2021 International Wireless Communications and Mobile Computing (IWCMC) | 978-1-7281-8616-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/IWCMC51323.2021.9498853

Federated Learning for Energy-balanced Client Selection in Mobile Edge Computing

Jingjing Zheng
CISTER Research Centre
Porto, Portugal
zheng@isep.ipp.pt

Kai Li*
CISTER Research Centre
Porto, Portugal
kai@isep.ipp.pt

Eduardo Tovar
CISTER Research Centre
Porto, Portugal
emt@isep.ipp.pt

Mohsen Guizani
Qatar University
Doha, Qatar
mguizani@ieee.org

***Abstract*—Mobile edge computing (MEC) has been considered as a promising technology to provide seamless integration of multiple application services. Federated learning (FL) is carried out at edge clients in MEC for privacy-preserving training of data processing models. Despite that the edge clients with small data payloads consume less energy on FL training, the small data payload gives rise to a low learning accuracy due to insufficient input to the FL training. Inadequate selection of the edge clients can result in a large energy consumption at the edge clients, or a low learning accuracy of the FL training. In this paper, a new FL-based client selection optimization is proposed to balance the trade-off between energy consumption of the edge clients and the learning accuracy of FL. We first show that this optimization problem is NP-complete. Next, we propose a FL-based energy-accuracy balancing heuristic algorithm to approximate the optimal client selection in polynomial time. The numerical results show the advantage of our proposed algorithm.**

***Index Terms*—client selection, mobile edge computing, federated learning, heuristic algorithm**

I. Introduction

Mobile edge computing (MEC) has been considered as a promising technology to enable mobile cloud computing, network control and storage [1]. In MEC, an edge server provides data processing services for computation-intensive and latency-critical applications of edge clients, such as augmented reality [2], unmanned aerial vehicles [3], and healthcare [4] etc. Fig. 1 presents a typical MEC system, where the edge clients collect patient physiological information, e.g., pulse rate, body temperature, and blood pressure. The data are sent to the server in the cloud for health analysis and reporting. Federated learning (FL) [5] is developed to train a shared MEC model at the central server without collecting the edge clients' data, thereby preserving data privacy of the clients. FL, as an emerging distributed machine learning method, is used by the edge clients in MEC to analyze the data. Specifically, a local model of FL is trained at the edge clients to classify the patient physiological information. The FL server aggregates the local models of the edge clients to train a shared global model of FL, while keeping data localized at the edge clients. Next, the global model is broadcasted to the edge clients who train and update the global model by using their local data. The new local models will be uploaded to the server for generating the updated global model in the next iteration. The process is iterated until the MEC achieves a desirable classification accuracy of the patient physiological analysis.

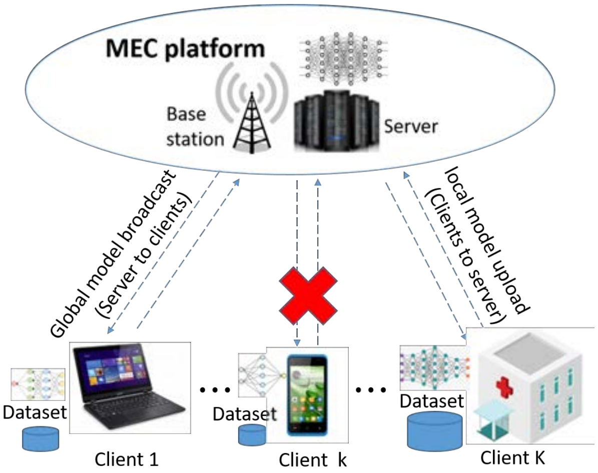

Fig. 1. Federated learning in MEC systems, where the selected clients upload their local models to the server.

Due to time-varying data quantity and channels, the energy consumption of the edge clients [6], [7] on data training and transmission can be greatly different from each other. Although scheduling the edge clients with a large dataset improves the learning accuracy of FL, analyzing large datasets rises energy consumption at the edge clients. Moreover, scheduling the edge clients with poor channel condition [8] to transmit requires a high transmit power at the client due to packet retransmissions. Therefore, client selection for balancing the energy consumption and the learning accuracy of FL is crucial in MEC.

In this paper, a client selection optimization is proposed to minimize the ratio of energy consumption and learning accuracy of FL in MEC. The optimization problem is formulated by nonlinear integer programming, with consideration of training time and required learning accuracy of FL. The contributions of this paper are as follows:

- To the best of our knowledge, this is the first attempt to investigate the client selection optimization in MEC to balance the tradeoff between the energy consumption

* Corresponding author.

978-1-7281-8616-0/21/$31.00 ©2021 IEEE

and learning accuracy of FL. We first demonstrate that the optimization problem is NP-complete.

- An energy-accuracy balancing heuristic algorithm, feadrated learning for accuracy-energy based client selection (FedAECS) is proposed to approximate the optimal solutions in polynomial time. In particular, FedAECS prioritizes the edge clients according to the learning time, data size, and channel quality. Furthermore, FedAECS recursively traverses the priority list and selects the edge clients, which minimizes the ratio of energy consumption and learning accuracy of FL.

The rest of this paper is organized as follows. Section II presents an overview of related works. The system model is studied in Section III. The proposed client selection optimization is formulated in Section IV. The FedAECS algorithm is developed in Section V. Numerical results are given in Section VI. Finally, the paper is concluded in Section VII.

II. RELATED WORK

This section presents the literature on client selection for FL in MEC framework. Federated Averaging (FedAvg) is studied in [5] to average the learning weights of the local models, where FL is used to reduce privacy and security risks. Since the client equiments are assumed to be homogeneous, FedAvg applies a random selection of the edge clients in MEC. In [9], FL-based client selection is developed to improve resource usage in MEC. Historical system dynamics and training outcomes are also used to analyze the convergence of the FL. It is difficult obtain the accurate resource information for all clients before the FL process is conducted, a multi-armed bandit-based client selection method is designed to to improve the trade-off between exploration an exploitation in the FL [10]. In [11], the central server is deployed to serve the edge clients, which transmit the training model of FL over shared channels. Since exchanging the channel state information results in large overheads, a multi-armed bandit-based framework is studied to schedule the edge clients to reduce the FL delay without the channel state information. In [12], federated edge learning coordinates global model training at the server and local model training at the edge clients over wireless links in MEC. Bandwidth allocation strategies are studied to reduce energy consumption of clients while improving learning efficiency of the FL.

A stateful FL heuristic is studied to schedule Internet-of-Things (IoT) devices to improve target accuracy of the FL [13]. The authors in [14] analyze the convergence rate of biased client selection. Based on the analysis results that biasing client selection towards clients with higher local loss achieves faster error convergence, an energy-efficient client selection framework is developed to flexibly span the trade-off between convergence speed and solution bias. Due to the differential computation capabilities of the clients' equipments, computation time of the local FL model can be highly different. [15] focuses on the training synchronization of the FL, where the length of each training iteration depends on the client with the longest training delay. FedCS selects the edge clients which can complete FL in one training iteration. In contrast, we focus on a new client selection optimization, which balances the tradeoff between the energy consumption and the learning accuracy of FL in MEC. We propose an energy-accuracy heuristic algorithm with FL to approximate the optimal client selection in polynomial time. In addition, FedCS and FedAvg are added as benchmarks in this paper for the performance evaluation of our proposed FedAECS. Details will be studied in Section VI.

III. SYSTEM MODEL

We consider a MEC system that consists of one server and K number of clients, as shown in Fig. 1. Each client k has a local dataset with $D_{i,k}$ samples in epoch i, where $i \in \{1, \cdots, I\}$. In epoch i, the edge clients firstly downloads the global model from the server. The client trains the local FL model with the updated parameters in the global model.

A. Energy Consumption Model

Let c_k denote the number of CPU cycles which is used by client k to train the local model in one iteration. Given $D_{i,k}$ data samples in one iteration, the number of CPU cycles required for client k to run one local iteration is $c_k D_{i,k}$ in every epoch i. Let f_k denote the CPU frequency of client k. The energy consumption of client k on the local model computation is:

$$E_{i,k}^{cmp} = U_{i,k} \zeta_k c_k D_{i,k} f_k^2 \tag{1}$$

where $U_{i,k}$ is the lower bound of the local iterations for achieving the training accuracy [16]. ζ_k is the effective capacitance coefficient of computing chipset for client k.

All selected clients upload their local FL parameters to the server once the training of the local model is completed. The transmission rate of client k in epoch i can be given by:

$$R_{i,k} = b_{i,k} \log_2 \left(1 + \frac{P_{i,k} G_k}{N_0 b_{i,k}} \right) \tag{2}$$

where $b_{i,k}$ is the bandwidth allocated to client k. $P_{i,k}$ denotes the transmission power consumption of client k in epoch i. G_k is the channel gain between client k and the server, and N_0 is the power spectral density of the Gaussian noise.

The data size that each client needs to upload is denoted by S_k, the transmit time $t_{i,k}^{up}$ can be given by:

$$t_{i,k}^{up} = \frac{S_k}{R_{i,k}} \tag{3}$$

By substituting (2) to (3) and multiplies transmission power, the energy consumption of client k on the transmission in epoch i is:

$$E_{i,k}^{up} = \frac{P_{i,k} S_k}{b_{i,k} \log_2 \left(1 + \frac{P_{i,k} G_k}{N_0 b_{i,k}} \right)} \tag{4}$$

Therefore, the total energy consumption $E_{i,k}$ of client that participating in FL process in epoch i:

$$E_{i,k} = V_{i,k} (E_{i,k}^{cmp} + E_{i,k}^{up}) \tag{5}$$

where $V_{i,k}$ is the number of global iteration of client k in epoch i, which is depicted in section III-B equation (7).

B. Traning Time of FL

We denote $t_{i,k}^{train}$ as the data processing time of client k in epoch i, which is:

$$t_{i,k}^{train} = U_{i,k}\frac{c_k D_{i,k}}{f_k} \tag{6}$$

The total completion time $T_{i,k}$ of client k in each epoch i can be given by:

$$T_{i,k} = V_{i,k}(t_{i,k}^{train} + t_{i,k}^{up}) \tag{7}$$

where $V_{i,k}$ is the number of global iteration of client k in epoch i increases with the local accuracy and guarantees the FL algorithm converge [16]. Note that the training time at the server is neglected since the server typically supports more powerful CPUs than the clients.

C. FL Accuracy

According to [17]–[19], the training accuracy of FL depends on the data size of the client.

$$\Gamma(\sum_{k=1}^{K}\beta_{i,k}D_{i,k}) = \log(1 + \sum_{k=1}^{K}\mu\beta_{i,k}D_{i,k}) \quad \forall i \in \mathcal{I} \tag{8}$$

where, $\beta_{i,k}$ is a binary indicator, $\beta_{i,k} = 1$ indicates that client k is selected for participating in the model training process in epoch i. Otherwise, $\beta_{i,k} = 0$. $\mu > 0$ is the system parameter.

IV. Problem Formulation

In this section, we formulate the optimization of balancing overall energy consumption of the edge clients and learning accuracy of FL:

$$\min_{\beta_{i,k}} \quad \sum_{i=1}^{I}\frac{\sum_{k=1}^{K}\beta_{i,k}E_{i,k}}{\Gamma(\sum_{k=1}^{K}\beta_{i,k}D_{i,k})} \tag{9}$$

$$\text{s.t.} \qquad 1 \leq \sum_{k=1}^{K}\beta_{i,k} \leq K \quad \forall i \in [1, I] \tag{10}$$

$$\beta_{i,k}T_{i,k} \leq T_{max} \quad \forall i \in [1, I], \quad \forall k \in [1, K] \tag{11}$$

$$\sum_{k=1}^{K}\beta_{i,k}b_{i,k} \leq B \qquad \forall i \in [1, I] \tag{12}$$

$$\Gamma(\sum_{k=1}^{K}\beta_{i,k}D_{i,k}) \geq \epsilon_o \quad \forall i \in [1, I], \quad \forall k \in [1, K] \tag{13}$$

$$P_k^{min} \leq P_{i,k} \leq P_k^{max} \quad \forall i \in [1, I], \quad \forall k \in [1, K] \tag{14}$$

$$f_k^{min} \leq f_k \leq f_k^{max} \quad \forall k \in [1, K] \tag{15}$$

$$0 \leq \epsilon_o \leq 1 \tag{16}$$

$$\text{var.} \qquad \beta_{i,k} \in \{0, 1\} \tag{17}$$

Constraint (10) ensures that the number of clients selected for the FL training is smaller than K, while at least one client is selected in each epoch. Constraint (11) defines that the total training time of FL in epoch i has to be shorter than the length of the epoch T_{max}. Constraint (12) guarantees that the total bandwidth of the selected clients should be within the bandwidth capacity B. Constraint (13) indicates that the selected clients enable the model accuracy of the server satisfies the lower bound value, which ranges in accuracy from 0 to 1. Moreover, the constraints of client's transmission power and CPU frequency are given in (14) and (15) , respectively.

V. Client selection based on Heuristic Algorithm

The proposed optimization problem in (9) — (17) is a typical 0-1 Multidimensional Knapsack Problem (MKP) [20]. Specifically, the items to be put in knapsacks are the clientswith energy consumption $E_{i,k}$, data size $D_{i,k}$, and bandwidth $b_{i,k}$. The capacity of the knapsack is equal to total bandwidth, the variable $\beta_{i,k}$ is an binary indicator of item(client) k, $\beta_{i,k}$ is set to 1 indicates that item k is selected. Otherwise, $\beta_{i,k}$ is set to 0. The total weight of the knapsack has a lower bound which is equal to the minimum requirement of accuracy constraint (13). In addition, it is note worthy that every item to be put into the backpack must obey the time constraint (11). Our goal is to select which clients for the training of FL to minimize the ratio of the energy consumption and learning the accuracy of the FL. Therefore, the proposed optimization is NP-complete.

For each epoch i, let $\Phi_i(\boldsymbol{\beta_i}) = \frac{\sum_{k=1}^{K}\beta_{i,k}E_{i,k}}{\Gamma(\sum_{k=1}^{K}\beta_{i,k}D_{i,k})}$, where $\boldsymbol{\beta_i}$ is a vector of clients selection. To balance the energy consumption and learning accuracy, a heuristic, named FL for accuracy-energy based client selection(FedAECS) algorithm is proposed, which is presented in Algorithm 1. Specifically, the clients that fulfill time constraint (11) and accuracy constraint (13) are selected and sorted into a list η_i ascendingly according to the ratio of the energy consumption and the FL accuracy. The unselected clients are saved into a vector $\boldsymbol{\beta_i'}$. FedAECS algorithm returns once a client can satisfy the accuracy constraint (13) of the optimization. Otherwise, FedAECS continues to traverses the list η_i until the client $\beta_{i,m}$ satisfies the accuracy constraint (13). The corresponding objective function value $\Phi_i(\boldsymbol{\beta_i^p})$ will be determined; meanwhile, FedAECS further explore the possible combinations of the client $\beta_{i,m}$ to find the optimal combination $\boldsymbol{\hat{\beta}_i}$, the corresponding objective function value $\Phi_i(\boldsymbol{\hat{\beta}_i})$ will be calculated, finally, the optimal objective function value $\Phi_i(\boldsymbol{\beta_i^*})$ will be chosen from $\Phi_i(\boldsymbol{\beta_i^p})$ and $\Phi_i(\boldsymbol{\hat{\beta}_i})$.

VI. Numerical Results

In this section, we evaluate the proposed FedAECS algorithm. We compare the performance with the optimal solution as well as the existing solutions in the literature. We also study the impact of the number of clients and T_{max} on the runtime of FedAECS.

Algorithm 1: FedAECS—Federated Learning for Accuracy-energy Based Client Selection Algorithm

```
Input: $f_k$, $D_{i,k}$, $P_{i,k}$, $b_{i,k}$ ;
Initialization;
for (each client $k$) do
    if ((11) and (12) hold) then
        Calculate $E_{i,k}$ according to (5);
        Calculate $\Gamma(\sum_{k=1}^{K} \beta_{i,k} D_{i,k})$ according to (8);
        Calculate $\{\frac{E_{i,k}}{\Gamma(\sum_{k=1}^{K} \beta_{i,k} D_{i,k})}\}$ and store it into
          $\eta_i$ one by one;
    else
        Store client $k$ into $\boldsymbol{\beta}_{\boldsymbol{i}}^{'}$;
    end
end
$\eta_i$ are sorted in descending order and $m = 1$;
while ($m \leq |\eta_i|$) do
    if ((13) holds for client $m$ ) then
        $\boldsymbol{\beta}_{\boldsymbol{i}}^{*} = (\beta_{i,m} = 1, \bigcup_{j \in |\eta_i| \setminus m} \beta_{i,j} = 0) \bigcup \boldsymbol{\beta}_{\boldsymbol{i}}^{'}$;
        $\Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}}^{*}) = \eta_{i,m}$;
        Break;
    else
        if ($m <= |\eta_i| - 1$) then
            $m = m + 1$ ;
            while ($m \geq 2$ and $m \leq |\eta_i|$) do
                if (Client $m$ satisfies (13) ) then
                    $\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{p}} = (\beta_{i,m} = 1, \bigcup_{j \in |\boldsymbol{\eta}_{\boldsymbol{i}}| \setminus m} \beta_{i,j} = 0)$;
                    $\Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{p}}) = \eta_{i,m}$;
                    if ( $(\beta_{i,1}, \cdots, \beta_{i,m-1}, \beta_{i,m} \equiv 1)$
                      satisfy (13) and (12) ) then
                        $\hat{\boldsymbol{\beta}}_{\boldsymbol{i}} =$
                          $\arg\min \Phi_i(\beta_{i,1}, \cdots, \beta_{i,m-1}, \beta_{i,m} \equiv$
                          $1)$;
                    else
                        Break;
                    end
                    $\Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}}^{*}) = \min\{\Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{p}}), \Phi_i(\hat{\boldsymbol{\beta}}_{\boldsymbol{i}})\}$;
                    if $\Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}}^{*}) = \Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{p}})$,
                      $\boldsymbol{\beta}_{\boldsymbol{i}}^{*} = \boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{p}} \bigcup \boldsymbol{\beta}_{\boldsymbol{i}}^{'}$; otherwise,
                      $\boldsymbol{\beta}_{\boldsymbol{i}}^{*} = \hat{\boldsymbol{\beta}}_{\boldsymbol{i}} \bigcup \boldsymbol{\beta}_{\boldsymbol{i}}^{'}$.
                else
                    $m = m + 1$;
                end
            end
        end
    end
    Output: $\boldsymbol{\beta}_{\boldsymbol{i}}^{*}$, $\Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}}^{*})$;
end
```

A. Simulation Configuration

For our simulations, we set $I = 1000$. The number of clients $K = \{20, 40, 60, 80, 100\}$, correspondingly, the total bandwidth $B = \{1, 3, 5, 7, 9\}$ MHz. For each client, $D_{i,k}$, b_k, f_k, $P_{i,k}$, c_k are uniformly distributed in $[2, 10]$ MB, $[50, 150]$ KHz, $[2, 4]$ Hz, $[4, 10]$ dBm, $[1, 3]$ cycles/bit respectively. A transmit data size $S_k = 100$ kbits. The number of local iterations $U_{i,k} = 10$ and global iterations $V_{i,k} = 4$. In addition, the noise power spectral density is $N_0 = -174$ dBm/Hz. The effective switched capacitance in local computation is $\zeta_k = 10^{-28}$ and the the system parameter $\mu = 1.7 \times 10^{-8}$. The proposed FedAECS is implemented in MATLAB R2015a, running on 3.0 GHz Intel core processor with 24 GB of memory.

We compare FedAECS with the following four client selection schemes, i.e., FedAvg [5], FedCS [15], Dataset-based and Energy-based policies.

FedAvg: the clients are randomly selected participating in FL training, while satisfying the bandwidth constraint.

FedCS: the FL server selects the clients as long as the bandwidth capacity constraint holds.

Energy-based client selection: the clients are prioritized by energy consumption, FL server selects a few number of clients while satisfying the bandwidth constraint.

Data-based client selection: the clients are prioritized by data size, FL server selects a few number of clients while satisfying the bandwidth constraint.

B. Simulation Results and Analysis

Comparing to optimal client selections. Table I summarizes running time, the ratio value of $\Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}})$ and Error($= \frac{|\Phi_i \text{ of FedAECS} - \Phi_i \text{ of CPLEX}|}{\Phi_i \text{ of CPLEX}} \times 100\%$). We assess the ratio value of FedAECS when the number of clients are increased from 2 to 10. Meanwhile, we conducted in ILOG CPLEX 12.10, and obtained optimal results, which are compared with the results that obtained by FedAECS. It is found that the optimal solution have the maximum difference which is 58.699% when $K = 10$. On average, FedAECS guarantees exactly the client scheduling schemes as optimal schedules. Moreover, FedAECS is much more efficient than CPLEX on runtime.

TABLE I
COMPARISON RESULTS WITH CPLEX.

# Clients	Cplex(Φ_i)	Runtime	FedAECS(Φ_i)	Runtime	Error
2	5.2786 W	19 s	5.3242 W	0.061026 s	0.864%
3	12.156 W	27 s	12.1782 W	0.034008 s	0.183%
4	14.478 W	44 s	14.5272 W	0.067833 s	0.340%
5	6.3921 W	21 s	8.8479 W	0.095634 s	38.419%
6	6.0202 W	27 s	6.0669 W	0.097590 s	0.776%
7	4.4662 W	19 s	4.4879 W	0.082551 s	0.486%
8	5.8882 W	21 s	5.9214 W	0.063147 s	0.564%
9	4.8528 W	21 s	4.9042 W	0.039516 s	1.059%
10	3.1643 W	33 s	5.0217 W	0.071169 s	58.699%

Fig. 2 demonstrates the advantage of FedAECS with respect to the objective function value $\Phi_i(\boldsymbol{\beta}_{\boldsymbol{i}})$. Considering the case of twenty clients and a edge server. For wireless communication model, the total bandwidth $B = 1$ MHZ, in addition, the threshold value of delay $T_{max} = 5$ s. As an illustration, it is evident that $\Phi_i(\boldsymbol{\beta_i})$ of FedAECS client selection scheme keeps it to a minimum at all times with the change of epoch. In particular, $\Phi_i(\boldsymbol{\beta_i})$ of FedCS-based client selection is about 5-folds as compared with FedAECS. The principal reason is that FedCS-based method can obtain more clients, whilst it incurs a substantial of energy consumption.

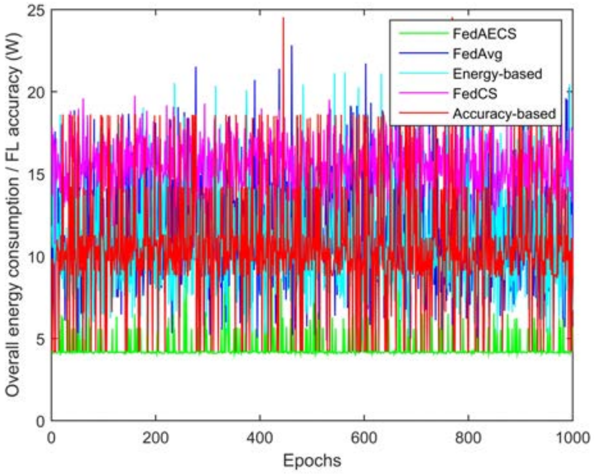

Fig. 2. The objective function value $\Phi_i(\boldsymbol{\beta_i})$ varying with epochs.

Fig. 3 shows that the objective function value $\Phi_i(\boldsymbol{\beta_i})$ varies with the total number of clients $I = \{20, 40, 60, 80, 100\}$, in response, total bandwidth is also increasing with the number of clients, that is $B = \{1, 3, 5, 7, 9\}$ MHZ. We calculate the mean value and standard deviation of ratio for 1000 epochs. Obviously, $\Phi_i(\boldsymbol{\beta_i})$ of FedAECS always keep the minimum, while $\Phi_i(\boldsymbol{\beta_i})$ of FedCS increasing with the more number of participating clients and more energy consumption. Moreover, with the participating clients increasing, the increasing of energy consumption is more significant than the increasing of FL model accuracy. In contrast, the ratio values of the others client selection methods not significantly affected by the number of clients.

Fig. 4 shows that the relationship among objective function value $\Phi_i(\boldsymbol{\beta_i})$, the number of clients and length of the epoch T_{max}. With the increasing of the total number of clients and T_{max}, the more potential clients will be selected to participate in the process of model training. However, When T_{max} increases to 6 seconds, $\Phi_i(\boldsymbol{\beta_i})$ will not change significantly with the increase of the number of users. This is mainly because the number of users available no longer changes significantly.

Fig. 5 shows that the relationship among objective function value $\Phi_i(\boldsymbol{\beta_i})$, T_{max} and lower bound of accuracy ϵ_0. It can be clearly seen that $\Phi_i(\boldsymbol{\beta_i})$ decreasing with the increasing of T_{max} and the decreasing of ϵ_0. Note that when $T_{max} = 1$ s and $\epsilon_0 = 0.07$ or $\epsilon_0 = 0.09$ the ratio values of $\Phi_i(\boldsymbol{\beta_i})$ are non-exist because the constraints are too strict to obtain available clients.

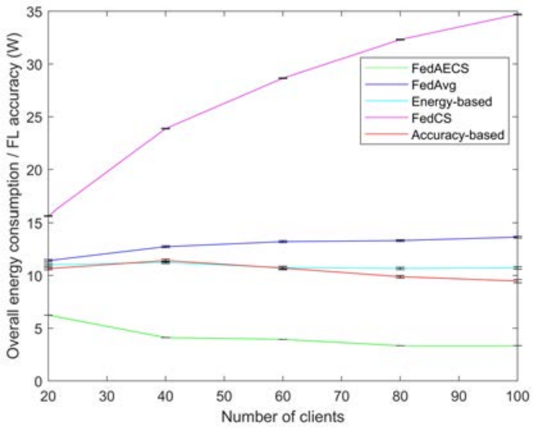

Fig. 3. The objective function value $\Phi_i(\boldsymbol{\beta_i})$ varying with the number of clients.

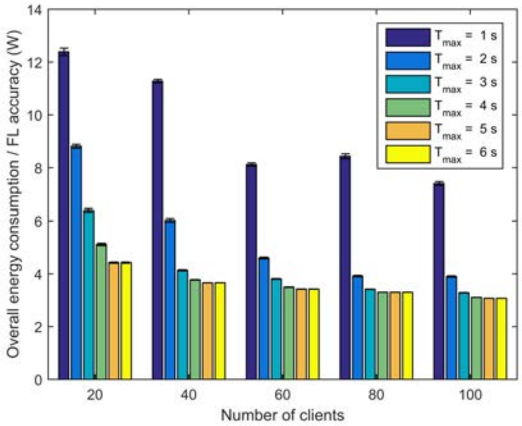

Fig. 4. The relationship among objective function value $\Phi_i(\boldsymbol{\beta_i})$, the number of clients, and length of the epoch T_{max}.

Fig. 6 illustrates that the runtime changes with length of the epoch T_{max} and the number of clients changes. The runtime is first decline and then moderate rise with the increase of T_{max} when the number of clients equal to 60, 80, 100. In particular, when T_{max} is greater than 3 seconds, for higher T_{max}, a longer run-time is required. On the other hand, the number of clients equal to 20 and 40, the runtime increases correspondingly from $T_{max} = 1$ s to $T_{max} = 2$ s in that more available candidate clients, and afterwards follows the same law as the number of clients equal to 60, 80, 100.

VII. Conclusion

In this paper, the FL-based client selection optimization is formulated to balance the total energy consumption of edge clients and learning accuracy of the FL. To solve the problem in polynomial time, we propose the suboptimal heuristic, FedAECS algorithm, which recursively assesses the potential

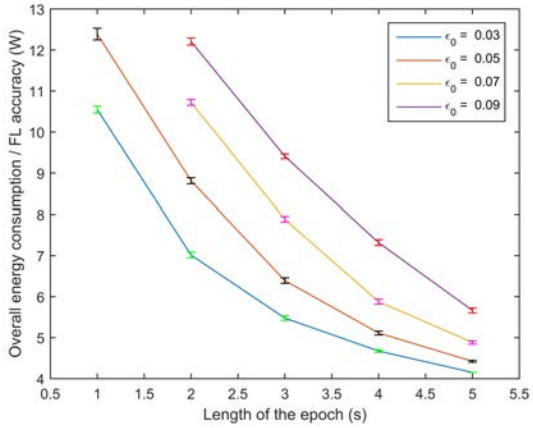

Fig. 5. The relationship among objective function value $\Phi_i(\boldsymbol{\beta_i})$, length of the epoch T_{max}, and model accuracy of lower bound value ϵ_0.

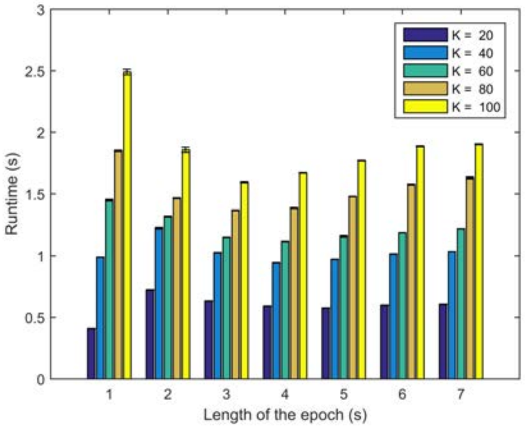

Fig. 6. The relationship among runtime, length of the epoch T_{max}, and the number of clients.

clients in FL to balance the energy consumption and learning accuracy. The numerical results show that the proposed FedAECS outperforms the existing solutions in the literature.

ACKNOWLEDGMENT

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (CEC/04234); also by the Operational Competitiveness Programme and Internationalization (COMPETE 2020) through the European Regional Development Fund (ERDF) and by national funds through the FCT, within project POCI-01-0145-FEDER-029074 (ARNET).

REFERENCES

[1] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4):2322–2358, 2017.

[2] Esther Z Barsom, Maurits Graafland, and Marlies P Schijven. Systematic review on the effectiveness of augmented reality applications in medical training. *Surgical endoscopy*, 30(10):4174–4183, 2016.

[3] C. Sun, W. Ni, and X. Wang. Joint computation offloading and trajectory planning for uav-assisted edge computing. *IEEE Transactions on Wireless Communications*, pages 1–1, 2021.

[4] Sawsan Abdul Rahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 2020.

[5] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[6] Lingyun Lu, Tian Wang, Wei Ni, Kai Li, and Bo Gao. Fog computing-assisted energy-efficient resource allocation for high-mobility MIMO-OFDMA networks. *Wirel. Commun. Mob. Comput.*, 2018:5296406:1–5296406:8, 2018.

[7] Kai Li, Branislav Kusy, Raja Jurdak, Aleksandar Ignjatovic, Salil S. Kanhere, and Sanjay K. Jha. κ-fsom: Fair link scheduling optimization for energy-aware data collection in mobile sensor networks. In Bhaskar Krishnamachari, Amy L. Murphy, and Niki Trigoni, editors, *Wireless Sensor Networks - 11th European Conference, EWSN 2014, Oxford, UK, February 17-19, 2014, Proceedings*, volume 8354 of *Lecture Notes in Computer Science*, pages 17–33. Springer, 2014.

[8] Kai Li, Wei Ni, Lingjie Duan, Mehran Abolhasan, and Jianwei Niu. SWPT: A joint-scheduling model for wireless powered sensor networks. In *2017 IEEE Global Communications Conference, GLOBECOM 2017, Singapore, December 4-8, 2017*, pages 1–6. IEEE, 2017.

[9] Yibo Jin, Lei Jiao, Zhuzhong Qian, Sheng Zhang, Sanglu Lu, and Xiaoliang Wang. Resource-efficient and convergence-preserving online participant selection in federated learning. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2020.

[10] Naoya Yoshida, Takayuki Nishio, Masahiro Morikura, and Koji Yamamoto. Mab-based client selection for federated learning with uncertain resources in mobile networks. *CoRR*, abs/2009.13879, 2020.

[11] Wenchao Xia, Tony QS Quek, Kun Guo, Wanli Wen, Howard H Yang, and Hongbo Zhu. Multi-armed bandit-based client scheduling for federated learning. *IEEE Transactions on Wireless Communications*, 19(11):7108–7123, 2020.

[12] Qunsong Zeng, Yuqing Du, Kaibin Huang, and Kin K Leung. Energy-efficient radio resource allocation for federated edge learning. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2020.

[13] Ihab Mohammed, Shadha Tabatabai, Ala Al-Fuqaha, Faissal El Bouanani, Junaid Qadir, Basheer Qolomany, and Mohsen Guizani. Budgeted online selection of candidate iot clients to participate in federated learning. *IEEE Internet of Things Journal*, 2020.

[14] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *CoRR*, abs/2010.01243, 2020.

[15] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[16] Zhaohui Yang, Mingzhe Chen, Walid Saad, Choong Seon Hong, and Mohammad Shikh-Bahaei. Energy efficient federated learning over wireless communication networks. *IEEE Transactions on Wireless Communications*, 2020.

[17] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 2020.

[18] Wei Yang Bryan Lim, Jianqiang Huang, Zehui Xiong, Jiawen Kang, Dusit Niyato, Xian-Sheng Hua, Cyril Leung, and Chunyan Miao. Towards federated learning in uav-enabled internet of vehicles: A multi-dimensional contract-matching approach. *CoRR*, abs/2004.03877, 2020.

[19] Latif U Khan, Shashi Raj Pandey, Nguyen H Tran, Walid Saad, Zhu Han, Minh NH Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine*, 58(10):88–93, 2020.

[20] Jakob Puchinger, Günther R Raidl, and Ulrich Pferschy. The multi-dimensional knapsack problem: Structure and algorithms. *INFORMS Journal on Computing*, 22(2):250–265, 2010.