of two-tier model aggregation structure is similar to a modular process that was used for robust FL in heterogeneous environment [24]. The convergence of the modular learning algorithm has been proven theoretically in [24]. Hence, the proposed D2D-FedAvg is also a FL scheme that converges. Simulation results in Section IV also verified this.

Algorithm 1 D2D-Assisted Federated Averaging

```

1: Initialize selected clients with parameter  $w_0$ , learning rate
    $\eta$ , noise scale  $\sigma$ , group size  $M$ , gradient norm bound  $C$ 
2: for  $k = 1, 2, \dots, T$  do
3:   Form the D2D learning group
4:   for each UE  $i = 1, 2, \dots, N$  in parallel do
5:     Add noise using Eq.(16)
6:     Descent  $w_i^l(k) \leftarrow w_i^l(k-1) - \eta \tilde{g}_{k-1}$ 
7:   end for
8:   if  $k$  is an integer multiple of  $k_1$  then
9:     if  $|U^l| \geq 2$  then
10:      Select MUE for each group using Eq.(14)
11:    end if
12:    Establish the communication session
13:    for each MUE  $l = 1, \dots, L$  in parallel do
14:       $w^l(k) \leftarrow$  perform MUE aggregation in Step 3
15:      if  $k$  is not an integer multiple of  $k_1 k_2$  then
16:        for each UE  $i \in U^l$  in parallel do
17:           $w_i^l(k) \leftarrow w_i^l(k)$ 
18:        end for
19:      end if
20:    end for
21:  end if
22:  if  $k$  is an integer multiple of  $k_1 k_2$  then
23:    if  $|U^l| \geq 2$  and UE exit then
24:      Re-select MUE for each group using Eq.(14)
25:    end if
26:     $w(k) \leftarrow$  perform edge aggregation in Step 4
27:    for each UE  $i = 1 \dots N$  in parallel do
28:       $w_i^l(k) \leftarrow w(k)$ 
29:    end for
30:  end if
31: end for

```

IV. SIMULATIONS

We simulated D2D-FedAvg using the network simulation platform NS-3 [25] and PySyft [26]. We implemented a generic interface between NS-3 and PySyft. D2D networking is simulated in NS-3 and with a callback of FL execution for model training in PySyft via the interface. For our evaluation, the typical Federated Averaging (FedAvg) [6] with adaptive model updates [13] is used as the reference scheme for comparison.

A. Simulation Setup

For our simulation topology we considered a MEC server which is located in the center of an area with a size of $1000\text{m} \times 1000\text{m}$. The D2D-FedAvg system was tested in this area with a number of 50 UEs that meet FedCS, including 7 MUE groups, that each consists of a number of UEs varying from 3 to 7, and also 12 CUEs. The simulation area was divided into seven sectors. For each sector, the positions of CUEs, SUEs and MUEs are randomly deployed with a uniform distribution. The specific parameters used for the D2D networking simulation are presented in Table I.

The simulations were verified on the FEMNIST [27] dataset. Every UE is assigned an equally sized random subset of the total training data. We used a simple feedforward neural network with ReLU units and softmax of 62 classes (corresponding to 62 digits) with negative log likelihood (NLL) loss and an optional MaxPool2d input layer. We

TABLE I
SIMULATION PARAMETERS OF D2D NETWORKING

Parameter	Value
Max.UE's transmit power P_{\max} [dB]	23dBm
Noise power Level	-174dBm/Hz
Edge Server's transmit power	43dBm
Maximum distance between MUE and SUE d_{max}	30m
Spectral power density P_0 [dB]	-70dBm
Path-loss compensation factor ρ	0.7
UE-specific parameter Δ_{TF}	0
Weighting factor λ	0.5

adopted the mini-batch SGD with a mini-batch size of 64, and an initial learning rate of 0.01. To avoid model overfitting, weight_decay was set to 0.0001.

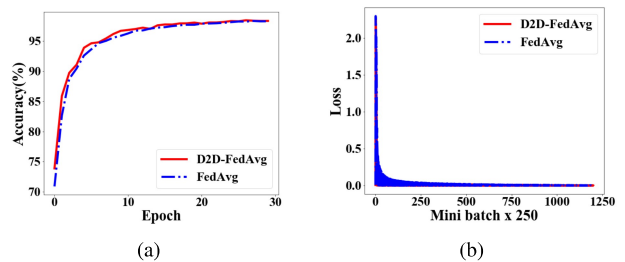


Fig. 2. (a) The accuracies of the D2D-FedAvg and FedAvg schemes. (b) The training loss for every 250 mini-batches per epoch.

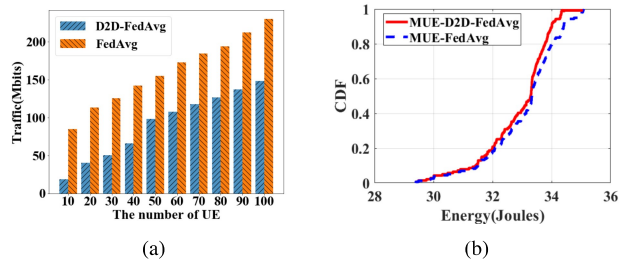


Fig. 3. (a) Communication overhead in MEC network for increasing UEs. (b) Energy consumption of MUEs in D2D-FedAvg and FedAvg systems.

B. Simulation Results

We first assess the convergence and accuracy of D2D-FedAvg. Fig.2(a) shows the learning accuracy of the proposed D2D-FedAvg compared to that of the traditional FedAvg. It can be seen from Fig.2(a) that the proposed D2D-FedAvg converges after 25 training epochs. It shows that at least 8 epochs of training are needed so that the D2D-FedAvg scheme achieves an accuracy above 95% and it can achieve accuracy 98.46% finally. Using the same number of training epochs, D2D-FedAvg achieves almost the same accuracy with FedAvg. This is also validated in Fig.2(b). Fig.2(b) shows that, similar to the FedAvg scheme, the proposed D2D-FedAvg scheme finally achieves a training loss close to 0.

The number of necessary communication rounds of the parameters in the MEC network for the two schemes are summarized in Table II. It can be seen from Table II that the communication overhead over the cellular link required by FedAvg is about 33 times that of D2D-FedAvg. Recall that in our simulation the maximum number of total UEs was set to 100. Fig.3(a) shows the learning communication traffic

in the cellular network for the two schemes. It can be seen from Fig.3(a) that the communication traffic of the proposed scheme is significantly lower than that of the FedAvg scheme. With an increasing number of UEs (e.g., beyond 50 in our case), the gap of cellular link traffic between the two schemes is gradually increasing. This is because the number of D2D UEs is also dynamically increased along with the increase of the total number of UEs.

TABLE II
COMMUNICATION OVERHEAD

Scheme	Accuracy (%)	Epoch	Communication rounds
D2D-FedAvg	95.92	8	17920
	98.46	30	252002
FedAvg	95.60	8	582406
	98.16	30	8190032

Besides the communication overhead, we also investigated another critical quantity in training process, namely, the energy consumption of MUEs. We recorded the energy consumption of MUEs in 100 simulation rounds. Fig.3(b) shows the experimental cumulative distribution function curves (CDF) of the energy consumption of the MUEs (UEs in FedAvg) in the D2D-FedAvg and FedAvg systems. It can be seen from Fig.3(b) that, approximately 60% of MUEs consume energy lower than 33.5 Joules in the two systems. Though the MUE in D2D-FedAvg consumes more energy than in FedAvg, the increased energy consumption for each MUE is less than 2 Joules in D2D-FedAvg when compared to FedAvg, which is usually acceptable in FL applications. The MUE can give up its master role at any time, and when the energy consumption is larger than its expectation, it will select to quit the role of MUE. Thus on average, the increment of energy consumption of MUE in D2D-FedAvg is not very much.

V. CONCLUSION

In this paper, we proposed a D2D-assisted Federated Learning (FL) scheme over MEC networks for edge intelligent applications. The proposed scheme exploits the benefits of D2D communication in FL for reducing the communication cost in the cellular network. D2D communication was embedded into a FL system by introducing new algorithms for D2D grouping, MUE selection, and D2D exit. The proposed scheme allows for low-cost integration of FL into mobile edge networks for a plethora of applications. The proposed D2D-FedAvg system was implemented over a MEC network and the simulation results showed that the proposed learning scheme can reduce significantly the traffic load while keeping the same model accuracy with the traditional FL scheme.

REFERENCES

[1] Ericsson. Ericsson mobility report. 2019. <https://www.ericsson.com/en/mobility-report/reports>.
[2] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1628-1656, thirdquarter 2017.
[3] D. Xu, T. Li, Y. Li, et al., "Edge Intelligence: Architectures, Challenges, and Applications," *ArXiv Preprint ArXiv:2003.12172*, 2020.

[4] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proc. of the IEEE*, vol. 107, no. 8, pp. 1738-1762, Aug. 2019.
[5] J. Park, S. Samarakoon, M. Bennis and M. Debbah, "Wireless Network Intelligence at the Edge," *Proc. of the IEEE*, vol. 107, no. 11, pp. 2204-2239, Nov. 2019.
[6] H. B. McMahan, E. Moore, D. Ramage, B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, abs/1602.05629, 2016.
[7] R. Fantacci and B. Picano, "Federated learning framework for mobile edge computing networks," *CAAI Trans. Intell. Tech.*, vol. 5, no. 1, pp. 15-21, March 2020.
[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data", *Proc. 20th Int. Conf. Artif. Intell. Stat.*, vol. 54, pp. 1273-1282, Apr. 2017.
[9] W. Y. B. Lim et al., "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 2031-2063, thirdquarter 2020.
[10] L. Wang, W. Wang and B. Li, "CMFL: Mitigating Communication Overhead for Federated Learning," 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 954-964.
[11] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," 2019 IEEE Int. Conf. Commun., pp. 1-7.
[12] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," *ArXiv Preprint ArXiv:1610.05492*, 2018.
[13] S. Wang, T. Tuor, T. Salonidis, et al., "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205-1221, June 2019.
[14] J. Qian, S. Sengupta, L. Hansen, "Active Learning Solution on Distributed Edge Computing," *ArXiv Preprint ArXiv:1906.10718*, 2019.
[15] L. Liu, J. Zhang, S. H. Song, K. B. Letaief, "Client-edge-cloud hierarchical federated learning," 2020 IEEE Int. Conf. Commun., pp. 1-6.
[16] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155-2169, 2020.
[17] H. Xing, O. Simeone and S. Bi, "Decentralized Federated Learning via SGD over Wireless D2D Networks," *Proc. IEEE 21st Int. Workshop on Signal Processing Advances in Wireless Comm. (SPAWC)*, 2020, pp. 1-5.
[18] S. Hosseinalipour, et al., "Multi-stage hybrid federated learning over large-scale wireless fog networks," *arXiv preprint arXiv:2007.09511*, 2020.
[19] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive D2D collaboration for energy-efficient mobile edge computing," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 64-71, Aug. 2017.
[20] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," in *Proc. of COMPSTAT*. Springer, 2010, pp. 177-186.
[21] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma and J. Ott, "Security and Privacy in Device-to-Device (D2D) Communication: A Review," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 2, pp. 1054-1079, Secondquarter 2017.
[22] Y. Dai, M. Sheng, K. Zhao, L. Liu, J. Liu and J. Li, "Interference-aware resource allocation for D2D underlaid cellular network using SCMA: A hypergraph approach," 2016 IEEE Wireless Commun. and Netw. Conf., pp. 1-6.
[23] C. Dwork, A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211-407, 2014.
[24] G. Avishek, H. Justin, Y. Dong, and R. Kannan, "Robust Federated Learning in a Heterogeneous Environment," *ArXiv preprint ArXiv:1906.06629*, 2019.
[25] R. Richard, F. J. Cintrón, A. B. Mosbah, S. Gamboa, "Implementation and Validation of an LTE D2D Model for Ns-3," in *Proc. of the Workshop on Ns-3*, 2017, pp. 55-62.
[26] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *CoRR*, abs/1811.04017, 2018.
[27] S. Caldas, P. Wu, T. Li, H. B. McMahan, "Leaf: A benchmark for federated settings," *ArXiv preprint ArXiv:1812.01097*, 2018.