Client Selection in Federated Learning: A Dynamic Matching-Based Incentive Mechanism

Sai Sharanya Yellampalli†, Mikulas Chalupa*, Jingyi Wang†, Hyo Jung Song†, Xinyue Zhang*, Hao Yue†, Miao Pan‡
†Department of Computer Science, San Francisco State University, San Francisco, CA 94132
*Department of Computer Science, Kennesaw State University, Marietta, GA 30060
‡Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204

***Abstract*—Federated learning (FL) has rapidly evolved as a distributed learning paradigm, enabling clients to collaboratively train models while retaining data privacy on their devices, which can guarantee the privacy of the training data. However, it faces distinct challenges on both server and client fronts. On the server side, there is a lack of efficient strategies for selecting high-performing clients, leading to potential degradation in training accuracy due to subpar model updates. On the client's side, they are often deterred from participation due to significant energy consumption during both computation and data transmission processes. Existing incentive mechanisms in FL seldom consider both the energy consumption of the clients and the learning quality of the server. To bridge this gap, this paper introduces an adaptive incentive mechanism, which considers both the anticipated learning quality of clients and the associated energy costs during training. We propose a novel distributed Matching-based Incentive Mechanism (MAAIM) for client selection in FL. Leveraging a deferred acceptance algorithm, MAAIM facilitates stable client-server pairings, ensuring that both parties' primary concerns are addressed. Experimental results demonstrate the effectiveness of the proposed MAAIM.**

***Index Terms*—Federated Learning, Learning Quality, Matching, Optimization**

I. Introduction

During the last decade, with the rapid development of smart mobile devices such as cellphones, smartwatches, fitness trackers, computer tablets, etc, the Internet of Things (IoT) has fast evolved. Due to the ubiquitous mobile devices with embedded sensors and an extensive set of IoT applications, a massive amount of data has been generated and collected. Advanced machine learning techniques become an emerging paradigm, which can utilize the data from mobile devices to build promising high-performance models. Internet giants such as Google and Amazon have been offering "machine learning as a service" (MLaaS) [1]. The customer first uploads data to the cloud, the machine learning model will be trained on the cloud and the constructed model will be delivered to the customer as a black-box API.

However, the rising popularity of MLaaS is putting the learning pipelines at risk of cyber threats more than ever before [2]. The MLaaS users are supposed to upload their raw data to the server to receive service, whereas the raw data may contain users' sensitive and private information. For instance, the electronic health records (EHR) collected from mHealth applications contain personal private information like drug usage patterns of the individual patient. Due to the arising concerns of privacy leakage, federated learning (FL), a novel framework for distributed learning, is proposed to address the privacy and security concerns. Within the FL framework, the mobile edge devices referred to as FL clients train local models and only share the model parameters with the centralized cloud server instead of transmitting their original data samples. Based on the aggregated model parameters from each edge device, the server can update the global model with various averaging algorithms such as FedAvg [3].

In FL, the global model accuracy is highly dependent on how to select participating clients. Due to the data quality on mobile edge devices, further efforts are needed to design an effective client selection mechanism that can guarantee global training performance. Efforts have been made to improve FL performance with the client selection optimization problem. In [4], the authors proposed to deploy multi-agent reinforcement learning to solve the client selection optimization problem by jointly considering the global model accuracy and communication latency. In [5], the authors formulated a stochastic optimization problem for maximizing the global model performance. However, these works all use the centralized algorithm to solve the client selection optimization problem in FL. While the centralized method can solve the client selection optimization problem, practically, with an extremely large number of clients, a centralized algorithm is inefficient. Moreover, the cost on the clients' side is not taken into consideration. To motivate clients to participate in the FL, it is necessary to utilize incentive mechanisms to compensate for the extra energy cost to clients during the local training and transmission with the central FL server. In [6], Kang et al. studied the reputation of clients and designed the incentive mechanism based on contract theory. In [7], Deng et al. proposed a quality-aware incentive mechanism and formulated a reverse auction problem to select high-quality learning clients. Even though these works select the clients by incentive mechanisms according to the learning quality of the clients, none of them consider the extra energy cost of clients to work for different servers.

In this paper, we propose a novel distributed MAtching bAsed Incentive Mechanism (MAAIM) which jointly con-

This work was supported by the U.S. National Science Foundation under grants CNS-2318664 and CNS-2107057.

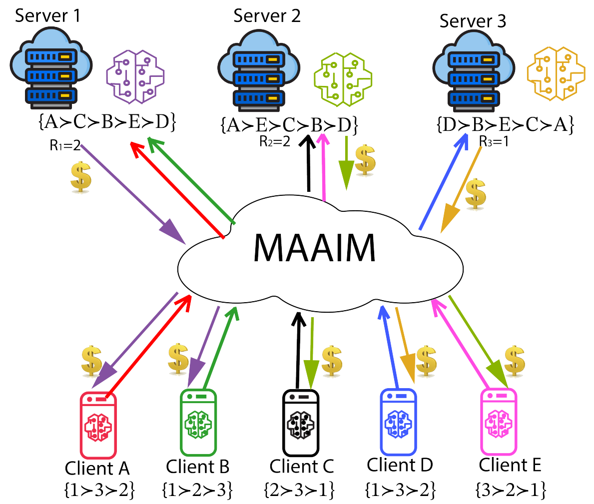


Fig. 1. System architecture.

siders the learning quality and task-based energy cost for the clients, and allows resource allocation between servers and clients in a distributed manner. In MAAIM, we employ dynamic matching to select different clients to maximize training performance. We mathematically model the problem by employing the deferred acceptance matching algorithm to solve the problem and conduct FL performance evaluations. Our salient contributions are listed as follows.

- We formulate an optimization problem to maximize the server's learning quality with consideration of the server's bid price and the client's energy consumption in each iteration.
- To obtain feasible solutions to the proposed optimization problem in a distributed manner, we propose a novel MAAIM scheme with a dynamic many-to-one matching deferred acceptance algorithm.
- In MAAIM, we formulate clients' preference lists based on their total monetary gain, which is related to the servers' bidding price and the task-based energy cost. We also formulate servers' preference lists based on the clients' learning quality estimation. The preference lists of servers will evolve during each iteration.
- Through extensive simulations, we show that the proposed MAAIM outperforms the FedAvg method.

II. System Description

A. Network Configuration

We consider a FL framework consisting of M clients in a set $\mathcal{M}$, and N servers in a set $\mathcal{N}$. We show the proposed MAAIM in Figure 1 with an example for $M=5$ and $N=3$. We assume that every server has a distinct learning task, each associated with a unique dataset for training. We also assume that each server can accommodate R_n clients during the FL process, with the capacity to accept an equivalent number of clients in subsequent iterations. We denote the $j \in \mathcal{N}$ server can accept total as quota $|R_j|$ clients to train the model. Additionally, we operate under the assumption that a client can collaborate with only one server at any given moment. In the MAAIM, to motivate clients to participate in the FL process, each server offers monetary awards to clients for the engagement of the specific learning task. Because of the limited budget, the goal of servers is to involve clients, who can provide better learning results in their learning tasks. Meanwhile, despite the monetary awards, client $i \in \mathcal{M}$ considers the energy efficiency for completing different FL tasks as well [8]. For example, in Fig. 1, Client A ranks Server 1 at the top of its preference list due to the maximum net profit it offers, calculated as the monetary reward from Server 1 minus the energy expenses for dataset training. For Server 1, Client A is the preferred choice due to its ability to provide the highest learning quality model. In conclusion, through the deployment of the MAAIM algorithm, servers prioritize clients with superior training quality to ensure model accuracy. Conversely, clients opt for servers that provide the greatest net revenue, factoring in both the monetary reward and energy costs.

B. Energy Consumption and Learning Quality Estimation

1) Client's Energy Consumption: The extra energy consumption of FL for each client takes place during the training and uploading of the local updates, where the former is based on computation and the latter is based on transmission. The energy consumption of computation is mainly dependent on the amount of local training data samples. The energy consumption of transmission is related to the data size of local updates and transmission channels between the client and server. In practice, the client energy consumption at the provided time slot when given a dataset with size $|\Pi_i|$ of client i, occupying bandwidth of B for updates transferring. the link e between client i and server j can be calculated by [9]:

$$E_{ij} = \rho_i |\Pi_i| + s_i \frac{\Upsilon}{B \log_2\left(1 + \frac{h_e s_i}{v_e}\right)}, \tag{1}$$

where Υ denotes the size of the local updates, ρ_i denotes the energy consumed to process one sample on client i which depends on the hardware components, s_i indicates the signal power, h_e and v_e define the wireless channel gains and the white noise of link e, respectively. On the right-hand side of Eq. 1, the first term indicates the energy consumption for model training and the second term expresses the energy consumption of local updates transmission, based on the Shannon-Hartley formula [10].

2) Learning Quality Estimation: The learning quality is highly related to the local training dataset quality of each client. To achieve better performance, the server prefers to select clients that can benefit more from the global model updates. To this end, the servers quantify each client's learning quality by evaluating the current global model on every client's local training results. The estimated learning quality is then calculated by subtracting the overall server evaluation loss from the evaluated client loss as follows [7],

$$q_i^j = loss_j(t) - loss_i^j(t+1), \tag{2}$$

where $loss_j$ is the average loss value of the global model on server j in round t and $loss_i^j$ is the average loss value of client i's local model in round $t+1$. The estimated learning quality for each client is then ranked in descending order, which is used to form each server's preference list. A higher evaluated loss implies there is higher learning utility available in a particular client's dataset, and thus the server ranks the client of higher priority on its preference list.

III. Optimizing Client Selection in Matching-Based Incentive Mechanism

A. Centralized Optimization Formulation of Client Selection in MAAIM

The goal of our proposed MAAIM is to optimize global training performance by selecting clients with high learning quality in each training iteration. Let the binary variable δ_i^j denotes whether client $i \in \mathcal{M}$ is chosen by server $j \in \mathcal{N}$ to work on the specific training task. We use $\delta_i^j = 1$ denotes that client i train model for server j and communicate to j, otherwise 0. The centralized optimization problem can be formulated as follows,

$$\text{Maximize} \quad \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} \delta_i^j q_i^j \tag{3}$$

s.t.:

$$\delta_i^j \in \{0,1\}, \forall i \in \mathcal{M}, j \in \mathcal{N}, \tag{4}$$

$$\sum_{j \in \mathcal{N}} \delta_i^j \leq | R_j | \quad \forall i \in \mathcal{M}, \tag{5}$$

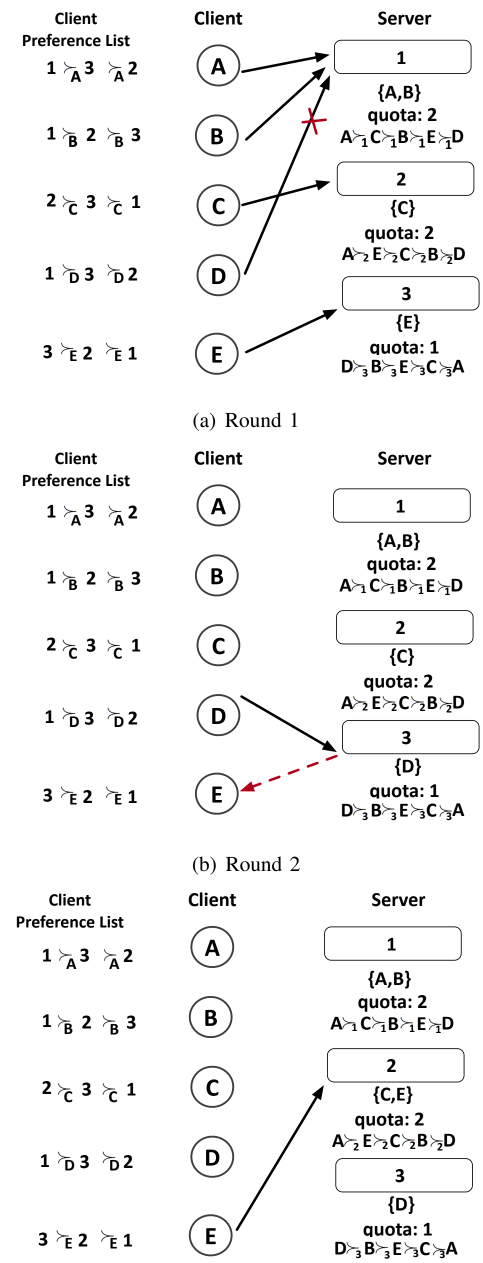
$$\delta_i^j \delta_i^{j'} = 0 \ \ \forall i \in \mathcal{M}, j, j' \in \mathcal{N}, \tag{6}$$

where q_i^j denotes the learning quality of client i for the task of server j that is defined in Eqn. (2). Eqn. (5) means the number of clients working for server j cannot be more than the number of the capacities $| R_j |$ that server j occupies. Eqn. (6) presents each time, one client can only train a model for one server. This formulated optimization problem is a mixed-integer nonlinear programming (MINLP) problem. We propose to utilize the matching algorithm to solve this proposed optimization problem in a decentralized way.

B. Matching Algorithm Preliminaries

1) Stable Marriage Matching: Matching theory is widely researched across various fields of study such as economics, mathematics, and computer science [11], [12]. The stable marriage matching problem (SMP) (*Man*, *Woman*, $\succ$) is the most fundamental one-to-one matching problem. Given two sets of elements of the same size, i.e., males and females, each person has a ranking of all members of the opposite sex by their personal preference. For instance, $man_i :\succ_{man_i}$ represents man i's preference list from the most favorite to the least favorite woman based on his preferences. Similarly, each woman has her preferences over men.

2) College Admissions Matching: The many-to-one college admissions matching model (*Student*, *College*, quota, $\succ$) consists of a finite set of colleges, a finite set of students and a finite non-negative quota $quota_{college_i}$ for $college_i \in College$.



(a) Round 1

(b) Round 2

(c) Round 3

Fig. 2. Toy Example of MAAIM.

Each college has its preferences over the firms, and based on the college's preferences for students, each college accepts a group of students below the college's quota.

3) Gale-Shapley Algorithm: The Gale-Shapley or "deferred acceptance" solution [13] to the SMP and College-Admission matching takes place in several rounds. Each member has a preference list that includes all members of the opposite group. In SMP, in the first round, each "single" male makes a proposal to their most preferred female based on their rankings. Each female tentatively thinks over the proposal by replying "maybe" and in the meantime while engaged, she rejects all other suitors. Each following round, while there are still males that are not engaged, each unengaged male proposes to his next most preferred female option regardless if the female is already engaged. The female then replies "maybe" if they are not currently engaged or if she is engaged and prefers the new male suitor over her current partner, she then rejects her current partner's proposal for the new one. This process continues until all males are engaged and by default all females as well.

C. Generating Preference Lists for Servers and Clients

The purpose of server j is to maximize its learning quality q_i shown as follows,

$$\text{Maximize} \quad \sum_{i \in \mathcal{M}} \delta_i^j q_i^j, \tag{7}$$

s.t.:

$$\delta_i^j \in \{0,1\}, \tag{8}$$

$$\sum_{i \in \mathcal{M}} \delta_i^j \leq \mid R_j \mid, \tag{9}$$

$$\delta_i^j \delta_i^{j'} = 0 \ \ \forall i \in \mathcal{M}, j, j' \in \mathcal{N}. \tag{10}$$

Therefore, the preference relation $\succ_j$ for j is established as

$$u \succ_j v \Leftrightarrow q_u^j \succ_j q_v^j, \ \ u, v \in \mathcal{M}. \tag{11}$$

Moreover, server j can accept as many as $\mid R_j \mid$ clients same time.

On the other hand, the goal of client i is to maximize its revenue, i.e., Maximize $\sum_{j \in \mathcal{N}} \delta_i^j (b_i - E_{ij})$, subject to $\delta_i^j \in \{0,1\}$ and $\delta_i^j \delta_i^{j'} = 0$. In the assumption, all servers have the same bid proposal, i.e., \$1. The unit energy cost is denoted as w. Therefore the client would prefer the server with the model which causes less energy consumption. Hence a preference relation $\succ_i$ for $i \in \mathcal{M}$ is established, as follows

$$k \succ_i l \Leftrightarrow \delta_i^k (b_i - E_{ik}) \succ_i \delta_i^l (b_i - E_{il}), \tag{12}$$

where $k, l \in \mathcal{N}$ and $\delta_i^k \delta_i^l = 0$.

D. Dynamic Many-to-One Matching Deferred Acceptance Algorithm

We employ the deferred acceptance algorithm for our proposed MAAIM mechanism as detailed in [13]. The specifics are outlined below.

1) Preference List Preparation: First of all, all clients and servers will start by evaluating the other side and preparing their preference lists. The client i constructs its preference list $\mathbb{PL}(i)$ according to (11) and server will construct its preference list $\mathbb{PL}(j)$ according to (12).

2) Tentative matching with clients' proposal: In the initial round, all clients propose to their most preferred server as per their preference list. Servers, upon receiving these proposals, assess the clients based on their preference lists. They then provisionally accept clients, considering both their preferences and their capacity quotas. ***Example***: In Figure 2(a), For demonstration, we've arbitrarily generated preference lists for clients and servers using Eqns. (11) and (12). Servers 1 and 2 can match with a maximum of 2 clients each, while Server 3 can only match with 1 client. In round 1: Server 1 receives proposals from clients A, B, and D. Given its quota and preference list, it accepts clients A and B, but declines client D. Client C proposes to and is matched with Server 2. Client E proposes to and is paired with Server 3.

3) Updated matching result with following round proposal: In subsequent stages of our matching process, servers re-evaluate their current list of clients when new proposals arrive. If a server hasn't filled its maximum client capacity, it will consider the new proposals based on its set preferences. However, if it's already at its limit and receives a proposal from a more preferred client, it will release its least preferred client. This released client will then approach the next server on its preference list. The process continues until every client is paired with a server or exhausts all its options. ***Continuted Example***: In Round 2, depicted in Figure 2(b), Client D which is still unmatched, approaches its next preferred server, Server 3. After evaluation, Server 3 decides it values Client D over its current match-Client E, leading to Client E being released. In Round 3, as shown in Figure 2(c), Client E which is now unmatched, approaches its subsequent choice, Server 2, which accepts it due to available capacity. This structured method ensures optimal matches for both servers and clients.

4) Stability: The stability of deferred acceptance matching is proved in [13].

5) Dynamic matching process: For each training iteration, servers will choose new clients by the matching algorithm. In each iteration, clients will broadcast training loss of each kind of data set to servers and calculate the energy lost. After that, the clients and servers will update their preference list by (12) and (11), and start a new round of matching.

IV. Performance Evaluation

A. Experimental Settings

In our experimental setup, we consider four independent servers, each of which is given a different learning task utilizing four distinct datasets: EMNIST (digits only), CIFAR10, SVHN, and Fashion MNIST (F-MNIST). We also assume that there are 48 clients available. We will consider more clients and servers in future work. Each client i has a local dataset with s_i data samples, where $s_i = p_i D_j$ and D_j is the size of each server's dataset, subject to $0 < p_i < 1$ and $\sum_i p_i = 0$. The four central servers aggregate the local updates performed on the clients and implement the MAAIM method that has been suggested to create reliable client-server pairings. We use distinct models for each dataset to achieve the best learning performance: LeNet-5 is used for training on EMNIST and F-MNIST, MobileNet for CIFAR 10, and EfficientNet for SVHN. In our experimental evaluations, we compare the accuracies of the random client selection-based FedAvg method ("random-FedAvg") with our suggested MAAIM approach. FedAvg [3] is the first aggregation approach used in federated learning, where the central server computes a weighted average of the local results from clients to update the global model. When implementing random client selection, we first randomize the order of all available clients before distributing clients to

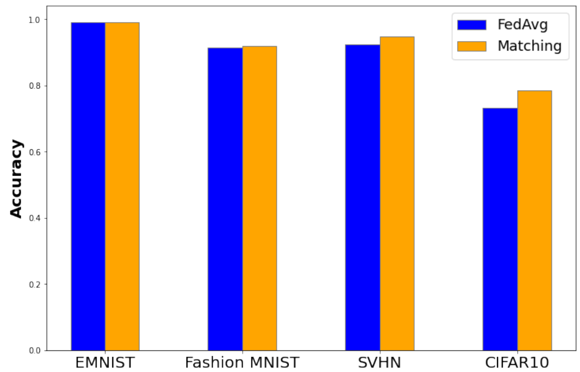


Fig. 3. Accuracy comparison between random-FedAvg and MAAIM.

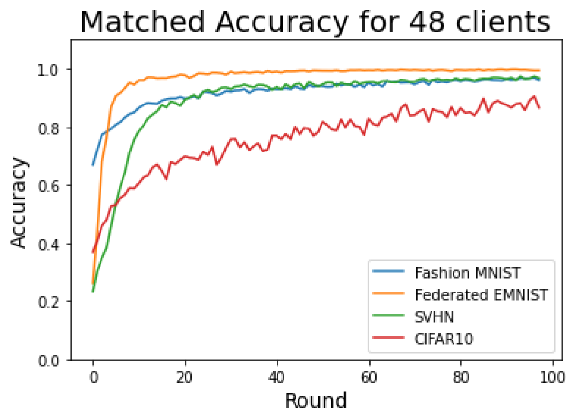


Fig. 4. Model accuracy of MAAIM scheme with 48 clients.

each server until we reach a predefined limit, ensuring a fair distribution of clients among the servers.

B. Experimental Results

Figure 3 compares the proposed MAAIM scheme to FedAvg in terms of model accuracy. The MAAIM technique starts a training round by calculating client preferences. According to Eqn. (12), each client evaluates the four servers' desirability based on elements including monetary gain and energy consumption. Servers then deliver the appropriate models to the chosen clients, who carry out local training. Using Eqn. (11), the servers determine their preference lists based on each client's level of learning. After assigning clients to servers using the matching algorithm, a training cycle takes place. We assume that each server chooses a total of 24 clients for the four servers in this experiment, dividing them evenly based on the MAAIM and random FedAvg algorithms. In other words, a total of $|R_j| = 6$ clients are chosen by each server. Across all four datasets, the experimental findings consistently show that the proposed MAAIM method works better than the FedAvg algorithm. The model performance of MAAIM method is shown in Figures 4. Each server is permitted to select $|R_j| = 12$ clients for local training in this particular trial. By keeping a check on the model's accuracy during each training round, we evaluate the effectiveness of each technique. The statistics undeniably show that the suggested MAAIM technique constantly outperforms the random-FedAvg method in terms of accuracy.

V. Conclusion

In this paper, we have proposed a distributive incentive mechanism MAAIM for FL, which considers learning quality, energy consumption, and economic revenue of the client. In MAAIM, we have formulated the learning quality comparison of the average test loss value to the averaging training loss value. We also formulate the energy cost with the consideration of the energy consumption of model training and communication energy consumption. We construct the preference lists for clients and servers based on their utility function, i.e. monetary gain and learning quality. We employ the many-to-one deferred acceptance algorithm in MAAIM and achieve stability. The experimental results show that the proposed MAAIM scheme can achieve better model accuracy than the benchmark scheme.

References

[1] M. Ribeiro, K. Grolinger, and M. A. Capretz, "Mlaas: Machine learning as a service," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Miami, FL, December 2015.
[2] X. Zhang, J. Ding, M. Wu, S. T. Wong, H. Van Nguyen, and M. Pan, "Adaptive privacy preserving deep learning algorithms for medical data," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Virtual, January 2021.
[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL, April 2017.
[4] S. Q. Zhang, J. Lin, and Q. Zhang, "A multi-agent reinforcement learning approach for efficient client selection in federated learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, March 2022.
[5] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1188–1200, October 2020.
[6] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.
[7] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, and Y. Zhang, "Fair: Quality-aware federated learning with precise user incentive and model aggregation," in *2021 IEEE Conference on Computer Communications (INFOCOM)*, Virtual, May 2021.
[8] R. Chen, Q. Wan, X. Zhang, X. Qin, Y. Hou, D. Wang, X. Fu, and M. Pan, "Eefl: High-speed wireless communications inspired energy efficient federated learning over mobile devices," in *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services (MobiSys'23)*, Helsinki, Finland, June 2023.
[9] Z. Meng, H. Xu, M. Chen, Y. Xu, Y. Zhao, and C. Qiao, "Learning-driven decentralized machine learning in resource-constrained wireless edge computing," in *IEEE INFOCOM-IEEE Conference on Computer Communications*, Vancouver, BC, Canada, May 2021.
[10] C. Ren, X. Lyu, W. Ni, H. Tian, and R. P. Liu, "Profitable cooperative region for distributed online edge caching," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4696–4708, 2019.
[11] J. Wang, W. Ding, Y. Guo, C. Zhang, M. Pan, and J. Song, "M3-step: Matching-based multi-radio multi-channel spectrum trading with evolving preferences," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 11, pp. 3014–3024, 2016.
[12] X. Zhang, R. Chen, J. Wang, H. Zhang, and M. Pan, "Energy efficient federated learning over cooperative relay-assisted wireless networks," in *2022 IEEE Global Communications Conference*, Rio de Janeiro, Brazil, December 2022.
[13] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.