Energy-Aware, Device-to-Device Assisted Federated Learning in Edge Computing

Yuchen Li, Weifa Liang, *Senior Member, IEEE*, Jing Li, Xiuzhen Cheng, *Fellow, IEEE*, Dongxiao Yu, *Senior Member, IEEE*, Albert Y. Zomaya, *Fellow, IEEE*, and Song Guo, *Fellow, IEEE*

***Abstract*— The surging of deep learning brings new vigor and vitality to shape the prospect of intelligent Internet of Things (IoT), and the rise of edge intelligence enables provisioning real-time deep neural network (DNN) inference services for mobile users. To perform efficient and effective DNN model training in edge computing environments while preserving training data security and privacy of IoT devices, federated learning has been envisioned as an ideal learning paradigm for this purpose. In this article, we study energy-aware DNN model training in edge computing. We first formulate a novel energy-aware, Device-to-Device (D2D) assisted federated learning problem with the aim to minimize the global loss of a training DNN model, subject to bandwidth capacity on an edge server and energy capacity on each IoT device. We then devise a near-optimal learning algorithm for the problem when the training data follows the i.i.d. data distribution. The crux of the proposed algorithm is to explore using the energy of neighboring devices of each device for its local model uploading, by reducing the problem to a series of weighted maximum matching problems in corresponding auxiliary graphs. We also consider the problem without the assumption of the i.i.d. data distribution, for which we propose an efficient heuristic algorithm. We finally evaluate the performance of the proposed algorithms through experimental simulations. Experimental results show that the proposed algorithms are promising.**

***Index Terms*—Edge computing, energy-aware federated learning, D2D-assisted learning, weighted maximum matching, budgeted-energy DNN model training, constrained optimization, decentralized machine learning.**

Manuscript received 27 August 2022; revised 30 April 2023; accepted 15 May 2023. Date of publication 18 May 2023; date of current version 5 June 2023. The work of Yuchen Li and Weifa Liang was supported by the Australian Research Council through Discovery Project Scheme under Grant DP200101985. The work of Weifa Liang was also supported by the City University of Hong Kong under Grant 9380137/CS. The work of Jing Li and Song Guo was supported in part by Hong Kong RGC Research Impact Fund (RIF) under Grant R5060-19, in part by General Research Fund (GRF) under Grants 152221/19E and 15220320/20E, in part by the National Natural Science Foundation of China under Grant 61872310, and in part by Shenzhen Science and Technology Innovation Commission under Grant R2020A045. Recommended for acceptance by Y. Yang. *(Corresponding author: Weifa Liang.)*

Yuchen Li is with the School of Computing, Australian National University, Canberra, ACT 2601, Australia (e-mail: yuchen.li@anu.edu.au).

Weifa Liang is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: weifa.liang@cityu.edu.hk).

Jing Li and Song Guo are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong (e-mail: jing.li5@anu.edu.au; song.guo@polyu.edu.hk).

Xiuzhen Cheng and Dongxiao Yu are with the School of Computer Science and Technology, Shandong University, Qingdao 266510, China (e-mail: xzcheng@sdu.edu.cn; dxyu@sdu.edu.cn).

Albert Y. Zomaya is with the School of Computer Science, University of Sydney, Sydney, NSW 2006, Australia (e-mail: albert.zomaya@sydney.edu.au).

Digital Object Identifier 10.1109/TPDS.2023.3277423

I. Introduction

THE explosive increase in the number of cutting-edge mobile and Internet of Things (IoT) devices results in the phenomenal growth of the data volume generated at the edge of networks [14]. The collected data is a double-edged sword, on one hand, the data is invaluable to many companies and government organizations for taking business advantages through business activity decision-making and government policy-making. On the other hand, the majority of data is privacy-sensitive, it is a high risk to hand the data to a cloud platform for processing and analysis.

Deep learning is a technique that utilizes deep neural networks to learn patterns of a group of data. The rapid development of deep learning has led to desperate demands on a large volume of data for more accurate predictions. Traditional training of deep neural networks needs users to upload their data to a centralized server, which has a high risk of privacy violation and lots of communication bandwidth consumption. Federated learning (FL), as a promising framework, provides a solution to utilizing the data on training while protecting privacy. In Federated Learning, Smart devices train DNN models locally, using their local data. The trained local models are periodically sent to a server for aggregation, and the aggregated global model is then sent back to the devices for further training. In FL paradigm, users only upload their local models rather than their collected data to the server, which protects the privacy of users.

As devices usually are powered by limited batteries, performing federated learning consumes lots of energy. A typical way is to train the model on an edge server, and each device uploads its local model to the server for aggregation, and finally a global model is formed in the edge server. Thus, mobile edge computing, which brings computing resource to the edge of the core network so that the data generated by mobile devices can be processed at the edge to reduce the burden of the back-haul network, has emerged as a promising paradigm for federated learning in the era of the Internet of Things [22], [34], [36]. However, devices with longer distances from the edge server need to use larger transmission powers to upload their local models, thereby consuming much more energy than those of devices with shorter distances. Also, training a large-scale DNN model needs a large volume of data transmissions between devices and the edge server, this introduces a heavy communication overhead [10]. Due to limited energy imposed on devices, not every device has sufficient energy to train its local model on its dataset. Instead, only a subset of the dataset is used for

local model training due to the energy budget of the device for training. Consequently, the solution accuracy and convergence of federated learning training might degrade because not all data are used for the model training.

There are extensive studies on the federated learning in edge computing environments. Some of them focused on limited network and computation resource allocations [8], [24], [28], [32], while others concentrated on client choices [4], [21], [35]. There were several investigations on the learning parameters of federated learning, including the frequency of local updates and global aggregations [28], [38]. Unlike the above studies, in this paper we introduce a device-to-device (D2D) assisted uploading concept to alleviate the communication overhead on uploading local models from devices to the edge server. That is, devices in the edge environments are paired according to their distances and energy availability. For each pair of devices, the one with less energy budget sends its trained local model to another with more energy budget, and the received device then aggregates the model with its local model and uploads the aggregated model to the edge server. Since the wireless bandwidth capacity at the edge server is limited, such a model aggregation and uploading can reduce the volume of data transmitted to the edge server. Performing energy-aware federated learning in edge environments thus poses several challenges.

First, by allocating more energy on computation, a device can train its local model on a large volume portion of its collected dataset, but this leaves the device less energy on its local model uploading or vice versa, how to strive for a non-trivial trade off of energy allocations between its local model training and local model transmission/uploading?

Second, due to the heterogeneity of computing power and volume of collected data, different devices have different amounts of energy budgets at different rounds, the selection of device pairings heavily impacts not only the transmission energy consumption of devices but also the amount of data for local model training, thereby affecting the accuracy and convergence of federated learning, how to pair devices such that the energy consumption is minimized while the convergence can be guaranteed is challenging.

Finally, the wireless bandwidth capacity of the edge server usually is bounded, which can support only up to K devices rather than all devices to upload their local models to the server simultaneously. As the contributions towards the global model of different devices are different, some devices are more important than others. Then, how to identify top contribution devices to upload their local models to the edge server? In the rest of this paper we will tackle the aforementioned challenges.

The novelty of this paper lies in that energy-aware device-to-device assisted federated learning in edge computing is considered, through fully making use of an energy-sufficient neighbor device of each device for its local model uploading. A novel energy-aware, D2D-assisted federated learning problem is formulated, and a near-optimal algorithm for the problem is devised when the training data distribution meets certain assumptions.

The main contributions of the paper are given as follows.

- We formulate a novel energy-aware, D2D-assisted federated learning problem in an edge computing environment with the aim to minimize the expected loss of a DNN model training, subject to the wireless bandwidth capacity on an edge server, and energy capacities and transmission ranges on devices.
- We devise a near-optimal learning algorithm for a special case of the problem where the training data follows the i.i.d. data distribution. The crux of the proposed algorithm is a non-trivial reduction to reduce the problem to a series of weighted maximum matching in corresponding auxiliary graphs.
- We propose an efficient heuristic algorithm for the problem without the i.i.d. data distribution assumption, by adopting the similar reduction technique as we did for the special case. However, the weights assigned to edges in the auxiliary graphs are positively related to the priorities of devices, where a device has a higher priority if the gradient-descent rate on its local model is faster, compared with the other devices.
- We evaluate the performance of the proposed algorithms through experimental studies. Experimental results demonstrate that the proposed algorithms are promising, and the solutions delivered improve by 15.4% of the loss function in comparison with benchmark solutions.

The rest of the paper is organized as follows. Section II reviews the state-of-the-date on federated learning in edge computing environments. Section III introduces the system model, notions and notations, and defines the problem formally. Section IV devises a near-optimal learning algorithm for a special case of the problem. Section V proposes an efficient algorithm for the problem without the training data distribution assumption. Section VI evaluates the proposed algorithms empirically, and Section VII concludes the paper.

II. Related Work

Federated Learning in edge computing environments has been extensively investigated recently. Most studies investigated energy consumption on devices and edge servers, others dealt with the non-trivial trades off between the communication cost, the accuracy of solutions, and the convergence speeds of different learning algorithms. For example, Wang et al. [28] focused on the trade-off between the communication cost and the convergence performance, they provided an upper bound of FL convergence that later is used to develop an approximation algorithm for the FL problem, with the aim to minimize the loss function under resource capacity constraints. Dinh et al. [8] concentrated on the trade-off between the energy consumption and the model convergence. They proposed a new FL algorithm (FEDL) with the assumption of strongly convex and smooth loss functions. The crux of their algorithm is a new local surrogate function for each device to train its local model approximately up to a local accuracy level. Sun et al. [24] considered a long-term energy-aware dynamic edge server scheduling problem. They developed an online scheduling algorithm based on the Lyapunov optimization, with the aim to maximize the average number of edge server participation in training at each training round. Tu et al. [27] studied federated learning in a fog computing environment, where devices are allowed to offload their local data to other devices for processing and discard

some of their data. They investigated the impact of data transfer between devices and data discarding on model training. Chen et al. [4] dealt with the optimization of user selections and network resource allocation in a wireless network for a set of users and a base station, with the aim to minimize the convergence time of federated learning. They developed a scheme to select users with high contributions to the convergence of the training, and formulated an integer linear programming for network resource allocation. Zhang et al. [38] considered a scenario where the data across different clients are non-independent and identically distributed (Non-IID), for which they proposed a deep reinforcement learning based method to decide the batch size of local training adaptively. Nishio and Yonetani [21] studied the federated learning within the deadline of the global aggregation. They proposed a federated learning protocol to select as many participants as possible from a set of heterogeneous devices with limited resources. Wu et al. [32] dealt with reducing the traffic volume of WAN transmissions. They proposed a hierarchical federated learning paradigm, which performs synchronous federated learning aggregation between clients and edge servers and asynchronous aggregation between edge servers and cloud servers. They also devised algorithms to decide the federated learning controllable factors, staleness threshold client-edge association, and data distribution. Chen et al. [5] studied the serverless decentralized federated learning, where workers only communicate with their neighbors. They enabled the exchange of intermediate results instead of the entire model between the workers. They also proposed an efficient algorithm to trade off between the communication cost and training performance. Lu et al. [17] tackled a heterogeneity problem of local data, by exploiting a clustered FL framework and an auction-based client selection strategy with constrained local resources. Pilla [22] aimed to minimize the energy consumption of FL training on heterogeneous devices. The author proposed an optimal pseudo-polynomial solution to find the optimal workload distribution. Chen et al. [2] aimed to mitigate the communication overhead of transferring DNN models, and proposed a novel framework that identifies and freeze stable parameters of DNN models to improve the communication efficiency. Tang et al. [25] sparsified the DNN model in a decentralized federated learning to deal with the communication bottleneck, and devised an algorithm to find the an peer selection that efficiently utilizes bandwidth resources. Tao et al. [26] proposed a Byzantine-resilient distributed gradient descent algorithm for FL that can handle the heavy-tailed data and converge under the standard assumptions. They also developed another algorithm to further reduce the communication overhead on learning process, using the gradient compression technique, and theoretical analysis shows that the proposed algorithms can achieve the statistical error rate that is order-optimal. Xu et al. [33] investigated the problem of energy minimization for a single FL request with uncertain availability of UEs, by proposing a novel optimization framework. They also devised an online learning algorithm with a bounded regret for the UE selection, by considering various contexts (side information) that influence energy consumption. Xu et al. [34] proposed a hierarchical FL framework for joint worker aggregator placement and UE assignment for a single FL request, and devised an approximation algorithm for it. They also considered the online worker aggregator placement and UE assignment for multiple FL request admissions with model uncertainty, by proposing a multi-armed bandit based online learning algorithm for it. Wu [31] studied on the federated learning client selection to overcome the negative impacts brought by low-quality data. They devised a novel algorithm to select clients over mixed quality and non-i.i.d. data to optimize the performance of federated learning.

The aforementioned studies mainly investigated resource allocation or client (user) selection for a set of devices that upload their local models to a server for aggregation, by striving for non-trivial trade-offs between energy consumption and accuracy of the solution obtained. However, none of these works considered using energy of neighbor devices of a device for its local model uploading. To the best of our knowledge, we are the first to study how to minimize the global loss of a model through D2D transmissions. The saved energy at each device can be used for local model training to further improve the performance of federated learning. It must be mentioned that this is an extension of a conference paper [16].

III. Preliminaries

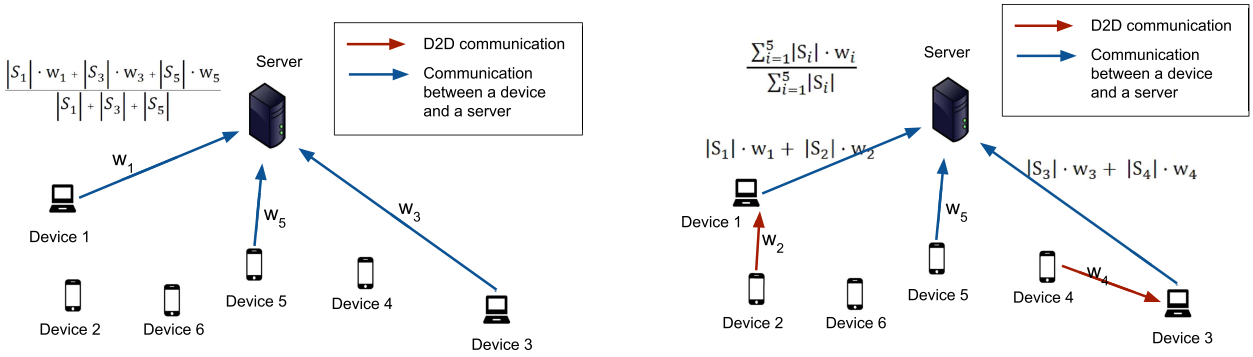
In this section, we first introduce the system model. We then introduce federated learning in edge computing, and energy metrics for device-assisted federated learning. We finally define the problem precisely.

A. System Model

We consider a set of devices $V = \{v_1, v_2, \ldots, v_{|V|}\}$ and an edge server s, assume that v_0 is the edge server s. Denote by $d_{i,j}$ the distance between devices v_i and v_j with $0 \leq i, j \leq |V|$. Assume that there is a DNN model deployed in the edge server s for training, and its training data comes from $|V|$ IoT devices. Each device $v_i \in V$ has a dataset $\mathcal{D}_i$ for the DNN model training. Denote by $\mathcal{D} = \cup_{i=1}^{|V|} \mathcal{D}_i$ the dataset of all devices. Each device v_i with energy capacity $\mathcal{E}_i$ has a set $\mathcal{P}$ of finite transmission power levels for communicating with other devices and the edge server, and such communication is conducted via D2D transmission, using one of its transmission power levels $p_i(t) \in \mathcal{P}$ at round t. We further assume that the federated learning process takes T training rounds, while at each round t, each device $v_i \in V$ performs its local model training for τ epochs, assuming that the energy budget $\mathcal{E}_i(t)$ $(\leq \mathcal{E}_i)$ of v_i at round t is given [3], where $1 \leq t \leq T$. Due to limited wireless bandwidth on edge server s, we assume that at most K devices can upload their local models to the server simultaneously at each round, where $1 \leq K \leq |V|$. An illustrative example of the system model is given in Fig. 1.

B. Federated Learning in Edge Computing

Let (x, y) be *one data point* in the dataset $\mathcal{D}$, where $x \in \mathbb{R}^d$ is the input features and y is the label of the data point. We aim to train a deep neural network (DNN) to minimize the error between the output of the neural network and label y under a



(a) Traditional federated learning

(b) D2D-assisted federated learning

Fig. 1. Comparisons of the uploading phase between the traditional federated learning and the energy-aware D2D-assisted federated learning in an edge computing environment at one round with $K = 3$.

given input x. Specifically, the error is defined by a loss function $l(w, x, y)$, which is the mean squared error or cross-entropy [38], and $w \in \mathbb{R}^d$ is the *model parameter* of the DNN model that is a d-dimensional real vector.

The loss function on a dataset $\mathcal{D}$ is defined as follows:

$$L(w \mid \mathcal{D}) = \frac{\sum_{(x,y)\in\mathcal{D}} l(w, x, y)}{|\mathcal{D}|}. \tag{1}$$

The training objective is to find an optimal model parameter w^* to minimize the value of $L(w \mid \mathcal{D})$. Since it is difficult to find the closed-form solution of w^*, the training of neural networks often applies the gradient descent method to find a parameter w to approximate w^* as much as possible.

To protect the privacy of users, instead of training the DNN model by uploading the full dataset from all devices to the server, devices will train the deep neural network on their local datasets. In the beginning of each round t with $1 \le t \le T$, server s distributes the global model parameter $w(t-1)$ obtained at round $t-1$ to all devices, assuming that $w(0)$ is randomly initialized.

Due to limited wireless bandwidth capacity B of edge server s in the defined system model, it is impossible to allow all devices to upload their models to the server at the same time. Instead, a subset of devices $V_{fed}^t \subset V$ is chosen to participate in model training at each round t. Among the participating devices, up to K of the devices are allowed to upload their training models to the server directly without violating its bandwidth capacity constraint, where $K \ge 1$ is a positive integer. Specifically, each device $v_i \in V_{fed}^t$ uniformly samples a subset $\mathcal{S}_i(t)$ of its dataset $\mathcal{D}_i$ for local model training under its energy budget $\mathcal{E}_i(t)$ for round t, and then transmits its trained local model to its partner that is also in V_{fed}^t or vice versa. Its partner finally aggregates its local model with the received model and uploads the aggregated model to server s. Within each round t, we further assume that each device v_i applies τ gradient descent steps to train its local model $w_i(t)$, and we refer to each step of gradient descent of local training as *one epoch*. Denote by $w_i^k(t)$ the local model parameter of device v_i after the kth local training epoch with $k = 1, 2, \ldots, \tau$. Then, $w_i^0(t) = w(t-1)$ is the global model parameter distributed by server s after finishing local model training at round $t-1$. During each epoch k of round t with $1 \le k \le \tau$, device v_i updates its local model as follows.

$$w_i^k(t) = w_i^{k-1}(t) - \eta \cdot \nabla L_i(w_i^{k-1}(t) \mid \mathcal{S}_i(t)), \tag{2}$$

and

$$\nabla L_i(w_i^{k-1}(t) \mid \mathcal{S}_i(t)) = \frac{\sum_{(x,y)\in\mathcal{S}_i(t)} \nabla l(w_i^{k-1}(t), x, y)}{|\mathcal{S}_i(t)|}, \tag{3}$$

where η is the learning rate, and $\nabla l(w_i^{k-1}(t), x, y)$ is the gradient of the loss function $l(w, x, y)$ with respect to $w_i^{k-1}(t)$ on each data point $(x, y) \in \mathcal{S}_i(t)$ at epoch $(k-1)$ within round t.

Having τ-epoch local training at round t, some device $v_i \in V_{fed}^t$ uploads its trained (or aggregated) local model $w_i(t)$ $(= w_i^\tau(t))$ to server s. The uploaded local models from devices in V_{fed}^t are then aggregated at server s as follows [28].

$$w(t) = \frac{\sum_{v_i\in V_{fed}^t} |\mathcal{S}_i(t)| \cdot w_i^\tau(t)}{\sum_{v_i\in V_{fed}^t} |\mathcal{S}_i(t)|}. \tag{4}$$

Server s finally distributes the aggregated global model $w(t)$ back to each device in V at the beginning of the next round $t+1$.

C. Energy-Budgeted D2D Assisted Uploading

Considering available energy heterogeneity on devices, some devices have less energy than others, due to more energy consumed on both uploading their local models to server s and their local model training. We here allow devices with sufficient energy to serve as relay nodes to help those devices with less energy to upload their local models to the server, thereby reducing the energy consumption of those less-energy devices. Meanwhile, the relay devices can aggregate the local models of relayed neighbors locally prior to uploading their aggregated local models to the server.

Specifically, each device $v_i \in V_{fed}^t$ has a destination device $\phi_{v_i}(t)$ at round t, which is either another device or server s. To avoid a long training delay, each device can only serve as either relay or relayed node at each round exclusively. The transmission

range $\theta_i(p_i(t))$ of device $v_i \in V$ at round t usually is determined by its transmission power level $p_i(t) \in \mathcal{P}$, a device v_j or server s can be the destination of device v_i only if it is within the transmission range of v_i, i.e., $d_{i,\phi_{v_i}(t)} \leq \theta_i(p_i(t))$. Denote by $\mathcal{C}_{v_i}(t)$ the set of nodes using v_i as their relays at training round t, we have $|\mathcal{C}_{v_i}(t)| \leq 1$. Similarly, $\mathcal{C}_s(t)$ is the set of nodes that can send their models to server s directly.

Since energy is the main constraint on devices, we assume that devices communicate with server s by adopting the Orthogonal Frequency-Division Multiplexing Access (OFDMA) mode. To avoid long delays on data transmission and aggregation and limited wireless bandwidth constraint on the edge server, we assume that at most K devices can send their local models to s with $1 \leq K \leq |V|$ at each round, that is,

$$|\mathcal{C}_s(t)| \leq K. \tag{5}$$

Having local training on its dataset $\mathcal{S}_i$ at round t, device $v_i \in V_{fed}^t$ then sends its local model $w_i^\tau(t)$ to its destination (a matching device of the device, which will be detailed later) $\phi_{v_i}(t)$ or server s. If its destination $\phi_{v_i}(t)$ is not to server s, device v_j $(= \phi_{v_i}(t))$ will aggregate its local model with its received local model from v_i if v_j, and transmits the aggregated result to server s to reduce transmission energy consumption. Denote by $w_i^g(t)$ the aggregated model uploaded by v_i. The aggregation at device v_i is

$$w_i^g(t) = |\mathcal{S}_i(t)| \cdot w_i^\tau(t) + \sum_{v_j \in \mathcal{C}_{v_i}(t)} |\mathcal{S}_j(t)| \cdot w_j^\tau(t),\ v_i \in V_{fed}^t \tag{6}$$

Recall that $\mathcal{S}_i(t)$ is a subset of dataset $\mathcal{D}_i$ of device v_i, $\mathcal{C}_{v_i}(t)$ is the set of devices that utilize v_i as their relay, and $\mathcal{C}_{v_i}(t)$ contains at most one device. If v_i participates in training at round t, $w_i^g(t)$ is the sum of sampled-data-size-weighted model parameters of v_i and the device in $\mathcal{C}_{v_i}(t)$ (if $\mathcal{C}_{v_i}(t)$ is not an empty set). The global model at server s after round t is updated as follows:

$$w(t) = \frac{\sum_{v_i \in \mathcal{C}_s(t)} w_i^g(t)}{\sum_{v_i \in V_{fed}^t} |\mathcal{S}_i(t)|}, \tag{7}$$

where the numerator of the right hand side of (7) is the weighted sum of model parameters of chosen devices in V_{fed}^t, while the denominator in the right hand side of (7) is the total number of sampled data points used in training. It can be seen that $w(t)$ in (7) is equivalent to it in (4).

Fig. 1 provides a comparison of the uploading phase between the traditional federated learning and the energy-aware D2D-assisted federated learning, where at most three devices can communicate with a server at each round. For convenience, in Fig. 1(b), the model training consists of two stages: the red arrows indicate the D2D communication in the first stage, while the blue arrows represent the model uploading communication between devices and the server in the second stage. The numbers on the arrows indicate the data volume of transmitted models. It can be seen from Fig. 1(a) that all (six) devices participate in local model training but only three of them can upload their trained local models to the edge server at each round, the trained results of the rest three devices are not uploaded, implying no contributions towards the global model. In contrast, it can be seen from Fig. 1(b) in the proposed solution that there are five devices participating in local model training and their trained local models are uploaded to the edge server through three of the five devices.

D. Energy Consumption of Devices

Let $h_{i,\phi_{v_i}(t)}$ be the channel gain between device v_i and device $\phi_{v_i}(t)$. The channel gain $h_{i,\phi_{v_i}(t)}$ [18] is

$$h_{i,\phi_{v_i}(t)} = \frac{\alpha}{d_{i,\phi_{v_i}(t)}^2}, \tag{8}$$

where α is the channel gain at the reference distance of 1 meter.

Denote by C the size of the DNN model w. By uploading its local model $w_i^\tau(t)$ to device $\phi_{v_i}(t)$ at round t, the amount of transmission energy consumed by device v_i is

$$Tran_{v_i}(t) = \frac{C \cdot p_i(t)}{B \cdot \log_2\left(1 + \frac{p_i(t) \cdot h_{i,\phi_{v_i}(t)}}{\sigma^2}\right)}, \tag{9}$$

where σ is the white Gaussian noise power, and B is the wireless bandwidth capacity of server s.

Let ψ_i be the energy consumption of calculating the gradient on one data point at device v_i. As shown in (2) and (3), device v_i calculates the gradient of each data point in set $\mathcal{S}_i(t)$ for τ epochs at round t, the total computing energy consumption of v_i on local model training at round t is

$$Comp_{v_i}(t) = \psi_i \cdot |\mathcal{S}_i(t)| \cdot \tau. \tag{10}$$

The total energy consumption of device v_i at round t should be no greater than its energy budget $\mathcal{E}_i(t)$ for that round, i.e.,

$$Tran_{v_i}(t) + Comp_{v_i}(t) \leq \mathcal{E}_i(t). \tag{11}$$

E. Problem Definition

Given a set of devices V and an edge server s, an integer K that is determined by the bandwidth of server s, they collaboratively perform federated learning to train a DNN model with a parameter vector (global model) w, each device $v_i \in V$ has a set $\mathcal{P}$ of transmission power levels. Each device v_i can sample a subset $\mathcal{S}_i(t)$ of its dataset $\mathcal{D}_i$ at each round t, and may upload its trained local model to server s directly or via another device, assuming that there are T rounds for the DNN model training. The *energy-aware D2D-assisted federated learning problem* in edge computing is to determine devices to participate in training at each round t, the number of sampled data points and the transmission power level of each chosen device, and the destination (paired) device of each chosen device to which to upload its local model at each round t with $1 \leq t \leq T$, such that the expected loss $\mathbb{E}[L(w(T) \mid \mathcal{D})]$ over the dataset $\mathcal{D}$ $(= \cup_{i=1}^{|V|} \mathcal{D}_i)$ is minimized, subject to the wireless bandwidth capacity B on server s, energy capacities and the maximum transmission ranges on devices.

The objective of the problem is to

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}[L(w(T) \mid \mathcal{D})], \\ \text{s.t.} \quad & (5)-(11), \end{aligned} \tag{12}$$

where $\mathbb{E}[L(w(T) \mid \mathcal{D})]$ is the expectation of $L(w \mid \mathcal{D})$ after T round iterations in (1).

IV. Algorithm for a Special Energy-Aware D2D-Assisted Federated Learning Problem

In this section, we consider a special case of the energy-aware D2D-assisted federated learning problem when $\tau = 1$ at each round t with $1 \leq t \leq T$. We devise a near-optimal learning algorithm for the problem, provided that the dataset $\mathcal{D}_i$ of each device $v_i \in V$ follows i.i.d. data distribution. We also analyze the convergence of the proposed algorithm and the quality of the solution obtained.

A. Overview of the Algorithm

Intuitively, if more data are used in the DNN model training, the trained DNN model can learn more accurate patterns on the dataset. We thus use as many sampled data points as possible to train the model with the aim to minimize the expected loss $\mathbb{E}[L(w(T) \mid D)]$ of the optimization objective in formula (12).

Following (10) and (11), when $\tau = 1$, the number $|\mathcal{S}_i(t)|$ of sampled data points that device v_i can use for its local model training at round t is upper bounded by

$$|\mathcal{S}_i(t)| \leq \frac{\mathcal{E}_i(t) - Tran_{v_i}(t)}{\psi_i}. \tag{13}$$

It can be seen from Inequality (13) that $|\mathcal{S}_i(t)|$ is bounded by the energy budget $\mathcal{E}_i(t)$ and the transmission energy consumption $Tran_{v_i}(t)$ of device v_i at round t. We then need to determine which devices to participate in model training at each round and the number of sample data points, the destination device, and the transmission power level of each chosen device.

According to Inequality (13), the decision at round t does not affect the number of sampled data points at any other round because the energy budget of each device for each round is given. Thus, the maximum number $|\cup_{v_i \in V_{fed}^t} \mathcal{S}_i(t)|$ of sampled data points used for local model training at each round t can be maximized, subject to energy budget $\mathcal{E}_i(t)$ for each $v_i \in V_{fed}^t$, where $1 \leq t \leq T$.

Since each device can help at most another device to relay its local model at each round, we can put devices in V into pairs so that the number of sampled data points for local model training at that round is maximized. To this end, we can reduce the problem to the maximum weight matching problem in an auxiliary graph. The detailed reduction is presented as follows.

B. Algorithm

From (9), the transmission energy consumption of a device increases with the increase on its transmission power level. To save energy of devices on transmissions, one device should select a minimum transmission power level in $\mathcal{P}$ such that its destination node is within the transmission range when it adopts the chosen transmission power.

For the sake of convenience, denote by $p_i^{min}(v_j)$ and $p_i^{min}(s)$ the minimum transmission power levels of v_i that v_j or s are within the transmission range of v_i. As device v_i has only limited power levels, $p_i^{min}(v_j)$ and $p_i^{min}(s)$ can be identified by binary search. Denote by $tran(v_i, v_j)$ and $tran(v_i, s)$ the amounts of energy consumed by transmitting the local model of v_i to device v_j or server s by adopting its minimum transmission powers, respectively, then

$$tran(v_i, v_j) = \frac{C \cdot p_i^{min}(v_j)}{B \cdot \log_2\left(1 + \frac{p_i^{min}(v_j) \cdot h_{i,v_j}}{\sigma^2}\right)} \tag{14}$$

and

$$tran(v_i, s) = \frac{C \cdot p_i^{min}(s)}{B \cdot \log_2\left(1 + \frac{p_i^{min}(s) \cdot h_{i,s}}{\sigma^2}\right)}, \tag{15}$$

where C is the size of the model w, and B is the bandwidth capacity of server s.

The proposed algorithm proceeds as follows. For each round t, a weighted auxiliary graph $G(t) = (U, E; w(\cdot, \cdot))$ is constructed, where $U = \{u_i, u_i' \mid v_i \in V\} \cup \{u_j^\nu \mid 1 \leq j \leq 2|V| - 2K\}$. The edge set E of $G(t)$ is defined as follows. (i) For each device $v_i \in V$, if $\mathcal{E}_i(t) > tran(v_i, s)$, an edge $e(u_i, u_i')$ between u_i and u_i' is added to E, and its weight $w(u_i, u_i')$ is the maximum number of sampled data points transferred if v_i directly sends its local model to server s, where $w(u_i, u_i') = \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, s)}{\psi_i} \rfloor$. Recall that the minimum transmission power of v_i is $tran(v_i, s)$ if v_i sends its local model to s, and the maximum amount of energy consumed for local model training is $\mathcal{E}_i(t) - tran(v_i, s)$.

(ii) For each pair of nodes $v_i \in V$ and $v_j \in V$ with $i \neq j$, denote by $S_{i,j}$ the maximum number of sampled data points from both devices and v_i is the relay node of v_j, we aim to find the maximum number of sampled data points if they both participate in model training at round t and one of them makes use of another as its relay vertex, then, $S_{i,j}$ is defined as follows:

$$S_{i,j} = \left\lfloor \frac{\mathcal{E}_i(t) - tran(v_i, v_j)}{\psi_i} \right\rfloor + \left\lfloor \frac{\mathcal{E}_j(t) - tran(v_j, s)}{\psi_j} \right\rfloor, \tag{16}$$

assuming that v_i is the relay device of v_j; otherwise, $S_{j,i}$ can be calculated similarly. If $\max\{S_{i,j}, S_{j,i}\} > 0$, we add an edge (u_i, u_j) between u_i and u_j with weight $w(u_i, u_j) = \max\{S_{i,j}, S_{j,i}\}$ to E, i.e., the maximum number of sampled data points of devices v_i and v_j is $w(u_i, u_j)$ when v_i is the relay node of v_j or vice versa.

(iii) For each pair of vertex u_i and dummy vertex u_j^ν, an edge $e(u_i, u_j^\nu)$ with an infinite weight $w(u_i, u_j^\nu)$ is added to E.

The edge set E of $G(t)$ can be partitioned into two disjoint subsets E_{dum} and E_{dev}, where E_{dum} consists of edges incident to at least one dummy vertex, i.e., $E_{dum} = \{e(u_i^\nu, u_j^\nu) \mid i \neq j,\ 1 \leq i, j \leq 2|V| - 2K\} \cup \{(u_i, u_j^\nu) \mid v_i \in V, 1 \leq j \leq 2|V| - 2K\}$, while E_{dev} contains only edges by device vertices, i.e., $E_{dev} = \{e(u_i, u_j) \mid v_i, v_j \in V, i \neq j\}$.

The construction of an auxiliary graph $G(t)$ is to find a set M of edges in $G(t)$ such that (i) M is a subset of E_{dev}; (ii) the cardinality $|M|$ of M is no larger than K; and (iii) M is a matching in $G(t)$ and the weighted sum of the edges in M is maximized. A set V_{fed}^t then can be derived from M, which is the set of endpoints of edges in M, i.e., $|V_{fed}^t| \leq 2\,K$.

Denote by $\mathcal{M}$ a weighted maximum matching in $G(t)$. Assume that M consists of only edges in $\mathcal{M}$ but not in E_{dum}, that is, $M = \mathcal{M} \backslash (E_{dum} \cap \mathcal{M})$. It can be seen that for device $v_i \in V$, the weight $w(u_i, u_i')$ of edge $e(u_i, u_i') \in \mathcal{M}$ represents the maximum number of sampled data points for its local model training if device v_i sends its local model to server s directly. For a pair of devices v_i and v_j, the weight $w(u_i, u_j)$ of edge $e(u_i, u_j) \in \mathcal{M}$ is the maximum number of sampled data points if both devices participate in the training at round t and one of them uses the other as its relay node to upload their aggregated local model to s.

Since $\mathcal{M}$ is a maximum weight matching in $G(t)$ that contains $4|V| - 2K$ vertices in total, there must have a matching edge in $\mathcal{M}$ incident to each of the $2|V| - 2K$ dummy vertices. Thus, there are $4|V| - 4K$ vertices in $\mathcal{M}$ with one endpoint being a dummy vertex.

Let V' be the set of the rest vertices in $G(t)$, clearly, $|V'| = 2K$. Let M' be the weighted maximum matching in the subgraph $G'(t) = (\{u_i, u_i' \mid v_i \in V'\}, E \cap \{(u_i, u_i') \cup (u_i, u_j)\} \mid v_i \in V', v_j \in V'\})$ of $G(t)$ induced by vertices in V'. Then, the cardinality of M' is no greater than K, i.e., $|M'| \leq K$, it can be seen that $M' \cup (\mathcal{M} \setminus M)$ is another matching of $G(t)$. As the weighted sum of edges in $\mathcal{M}$ is the maximum one, which equals the maximum number of sampled data points used for model training at round t, the weighted sum of edges in $M' \cup (\mathcal{M} \setminus M)$ must equal the weighted sum of edges in $\mathcal{M}$; otherwise, $\mathcal{M}$ is not the weighted maximum matching of $G(t)$. Thus, the set V_{fed}^t can be obtained from M', i.e., $|V_{fed}^t| \leq 2K$ as $|M'| \leq K$.

Let $\mathcal{M}$ be a weighted maximum matching in the auxiliary graph $G(t)$, by applying the weighted maximum matching algorithm in [11]. Let $V_{fed}^t = \emptyset$ initially, its construction is given as follows. For each edge $e(a, b) \in \mathcal{M}$, (i) if $a = u_i$ and $b = u_i'$, add device v_i to V_{fed}^t and set server s as the destination $\phi_{v_i}(t)$ of v_i; (ii) if $a = u_i$ and $b = u_j$ with $i \neq j$, both devices v_i and v_j are added to V_{fed}^t. Furthermore, if $S_{i,j} > S_{j,i}$, device v_j is the destination of v_i, v_i trains its local model and transmits its local model to v_j, v_j finally aggregates its local model and v_i's local model and uploads the aggregated model to server s; otherwise, device v_i is the destination of v_j, the similar operations can be performed, omitted; and (iii) if $a = u_i$ and $b = u_j^\nu$, then device v_i will not participate in training at round t.

Finally, each device $v_i \in V_{fed}^t$ sets its power level at $p_i^{min}(\phi_{v_i}(t))$, and samples $\lfloor \frac{\mathcal{E}_i(t) - Tran_{v_i}(t)}{\psi_i} \rfloor$ data points in set $\mathcal{D}_i$ for training at round t. The detailed algorithm for the special energy-aware D2D-assisted federated learning problem is shown in `Algorithm` 1.

We here use an example in Fig. 2 to illustrate how the proposed algorithm works at round t, where there are four devices ($|V| = 4$), and server s only allows two devices ($K = 2$) to upload its local model at each round. There are 4 ($=2|V| - 2K$) dummy vertices. As shown in Fig. 2, edges (u_1, u_2) and (u_2, u_3) represent device v_1 and v_3 can pair with v_2 to upload the trained local model, and edges (u_1, u_1'), (u_2, u_2') and (u_4, u_4') represent that device v_1, v_2 and v_4 can send their model to s alone. The red edges in Fig. 2 represent the edges in the weighted maximum matching $\mathcal{M}$. In Fig. 2, we

Algorithm 1: Algorithm for the Special Energy-Aware, D2D-Assisted Federated Learning Problem.

Input: A set of devices V, a server s, a local dataset $\mathcal{D}_i$ for each device $v_i \in V$, a set of transmission power level $\mathcal{P}$ and a DNN model w to be trained at each round t with $1 \leq t \leq T$.

Output: The set V_{fed}^t of devices participating in the training, the offloading destination $\phi_{v_i}(t)$, the set of sampled data points $\mathcal{S}_i(t)$, and the transmission power level $p_i(t)$ of each device v_i at each round t.

for $t \leftarrow 1$ to T **do**
  Initialize $G(t) \leftarrow (U, E)$ where $U = \emptyset$ and $E = \emptyset$;
  for $i \leftarrow 1$ to $|V|$ **do**
    $U \leftarrow U \cup \{u_i, u_i'\}$;
    Calculate $p_i^{min}(s)$ and $tran(v_i, s)$
    if $\mathcal{E}_i(t) > tran(v_i, s)$ **then**
      $E \leftarrow E \cup \{e(u_i, u_i')\}$;
      $w(u_i, u_i') \leftarrow \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, s)}{\psi_i} \rfloor$;
  for $i \leftarrow 1$ to $|V|$ **do**
    for $j \leftarrow i + 1$ to $|V|$ **do**
      Calculate $p_j^{min}(s)$, $tran(v_i, v_j)$, $tran(v_j, s)$, $p_i^{min}(s)$, $tran(v_j, v_i)$, and $tran(v_i, s)$;
      if $tran(v_i, v_j) < \mathcal{E}_i(t)$ and $tran(v_j, s) < \mathcal{E}_j(t)$ **then**
        $S_{i,j} \leftarrow \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, v_j)}{\psi_i} \rfloor + \lfloor \frac{\mathcal{E}_j(t) - tran(v_j, s)}{\psi_j} \rfloor$;
      else
        $S_{i,j} \leftarrow 0$;
      if $tran(v_j, v_i) < \mathcal{E}_j(t)$ and $tran(v_i, s) < \mathcal{E}_i(t)$ **then**
        $S_{j,i} \leftarrow \lfloor \frac{\mathcal{E}_j(t) - tran(v_j, v_i)}{\psi_j} \rfloor + \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, s)}{\psi_i} \rfloor$;
      else
        $S_{j,i} \leftarrow 0$;
      $E \leftarrow E \cup \{e(u_i, u_j)\}$;
      $w(u_i, u_j) \leftarrow \max\{S_{i,j}, S_{j,i}\}$;
  $U_{dum} \leftarrow \{u_j^\nu \mid 1 \leq j \leq 2|V| - 2K\}$
  $U \leftarrow U \cup U_{dum}$ /* U_{dum} is the set of dummy nodes */;
  for each dummy node $u_j^\nu \in U_{dum}$ **do**
    for each node $u_i \in U \backslash U_{dum}$ **do**
      $E \leftarrow E \cup \{e(u_i, u_j^\nu)\}$;
      $w(u_i, u_j^\nu) \leftarrow \infty$;
  Find a weighted maximum matching $\mathcal{M}$ in $G(t)$, by applying the algorithm in [11];
  for $i \leftarrow 1$ to $|V|$ **do**
    if $e(u_i, u_i') \in \mathcal{M}$ **then**
      $\phi_{v_i}(t) \leftarrow s$;
      $V_{fed}^t \leftarrow V_{fed}^t \cup \{v_i\}$;
      $|\mathcal{S}_i(t)| \leftarrow \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, s)}{\psi_i} \rfloor$;
    if $\exists u_j \in U, e(u_i, u_j) \in \mathcal{M}$ and $S_{i,j} \geq S_{j,i}$ **then**
      $\phi_{v_j}(t) \leftarrow s$; $\phi_{v_i}(t) \leftarrow v_j$;
      $V_{fed}^t \leftarrow V_{fed}^t \cup \{v_i, v_j\}$;
      $|\mathcal{S}_i(t)| \leftarrow \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, v_j)}{\psi_i} \rfloor$;
      $|\mathcal{S}_j(t)| \leftarrow \lfloor \frac{\mathcal{E}_j(t) - tran(v_j, s)}{\psi_j} \rfloor$;
  return V_{fed}^t, $\phi_{v_i}(t)$, $|\mathcal{S}_i(t)|$ for each device v_i at round t.

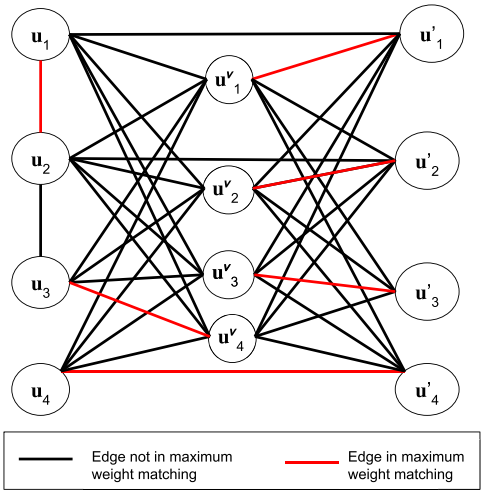


Fig. 2. An example of auxiliary graph $G(t)$ when there are four devices and $K = 2$, two devices (either u_1 or u_2, and u_4) can upload their aggregated local models to server s at round t.

assume that the weighted maximum matching of $G(t)$ consists of edges $e(u_1, u_2), e(u_4, u_4'), e(u_1', u_2^\nu), e(u_2', u_2^\nu), e(u_3', u_3^\nu)$, and $e(u_3, u_4^\nu)$ respectively. Edge $e(u_4, u_4')$ is in the weighted maximum matching means that device v_4 participates in training at round t, and the number of sampled data points used for its local model training is determined by the weight of $e(u_1, u_1')$. Similarly, edge $e(u_1, u_2)$ is in the weighted maximum matching represents that both devices v_1 and v_2 participate in training at round t, and v_1 uses v_2 as its relay or vice versa. Device v_3 does not participate in training at round t as vertex u_3 is connected to a dummy vertex through edge $e(u_3', u_4^\nu)$ in the weighted maximum matching.

C. Algorithm Analysis

The rest is to analyze the convergence rate and time complexity of the proposed algorithm, `Algorithm 1`, and show that the solution delivered by the proposed algorithm is a near-optimal solution under the well adopted assumptions on the loss function $L(\cdot)$ [28].

Assumption 1: The loss function $L(w)$ for the federated learning with model parameter w on a dataset $\mathcal{D}$ meets the following assumptions.

1) The loss function $L(w)$ is convex.
2) The loss function $L(w \mid \mathcal{D})$ is μ-smooth, that is, for any w_1 and w_2, we have $L(w_1 \mid \mathcal{D}) \leq L(w_2 \mid \mathcal{D}) + (w_1 - w_2) \cdot \nabla L(w_2|\mathcal{D}) + \frac{\mu}{2}||w_1 - w_2||^2$.
3) The second derivative of $L(w)$ has a lower bound, that is, there exists a constant β such that $\nabla^2 L(w) \leq \beta \cdot I$, where I is an identity matrix.
4) The learning rate η is smaller than β. i.e., $\eta < \beta$.
5) $\tau = 1$.
6) The datasets at the devices follow i.i.d distributions.

Notice that Assumption 1 from (1) to (4) holds when the DNN model is used for regression problems [28]. The later experimental result in Section VI will show that the proposed algorithm is still applicable, even if the loss function $L(\cdot)$ does not abide by the mentioned assumptions in Assumption 1. Under Assumption 1, we have the following lemmas and theorems.

Lemma 1: The solution delivered by the proposed algorithm, `Algorithm 1`, is feasible. Let V_{fed}^t be the set of chosen devices at round t in the solution delivered by `Algorithm 1`, then set $|\mathcal{C}_s(t)|$ of devices uploading local model to s is no more than K as $|\mathcal{C}_s(t)| \leq K \leq |V_{fed}^t(t)| \leq 2\,K$, where $1 \leq t \leq T$.

Proof: It can be seen that the energy constraint on devices at each round has not been violated. The rest is to show that the number of devices uploading their local models to edge server s at each round is no larger than K.

For a given round t, an auxiliary graph $G(t) = (U, E; w(\cdot, \cdot))$ is constructed, in which there are $2|V|$ device vertices corresponding to the $|V|$ devices, and $2|V| - 2\,K$ dummy vertices. There is an edge between a dummy vertex and a device vertex and the edge is assigned with an infinite weight. Thus, each dummy vertex is adjacent to exactly one edge in the weighted maximum matching $\mathcal{M}$ of $G(t)$, which implies that there are $2|V| - 2\,K$ device vertices connected to the $2|V| - 2\,K$ dummy vertices by the edges in $\mathcal{M}$. Then, there are $2\,K$ device vertices in $G(t)$ that are not connected to any dummy vertices by the edges in $\mathcal{M}(t)$. We claim that these $2\,K$ device vertices form at most K edges in $\mathcal{M}$ as follows. Since edges in any matching of $G(t)$ do not share any common vertex, for each of these no more than K matching edges in $\mathcal{M}$, one of its corresponding two devices will upload its local model to server s directly, following the construction of $G(t)$. Thus, there are at most K devices communicating with server s at each round, i.e., $|\mathcal{C}_s(t)| \leq K$, where the participating device set V_{fed}^t is driven from the K device-to-device matching edges in $\mathcal{M}$.

Lemma 2: In the solution delivered by `Algorithm 1` for the special energy-aware D2D-assisted federated learning problem, the number of sampled data points from chosen devices for local model learning at each round t is maximized, where $1 \leq t \leq T$.

Proof: We show the claim by contradiction. Assume that there is another solution for the problem, in which the number of sampled data points is larger than the number of sampled data points in the solution delivered by `Algorithm 1` at each round t. Let V_{fed}^{*t} be the set of devices participating in the training at round t, which is derived from the optimal solution with the maximum number of data points, and let $\phi_{v_i}^*(t)$, $p_i^*(t)$, and $\mathcal{S}_i^*(t)$ be the destination, the transmission power level, and the number of the sampled data points of device $v_i \in V_{fed}^t$.

We first show that the maximum number of sampled data points in $\cup_{v_i \in V_{fed}^{*t}} \mathcal{S}_i(t)$ equals the weighted sum of edges in a maximum weight matching of $G(t)$. For any device $v_i \in V^*{}_{fed}(t)$, if its destination $\phi_{v_i}^{*t}$ is server s, its sampled data volume $|\mathcal{S}_i^*(t)|$ must equal the weight of edge $e(u_i, u_i')$ as $w(u_i, u_i')$ is the maximum number of sampled data points of device v_i if v_i sends its local model to s directly; otherwise, if $|\mathcal{S}_i^*(t)| > w(u_i, u_i')$, we have $|\mathcal{S}_i^{*t}| \cdot \psi_i > \mathcal{E}_i(t) - Tran_{v_i}(t)$, which violates the energy constraint on v_i. Assume that $|\mathcal{S}_i^*(t)| < w(u_i, u_i')$, then device v_i can sample more data points than the current one without affecting other devices. This contradicts the assumption of the maximum number of sampled data points as

the weight of edge (u_i, u_i'). Similarly, if v_i is the destination of device v_j with $i \neq j$, then $w(u_i, u_j) = |\mathcal{S}_i^*(t)| + |\mathcal{S}_j^*(t)|$. Therefore, there is a subset of edges in $G(t)$ such that the weighted sum of the edges in it is equal to the number of sampled data points $|\cup_{v_i \in V_{fed}^{*t}} \mathcal{S}_i^*(t)|$. Since a device can only serve as either the relay node of another device or the relayed node, not both of the roles at the same time. The set of edges through pairing relay and relayed nodes forms a matching M^* of $G(t)$.

We then show that the weighted sum of edges in M^* is no larger than that of another matching $M' = \mathcal{M} \backslash E_{num}$ delivered by `Algorithm 1`. M^* consists of at most K edges due to constraint (5), which means $2K$ vertices in $\{u_i, u_i' \mid v_i \in V\}$ are adjacent to edges in M^*. Consequently, there are $2|V| - 2K$ vertices in $G(t)$ that are not adjacent to any edges in M^*. We can find a subset of edges $E_{dum}^* \subset E_{dum}$ with cardinality $2|V| - 2K$ such that $M^* \cup E_{dum}^*$ is still a matching in $G(t)$. Recall that $\mathcal{M}$ is a maximum weight matching in $G(t)$. Since the cardinality of the intersection of $\mathcal{M}$ and E_{num} is also $2|V| - 2K$ and the weights of edges in E_{num} are identical, the weighted sums of edges in $\mathcal{M} \cap E_{num}$ and E_{dum}^* are equal. As $M^* \cup E_{dum}^*$ is a matching, the weighted sum of edges in $M^* \cup E_{dum}^*$ is no larger than that of edges in $\mathcal{M}$. The weighted sum of edges in M^* thus is no larger than the weighted sum of edges in M'. The number of sampled data points in the solution delivered by `Algorithm 1` equals the weighted sum of edges in M^*. This contradicts the initial assumption that there is another solution that has more sampled data points than that of the solution delivered by `Algorithm 1`.

Lemma 3: The expectation of loss $\mathbb{E}[L(w(t+1) \mid \mathcal{D})]$ is upper bounded by an additive value $\mathbb{E}[||w(t+1) - w^*||^2]$.

$$\mathbb{E}[L(w(t+1) \mid \mathcal{D})] \leq L(w^* \mid \mathcal{D}) + \frac{\mu}{2} \cdot \mathbb{E}[||w(t+1) - w^*||^2],$$

where w^* is the optimal model parameter and $L(w^* \mid \mathcal{D})$ is the minimum value of the loss function, which is constant.

Proof: From Assumption 1.(2), we have

$$\begin{aligned} & L(w(t+1) \mid D) \\ \leq\ & L(w^* \mid \mathcal{D}) + (w(t) - w^*) \cdot \nabla L(w^* \mid \mathcal{D}) \\ & + \frac{\mu}{2}||w(t+1) - w^*||^2 \\ =\ & L(w^* \mid \mathcal{D}) + \frac{\mu}{2}||w(t+1) - w^*||^2, \end{aligned}$$

where $\nabla L(w^* \mid \mathcal{D}) = 0$, since $L(w^* \mid \mathcal{D})$ is the minimum value and the gradient at a minimum point is 0. Therefore,

$$\begin{aligned} & \mathbb{E}[L(w(t+1) \mid \mathcal{D})] \\ \leq\ & \mathbb{E}[L(w^* \mid \mathcal{D}) + \frac{\mu}{2}||w(t+1) - w^*||^2] \\ =\ & L(w^* \mid \mathcal{D}) + \frac{\mu}{2} \cdot \mathbb{E}[||w(t+1) - w^*||^2], \end{aligned}$$

as the minimum loss $L(w^* \mid \mathcal{D})$ is a constant.

Lemma 4: Let $w(t)$ be the result delivered by `Algorithm 1` at round t with $1 \leq t \leq T$. Then, the expected gap between $w(t)$ and the optimal one w^* is upper bounded, i.e.,

$$\mathbb{E}[||w(t+1) - w^*||^2] \leq (1 - \eta \cdot \beta) \cdot \mathbb{E}[||w(t) - w^*||^2] + \frac{\eta^2 \cdot \Delta^2}{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|}, \quad (17)$$

where Δ is the variance of $\nabla l(w(t), x, y)$.

Proof: As there is only one epoch local training ($\tau = 1$) at each round t by `Algorithm 1`, we have

$$w_i^\tau(t) = w_i^1(t) = w(t-1) - \eta \cdot \nabla L_i(w(t-1) \mid \mathcal{S}_i(t)).$$

Recall that w^* is the optimal parameter, we have

$$\nabla L(w^* \mid \mathcal{D}) = 0.$$

Also, for any two models w_1 and w_2, there is a model w' such that

$$\begin{aligned} & \nabla L(w_1 \mid \mathcal{D}) - \nabla L(w_2 \mid \mathcal{D}) \\ & = \nabla^2 L(w' \mid \mathcal{D}) \cdot (w_1 - w_2) \quad \forall w_1, w_2,\ \exists w', \end{aligned}$$

due to the property of gradients. By (3),

$$\begin{aligned} & \nabla L(w(t) \mid \cup_{v_i \in V_{fed}^t} \mathcal{S}_i(t)) \\ & = \frac{\sum_{v_i \in V_{fed}^t} |\mathcal{S}_i(t)| \cdot \nabla L_i(w(t) \mid \mathcal{S}_i(t))}{\sum_{v_i \in V_{fed}^t} |\mathcal{S}_i(t)|} \end{aligned}$$

Therefore,

$$\begin{aligned} w(t+1) - w^* &= \frac{\sum_{v_i \in V_{fed}^{t+1}} |\mathcal{S}_i(t+1)| \cdot w_i^\tau(t+1)}{\sum_{v_i \in V_{fed}^{t+1}} |\mathcal{S}_i(t+1)|} - w^* \\ &= \frac{\sum_{v_i \in V_{fed}^{t+1}} |\mathcal{S}_i(t+1)| \cdot (w(t) - \eta \cdot \nabla L_i(w(t) \mid \mathcal{S}_i(t+1)))}{\sum_{v_i \in V_{fed}^{t+1}} |\mathcal{S}_i(t+1)|} \\ & \quad - w^* \\ &= w(t) - \eta \cdot \nabla L(w(t) \mid \cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)) - w_i^* \\ &= w(t) - w^* - \eta \cdot (\nabla L(w(t) \mid \mathcal{D}) - \nabla L(w^* \mid \mathcal{D}) \\ & \quad - L(w(t) \mid \cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)) - \nabla L(w(t) \mid \mathcal{D})), \\ &= w(t) - w^* - \eta(\nabla^2 L(w' \mid \mathcal{D}) \cdot (w(t) - w^*)) \\ & \quad - \eta(\nabla L(w(t) \mid \cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)) - \nabla L(w(t) \mid \mathcal{D})). \end{aligned}$$

Since we assume that the training data follows i.i.d, the gradient over the sampled data set can be regarded as the data set $\mathcal{D}$ [27]. We then apply central limit rules,

$$\frac{\nabla L(w(t) | \cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1))) - \nabla L(w(t) \mid \mathcal{D})}{\Delta / \sqrt{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1))|}} \sim \mathcal{N}(0, 1),$$

where $\nabla l(w(t), x, y)$ is the gradient of the loss function $l(w(t), x, y)$ with respect to $w(t)$, and Δ^2 is the variance of $\nabla l(w(t), x, y)$.

The variance of $w(t+1) - w^*$ thus is

$$Var[w(t+1) - w^* \mid w(t)]$$

$$= Var[w(t) - w^* - \eta(\nabla L(w(t) \mid \mathcal{D}) - \nabla L(w^* \mid \mathcal{D}))$$
$$-\eta(\nabla L(w(t) \mid \cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)) - \nabla L(w(t) \mid \mathcal{D}) \mid w(t)]$$
$$= \eta^2 \cdot Var[(\nabla L(w(t) \mid \cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)))$$
$$- \nabla L(w(t) \mid \mathcal{D}) \mid w(t)]$$
$$= \frac{\eta^2 \cdot \Delta^2}{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1))|},$$

We then have

$$||\mathbb{E}[w(t+1) - w^* \mid w(t)]||^2$$
$$= ||\mathbb{E}[w(t) - w^* - \eta(\nabla^2 L(w' \mid \mathcal{D}) \cdot (w(t) - w^*)) \mid w(t)]$$
$$+ \mathbb{E}[-\eta(\nabla L(w(t)| \cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)$$
$$- \nabla L(w(t) \mid \mathcal{D})) \mid w(t)]||^2$$
$$= ||\mathbb{E}[(I - \eta \cdot \nabla^2 L(w' \mid \mathcal{D})) \cdot (w(t) - w^*) \mid w(t)]||^2$$
$$\leq ||\mathbb{E}[(1 - \eta \cdot \beta) \cdot (w(t) - w^*) \mid w(t)]||^2$$
$$= (1 - \eta \cdot \beta) \cdot ||w(t) - w^*||^2 \quad (18)$$

where Inequality (18) is due to Assumption 1.2. We then have

$$\mathbb{E}[||w(t+1) - w^*||^2 \mid w(t)]$$
$$= ||\mathbb{E}[w(t+1) - w^* \mid w(t)]||^2 + Var[w_s(t+1) - w^* \mid w(t)]$$
$$\leq (1 - \eta \cdot \beta) \cdot ||w(t) - w^*||^2 + \frac{\eta^2 \cdot \Delta^2}{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|}.$$

By the law of the total expectation,

$$\mathbb{E}[||w(t+1) - w^*||^2]$$
$$\leq (1 - \eta \cdot \beta) \cdot \mathbb{E}[||w(t) - w^*||^2] + \frac{\eta^2 \cdot \Delta^2}{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|}.$$

Theorem 1: Assuming that loss function $L(\cdot)$ is μ-smooth and β-Lipschitz. The proposed algorithm, `Algorithm 1`, delivers a feasible solution for the special energy-aware D2D assisted federated learning problem with the expected loss $\mathbb{E}[L(w(T) \mid \mathcal{D})]$, which is defined as follows:

$$\mathbb{E}[L(w(T) \mid \mathcal{D})]$$
$$\leq L(w^* \mid \mathcal{D}) + \frac{\mu}{2} \cdot \left((1 - \eta \cdot \beta)^T \cdot \mathbb{E}[||w(0) - w^*||^2] \right.$$
$$\left. + \sum_{t=0}^{T-1} (1 - \eta \cdot \beta)^{T-t-1} \cdot \frac{\eta^2 \cdot \Delta^2}{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|} \right), \quad (19)$$

where μ, η, and β are given constants that are defined in Assumption 1, and Δ^2 is the variance of $\nabla l(w(t), x, y)$ for all data points $(x, y) \in \mathcal{D}$.

Proof: Having Lemmas 3 and 4, Theorem 1 is shown as follows.

From Lemma 3, we can see that the expectation of loss $\mathbb{E}[L(w(t+1) \mid \mathcal{D})]$ depends on $\mathbb{E}[||w(t+1) - w^*||^2]$. This implies that minimizing the value $\mathbb{E}[||w(T) - w^*||^2]$ is equivalent to minimizing the expectation loss $\mathbb{E}[L(w(T) \mid \mathcal{D})]$ at the last round T.

By Lemma 4, the expectation $\mathbb{E}[||w(t) - w^*||^2]$ at round t is upper bounded by $\mathbb{E}[||w(t-1) - w^*||^2]$ and $\frac{\eta^2 \cdot \Delta'^2}{|\cup_{v_i \in V_{fed}^{t}} \mathcal{S}_i(t)|}$. It can be seen that the value gap between the global model parameter and the optimal model parameter is bounded by the difference of the global model parameter to the optimal model parameter at round $t-1$ and the number of sampled data points at round t. By expanding Inequality (17) recursively, we then have

$$\mathbb{E}[||w(t+1) - w^*||^2]$$
$$\leq (1 - \eta \cdot \beta) \cdot \mathbb{E}[||w(t) - w^*||^2] + \frac{\eta^2 \cdot \Delta^2}{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|}$$
$$\leq (1 - \eta \cdot \beta)^{t+1} \cdot \mathbb{E}[||w(0) - w^*||^2]$$
$$+ \sum_{t'=0}^{t} \frac{\eta^2 \cdot \Delta^2 \cdot (1 - \eta \cdot \beta)^{t-t'}}{|\cup_{v_i \in V_{fed}^{t'+1}} \mathcal{S}_i(t'+1)|}. \quad (20)$$

By plugging (20) and the final round T into the optimization objective (12), we have

$$\mathbb{E}[L(w(T) \mid \mathcal{D})]$$
$$\leq L(w^* \mid \mathcal{D}) + \frac{\mu}{2} \cdot \left((1 - \eta \cdot \beta)^T \cdot \mathbb{E}[||w(0) - w^*||^2] \right.$$
$$\left. + \sum_{t=0}^{T-1} \frac{\eta^2 \cdot \Delta^2 \cdot (1 - \eta \cdot \beta)^{T-t-1}}{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|} \right).$$

It can be seen from Theorem 1 that the accuracy of the obtained solution increases with the increase in the number of sampled data points $|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|$, while the latter is proportional to the value of K, where a larger K implies more devices can participate in the training at each round t, thereby leading to more sampled data points for model learning.

To better understand the relationship between K and the number of data sample points $|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|$ in the solution delivered by the proposed algorithm, `Algorithm 1`, we consider an extreme case of the special energy-aware D2D-assisted federated learning problem, where each device v_i has the same number of data points $|\mathcal{D}'|$, i.e., $|\mathcal{D}_i| = |\mathcal{D}'|$ for every device v_i. We further assume that server s is within the transmission range of all devices but none of the devices is within the transmission range of others. Under this assumption, each device can only upload its local model to server s directly. If the energy budget $\mathcal{E}_i(t)$ of each device v_i suffices for training at most half the number of data points in $\mathcal{D}_i$ at round t, i.e., $|\mathcal{S}_i(t)| = \frac{\mathcal{D}'}{2}$. then the expected loss of the DNN model on dataset

$\mathcal{D}$ is

$$\begin{aligned}
&\mathbb{E}[L(w(T) \mid \mathcal{D})] \\
&\leq L(w^* \mid \mathcal{D}) + \frac{\mu}{2} \cdot \Bigg((1-\eta\cdot\beta)^T \cdot \mathbb{E}[||w(0)-w^*||^2] \\
&\quad + \sum_{t=0}^{T-1} (1-\eta\cdot\beta)^{T-t-1} \cdot \frac{\eta^2 \cdot \Delta^2}{|\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|} \Bigg) \\
&= L(w^* \mid \mathcal{D}) + \frac{\mu}{2} \cdot \Bigg((1-\eta\cdot\beta)^T \cdot \mathbb{E}[||w(0)-w^*||^2] \\
&\quad + \sum_{t=0}^{T-1} (1-\eta\cdot\beta)^{T-t-1} \cdot \frac{2\eta^2 \cdot \Delta^2}{K \cdot |\mathcal{D}'|} \Bigg). \qquad (21)
\end{aligned}$$

It can be seen from Inequality (21) that a more accurate solution can be obtained when the value of K is larger, provided that the number T of training rounds does not change at all.

Theorem 2: Given a set V of devices and an edge server s for a federated learning process with T round training, there is an algorithm, `Algorithm 1`, for the special energy-aware D2D-assisted federated learning problem, which can deliver a near optimal solution within $\mathcal{O}(T \cdot |V|^4)$ time.

Proof: By `Algorithm 1`, the number of sampled data points at each round t is maximized. While the number of sampled data points at one round is independent of other rounds, `Algorithm 1` delivers a solution with maximizing the value of $\sum_{t=0}^{T-1} |\cup_{v_i \in V_{fed}^{t+1}} \mathcal{S}_i(t+1)|$, which is equivalent to minimizing (19).

We then analyze the time complexity of `Algorithm 1`. The construction of auxiliary graph $G(t)$ at each round t takes $\mathcal{O}(|V|^2)$ time, as $G(t)$ consists of $4|V| - 2K$ vertices. Finding a maximum weight matching in $G(t)$ takes $\mathcal{O}((4|V| - 2K)^4) = \mathcal{O}(|V|^4)$ time [11]. The time complexity of the proposed algorithm is $\mathcal{O}(T \cdot |V|^4)$, since it takes T training rounds.

V. Algorithm for the Energy-Aware D2D-Assisted Federated Learning Problem

So far we assumed that the data distribution of each device follows i.i.d., and the proposed algorithm for a special energy-aware, D2D-assisted federal learning problem, trains local models within one epoch at each round. In this section, we present an efficient heuristic algorithm for the energy-aware, D2D-assisted federated learning problem, by removing the mentioned assumptions on data sets and allowing multiple-epoch local training. We start with choosing devices for uploading their local models at each round, we then give an overview of the proposed algorithm, followed by detailing the algorithm.

A. Device Choices at Each Round

In reality, the data generated by different devices are biased and may not follow the i.i.d. assumption. The global optimal model parameter w^* is different from the local optimal model parameter w_i^* with the minimum local loss, i.e., the local loss $L(w^* \mid \mathcal{D}_i)$ is not optimal in terms of the global model. Since each device v_i only samples data points in its dataset $\mathcal{D}_i$, device v_i can only train the model w towards w_i^*. Moreover, only up to K devices are chosen to upload their local models to server s at each round, which makes the global model w strongly biased towards these chosen devices. To mitigate this biased model training caused by some but all devices in V participating in training on the global model, we propose an efficient strategy for device choices as follows.

A device will have a large variance if it samples inadequate numbers of data points for local training. To ensure the quality of local training, it requires each chosen device to train on at least $\delta \cdot |\mathcal{D}_i|$ data points with $\delta \cdot |\mathcal{D}_i| \leq |\mathcal{S}_i(t)|$, where δ is a given threshold with $0 < \delta < 1$. Intuitively, a higher local loss on some devices implies that the model lacks of proper training on the data samples of the chosen devices. Cho et al. showed that devices with higher local losses make the learning convergence faster [7]. This implies that devices with higher local losses will have higher priorities to participate in training at that round.

We here adopt the similar principle as Cho et al. [7] did. The key is to choose a subset V_{fed}^t of devices with high local loss to participate in training at each round t. To this end, each device v_i samples a small portion of its dataset and makes use of the sampled data points to estimate the local loss $\hat{L}(w_i(t-1) \mid \mathcal{D}_i)$ in the beginning of the next round t. Note that the energy consumption of devices on inferences usually is much smaller than that of it on model training. We thus assume that the energy consumption of devices on inference estimation is negligible. A device v_i that does not participate in training at round t has either a lower local loss than those of devices in V_{fed}^t, or a higher local loss than some of devices in V_{fed}^t, but may cause that devices with higher local losses have inadequate energy to train. In the following we use an example to illustrate why the greedy-based service choice for set V_{fed}^t does not work.

Consider that there are three devices v_{i_1}, v_{i_2} and v_{i_3}, where v_{i_1} has a higher local loss than v_{i_2}, and v_{i_2} has a higher local loss than v_{i_3} but both v_{i_1} and v_{i_2} are far from server s (i.e., server s is not in their transmission ranges). We further assume that device v_{i_3} is the only device within the transmission ranges of both v_{i_1} and v_{i_2} while server s is within the transmission range of v_{i_3}. In this case, only either v_{i_1} or v_{i_2} can participate in training at a round, and v_{i_3} as the unique relay vertex of either of them to upload the local model to server s at round t. Therefore, devices v_{i_1} and v_{i_3} form a pair, and v_{i_3} is added to V_{fed}^t despite that v_{i_2} has a higher local loss than that of v_{i_3}, i.e., $V_{fed}^t = \{v_{i_1}, v_{i_3}\}$.

B. Algorithm Overview

Due to the energy budget on each device at each round, it does not form a feasible solution by adding devices to set V_{fed}^t greedily. Instead, finding a feasible solution to the problem needs taking both the estimated local loss and the energy budget of each device into consideration, when determining whether the device can be added to V_{fed}^t.

The basic idea behind the proposed algorithm is similar to the one in the previous section, i.e., we reduce the problem to

a series of maximum weight matching problems in different auxiliary graphs. Specifically, we start by sorting devices in V in non-decreasing order of their estimated local loss. For the sake of convenience, let $v_1, v_2, \ldots, v_{|V|}$ be the sorted devices, where v_1 and $v_{|V|}$ have the lowest and highest local losses, respectively.

We then construct an auxiliary graph $\mathcal{G}(t) = (U, E; w(\cdot,\cdot))$ that is similar to the auxiliary graph in the previous section at round t, where the vertex set U consists of vertices u_i and u_i' for each device $v_i \in V$ and $2|V| - 2K$ dummy vertices.

The construction of the edge set E of $\mathcal{G}(t)$ is as follows. For each device $v_i \in V$, to fulfill the number of sampled data points, v_i must spend at least the amount $\psi_i \cdot \delta \cdot |\mathcal{D}_i| \cdot \tau$ of energy on training. For the sake of convenience, denote by $\mathcal{E}_i'(t)$ $(= \mathcal{E}_i(t) - \psi_i \cdot \delta \cdot |\mathcal{D}_i| \cdot \tau)$ the energy budget of v_i at round t. The required transmission power $tran(v_i, s)$ of v_i then is calculated as follows.

If the required transmission energy constraint can be met (i.e., $tran(v_i, s) \leq \mathcal{E}_i(t) - \mathcal{E}_i'(t)$), add an edge $e(u_i, u_i')$ to E in $\mathcal{G}(t)$ with weight of $w(u_i, u_i')$ $(= 2^i)$, assuming that device v_i has the ith lowest estimated local loss. An edge $e(u_i, u_i')$ with weight 2^i indicates the high priority of v_i as a potential uploading device at round t, and v_i can upload its local model to s directly if it is chosen.

For each pair of devices v_i and v_j with $i > j$, if $tran(v_i, v_j) \leq \mathcal{E}_i(t) - \mathcal{E}_i'(t)$ and $tran(v_j, v_s) \leq \mathcal{E}_i(t) - \mathcal{E}_j'(t)$, add an edge $e(u_i, u_j)$ to E with weight of $w(u_i, u_j)$ $(= 2^i + 2^j)$. This means that v_i can send its trained local model to v_j, v_j then aggregates the received model from v_i with its own local model, and uploads the aggregated model to server s. Otherwise, if $tran(v_j, v_i) \leq \mathcal{E}_i(t) - \mathcal{E}_j'(t)$ and $tran(v_i, v_s) \leq \mathcal{E}_i(t) - \mathcal{E}_i'(t)$), we add an edge $e(u_i, u_j)$ to E with weight of $w(u_i, u_j)$ $(= 2^i + 2^j)$, the operations are similar, instead, v_i will upload the aggregated model to server s.

For a pair of a dummy vertex u_j^ν and a device vertex u_i, add an edge $e(u_j^\nu, u_i)$ with weight of ∞ to E.

Let $\mathcal{M}(t)$ be the maximum weight matching in $\mathcal{G}(t)$, the K chosen devices driven from $\mathcal{M}(t)$ form a solution of the problem, and the set V_{fed}^t of devices for uploading their local models to server s can also be obtained, too.

C. Algorithm

The proposed algorithm proceeds as follows.

In the beginning of each round t, each device v_i samples a small subset of its dataset $\mathcal{D}_i$ to estimate the local loss $\hat{L}(w(t-1) \mid \mathcal{D}_i)$. A weighted auxiliary graph $\mathcal{G}(t)$ then is constructed, and the estimated local loss and device ranking are used to assign weights to the edges in the graph. A weighted maximum matching $\mathcal{M}(t)$ in $\mathcal{G}(t)$ then is found, and a feasible solution to the problem finally is derived from the weighted maximum matching $\mathcal{M}(t)$. That is, if edge $e(u_i, u_i') \in \mathcal{M}(t)$, device v_i uploads its local model to server s directly; otherwise, if edge $e(u_i, u_j) \in \mathcal{M}(t)$ with $i > j$ and $tran(v_j, s) \leq \mathcal{E}_i(t) - \mathcal{E}_i'(t)$, device v_j is the destination of v_i, aggregates its local model with the local model of v_i and uploads the aggregated local model to s; Similarly, if $e(u_i, u_j) \in \mathcal{M}(t)$ with $i > j$ but $tran(v_j, s) > \mathcal{E}_i(t) - \mathcal{E}_i'(t)$, v_j sends its local model to v_i for aggregation and v_i uploads the aggregated model to server s. The rest devices that are not incident to any matching edge in $\mathcal{M}(t)$ will not participate in training at round t. As a result, the number of sampled data points $|\mathcal{S}_i(t)|$ of device v_i at round t is $\lfloor \frac{\mathcal{E}_i(t) - Tran_{v_i}(t)}{\psi_i} \rfloor$.

The detailed algorithm for the energy-aware, D2D assisted federated learning problem is given in `Algorithm 2`.

Theorem 3: Given a set V of IoT devices, and an edge server s that performs federated learning training within T rounds, there is an efficient algorithm, `Algorithm 2`, for the energy-aware D2D-assisted federated learning problem, which takes $\mathcal{O}(T \cdot |V|^4)$.

Proof: The analysis of time complexity of `Algorithm 2` is similar to `Algorithm 1`, while the latter has been analyzed in Theorem 2, omitted.

VI. Performance Evaluation

In this section, we evaluated the performance of the proposed algorithms through experimental simulations. We also investigated the impact of important parameters on the performance of the proposed algorithms.

A. Experimental Settings

We consider a sensor network that consists of 100 devices randomly deployed in a circular area with a 250 meter radius and an edge server co-located with an access point at the center of the area, where the server can communicate with up to 20% of the devices directly, i.e., $K = |V| * 20\%$. The devices and the server collaboratively conduct a federated learning model training with 50 rounds, and each training round consists of $\tau = 1$ epoch. The bandwidth at each AP is set at 10 MHz and the white noise σ^2 is set at 1×10^{-10} Watt [13]. The channel gain at the reference distance of 1 meter α is set as 10^{-3} [18]. The maximum transmission power of IoT devices in [13] is 0.5 Watt, The transmission power levels of each device are in $\{0.2, 0.3, 0.4, 0.5\}$ Watt and the minimum signal-to-noise ratio is at 14dB [29]. The corresponding transmission distances of these transmission power levels are 282, 346, 400, and 447 meters, respectively. The energy budget on each device at each round is randomly drawn in [0.01, 0.04] Joules [3]. The dataset in this experiment is the MNIST that consists of 70,000 images of handwritten digits, which is a well known dataset adopted by many federated learning works [3], [28]. We randomly distribute the training images of the dataset to 100 IoT devices. The DNN model used in this experiment is Le-net5, which has a size $\mathcal{C}$ of 1,960Kbits [12]. The energy consumption on training one data point for Le-net5 at a device is randomly drawn between 1×10^{-4} Joules and 5×10^{-4} Joules [3], [20]. The value in each figure is the mean of the results out of 50 network instances of the same size, and the running time is obtained based on a machine with a 3.79 GHz AMD Ryzen 5 CPU.

To evaluate the performance of the proposed algorithms, we first compared the proposed algorithm against `FedAvg` [19] - a

Algorithm 2: Algorithm for the Energy-Aware, D2D-Assisted Federated Learning Problem.

Input: A set of devices V, a server s, the dataset $\mathcal{D}_i$ of each device $v_i \in V$, an edge budget $\mathcal{E}_i(t)$ of v_i at round t, a set of transmission power level $\mathcal{P}$, a given percentage of sampled data threshold δ and a DNN model w to be trained at t round with $1 \leq t \leq T$.

Output: The set V_{fed}^t of devices that participate in training, the offloading destination $\phi_{v_i}(t)$, the set of sampled data $\mathcal{S}_i(t)$, and the transmission power level $p_i(t)$ of each chosen device v_i at each round t.

for $t \leftarrow 1$ to T **do**
  Each device $v_i \in V$ selects a small subset of its data to estimate the local loss $\hat{L}(w(t-1) \mid \mathcal{D}_i)$;
  Sort vertices in V in non-decreasing order of the estimated local loss;
  Initialize $G(t) \leftarrow (U, E)$ where $U = \emptyset$ and $E = \emptyset$;
  for $i \leftarrow 1$ to $|V|$ **do**
    $U \leftarrow U \cup \{u_i, u_i'\}$;
    Calculate $p_i^{min}(s)$, $tran(v_i, s)$;
    if $tran(v_i, s) + \psi_i \cdot \delta \cdot |\mathcal{D}_i| \cdot \tau \leq \mathcal{E}_i(t)$ **then**
      $E \leftarrow E \cup \{e(u_i, u_i')\}$; $w(u_i, u_i') \leftarrow 2^i$;
  for $i \leftarrow 1$ to $|V|$ **do**
    for $j \leftarrow i+1$ to $|V|$ **do**
      Calculate $p_j^{min}(s)$, $tran(v_i, v_j)$, and $tran(v_j, s)$;
      if $tran(v_i, v_j) + \psi_i \cdot \delta \cdot |\mathcal{D}_i| \cdot \tau < \mathcal{E}_i(t)$ && $tran(v_j, s) + \psi_j \cdot \delta \cdot |\mathcal{D}_j| \cdot \tau < \mathcal{E}_j(t)$ || $tran(v_j, v_i) + \psi_j \cdot \delta \cdot |\mathcal{D}_j| \cdot \tau < \mathcal{E}_j(t)$ && $tran(v_i, s) + \psi_i \cdot \delta \cdot |\mathcal{D}_i| \cdot \tau < \mathcal{E}_i(t)$ **then**
        $E \leftarrow E \cup \{e(u_i, u_j)\}$;
        $w(u_i, u_j) \leftarrow 2^i + 2^j$;
  $U_{dum} \leftarrow \{u_j^\nu \mid 1 \leq j \leq 2|V| - 2K\}$
  $U \leftarrow U \cup U_{dum}$ /* U_{dum} is the set of dummy nodes */;
  for each dummy vertex $u_j^\nu \in U_{dum}$ **do**
    for each vertex $u_i \in U \setminus U_{dum}$ **do**
      $E \leftarrow E \cup \{e(u_i, u_j^\nu)\}$; $w(u_i, u_j^\nu) \leftarrow \infty$;
  Find a weighted maximum matching $\mathcal{M}(t)$ in $G(t)$, by applying the algorithm in [11];
  for $i \leftarrow 1$ to $|V|$ **do**
    if $e(u_i, u_i') \in \mathcal{M}(t)$ **then**
      $\phi_{v_i}(t) \leftarrow s$; $V_{fed}^t \leftarrow V_{fed}^t \cup \{v_i\}$;
      $|\mathcal{S}_i(t)| \leftarrow \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, s)}{\psi_i} \rfloor$;
    if $\exists j > i, s.t. e(u_i, u_j) \in \mathcal{M}(t)$ **then**
      $V_{fed}^t \leftarrow V_{fed}^t \cup \{v_i, v_j\}$;
      if $tran(v_i, s) + \psi_i \cdot \delta \cdot |\mathcal{D}_i| \cdot \tau < \mathcal{E}_i(t)$ **then**
        $\phi_{v_i}(t) \leftarrow s$; $\phi_{v_j}(t) \leftarrow v_i$;
        $|\mathcal{S}_j(t)| \leftarrow \lfloor \frac{\mathcal{E}_j(t) - tran(v_j, v_i)}{\psi_j} \rfloor$;
        $|\mathcal{S}_i(t)| \leftarrow \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, s)}{\psi_i} \rfloor$;
      else
        $\phi_{v_j}(t) \leftarrow s$; $\phi_{v_i}(t) \leftarrow v_j$;
        $|\mathcal{S}_j(t)| \leftarrow \lfloor \frac{\mathcal{E}_j(t) - tran(v_j, s)}{\psi_j} \rfloor$;
        $|\mathcal{S}_i(t)| \leftarrow \lfloor \frac{\mathcal{E}_i(t) - tran(v_i, v_j)}{\psi_i} \rfloor$;
  return V_{fed}^t, $\phi_{v_i}(t)$, $\mathcal{S}_i(t)$ for each device v_i at round t.

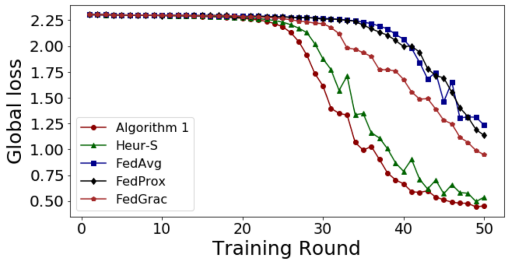


Fig. 3. Convergence of different algorithms for the special energy-aware D2D-assisted federated learning problem with i.i.d. data and $T = 50$.

classic federated learning algorithm where we randomly select K devices to participate in the model training at each round. We also compare the proposed algorithm against algorithm `FedProx` in [15] and a recent algorithm `FedGrac` in [30].

We then devised a heuristic algorithm `Heur`, which chooses K devices participating in training greedily at each round. Within each iteration, algorithm `Heur` always chooses the device with the highest local loss from those yet-to-be-chosen devices as one participant at the current round, provided that the device has adequate energy to upload its local model to the server directly, or transmits its local model to its relay partner device. This procedure continues until no more devices can be added or there are K chosen devices already. Similarly, let `Heur-S` be a heuristic that adds devices with the maximum number of sampled points for each round training for the special energy-aware, D2D-assisted federated learning problem. `Heur-S` selects top-K devices with the maximum number of sampled data points to send their local models to the server. It then calculates the number of sampled data points of those not yet-to-be chosen devices if they upload their local models, using the top K chosen devices.

B. Performance of Different Algorithms for the Special Energy-Aware, D2D-Assisted Federated Learning Problem

We first evaluated the performance of different algorithms for the special energy-aware D2D-assisted federated learning problem under the i.i.d data distribution. To ensure that the dataset at each device follows the i.i.d distribution, the data points with identical label are evenly distributed among devices [37]. Fig. 3 showed the convergence curves of the five comparison algorithms in the default setting. It can be seen that the global loss of all mentioned algorithms decreases with the growth on the number T of training rounds, and the decreasing rate on the global loss of `Algorithm 1` is faster and more steady compared with the other four algorithms. The rationale behind is that `Algorithm 1` makes use of more sampled data points for training at each round that leads to a lower variance on the training result, while `FedGrac` does not consider the energy capacity of devices and communication constraints on the server, it samples fewer data points for model training, which leads to a poor performance compared with `Algorithm 1` and `Heur-S`. Consequently, `Algorithm 1` converges to a lower global loss than those of the other four comparison algorithms,

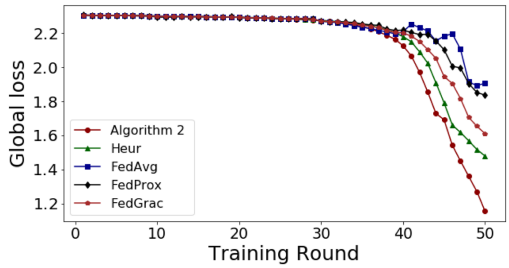
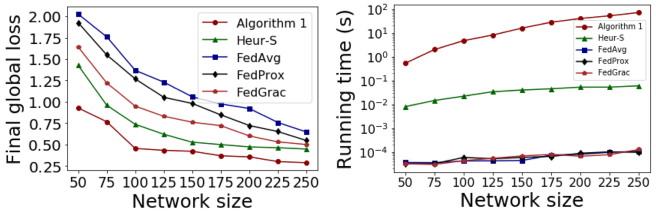
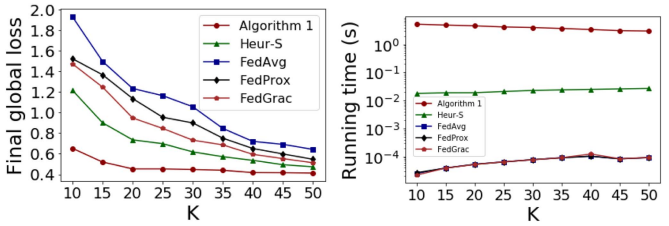


Fig. 4. Convergence of different algorithms for the energy-aware D2D-assisted federated learning problem when $\tau = 1$ and $T = 50$.



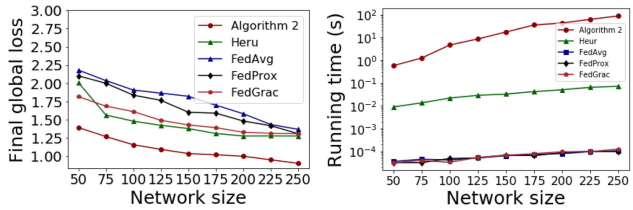
(a) The final global losses of different algorithms (b) Average running time of algorithms per training round

Fig. 5. Performance of different algorithms for the special D2D-assisted federated learning problem by varying network size.



(a) The final global losses of different algorithms (b) Average running time of algorithms per training round

Fig. 6. Performance of different algorithms for the special D2D assisted federated learning problem by varying the number K of devices communicating with the edge server simultaneously.



(a) The final global losses of different algorithms (b) Average running time of algorithms per training round

Fig. 7. Performance of different algorithms for the energy-aware D2D-assisted federated learning problem by varying network size.

and the final global loss of `Algorithm 1` is 15.4% lower than that of algorithm `Heur-S`.

C. Performance of Different Algorithms for the Energy-Aware, D2D-Assisted Federated Learning Problem

We then studied the performance of different algorithms for the energy-aware, D2D-assisted federated learning problem under the non-i.i.d data distribution. As the training data does not follow the i.i.d distribution any more, it can be seen from Fig. 4 that all comparison algorithms converge slower than that of themselves for the case with the i.i.d distribution, and the final global loss is higher compared with the one for the special case. Among the five algorithms, `Algorithm 2` has the minimum final global loss. The rationale is that `Algorithm 2` allows more devices to participate in each round training, and always chooses devices with the highest estimated local losses. Although `FedGrac` is able to mitigate the negative effect of non-i.i.d. data distribution, it only allows at most K devices to participate in each round training, thereby resulting in a higher global loss in comparison with that of `Algorithm 2`.

D. Impact of Parameters on the Algorithm Performance

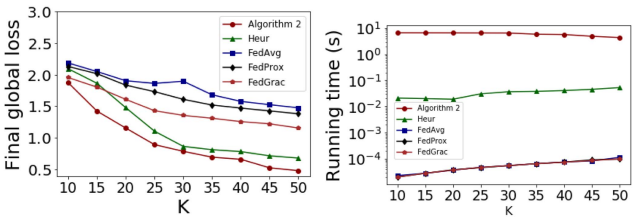
We finally investigated the impact of parameters on the performance of different algorithms for the energy-aware D2D-assisted federated learning problem. We start by evaluating the impacts of important parameters on the special case of the problem as follows.

The Impact of Network Size $|V|$. With the increase on network size, K $(= 20\% \cdot |V|)$ increases, too. Fig. 5 illustrates the performance of the five mentioned algorithms with different network sizes. It can be seen from Fig. 5(a) that `Algorithm 1` is superior to the other algorithms. It is noted that all the five algorithms have lower global losses with the increase on the number K of devices that communicate with the server simultaneously. The rationale behind this is that more devices are able to participate in training at each round. Fig. 5(b) shows the average running time of different algorithms per training round. Although `Algorithm 1` takes the longest running time among the five comparison algorithms, the global loss it delivers is the lowest among the five algorithms.

The Impact of K. We varied the value of K from 10 to 50 while fixing the network size at 100. Fig. 6(a) depicts the final global losses of different algorithms. It can be seen that `Algorithm 1` has the lowest final global loss. Specifically, when $K = 50$, the final global loss by `Algorithm 1` is 12.78% lower than that by `Heur-S`. This can be justified by that `Algorithm 1` achieves better decision of the destination of devices to ensure the total number of sampled data points is maximized. Fig. 6(b) depicts the running times of the comparison algorithms. With the growth of K, the number of dummy nodes in the auxiliary graph decreases, leading to decreasing on the running time of `Algorithm 1`.

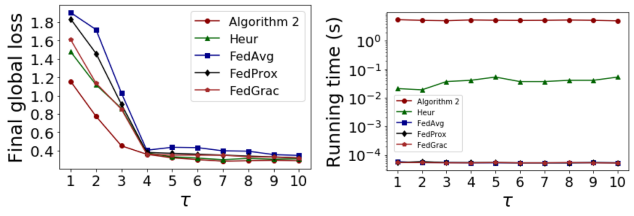
The rest is to study the impact of important parameters on the performance of the comparison algorithms for the energy-aware D2D-assisted federated learning problem as follows.

The Impact of Network Size $|V|$. We analyzed the impact of network size by varying it from 50 to 250. Fig. 7(a) shows that the final global losses of all algorithms decrease with the growth on network size. When $|V| = 250$, the final global loss of for `Algorithm 2` is 35% lower than itself when $|V| = 50$. Fig. 7(b) depicts the running times of the five algorithms.



(a) The final global losses of different algorithms (b) Average running time of algorithms per training round

Fig. 8. Performance of different algorithms for the energy-aware D2D-assisted federated learning problem by varying the number K of devices communicating with server s simultaneously.



(a) The final global losses of different algorithms (b) Average running time of algorithms per training round

Fig. 9. Performance of different algorithms for the D2D-assisted federated learning problem by varying τ.

The Impact of K. We evaluated the impact of the number K of devices that can communicate with the server simultaneously. Fig. 8(a) depicts the final global losses of different algorithms by varying K from 50 to 100. When $K = 50$, the global loss by `Algorithm 2` is 74% lower than that by itself when $K = 10$. Fig. 8(b) plots the running times of the comparison algorithms. It can be seen that both the final global loss and the running time decrease with the growth of K.

The Impact of Training Epochs τ. We investigated the impact of the number τ of training epochs per round. To better illustrate the impact of τ, we scale up the energy budget on each device v_i to $\tau \cdot \mathcal{E}_i(t)$ at each round accordingly. Fig. 9(a) plots the final global losses of different algorithms. It can be seen that `Algorithm 2` has the best performance, as it has a lower final global loss when τ is no greater than 4. This means that the federated learning can train the model better with fewer training epochs when applying `Algorithm 2`. Also, when $\tau = 10$, the final global loss of algorithm `Heur` is 10.7% larger than that of `Algorithm 2`.

VII. Conclusion and Future Work

In this paper, we investigated the energy-aware D2D-assisted federated learning problem in an edge computing environment, by making use of the energy of neighbor devices of a certain number of devices to help their local model uploading. Under the i.i.d. training data distribution, we devised a near optimal learning algorithm for the problem. We also developed an efficient heuristic algorithm for the problem by removing the i.i.d. data distribution assumption. We finally evaluated the performance of the proposed algorithms by experimental simulations. Simulation results demonstrated that the proposed algorithms are promising, and the performance of global loss improves 15.4% compared with that of the comparison algorithms. In our future work, we will extend the proposed algorithms by considering more than two IoT devices as a group to upload their aggregated model and examine whether they will outperform the proposed ones.

Acknowledgments

We appreciate the four anonymous referees and the associate editor for their constructive comments and valuable suggestions, which helped us improve the quality and presentation of the article greatly.

References

[1] A. Abdallah, M. M. Mansour, and A. Chehab, "Power control and channel allocation for D2D underlaid cellular networks," *IEEE Trans. Commun.*, vol. 66, no. 7, pp. 3217–3234, Jul. 2018.

[2] C. Chen et al., "Synchronize only the immature parameters: Communication-efficient federated learning by freezing parameters adaptively," *IEEE Trans. Parallel Distrib. Syst.*, to be published, doi: 10.1109/TPDS.2023.3241965.

[3] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[4] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.

[5] S. Chen, Y. Xu, H. Xu, Z. Jiang, and C. Qiao, "Decentralized federated learning with intermediate results in mobile edge computing," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2022.3221212.

[6] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[7] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," 2020, *arXiv: 2010.01243*.

[8] C. T. Dinh et al., "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no, 1, pp. 398–409, Feb. 2021.

[9] C. Hu, W. Bao, D. Wang, and F. Liu, "Dynamic adaptive DNN surgery for inference acceleration on the edge," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1423–1431.

[10] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-IID private data," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 191–205, Jan. 2023, doi: 10.1109/TMC.2021.3070013.

[11] E. Jack, "Maximum matching and a polyhedron with 0,1-vertices," *J. Res. Nat. Bur. Standards B*, vol. 69, no. 125–130, pp. 55–56, 1965.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[13] J. Li, W. Liang, Y. Li, Z. Xu, X. Jia, and S. Guo, "Throughput maximization of delay-aware DNN inference in edge computing by exploring DNN model partitioning and inference parallelism," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 3017–3030, May 2023.

[14] J. Li, W. Liang, W. Xu, Z. Xu, Y. Li, and X. Jia, "Service home identification of multiple-source IoT applications in edge computing," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 1417–1430, Mar./Apr. 2023.

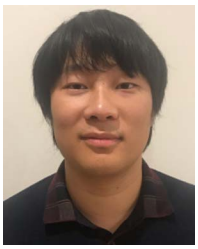
[15] T. Li, A. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2020.

[16] Y. Li et al., "Energy-constrained D2D assisted federated learning in edge computing," in *Proc. 25th Int. ACM Conf. Model. Anal. Simul. Wireless Mobile Syst.*, 2022, pp. 33–37.

[17] R. Lu et al., "Auction-based cluster federated learning in mobile edge computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1145–1158, Apr. 2023.

[18] Z. Lv, J. Hao, and Y. Guo, "Energy minimization for MEC-enabled cellular-connected UAV: Trajectory optimization and resource scheduling," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2020, pp. 478–483.

[19] B. McMahan, M. Eider, D. Ramage, S. Hampson, and A. Blaise, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
[20] T. Mohammed, C. Joe-Wong, R. Babbar, and M. D. Francesco, "Distributed inference acceleration with adaptive DNN partitioning and offloading," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 854–863.
[21] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.
[22] L. L. Pilla, "Scheduling algorithms for federated learning with minimal energy consumption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1215–1226, Apr. 2023.
[23] D. Reinsel, J. Gantz, and J. Rydning, "The digitization of the world for edge to core," IDC, Framingham, MA, USA, white paper #: US44413318, 2018, Art. no. 28.
[24] Y. Sun, S. Zhou, and D. Gündüz, "Energy-aware analog aggregation for federated learning with redundant data," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–7.
[25] Z. Tang, S. Shi, B. Li, and X. Chu, "GossipFL: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 3, pp. 909–922, Mar. 2023, doi: 10.1109/TPDS.2022.3230938.
[26] Y. Tao et al., "Byzantine-resilient federated learning at edge," *IEEE Trans. Comput.*, to be published, doi: 10.1109/TC.2023.3257510.
[27] S. Wang, Y. Ruan, Y. Tu, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2019–2032, Oct. 2021.
[28] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
[29] Z. Wang, L. Duan, and R. Zhang, "Adaptive deployment for UAV-aided communication networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4531–4543, Sep. 2019.
[30] F. Wu et al., "From deterioration to acceleration: A calibration approach to rehabilitating step asynchronism in federated optimization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1548–1559, May 2023.
[31] W. Wu, L. He, W. Lin, and C. Maple, "FedProf: Selective federated learning based on distributional representation profiling," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 6, pp. 1942–1953, Jun. 2023, doi: 10.1109/TPDS.2023.3265588.
[32] Q. Wu et al., "HiFlash: Communication-efficient hierarchical federated learning With adaptive staleness control and heterogeneity-aware client-edge association," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1560–1579, May 2023, doi: 10.1109/TPDS.2023.3238049.
[33] Z. Xu et al., "Energy or accuracy? Near-optimal user selection and aggregator placement for federated learning in MEC," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2023.3262829.
[34] Z. Xu et al., "HierFedML: Aggregator placement and UE assignment for hierarchical federated learning in mobile edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34. no. 1, pp. 328–345, Jan. 2023.
[35] W. Yang, X. Xiang, Y. Yang, and P. Cheng, "Optimizing federated learning with deep reinforcement learning for digital twin empowered industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1884–1893, Feb. 2023.
[36] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
[37] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL for wireless networks: Cooperative learning mechanism using non-IID data," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–7.
[38] J. Zhang et al., "Adaptive federated learning on Non-IID data with resource constraint," *IEEE Trans. Comput.*, vol. 71, no. 7, pp. 1655–1667, Jul. 2022, doi: 10.1109/TC.2021.3099723.



Yuchen Li received the BSc (first class honours) degree in computer science from the Australian National University, in 2018. He is currently working toward the PhD degree with the School of Computing, Australian National University. His research interests include the Internet of Things, mobile edge computing, and algorithm design.

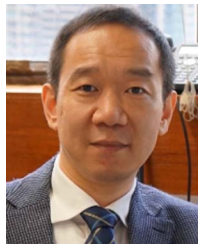
Weifa Liang (Senior Member, IEEE) received the BSc degree in computer science from Wuhan University, China, in 1984, the ME degree in computer science from the University of Science and Technology of China, in 1989, and the PhD degree in computer science from the Australian National University, in 1998. He is a professor with the Department of Computer Science, City University of Hong Kong. Prior to that, he was a professor with the Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, mobile edge computing (MEC), network function virtualization (NFV), Internet of Things (IoT) and Digital Twins (DTs), design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He currently serves as an associate editor of the *IEEE Transactions on Communications*.

Jing Li received the BSc (first class honours) and PhD degrees from the Australian National University, in 2018 and 2022, respectively. He is currently a postdoctoral fellow with the Hong Kong Polytechnic University. His research interests include mobile edge computing, Internet of Things, network function virtualization, and combinatorial optimization.

Xiuzhen Cheng (Fellow, IEEE) received the MS and PhD degrees in computer science from the University of Minnesota–Twin Cities, in 2000 and 2002, respectively. She is currently a professor with the School of Computer Science and Technology, Shandong University. Her current research interests include cyber-physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She is a member of the ACM. She has received the NSF CAREER Award in 2004. She has chaired several international conferences. She has served on the editorial boards for several technical journals and the technical program committees of various professional conferences/workshops.

Dongxiao Yu (Senior Member, IEEE) received the BSc degree from the School of Mathematics, Shandong University, in 2006, and the PhD degree from the Department of Computer Science, University of Hong Kong, in 2014. He became an associate professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2016. He is currently a professor with the School of Computer Science and Technology, Shandong University. His research interests include wireless networks and distributed computing.

Albert Y. Zomaya (Fellow, IEEE) is the chair professor of high performance computing & networking with the School of Information Technologies, University of Sydney, and he also serves as the director of the entire for Distributed and High Performance Computing. He has published more than 550 scientific papers and articles and is the author, coauthor or editor of more than 20 books. He is the founding editor in chief of the *IEEE Transactions on Sustainable Computing* and serves as an associate editor for more than 20 leading journals. He served as an editor in chief of the *IEEE Transactions on Computers* (2011–2014). He is the IEEE Computer Society Technical Achievement Award (2014), and the ACM MSWIM Reginald A. Fessenden Award (2017). He is a chartered engineer, a fellow of the AAAS and IET. His research interests include the areas of parallel and distributed computing and complex systems.



Song Guo (Fellow, IEEE) is a full professor with the Department of Computing, Hong Kong Polytechnic University. He also holds a Changjiang chair professorship awarded by the Ministry of Education of China. His research interests are mainly in the areas of Big Data, edge AI, mobile computing, and distributed systems. With many impactful papers published in top venues in these areas, he has been recognized as a Highly Cited Researcher (Web of Science) and received more than 12 Best Paper Awards from IEEE/ACM conferences, journals and technical committees. He is the editor-in-chief of IEEE Open Journal of the Computer Society. He has served on IEEE Communications Society Board of Governors, IEEE Computer Society Fellow Evaluation Committee, and editorial board of a number of prestigious international journals like the *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, *IEEE Internet of Things Journal*, etc. He has also served as chair of organizing and technical committees of many international conferences. He is an ACM distinguished member.