Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Review:

A survey of energy-efficient strategies for federated learning in mobile edge computing*

Kang YAN[1], Nina SHU[1], Tao WU†‡[1,2], Chunsheng LIU[1], Panlong YANG[3]

[1]*School of Electronic Engineering, National University of Defense Technology, Hefei 230009, China*

[2]*Department of Computing, Hong Kong Polytechnic University, Hong Kong 999077, China*

[3]*School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China*

†E-mail: wutao20@nudt.edu.cn

Received Mar. 14, 2023; Revision accepted Sept. 14, 2023; Crosschecked Apr. 24, 2024

Abstract: With the booming development of fifth-generation network technology and Internet of Things, the number of end-user devices (EDs) and diverse applications is surging, resulting in massive data generated at the edge of networks. To process these data efficiently, the innovative mobile edge computing (MEC) framework has emerged to guarantee low latency and enable efficient computing close to the user traffic. Recently, federated learning (FL) has demonstrated its empirical success in edge computing due to its privacy-preserving advantages. Thus, it becomes a promising solution for analyzing and processing distributed data on EDs in various machine learning tasks, which are the major workloads in MEC. Unfortunately, EDs are typically powered by batteries with limited capacity, which brings challenges when performing energy-intensive FL tasks. To address these challenges, many strategies have been proposed to save energy in FL. Considering the absence of a survey that thoroughly summarizes and classifies these strategies, in this paper, we provide a comprehensive survey of recent advances in energy-efficient strategies for FL in MEC. Specifically, we first introduce the system model and energy consumption models in FL, in terms of computation and communication. Then we analyze the challenges regarding improving energy efficiency and summarize the energy-efficient strategies from three perspectives: learning-based, resource allocation, and client selection. We conduct a detailed analysis of these strategies, comparing their advantages and disadvantages. Additionally, we visually illustrate the impact of these strategies on the performance of FL by showcasing experimental results. Finally, several potential future research directions for energy-efficient FL are discussed.

Key words: Mobile edge computing; Federated learning; Energy-efficient
https://doi.org/10.1631/FITEE.2300181 **CLC number:** TN929.5

1 Introduction

With the rapid development of fifth-generation (5G) network technology and the Internet of Things (IoT), the number of end-user devices (EDs, e.g., smartphones and IoT devices) is surging and the applications are becoming more and more diverse, which leads to a large amount of data generated at the edge of networks. The traditional centralized mobile cloud computing, which collects the uploaded data of EDs and is processed centrally on the cloud, is usually facing various shortcomings, such as high latency and privacy concerns, and thus is unable to efficiently address the massive data of EDs generated at the edge (Jararweh et al., 2016). To deal with these issues, the innovative mobile edge computing

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 62002377, 62072303, 62072424, 61872178, and 62272223), the Hong Kong Scholars Program (No. 2021-101), and the High-Level Talent Fund (No. 22-TDRCJH-02-013)

ORCID: Kang YAN, https://orcid.org/0000-0002-0258-0817; Tao WU, https://orcid.org/0000-0003-1344-835X

(MEC) framework has emerged to enable low latency and efficient computing close to the user traffic. In contrast with the prolonged propagation exhibited by cloud computing, MEC strives to relocate mobile computing, storage, and network control to the edge of networks, such as base stations and access points (Mao et al., 2017). Due to the potential to reduce latency and enhance service quality, MEC has attracted much attention and is widely regarded as a key technology to realize various visions of the next-generation Internet.

In recent years, there has been a surge of progress in artificial intelligence (AI) technology and corresponding intelligent applications, such as smart medical care (Zeng SG and Wu, 2019) and driverless cars (Yurtsever et al., 2020), which have brought much convenience to people and have been greatly favored. Since data serve as fuel for AI model training, the abundance of data generated at the network edge makes machine learning (ML) tasks a predominant workload in MEC. However, in light of growing privacy concerns, users are reluctant to upload their data to the server for model training. Encouragingly, the emergence of federated learning (FL) provides a promising solution and enables privacy-preserving distributed ML at the network edge (Wang SQ et al., 2018; Lim et al., 2020). In FL, EDs use local data for model training and send model parameters only to the edge server while keeping the user data locally. The edge server aggregates and updates the models uploaded by each device into a new model, and broadcasts it to each device again. After several iterations, a convergent AI model is finally obtained. In addition to the advantage of privacy protection, FL helps save bandwidth, because the number of intermediate parameters uploaded tends to be much smaller than that in the training dataset.

In this paper, we focus on the energy efficiency of FL, which studies how to efficiently perform energy-consuming learning tasks on energy-constrained EDs. On one hand, both computation (e.g., ML model training that usually involves millions of parameters) and communication (e.g., uploading intermediate results) require a large amount of energy. On the other hand, the computing and communication resources of EDs are constrained, which restricts devices from participating in more learning tasks, and eventually hurts FL performance. Moreover, model training involves multiple iterations, requiring EDs to execute numerous rounds of computation and communication. These factors pose significant challenges to the practical application of FL in MEC. Therefore, designing energy-efficient strategies to address these challenges is highly meaningful and has become an active research topic.

Existing reviews related to FL focus on framework design, wireless communication, and security and privacy issues, with little emphasis on the energy consumption of the system. Lim et al. (2020) reviewed resource allocation methods that lower the communication cost of FL in mobile edge networks, but they did not discuss in depth the energy efficiency optimization methods. Niknam et al. (2020) investigated prospective applications of FL in 5G networks and discussed major technical challenges of FL in the domain of wireless communications. Yu and Li (2021) investigated the state-of-the-art resource optimization methods for FL. Imteaj et al. (2022) presented the challenges arising from applying FL in resource-constrained IoT environments and discussed the potential solutions. Shi et al. (2022a) provided an overview of the cutting-edge methods of merging the FL process with energy-efficient learning techniques, focusing primarily on model compression techniques. Zhao BR et al. (2022) analyzed the energy consumption challenges of FL within sixth-generation (6G) networks and proposed several feasible green designs for the FL-based 6G network architecture. In sum, none of these works make a detailed classification and summary of the existing energy-efficient strategies for resource-constrained devices in FL. Different from these reviews, in this paper, we provide a comprehensive survey of the state-of-the-art energy-efficient strategies in FL within the context of MEC.

Specifically, the system model and common energy consumption models in FL are first discussed. Then we analyze the challenges associated with improving energy efficiency in FL systems. Additionally, we provide a comprehensive summary and introduction to the existing energy-efficient strategies, which can be categorized into learning-based strategies, resource allocation strategies, and client selection strategies. We conduct an elaborate analysis of these strategies, meticulously comparing their advantages and disadvantages. Furthermore, we visually illustrate the influence of these strategies on

the performance of FL through the presentation of experimental results. Finally, we present several potential research directions for energy-efficient FL.

2 System model in federated learning

In this section, we detail the basic architecture of the FL system and present the energy consumption models, including computation and communication energy consumption models. The main variables and their definitions are listed in Table 1.

Table 1 Main variables affecting energy efficiency

Variable	Definition
M	Number of global iterations
N	Number of local iterations
B	Batch size for local model training
K_i	Number of scheduled clients
$\lvert\mathcal{D}_k\rvert$	Number of training data samples for device k
τ_k	Local model training duration for device k
f_k^{CPU}	CPU frequency for device k
P_k^{CPU}	CPU power for device k
V_k^{Gc}	GPU core voltage for device k
f_k^{Gc}	GPU core frequency for device k
f_k^{Gm}	GPU memory frequency for device k
P_k^{GPU}	GPU power for device k
r_k	Achievable transmission rate for device k
W_k	Bandwidth allocation for device k
p_k	Transmission power for device k
h_k	Channel gain for device k
t_k^{up}	Time of uploading model parameters for device k
s_k	Data size of model parameters for device k

2.1 Federated learning model

As shown in Fig. 1, an FL system in MEC usually consists of a central unit (model owner), e.g., the edge server, and a group of EDs (data owner), e.g., mobile phones. The edge server manages the collaboration of EDs to train a shared ML model without exchanging user data. The shared ML model is called the global model and the ML model trained by each device with its local dataset is called the local model.

Consider the FL system composed of an edge server and a group of EDs $\mathcal{K}=\{1,2,...,K\}$. Let $\mathcal{D}_k$ $(k=1,2,...,K)$ denote the local dataset of device k, and $|\mathcal{D}_k|$ refers to its dataset size. The global FL model is denoted by the parameter set ω. For each data sample $s\in\mathcal{D}_k$, let x_k^s denote its features and y_k^s its label. The loss function is represented by $f(\omega;x_k^s,y_k^s)$, measuring the prediction error of model

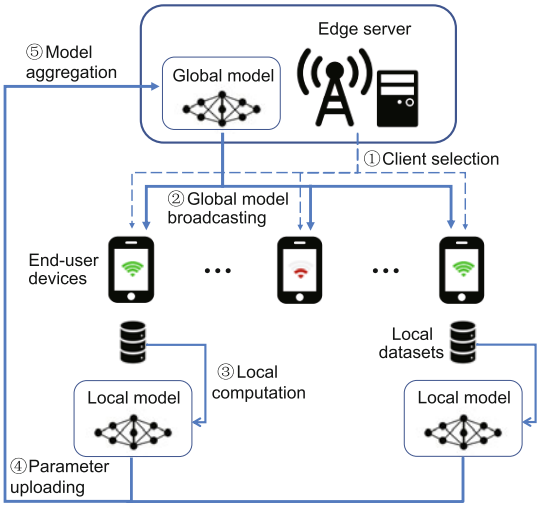

Fig. 1 Overview of the federated learning (FL) system in mobile edge computing (MEC)

ω on data sample s. Thus, the local loss function is

$$F_k(\omega)=\frac{1}{|\mathcal{D}_k|}\sum_{s\in\mathcal{D}_k} f\left(\omega;x_k^s,y_k^s\right). \tag{1}$$

Accordingly, the global loss function across all distributed datasets is denoted as

$$\begin{aligned} F(\omega)&=\frac{1}{|\cup_k\mathcal{D}_k|}\sum_{s\in\cup_k\mathcal{D}_k} f\left(\omega;x_k^s,y_k^s\right)\\ &=\frac{1}{|\cup_k\mathcal{D}_k|}\sum_{k\in\mathcal{K}}|\mathcal{D}_k|\,F_k(\omega). \end{aligned} \tag{2}$$

The FL process aims to find a desired model parameter set ω^* to minimize the global loss function $F(\omega)$.

The optimization process of the parameter set involves multiple rounds of global and local iterations. We use M to represent the number of global iterations and N the number of local iterations. For each global iteration $i\in\{1,2,...,M\}$, let $\omega^{(i)}$ represent the global model at the edge server after i rounds of global iteration, and $\omega_k^{(i,j)}$ the local model at device k after j rounds of local iteration of the global iteration i. Suppose that K_i devices are selected for participation in the global iteration i. The detailed process at each global iteration i is described as follows:

1. Client selection: The edge server selects K_i clients from a subset of its clients, namely the EDs, which meet requirements to participate in this round of training, e.g., mobile phones that currently have a wireless connection.

2. Broadcasting: The edge server sends the global model $\omega^{(i-1)}$ and selection indicators $\{\rho_k\}$ to all clients. When device k is selected, the indicator $\{\rho_k\}=1$; otherwise, $\{\rho_k\}=0$. The local model at each client is set as $\omega^{(i-1)}$, i.e., $\omega_k^{(i,0)}=\omega^{(i-1)}, \forall k \in \mathcal{K}_i$, where $\mathcal{K}_i$ denotes the set of selected clients.

3. Local model training: Each selected client iteratively trains its local model based on the gradient of the local loss function, i.e., $\nabla F_k(\omega)$, with its local dataset. In each local iteration $j \in \{1,2,...,N\}$, we have $\omega_k^{(i,j)}=\omega_k^{(i,j-1)}+\eta_1 \nabla F_k(\omega_k^{(i,j-1)})$, where η_1 denotes the learning rate.

4. Parameter uploading: After local computation, all the selected clients upload the resultant parameters to the server in the form of either trained parameter sets, i.e., $\omega_1^{(i,N)}, \omega_2^{(i,N)}, ..., \omega_{K_i}^{(i,N)}$, or gradients, i.e., $g_1^{(i)}, g_2^{(i)}, ..., g_{K_i}^{(i)}$, where $g_k^{(i)}$ $(k=1,2,...,K_i)$ represents the change of the model on device k after N rounds of local iteration.

5. Model aggregation: The server aggregates the parameters uploaded by clients, generally by a weighted average, and updates the global model. For the parameter set form, the server aggregates all the parameter sets $\omega_k^{(i,N)}$ from K_i clients and updates the global parameter set by $\omega^{(i)}=\frac{1}{|\cup_k \mathcal{D}_k|}\sum_{k\in\mathcal{K}_i}|\mathcal{D}_k|\ \omega_k^{(i,N)}$. For the gradient form, the server will aggregate the model gradients by $G^{(i)}=\frac{1}{|\cup_k \mathcal{D}_k|}\sum_{k\in\mathcal{K}_i}|\mathcal{D}_k| g_k^{(i)}$, and update the global model by $\omega^{(i)}=\omega^{(i-1)}+\eta_2 G^{(i)}$, where η_2 denotes the learning rate.

After M rounds of global iteration, the global model parameter set $\omega^{(M)}$ is set as the desired solution for FL, i.e., $\omega^* \leftarrow \omega^{(M)}$.

In this paper, we emphasize the energy efficiency of EDs, because they are energy-constrained. For the server, the energy consumption and delay of the aggregation in step 5 can be ignored, due to its sufficient power and high performance. For EDs, the energy consumption includes the computation energy consumption for training the local model in step 3 and the communication energy consumption for uploading parameters in step 4. Detailed descriptions of the energy consumption models for computation and communication will be provided in the following subsections.

2.2 Computation energy consumption models

With the increasing prevalence of high-performance co-processors in EDs, computing tasks can be performed on the central processing unit (CPU) or any of the co-processors, such as the graphics processing unit (GPU), in the system. Since most of the related works in FL consider training the local model on the CPU or GPU, we introduce the CPU and GPU energy consumption models, separately.

2.2.1 CPU energy consumption model

Let f_k^{CPU} be the CPU frequency of device k, which indicates the computation capacity. The computation time for local model training at device k is

$$\tau_k^{\mathrm{CPU}}=\frac{I_k C_k |\mathcal{D}_k|}{f_k^{\mathrm{CPU}}},\ \ \forall k \in \mathcal{K}, \tag{3}$$

where I_k represents the number of local iterations at device k and C_k denotes the number of CPU cycles required to process a data sample (Yang ZH et al., 2021). The power consumption of CPU is proportional to $f_k^{\mathrm{CPU}}(V_k^{\mathrm{CPU}})^2$, and f_k^{CPU} is approximately linearly proportional to V_k^{CPU}, where V_k^{CPU} is the circuit-supplied voltage (Mao et al., 2017). Thus, the power consumption of CPU at device k for local model training is

$$P_k^{\mathrm{CPU}}=\kappa\left(f_k^{\mathrm{CPU}}\right)^3, \tag{4}$$

where κ is a constant coefficient that depends on the CPU chip architecture (Yang ZH et al., 2021). Accordingly, the energy consumption of local model training on CPU is

$$\begin{aligned} E_k^{\mathrm{CPU}} &= \tau_k^{\mathrm{CPU}} P_k^{\mathrm{CPU}} \\ &= \kappa I_k C_k |\mathcal{D}_k| \left(f_k^{\mathrm{CPU}}\right)^2. \end{aligned} \tag{5}$$

2.2.2 GPU energy consumption model

Simulating the performance and energy usage of GPUs is challenging due to their complex hardware architecture and dynamic power characteristics (Abe et al., 2014; Mei et al., 2017b). Previous studies have used empirical and statistical methods to model GPU power consumption. In this paper, we introduce a concise GPU power model (Mei et al., 2017a). Different from CPU energy consumption, which is related mainly to CPU voltage or frequency, the major factors affecting the GPU power consumption include GPU core voltage, GPU core frequency,

and GPU memory frequency. We use V_k^{Gc}, f_k^{Gc}, and f_k^{Gm} to represent them, separately. Then, the energy consumption of GPU is

$$P_k^{\mathrm{GPU}} = P_k^{\mathrm{G0}} + \gamma_k f_k^{\mathrm{Gm}} + c_k^{\mathrm{G}} \left(V_k^{\mathrm{Gc}}\right)^2 f_k^{\mathrm{Gc}}, \tag{6}$$

where P_k^{G0} is the summation of the power consumption unrelated to the GPU voltage/frequency scaling, and γ_k and c_k^{G} are constant coefficients that indicate the sensitivity to memory frequency scaling and core voltage/frequency scaling, respectively (Hong and Kim, 2010).

GPU execution time for processing one data sample can be formulated as

$$t_k = t_0 + \frac{u}{f_k^{\mathrm{Gm}}} + \frac{v}{f_k^{\mathrm{Gc}}}, \tag{7}$$

where t_0 denotes the other component in task execution time, and the constant factors u and v represent how sensitive the task execution is to changes in GPU memory and core frequency, respectively. Due to the parallelism property of GPU, the computation time does not linearly increase as the batch size increases (Shi et al., 2022b). Thus, we use $d(B)$ to describe the relationship between the batch size and time consumption, where B denotes the batch size for local model training. Correspondingly, GPU execution time for local model training is

$$\begin{aligned}\tau_k^{\mathrm{GPU}} &= \frac{t_k I_k \left|\mathcal{D}_k\right| d(B)}{B} \\ &= \left(t_0 + \frac{u}{f_k^{\mathrm{Gm}}} + \frac{v}{f_k^{\mathrm{Gc}}}\right) \frac{I_k \left|\mathcal{D}_k\right| d(B)}{B}.\end{aligned} \tag{8}$$

Then, the energy consumption of local model training on GPU is

$$E_k^{\mathrm{GPU}} = \tau_k^{\mathrm{GPU}} P_k^{\mathrm{GPU}}. \tag{9}$$

2.3 Communication energy consumption models

The communication energy consumption is caused mainly by the model broadcasting of the edge server and the parameter uploading of EDs after local computation. We focus on the latter because energy-hungry EDs are the main object of energy consumption optimization. In particular, we consider three transmission schemes that are commonly used for EDs to upload their ML parameters, namely time division multiple access (TDMA), frequency division multiple access (FDMA), and non-orthogonal multiple access (NOMA).

2.3.1 TDMA-based transmission

In TDMA-based transmission schemes, EDs transmit their parameters to the edge server over different time slots. Let t_k^{up} represent the uploading duration allocated to device k. The achievable transmission rate of device k is

$$r_k^{\mathrm{TDMA}} = W \ln\left(1 + \frac{p_k h_k}{N_0}\right), \tag{10}$$

where W represents the channel bandwidth, N_0 denotes the background noise, p_k denotes the transmission power at device k, and h_k represents the channel gain between device k and the edge server (Tran NH et al., 2019). Let s_k represent the data size of the uploaded parameters at device k (in bits). Then the transmission rate of device k is

$$r_k^{\mathrm{TDMA}} = s_k / t_k^{\mathrm{up}}, \tag{11}$$

by which we can express the minimum transmission power for device k to upload its parameters with duration t_k^{up} as

$$p_k = \frac{N_0}{h_k}\left(\mathrm{e}^{\frac{s_k / t_k^{\mathrm{up}}}{W}} - 1\right). \tag{12}$$

Correspondingly, device k's energy consumption for uploading its parameters to the edge server is

$$E_k^{\mathrm{TDMA}} = t_k^{\mathrm{up}} p_k = \frac{t_k^{\mathrm{up}} N_0}{h_k}\left(\mathrm{e}^{\frac{s_k / t_k^{\mathrm{up}}}{W}} - 1\right). \tag{13}$$

2.3.2 FDMA-based transmission

In FDMA-based transmission schemes, EDs transmit their parameters to the edge server over different bandwidths. Let W_k denote the bandwidth allocation for device k and $\sum_{k\in\mathcal{K}_i} W_k = W$, where W is the total uplink bandwidth (Yang ZH et al., 2021). Then, the achievable rate of device k is

$$r_k^{\mathrm{FDMA}} = W_k \log_2\left(1 + \frac{p_k h_k}{W_k N_0}\right). \tag{14}$$

The time duration of uploading parameters can be expressed as $t_k^{\mathrm{up}} = s_k / r_k^{\mathrm{FDMA}}$. Thus, device k's energy consumption for uploading its parameters to the edge server is

$$E_k^{\mathrm{FDMA}} = p_k t_k^{\mathrm{up}} = \frac{p_k s_k}{W_k \log_2\left(1 + \frac{p_k h_k}{W_k N_0}\right)}. \tag{15}$$

2.3.3 NOMA-based transmission

In NOMA-based transmission schemes, EDs are allowed to send their parameters to the edge server over a common resource block. Because the uplink-NOMA scheme can enable an arbitrary decoding order, we assume that the EDs are decoded in the order of their indices in $\mathcal{K}_i$, from K_i to 1 (Wu Y et al., 2022). Thus, device k's throughput to the edge server can be written as

$$r_k^{\mathrm{NOMA}} = W \log_2 \left(1 + \frac{p_k h_k}{\sum_{j=1}^{k-1} p_j h_j + W N_0} \right). \quad (16)$$

Assuming that in a given duration t^{up}, all EDs $k \in \mathcal{K}_i$ must complete their task of uploading parameters. Then, with Eq. (16), and after some manipulations, we can express device k's minimum transmission power for uploading its parameters s_k to the edge server with duration t^{up} as

$$p_k = \frac{W N_0}{h_k} \left(2^{\frac{s_k}{t^{\mathrm{up}} W}} - 1 \right) 2^{\frac{1}{t^{\mathrm{up}} W} \sum_{j=1}^{i-1} s_j}. \quad (17)$$

Correspondingly, the energy consumption for device k to upload parameters is

$$\begin{aligned} E_k^{\mathrm{NOMA}} =& t^{\mathrm{up}} p_k \\ =& t^{\mathrm{up}} \frac{W N_0}{h_k} \left(2^{\frac{s_k}{t^{\mathrm{up}} W}} - 1 \right) 2^{\frac{1}{t^{\mathrm{up}} W} \sum_{j=1}^{i-1} s_j}. \end{aligned} \quad (18)$$

3 Energy-efficient strategies for federated learning

As previously mentioned, the two primary energy-consuming components of FL on EDs are local computation and wireless communications. As shown in the system model, energy efficiency can be affected by many factors including, but not limited to, the parameters in Table 1. In this section, we analyze the challenges regarding improving energy efficiency in FL systems and survey the recent progress of energy-efficient strategies in FL.

3.1 Challenges in improving energy efficiency

Since FL is dynamic and heterogeneous, traditional distributed ML energy-efficient methods cannot be directly applied. It is necessary to design new energy-efficient methods according to the characteristics of the FL system to meet new challenges. The main challenges affecting the energy efficiency of FL systems are as follows:

1. Computing capability and communication capability heterogeneity: More than 10 000 distinct smart devices at the edge are equipped with over 2000 different systems-on-a-chip featuring varying computing resources, such as CPUs, GPUs, and digital signal processors (Kim YG and Wu, 2020, 2021). Heterogeneity has a substantial impact on the energy efficiency and performance of FL systems, presenting significant challenges for optimizing energy utilization (Abdelmoniem et al., 2023; Yang CX et al., 2024). The heterogeneity within various EDs can lead to notable disparities in computing capabilities. Additionally, heterogeneity in communication capabilities among EDs arises due to uncertainties in wireless channels and the dynamic nature of EDs. Consequently, significant differences in latency and energy consumption exist among different EDs. Furthermore, the presence of diverse power constraints and local data sizes exacerbates these disparities. Considering all these factors, the straggler problem arises. As shown in Fig. 2, straggler presence degrades FL performance as all devices must wait for the slowest device, leading to inefficient utilization of time and energy resources.

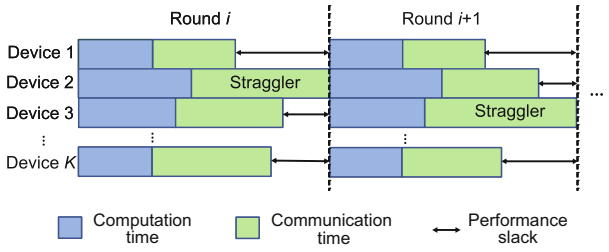

Fig. 2 Example of the straggler problem

2. Data heterogeneity: In the FL setting, EDs may have local datasets that follow different distributions; i.e., the datasets of EDs are not independent and identically distributed (non-IID) (McMahan et al., 2017). Due to the inconsistent data distribution, the local objectives of each ED are inconsistent with those of the global optimal solution, which ultimately leads to an average global model with much lower accuracy than that of the IID setting. Furthermore, under the non-IID setting, the model needs many more training rounds to reach convergence or even fails to converge, resulting in a large increase in energy consumption (Kim YG and Wu,

2021). Moreover, the data volume of devices can be heterogeneous. Devices with larger datasets require more computation resources, resulting in higher energy consumption.

3. Highly dynamic: ED computing and communication are highly dynamic due to the stochastic execution environment (Gaudette et al., 2016, 2019; Wu T et al., 2024). For local computation, FL tasks may experience reduced execution efficiency due to resource contention of co-running applications, resulting in longer computation time and higher energy usage. For wireless communication, the positions of EDs, such as mobile phones and smart cars, are dynamic and may change constantly, which leads to unstable network connections. As network conditions are deteriorated, devices will experience high bit error rates or encounter frequent dropouts and need to make more communication attempts or use higher transmission power to maintain a stable connection, leading to increased energy consumption in communication.

4. Increasing model complexity: The complexity of ML models has increased significantly with the development of deep learning models and the continuous improvement of the accuracy requirements of intelligent applications. An ML model usually contains millions of parameters. On one hand, highly complex computation tasks lead to high local computation energy consumption on EDs. On the other hand, the growing model parameter size significantly increases communication loads and energy consumption. Thus, striking a balance between model performance and complexity poses a significant challenge in improving energy efficiency in FL.

In conclusion, FL faces several challenges in improving energy efficiency. Heterogeneity in computing and communication capabilities across EDs creates variations in energy consumption and performance, resulting in the straggler problem. The heterogeneity of data, particularly non-IID data, leads to inconsistencies with the global optimal solution and necessitates more training rounds, increasing energy consumption. The highly dynamic execution environment with co-running applications and dynamic network connections adds to the energy efficiency variability. Additionally, the increasing complexity of ML models introduces energy-intensive local computations and communication loads. To address these challenges, existing works focus on optimizing energy efficiency in FL from two perspectives, namely, distributed learning and system design. The former seeks to employ learning-based strategies to increase the convergence rate of FL and enhance energy efficiency in the learning process. The latter concentrates on developing resource allocation strategies and client selection strategies for energy-efficient FL. The classification of these strategies is shown in Table 2. In the following, we will elaborate on these strategies.

3.2 Learning-based strategies

Since FL is a form of distributed learning, many optimization methods employed in distributed learning can be adopted in FL to enhance energy efficiency, while taking into account the unique properties of FL. These methods can be divided into three categories: model compression, hyperparameter optimization, and training algorithm improvement.

1. Model compression: Model compression techniques aim to reduce computation and communication costs by reducing the size of the model while ensuring its accuracy. Fig. 3 illustrates the impact of different quantization levels on energy consumption. In this experiment, simultaneous quantization of both model parameters and uploaded training results is performed. It can be observed that as the quantization level increases, the overall energy consumption of FL decreases accordingly. As depicted in Fig. 4, it is evident that moderate quantization has a minor impact on the accuracy of the model. However, excessively high levels of quantization have a detrimental impact on the model's accuracy, specifically when employing an 8-bit precision level. Therefore, when performing model compression, it is imperative to strike a balance between energy consumption and accuracy. Furthermore, considering the significant heterogeneity among EDs, it becomes crucial to develop tailored compression strategies that suit the characteristics of each specific device.

Li et al. (2021) developed an energy-efficient FL algorithm enabling flexible communication compression that allows participants to compress the gradients to different levels before uploading. The algorithm minimizes the energy consumption of FL training by controlling the compression parameter and the number of local iterations for each participant according to the communication condition and GPU capacity. Chen R et al. (2023) proposed

Table 2 Classification of energy efficiency optimization strategies

Strategy	Scheme	Reference	Advantage	Disadvantage
Learning-based strategies	Model compression	Abdelmoniem and Canini, 2021; Li et al., 2021; Prakash et al., 2022; Chen R et al., 2023	Significantly reducing computation and communication costs by reducing the complexity of the model	May result in compression errors and impact performance accuracy
	Hyperparameter optimization	Luo et al., 2021; Prakash et al., 2022; Shi et al., 2022b; Sun et al., 2024	Improving the model's accuracy, convergence speed, and generalization capability while enhancing resource utilization efficiency	Most of the existing works lack consideration for device heterogeneity
	Training algorithm improvement	Albaseer et al., 2021; Nguyen et al., 2021	Improving the quality of model updates and accelerating model convergence	Introducing additional computation overhead to the training process
Resource allocation based strategies	Computation resource allocation	Li et al., 2019; Zhan et al., 2020; Kim J et al., 2022	Increasing the utilization of computation resources and reducing training costs	1. Most existing strategies are based on static models and assumptions, which may not effectively adapt to the dynamic energy consumption requirements in real-world environments. 2. Optimizing the allocation of resources requires additional data transmission and sharing for the participating parties, which may introduce potential risks of privacy breaches and security vulnerabilities. 3. Most resource allocation strategies assume an ideal interference-free environment, but in reality, interference is inevitable, which can undermine their effectiveness.
	Communication resource allocation	Zeng QS et al., 2020; Hu et al., 2022	Increasing the utilization of communication resources and reducing communication costs	
	Joint C^2 resource allocation	Tran NH et al., 2019; Mo and Xu, 2021; Yang ZH et al., 2021; Zeng QS et al., 2021a; Battiloro et al., 2023	Increasing the utilization of communication and computation resources and reducing overall costs	
Client selection strategies	Direct energy-efficient client selection	Li et al., 2020; Zeng QS et al., 2020; Kim YG and Wu, 2021; Zheng et al., 2021; Albelaihi et al., 2022; Arouj and Abdelmoniem, 2022; Peng et al., 2023; Wu T et al., 2023	Reducing overall energy consumption by considering the heterogeneity among devices and deliberately selecting devices with better performance to participate in training	1. Bias in device selection may limit training data diversity and hinder the model's ability to generalize to diverse scenarios. 2. The majority of client selection strategies have failed to consider the dynamic characteristics of clients, such as their battery status and concurrent applications. 3. Client combinations and their impact on federated learning performance are disregarded in most strategies.
	Indirect energy-efficient client selection	Cho et al., 2020; Xu and Wang, 2021; Perazzone et al., 2022; Tang et al., 2022; Zhao JX et al., 2022	Accelerating model convergence and indirectly reducing overall energy consumption by selecting devices that contribute more significantly to model updates to participate in training	

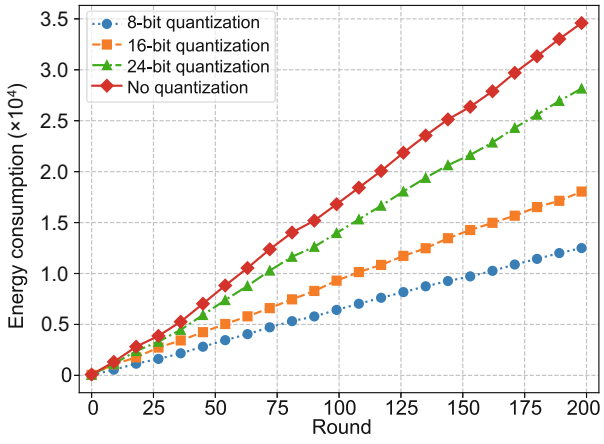

Fig. 3 Comparison of energy consumption at different quantization levels

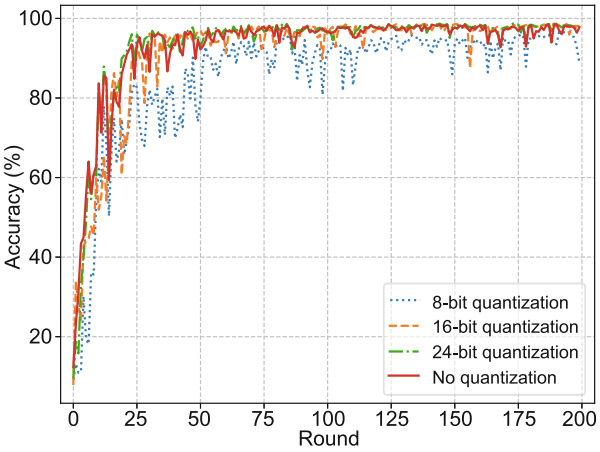

Fig. 4 Comparison of model accuracy at different quantization levels

flexible weight quantization (FWQ) methods to support efficient model training on heterogeneous EDs. According to the current storage budget, EDs are allowed to compress the shared model and train the model with low-precision weight (e.g., int8) to reduce the computation demand and the memory access frequency, resulting in lower energy consumption. The gradients uploaded by the EDs maintain high precision so that the global model can be updated in full precision. FWQ jointly determines bandwidth allocation and compression strategies for each participant to minimize energy consumption. Prakash et al. (2022) proposed a model compression based FL method called GWEP, which uses joint quantization and model pruning to reduce model redundancy and computation complexity. GWEP consists of three major components. First, weight quantization of the global model is performed to enable downlink compression. Second, the local model is pruned to desired sparsity to alleviate the computation burden and reduce the latency of model training. Third, the uplink load is alleviated by quantifying the gradients uploaded by EDs. In addition, error feedback is used to mitigate the impact of compression errors and recover the actual performance accuracy. Abdelmoniem and Canini (2021) investigated the influence of device heterogeneity on FL and showed that it has a substantial impact on model quality, resulting in reduced accuracy and potential convergence difficulties. To overcome these challenges, the AQFL approach was proposed which uses adaptive model quantization to accommodate the varying computing capabilities of clients and mitigate the negative effects of device heterogeneity on model performance.

2. Hyperparameter optimization: In Luo et al. (2021), an adaptive FL approach was presented which optimizes the number of participants (K) and the number of local iterations (E) per global round to minimize the total cost. The obtained theoretical properties show that a large K is beneficial in reducing the training time, while a small K conserves energy. It also shows that excessively small or large values of E are not cost-efficient, with the ideal value up to the correlation between communication and computation expenses. However, it is inappropriate to set the same number of iterations for all clients in this method, especially in scenarios with highly heterogeneous device resources. Sun et al. (2024) used multi-agent reinforcement learning to enable adaptive adjustment of the local iteration number by edge nodes, optimizing the tradeoff among model performance, energy consumption, and time delay in response to network dynamics. Prakash et al. (2022) adopted a distributed adaptive stochastic gradient method with an adaptive learning rate to accelerate the FL training process. Additionally, they established a theoretical proof of convergence of the proposed method. Shi et al. (2022b) proposed a dynamic batch size approach, called DBFL, to assist FL. DBFL allows users to exponentially increase batch sizes with an incremental factor, which can significantly reduce time and energy consumption by lowering the required number of communication rounds. In addition, experiments showed that the utilization of large batch training in the later stages can effectively improve the time efficiency and energy efficiency of the computation due to the parallel computing capabilities of GPUs.

3. Training algorithm improvement: Nguyen et al. (2021) proposed an FL algorithm that tackles data heterogeneity and system heterogeneity of devices by incorporating an appropriate weight-based proximal term into each local loss function. The proximal term can guarantee that the new local models do not deviate too much from the global model, allowing devices to produce valuable local models and speeding up convergence. In addition, an efficient sampling strategy was developed to substitute partial user participation. Albaseer et al. (2021) proposed a refined local training approach for intelligently identifying data samples that can improve the quality of the model. The algorithm uses the global model to filter the local data samples by excluding those with the classification probability exceeding a pre-set threshold, as they make insignificant contributions to the learned model. As the number of low-quality samples decreases, the time efficiency and energy efficiency of training are improved.

In conclusion, learning-based strategies aim to improve training efficiency and reduce energy consumption in FL by applying training and optimization techniques from ML. Among these strategies, model compression aims to reduce training and communication energy consumption by reducing model complexity. However, it is crucial to strike a balance between compression level and model accuracy, as excessive compression may increase the number of communication rounds and compromise

accuracy. Therefore, research should focus on minimizing model complexity while maintaining acceptable accuracy levels. Furthermore, it is crucial to delve into the research of adaptive model compression strategies that address the heterogeneous nature of devices, considering factors such as computation resources, communication conditions, and data volume. Hyperparameter optimization is a strategy to improve the energy efficiency of FL by adjusting hyperparameters such as the number of iterations and learning rate. Optimizing hyperparameters can accelerate the convergence of the model and reduce energy consumption. However, due to the heterogeneity of EDs, different EDs may have different optimal hyperparameters. Existing hyperparameter optimization strategies mostly do not consider the heterogeneity of devices, and further research is needed in this area. In addition, some strategies enhance the efficiency of FL by improving the model training algorithm, such as adding proximal terms to the loss function and filtering low-quality samples. However, these methods introduce additional computation overhead to the training process. Moreover, the current model aggregation methods in most FL algorithms are still based on the weighted aggregation of data volume. More efficient aggregation methods are yet to be researched.

3.3 Resource allocation based strategies

As EDs typically have limited resources, optimizing resource allocation is crucial to enhance the energy efficiency of the FL system. Through effective resource management, it is possible to maximize resource utilization and energy efficiency while minimizing the negative impact caused by stragglers. For instance, the presence of stragglers may result in idle time for faster clients during training. To fully use this idle time and reduce training energy consumption, we can adjust the CPU frequency of faster clients in each training round to slow down their training. The effectiveness of this approach in energy savings is illustrated in Fig. 5.

FL involves two primary resources: computing resources (e.g., clock frequency) and communication resources (e.g., transmission power and bandwidth). Consequently, existing works on energy-efficient strategies for resource allocation can be categorized into computation resource allocation strategies, communication resource allocation strategies,

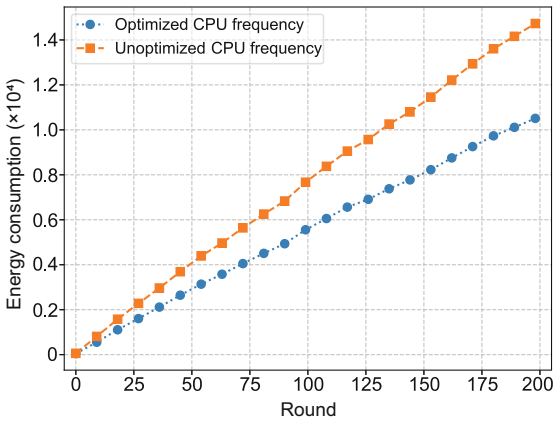

Fig. 5 Comparison of energy consumption with or without optimized CPU frequency

and joint computation-and-communication (C^2) resource allocation strategies.

1. Computation resource allocation strategies: Li et al. (2019) developed SmartPC, a framework that achieves energy-efficient FL by optimizing the balance between training time and model accuracy. SmartPC intelligently sets the training deadline in each global round, based on the EDs' hardware configuration and runtime behavior, to allow a statistically significant proportion of the devices to complete their global iterations. Then, SmartPC minimizes the energy consumption of the participating devices by determining the optimal CPU frequency while meeting the training deadline. Zhan et al. (2020) suggested enhancing energy efficiency in FL by reducing the CPU frequency of the EDs that are faster within the training group. The FL speed was preserved because the duration of the global iteration is affected only by the slowest device. Based on this idea, an optimization problem was formulated, whose objective is to minimize a weighted sum of training time and energy consumption. Then, an experience-driven algorithm was designed to converge on a nearly optimal solution. Kim J et al. (2022) proposed an energy-efficient method that finds the optimal dataset quantity and the CPU frequency used for local training over multiple FL clients, considering the tradeoff between the learning speed and the total energy consumption of participants.

2. Communication resource allocation strategies: Zeng QS et al. (2020) proposed radio-resource-management approaches for combined bandwidth allocation and scheduling to minimize the total

energy consumption under a learning speed constraint. The derived optimal policy for energy minimization suggests that devices with weaker channels or inferior computation capabilities, which often become stragglers in FL, should receive more bandwidth allocation. Furthermore, participating devices are suggested to make use of all allowable uploading time to decrease transmission power, which leads to less energy consumption. However, the bandwidth model in this work is the ideal case where there is no interference problem. Hu et al. (2022) considered the application of FL in a general cellular network characterized by inter-cell interference, and put forth a device scheduling and channel allocation approach that simultaneously guarantees model performance and optimizes energy efficiency.

3. Joint C^2 resource allocation strategies: Tran NH et al. (2019) formulated the FL over wireless networks as an optimization problem, FEDL. The objective of FEDL is to minimize the total learning time and energy consumption across all devices by adjusting several factors, including the CPU cycle frequency, local iteration accuracy threshold, communication time fraction of each device, local iteration time, and communication time in a global iteration. Because minimizing both FL time and energy consumption can conflict with each other, they employed a Joules-per-second weight coefficient to determine a Pareto-optimal tradeoff between the two objectives. The closed-form optimal solution is obtained by decomposing and transforming the non-convex FEDL problem into three convex subproblems. However, FEDL mandates synchronized uploading of local models across all devices. Yang ZH et al. (2021) proposed a joint C^2 resource allocation strategy for FL, where the transmission is based on FDMA so that devices are not required to upload parameters synchronously. They first derived the convergence rate for the considered FL algorithm. Then, an iterative algorithm was developed to derive the optimal time allocation, bandwidth allocation, power control, computation frequency, and learning accuracy to minimize the overall energy consumption of the system. Zeng QS et al. (2021a) designed a joint C^2 resource management framework on the heterogeneous mobile architecture where parallel computing uses both CPU and GPU. The framework aims to minimize the total energy consumption of EDs under delay and accuracy constraints by jointly controlling four dimensions: bandwidth allocation, C^2 time division, CPU–GPU workload partitioning, and CPU–GPU frequency scaling. Battiloro et al. (2023) proposed a method that dynamically optimizes C^2 resource based on Lyapunov stochastic optimization tools, to minimize the power expenditure of the FL system under time and model accuracy constraints. Mo and Xu (2021) considered resource allocation under two transmission schemes for EDs to upload parameters, based on NOMA and TDMA, separately. Under both schemes, the transmission power and rates, as well as CPU frequencies at EDs, are jointly adjusted to minimize the energy consumption under a latency constraint. According to the derived conclusions and experimental verification, NOMA performs better than TDMA in terms of delay and energy consumption in this design. Numerical results showed that the joint C^2 design can better enhance energy efficiency in FL by appropriately balancing the energy between communication and computation, compared to benchmark schemes considering only communication design or only computation design.

In summary, existing resource allocation strategies in FL primarily focus on optimizing CPU frequency, bandwidth allocation, transmission power, and other computing and communication resources to improve energy efficiency. By effectively allocating these resources, the utilization rate can be maximized and the impact of stragglers can be reduced. However, most existing strategies are based on static models and assumptions, which may not adapt well to dynamic energy consumption requirements in real-world environments. Additionally, to optimize resource allocation, clients may need to engage in additional data transmission and sharing, which can potentially expose them to risks of privacy breaches and security vulnerabilities. Furthermore, there is a lack of practical validation and research on the adaptability and scalability of these strategies in different scenarios and applications. Therefore, further research and improvement of resource allocation strategies are crucial. One potential direction is to incorporate dynamic factors inherent in real-world environments into the resource allocation process. This could involve considering variables like fluctuating workloads, varying energy availability, and evolving user behavior disruptions during the allocation of resources. Additionally, exploring methods

for preserving user privacy during resource allocation without the need to disclose personal device information is an area that merits further investigation. This research endeavor could involve techniques such as differential privacy, secure multi-party computation, or encryption-based approaches to ensure that sensitive user data remain protected throughout the resource allocation process.

3.4 Client selection strategies

Because of the system heterogeneity and data heterogeneity of EDs, it is unreasonable to randomly select EDs to participate in training or directly let all EDs participate in training in FL. Selecting inappropriate clients, e.g., clients with poor communication conditions, to participate in the training will slow down the learning process and increase energy consumption. Thus, the selection of clients is a critical factor for the practical and energy-efficient deployment of the FL system. The client selection strategies for energy efficiency optimization can be divided into a direct energy-efficient strategy, which aims to reduce the energy consumption of each round of global iteration, and an indirect energy-efficient strategy, whose goal is to increase the convergence rate of FL, indirectly reducing energy consumption.

1. Direct energy-efficient client selection strategies: One straightforward approach for saving energy is to directly exclude devices with high energy consumption. Fig. 6 presents the energy performance at various filtering levels, with λ representing the filtering level. For example, $\lambda = 0.2$ indicates that only 20% of devices with the lowest energy consumption are allowed to participate in training, while $\lambda = 1.0$ means that all clients can participate. From Fig. 6, it can be observed that by excluding clients with high energy consumption, the overall energy usage of the FL system can be effectively reduced. Fig. 7 shows the impact of different filtering levels on model accuracy. It can be observed that at lower filtering levels, there is no significant effect on model accuracy. However, when the filtering level is high, such as $\lambda = 0.2$, there is a noticeable decrease in model accuracy. This is due to excessive filtering, which leads to the exclusion of numerous clients from training, thereby reducing data diversity and compromising the accuracy of the model. Therefore, when selecting clients for participation, it is crucial to prioritize the maximum device inclusion while simultaneously enhancing energy efficiency. This requires careful consideration of the tradeoff between energy consumption and accuracy to strike the right balance.

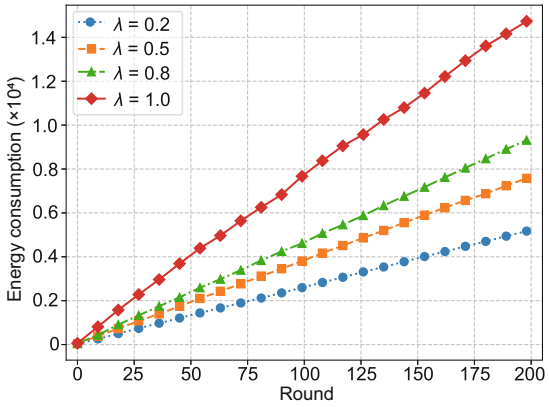

Fig. 6 Comparison of energy consumption at different filtering levels

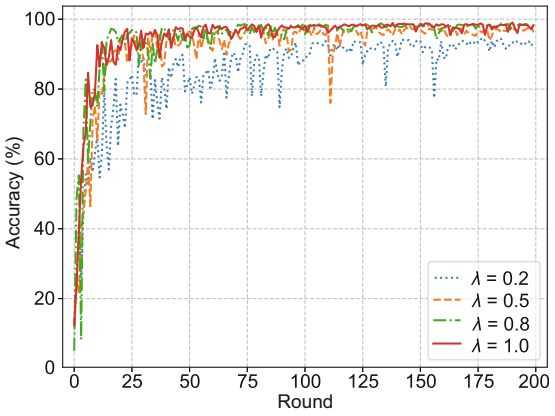

Fig. 7 Comparison of model accuracy at different filtering levels

Zeng QS et al. (2020) proposed a bandwidth allocation and scheduling method to improve energy efficiency for heterogeneous devices. Client scheduling is formulated as an optimization problem that minimizes the total energy consumption and maximizes the learning speed measured by the number of scheduled clients. The derived solution indicates that devices with better channel quality and computation power should have higher scheduling priorities. Zheng et al. (2021) developed an energy–accuracy balancing scheme called FedAECS to optimize client selection in FL. FedAECS prioritizes clients based on data size, training time, and the communication condition, and selects clients to minimize energy consumption while maximizing model

accuracy. However, these works select devices mainly from the perspectives of computing power, network quality, and data volume, while ignoring the impact of device runtime behavior (e.g., concurrent applications).

Li et al. (2020) experimentally observed that a relatively small number of devices involved in training does not have a great impact on the accuracy of the model, and proposed a control framework for energy-efficient FL called MCFL. MCFL first estimates the training capability of each device based on its hardware configuration and the impact of resource competition with concurrent applications, which is predicted by a long short-term memory model trained with the history training ability information. Then, MCFL selects devices to participate in training to minimize the training time limit based on training capability and dataset size. Moreover, MCFL minimizes the training energy consumption of each participant by controlling the memory bandwidth and CPU frequency while meeting the time limit. Kim YG and Wu (2021) proposed a framework called AutoFL that simultaneously optimizes the learning speed and energy efficiency of FL by considering the stochastic nature of edge execution. AutoFL is based on a reinforcement learning algorithm that learns the optimal device scheduling and per-device execution targets to minimize the system energy consumption in different execution environments. Peng et al. (2023) proposed a client selection scheme called FCE^{2}DS to jointly optimize the total energy consumption and the learning speed, subject to constraints on the data size and time consumption. To obtain more accurate energy consumption calculations, an online bandit learning method was incorporated to enhance the estimation of the CPU-cycle frequency of each device. Then, FCE^{2}DS selects devices in each round of global training according to energy consumption and local loss values. Devices with low energy consumption and large local losses, which contribute to quick convergence, are preferred. Arouj and Abdelmoniem (2022) revealed the significant influence of client dropout, resulting from battery constraints, on the performance and practical implementation of FL. To mitigate this issue, an energy-aware client selection method was proposed which strategically weighs the tradeoff between remaining power and time. By doing so, it effectively reduces client dropout in battery-powered scenarios while maintaining low convergence time and achieving energy savings. Albelaihi et al. (2022) proposed a green FL framework in which devices are powered by batteries and collect green energy from the environment to prolong battery life. They further introduced an energy-aware client selection method aimed at optimizing the balance between maximizing the number of selected clients and minimizing the energy consumption of their batteries, while ensuring that all selected clients have sufficient energy to upload their local models before the deadline. Wu T et al. (2023) considered the problem of joint edge aggregation and association in a multi-cell FL scenario, aiming to determine the aggregation location of the global model and the association of edge base stations with edge devices, as well as the resource allocation, to minimize the total cost.

2. Indirect energy-efficient client selection strategies: Cho et al. (2020) analyzed the convergence of federated optimization under biased client scheduling schemes and found that scheduling clients having high local loss can lead to a quick convergence of the model. Accordingly, an efficient client scheduling framework called POWER-OF-CHOICE was proposed. In POWER-OF-CHOICE, the server first samples a candidate set of clients and lets them compute their local loss based on the current global model. Then, the server schedules the clients that have the largest loss values to participate in the training during the next round. Xu and Wang (2021) experimentally showed that client selection for different periods of learning rounds has different effects on learning performance. The ascending client selection pattern, which selects fewer clients in the early learning rounds and gradually adds more clients in the later rounds, achieves higher accuracy and is more robust than other patterns. The reason is that the learning performance of early learning rounds is less sensitive to the number of selected clients; on the contrary, pushing accuracy even higher in the late learning rounds requires more clients to participate in the training. Accordingly, a scheme called OCEAN was developed to optimize client selection and bandwidth allocation under long-term client energy constraints. Zhao JX et al. (2022) proposed FedNorm, a client selection framework that measures the priority of a client with the local change of weight. This strategy gives preference to clients with large local weight changes because

more weight changes contribute to quicker convergence. Tang et al. (2022) investigated the correlations between clients and proposed a correlation-based client scheduling scheme called FedGP to improve the convergence speed. The loss correlation between clients was modeled using a Gaussian process. In this scheme, clients with similar datasets are considered redundant because they produce similar local updates, and the clients that are less correlated with other scheduled clients are preferred for selection. Perazzone et al. (2022) investigated the client selection method to minimize the communication costs in a fading communications channel while ensuring convergence. The convergence bound for non-convex functions using FL with arbitrary selection probabilities was first developed. Based on this bound, an optimization problem that minimizes the communication time was formulated and a stochastic optimization method was used to solve the problem. By intelligently selecting clients, the model can converge in less time and thus use fewer network resources even without direct knowledge of the underlying channel distribution. Simulation results showed that the higher the heterogeneity of wireless connections, the greater the gain of adopting this selection strategy.

In summary, existing energy-efficient client selection strategies can be classified into two categories. The first category focuses on selecting devices with better performance based on system heterogeneity, aiming to directly reduce energy consumption. The second category considers data heterogeneity and selects devices that are more beneficial for improving model accuracy, thereby accelerating model convergence and indirectly reducing energy consumption. However, there are limitations to existing client selection strategies. First, the bias in client selection preferences can lead to a situation where certain clients are frequently chosen while others are neglected. This imbalance can have a detrimental effect on the diversity of training data and ultimately impair the performance of the model. Hence, it is crucial to ensure fairness in the selection process when designing client selection strategies. Moreover, most of these strategies consider mainly factors such as computing capability, network quality, and data volume, while ignoring the impact of device runtime behavior, such as concurrent applications. Additionally, most of the existing client selection strategies rely on fixed priority-based selection without considering the impact of client combinations on FL performance, which may result in suboptimal selections. Therefore, further research and improvement of client selection strategies are necessary to enhance the efficiency of FL systems. These strategies should consider factors such as dynamic device characteristics and client combinations.

3.5 Other energy-efficient strategies

In addition to these strategies that we have described in detail, several works introduced the latest techniques to address the energy consumption challenges in FL.

da Silva et al. (2021) presented a method using simultaneous wireless information and power transfer (SWIPT) to enable both wireless power charging and communication for FL. They also investigated the relationship between the number of communication rounds and communication round time, aiming to achieve efficient model learning with fewer communication rounds and minimal battery depletion. Zeng QS et al. (2021b) suggested using wireless power transfer (WPT) to power-energy-constrained EDs in FL and presented the optimal tradeoff between power source settings and model convergence. Wu Y et al. (2022) studied NOMA-assisted FL via WPT and provided a joint optimization solution to minimize the system cost. Tran HV et al. (2020) used light energy harvesting to address the problem of energy constraints in EDs and developed a resource allocation method to optimize the power efficiency of the network. Zhang and Mao (2022) considered leveraging the intelligent reflecting surface (IRS) to assist the FL system, which can restructure the wireless channel and enhance signal strength. An iterative resource allocation algorithm was developed to minimize the energy consumption of the system. Vu et al. (2022) proposed transmission designs for energy-efficient FL in massive multiple-input multiple-output (MIMO) networks and developed a resource allocation algorithm to minimize the overall energy consumption.

Since device-to-device (D2D) communication has the potential to improve energy efficiency, reliability, and data rate, many researchers have introduced D2D communication technology into FL to improve energy efficiency. Lin et al. (2021) proposed the semi-decentralized FL that integrates

the conventional device-to-server communication paradigm with D2D communications to reduce network energy consumption. Al-Abiad et al. (2022) proposed a decentralized FL method that exploits D2D communications and overlapping clustering, eliminating the requirement for a central aggregator. Chen MZ et al. (2020) proposed collaborative FL, a framework that enables EDs to perform FL with less reliance on the server. Khowaja et al. (2021) proposed a distributed FL framework that addresses communication and energy efficiency issues for remote devices.

In conclusion, the introduction of novel technologies such as WPT, IRS, and D2D communication holds promise for addressing the issue of low device battery capacity and improving the energy efficiency of FL. However, it is important to acknowledge that these methods also come with certain limitations. When employing techniques like WPT or SWIPT, there may be constraints on the efficiency of energy transfer. Energy loss during transmission and limitations imposed by the transmission distance can result in a reduction of energy efficiency. Furthermore, implementing these strategies might necessitate additional hardware devices or complex system configurations, thus increasing both the cost and complexity of the system. Therefore, it is imperative to explore avenues for enhancing the energy transmission efficiency while simultaneously reducing resource requirements and system complexity. D2D communication mitigates the energy consumption of FL systems by facilitating data sharing and model aggregation among devices, effectively reducing the frequency and volume of direct communication between each device and the central server. Nevertheless, the efficacy of D2D communication is contingent upon device mobility and communication distance. In the case of frequent device movement or long communication distance, extra energy may be required to maintain the communication connection, which affects the efficiency of energy usage. Moreover, D2D communication could potentially increase communication overhead, particularly in large-scale FL systems. Direct communication between devices necessitates additional resource allocation and protocol support, which can augment energy consumption and introduce latency. Consequently, when applying these emerging technologies, it is crucial to conduct a comprehensive analysis and make necessary adjustments based on specific application scenarios and requirements.

3.6 Future directions

As energy-efficient FL continues to evolve and expand, several promising avenues emerge for future research and improvement. This subsection discusses three key areas that hold significant potential for shaping the future direction of FL: leveraging hardware advancements, advancements in communication technology, and integration of state-of-the-art AI methods. By considering these aspects, researchers and practitioners can pave the way for more energy-efficient and advanced FL systems.

1. Leveraging hardware advancements: As technology advances, hardware resources have witnessed significant improvements, providing opportunities to enhance the energy efficiency of FL systems (Capra et al., 2020). One avenue of research is the design of energy-efficient EDs. This involves developing hardware components such as processors, memory, and sensors that are optimized for low power consumption. For example, researchers can explore the use of low-power micro-controllers or application-specific integrated circuits (ASICs) specifically designed for edge computing and machine learning tasks (Wheeldon et al., 2020; Zaman et al., 2022).

Another area of focus is optimizing computing architectures for FL. Traditional computing architectures may not be well-suited for the resource-constrained EDs used in FL. Future research can explore the development of novel architectures that strike a balance between performance and energy efficiency (Hosseini and Mohsenin, 2021). This can include designing architectures that allow for efficient parallel processing, reducing unnecessary data movement, and minimizing memory access energy. Furthermore, specialized hardware accelerators tailored for distributed learning tasks can play a significant role in improving energy efficiency. These accelerators can be designed to offload computationally intensive tasks from the EDs, reducing their power consumption (Mazumder et al., 2021). For instance, dedicated accelerators for matrix operations, neural network computations, or encryption tasks can be integrated into EDs to improve the overall energy efficiency of FL systems. Research in this area can focus on developing efficient and scalable hardware designs for these accelerators.

2. Enhancing communication technologies: With the continuous advancements in communication technology, future FL systems can benefit from leveraging the capabilities of high-speed and low-latency networks (You et al., 2021). One area of research is the development of efficient communication protocols tailored for FL. Traditional protocols may not be well-suited for the unique characteristics of EDs, such as limited computation power and energy resources. Future research can explore the design of lightweight protocols that minimize overhead and communication energy consumption. These protocols can incorporate techniques like data compression, adaptive message size optimization, and smart transmission scheduling to reduce the energy required for communication in FL systems. In addition to these protocol-level advancements, incorporating technologies like NOMA and MIMO can further elevate the performance of wireless communication systems in terms of spectrum efficiency, system capacity, and energy efficiency (Vu et al., 2022; Wu Y et al., 2022). These advancements provide opportunities for substantial improvements in network performance, enabling efficient and scalable communication in the context of FL.

3. Integration of AI methods: To further enhance energy optimization in FL, future research should explore the application of state-of-the-art AI methods in FL settings. One promising approach is model compression, which aims to reduce the complexity and size of ML models without significant loss in performance (Deng et al., 2020). When applying model compression in FL, in addition to considering factors such as communication cost, computation complexity, and accuracy preservation, it is crucial to fully address the heterogeneity among devices in terms of their computation capabilities, memory constraints, and energy profiles (Abdelmoniem and Canini, 2021). By incorporating model compression techniques into the FL environment, the compressed models can be tailored to the specific characteristics of individual devices, thus optimizing energy efficiency and performance.

Another AI technique that can benefit FL is meta-learning, which enables models to acquire knowledge and learn new tasks or domains more efficiently (Hospedales et al., 2022). In the context of FL, meta-learning can be leveraged to facilitate knowledge transfer among nodes and enhance their learning capabilities (Sun et al., 2024). When a new node joins the FL system, instead of starting from scratch, it can leverage the knowledge gained from other nodes to initialize its model and transfer relevant information. By learning from the models and experiences of other nodes, the new node can quickly adapt to the target task and reduce the number of iterations required for convergence. Moreover, meta-learning can be used to improve the generalization ability of nodes in FL. Nodes can learn to extract and transfer latent knowledge across tasks or domains, enabling them to quickly adapt to new scenarios with limited local data. This allows FL models to generalize better across different nodes and tasks, enhancing the overall performance and convergence speed.

Furthermore, reinforcement learning's capacity to address complex decision-making problems, along with its flexibility and generalization abilities, makes it a valuable tool for enhancing energy efficiency in FL. By leveraging reinforcement learning techniques, FL systems empower intelligent decision-making, encompassing adaptive resource allocation, optimized device scheduling, and enhanced client training strategies (Wang H et al., 2020; Zhan et al., 2020; Sun et al., 2021). These advancements ultimately culminate in the achievement of more efficient resource utilization and a significant reduction in energy consumption.

4 Summary

In this paper, we concentrate on the energy consumption optimization problem of FL in MEC. First, we introduce the system model and energy consumption models, encompassing the computation and communication energy consumption models. Subsequently, we provide a comprehensive overview of the primary challenges in terms of improving energy efficiency. We categorize the existing energy-efficient strategies for FL into three main categories: learning-based strategies, resource allocation strategies, and client selection strategies. For each category, we provide detailed introductions and summaries of the strategies. Finally, we discuss potential research directions for achieving energy-efficient FL.

Contributors

Kang YAN drafted the paper. Nina SHU, Tao WU, Chunsheng LIU, and Panlong YANG helped organize the

paper. Kang YAN, Nina SHU, and Tao WU revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

References

Abdelmoniem AM, Canini M, 2021. Towards mitigating device heterogeneity in federated learning via adaptive model quantization. Proc 1st Workshop on Machine Learning and Systems, p.96-103. https://doi.org/10.1145/3437984.3458839

Abdelmoniem AM, Ho CY, Papageorgiou P, et al., 2023. A comprehensive empirical study of heterogeneity in federated learning. *IEEE Int Things J*, 10(16):14071-14083. https://doi.org/10.1109/JIOT.2023.3250275

Abe Y, Sasaki H, Kato S, et al., 2014. Power and performance characterization and modeling of GPU-accelerated systems. Proc 28th Int Parallel and Distributed Processing Symp, p.113-122. https://doi.org/10.1109/IPDPS.2014.23

Al-Abiad MS, Obeed M, Hossain J, et al., 2022. Decentralized aggregation for energy-efficient federated learning via overlapped clustering and D2D communications. https://arxiv.org/abs/2206.02981

Albaseer A, Abdallah M, Al-Fuqaha A, et al., 2021. Threshold-based data exclusion approach for energy-efficient federated edge learning. Proc IEEE Int Conf on Communications Workshops, p.1-6. https://doi.org/10.1109/ICCWorkshops50388.2021.9473806

Albelaihi R, Yu LK, Craft WD, et al., 2022. Green federated learning via energy-aware client selection. Proc IEEE Global Communications Conf, p.13-18. https://doi.org/10.1109/GLOBECOM48099.2022.10001569

Arouj A, Abdelmoniem AM, 2022. Towards energy-aware federated learning on battery-powered clients. Proc 1st ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network, p.7-12. https://doi.org/10.1145/3556557.3557952

Battiloro C, di Lorenzo P, Merluzzi M, et al., 2023. Lyapunov-based optimization of edge resources for energy-efficient adaptive federated learning. *IEEE Trans Green Commun Netw*, 7(1):265-280. https://doi.org/10.1109/TGCN.2022.3186879

Capra M, Bussolino B, Marchisio A, et al., 2020. An updated survey of efficient hardware architectures for accelerating deep convolutional neural networks. *Fut Int*, 12(7):113. https://doi.org/10.3390/fi12070113

Chen MZ, Poor HV, Saad W, et al., 2020. Wireless communications for collaborative federated learning. *IEEE Commun Mag*, 58(12):48-54. https://doi.org/10.1109/MCOM.001.2000397

Chen R, Li L, Xue KP, et al., 2023. Energy efficient federated learning over heterogeneous mobile devices via joint design of weight quantization and wireless transmission. *IEEE Trans Mob Comput*, 22(12):7451-7465. https://doi.org/10.1109/TMC.2022.3213766

Cho YJ, Wang JY, Joshi G, 2020. Client selection in federated learning: convergence analysis and power-of-choice selection strategies. https://arxiv.org/abs/2010.01243

da Silva JMB, Ntougias K, Krikidis I, et al., 2021. Simultaneous wireless information and power transfer for federated learning. Proc 22nd Int Workshop on Signal Processing Advances in Wireless Communications, p.296-300. https://doi.org/10.1109/SPAWC51858.2021.9593160

Deng L, Li GQ, Han S, et al., 2020. Model compression and hardware acceleration for neural networks: a comprehensive survey. *Proc IEEE*, 108(4):485-532. https://doi.org/10.1109/JPROC.2020.2976475

Gaudette B, Wu CJ, Vrudhula S, 2016. Improving smartphone user experience by balancing performance and energy with probabilistic QoS guarantee. Proc IEEE Int Symp on High Performance Computer Architecture, p.52-63. https://doi.org/10.1109/HPCA.2016.7446053

Gaudette B, Wu CJ, Vrudhula S, 2019. Optimizing user satisfaction of mobile workloads subject to various sources of uncertainties. *IEEE Trans Mob Comput*, 18(12):2941-2953. https://doi.org/10.1109/TMC.2018.2883619

Hong S, Kim H, 2010. An integrated GPU power and performance model. *ACM SIGARCH Comput Archit News*, 38(3):280-289. https://doi.org/10.1145/1816038.1815998

Hospedales T, Antoniou A, Micaelli P, et al., 2022. Meta-learning in neural networks: a survey. *IEEE Trans Patt Anal Mach Intell*, 44(9):5149-5169. https://doi.org/10.1109/TPAMI.2021.3079209

Hosseini M, Mohsenin T, 2021. QS-NAS: optimally quantized scaled architecture search to enable efficient on-device micro-AI. *IEEE J Emerg Sel Top Circ Syst*, 11(4):597-610. https://doi.org/10.1109/JETCAS.2021.3127932

Hu YQ, Huang HJ, Yu N, 2022. Device scheduling and channel allocation for energy-efficient federated edge learning. *Comput Commun*, 189:53-66. https://doi.org/10.1016/j.comcom.2022.03.008

Imteaj A, Thakker U, Wang SQ, et al., 2022. A survey on federated learning for resource-constrained IoT devices. *IEEE Int Things J*, 9(1):1-24. https://doi.org/10.1109/JIOT.2021.3095077

Jararweh Y, Doulat A, AlQudah O, et al., 2016. The future of mobile cloud computing: integrating cloudlets and mobile edge computing. Proc 23rd Int Conf on Telecommunications, p.1-5. https://doi.org/10.1109/ICT.2016.7500486

Khowaja SA, Dev K, Khowaja P, et al., 2021. Toward energy-efficient distributed federated learning for 6G networks. *IEEE Wirel Commun*, 28(6):34-40. https://doi.org/10.1109/MWC.012.2100153

Kim J, Kim D, Lee J, et al., 2022. A novel joint dataset and computation management scheme for energy-efficient federated learning in mobile edge computing. *IEEE Wirel Commun Lett*, 11(5):898-902. https://doi.org/10.1109/LWC.2022.3147236

Kim YG, Wu CJ, 2020. AutoScale: energy efficiency optimization for stochastic edge inference using reinforcement learning. Proc 53rd Annual IEEE/ACM Int Symp on Microarchitecture, p.1082-1096. https://doi.org/10.1109/MICRO50266.2020.00090

Kim YG, Wu CJ, 2021. AutoFL: enabling heterogeneity-aware energy efficient federated learning. Proc 54th Annual IEEE/ACM Int Symp on Microarchitecture, p.183-198. https://doi.org/10.1145/3466752.3480129

Li L, Xiong HY, Guo ZS, et al., 2019. SmartPC: hierarchical pace control in real-time federated learning system. Proc IEEE Real-Time Systems Symp, p.406-418. https://doi.org/10.1109/RTSS46320.2019.00043

Li L, Wang J, Chen X, et al., 2020. Multi-layer coordination for high-performance energy-efficient federated learning. Proc 28th Int Symp on Quality of Service, p.1-10. https://doi.org/10.1109/IWQoS49365.2020.9212881

Li L, Shi D, Hou RH, et al., 2021. To talk or to work: flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices. Proc IEEE Conf on Computer Communications, p.1-10. https://doi.org/10.1109/INFOCOM42981.2021.9488839

Lim WYB, Luong NC, Hoang DT, et al., 2020. Federated learning in mobile edge networks: a comprehensive survey. *IEEE Commun Surv Tut*, 22(3):2031-2063. https://doi.org/10.1109/COMST.2020.2986024

Lin FPC, Hosseinalipour S, Azam SS, et al., 2021. Semi-decentralized federated learning with cooperative D2D local model aggregations. *IEEE J Sel Areas Commun*, 39(12):3851-3869. https://doi.org/10.1109/JSAC.2021.3118344

Luo B, Li X, Wang SQ, et al., 2021. Cost-effective federated learning design. Proc IEEE Conf on Computer Communications, p.1-10. https://doi.org/10.1109/INFOCOM42981.2021.9488679

Mao YY, You CS, Zhang J, et al., 2017. A survey on mobile edge computing: the communication perspective. *IEEE Commun Surv Tut*, 19(4):2322-2358. https://doi.org/10.1109/COMST.2017.2745201

Mazumder AN, Meng J, Rashid HA, et al., 2021. A survey on the optimization of neural network accelerators for micro-AI on-device inference. *IEEE J Emerg Sel Top Circ Syst*, 11(4):532-547. https://doi.org/10.1109/JETCAS.2021.3129415

McMahan HB, Moore E, Ramage D, et al., 2017. Communication-efficient learning of deep networks from decentralized data. Proc 20th Int Conf on Artificial Intelligence and Statistics, p.1273-1282.

Mei XX, Chu XW, Liu H, et al., 2017a. Energy efficient real-time task scheduling on CPU-GPU hybrid clusters. Proc IEEE Conf on Computer Communications, p.1-9. https://doi.org/10.1109/INFOCOM.2017.8057205

Mei XX, Wang Q, Chu XW, 2017b. A survey and measurement study of GPU DVFS on energy conservation. *Dig Commun Netw*, 3(2):89-100. https://doi.org/10.1016/j.dcan.2016.10.001

Mo XP, Xu J, 2021. Energy-efficient federated edge learning with joint communication and computation design. *J Commun Inform Netw*, 6(2):110-124. https://doi.org/10.23919/JCIN.2021.9475121

Nguyen VD, Sharma SK, Vu TX, et al., 2021. Efficient federated learning algorithm for resource allocation in wireless IoT networks. *IEEE Int Things J*, 8(5):3394-3409. https://doi.org/10.1109/JIOT.2020.3022534

Niknam S, Dhillon HS, Reed JH, 2020. Federated learning for wireless communications: motivation, opportunities, and challenges. *IEEE Commun Mag*, 58(6):46-51. https://doi.org/10.1109/MCOM.001.1900461

Peng C, Hu Q, Wang ZL, et al., 2023. Online-learning-based fast-convergent and energy-efficient device selection in federated edge learning. *IEEE Int Things J*, 10(6):5571-5582. https://doi.org/10.1109/JIOT.2022.3222234

Perazzone J, Wang SQ, Ji MY, et al., 2022. Communication-efficient device scheduling for federated learning using stochastic optimization. Proc IEEE Conf on Computer Communications, p.1449-1458. https://doi.org/10.1109/INFOCOM48880.2022.9796818

Prakash P, Ding JH, Chen R, et al., 2022. IoT device friendly and communication-efficient federated learning via joint model pruning and quantization. *IEEE Int Things J*, 9(15):13638-13650. https://doi.org/10.1109/JIOT.2022.3145865

Shi D, Li L, Chen R, et al., 2022a. Toward energy-efficient federated learning over 5G+ mobile devices. *IEEE Wirel Commun*, 29(5):44-51. https://doi.org/10.1109/MWC.003.2100028

Shi D, Li L, Wu MQ, et al., 2022b. To talk or to work: dynamic batch sizes assisted time efficient federated learning over future mobile edge devices. *IEEE Trans Wirel Commun*, 21(12):11038-11050. https://doi.org/10.1109/TWC.2022.3189320

Sun W, Lei SY, Wang L, et al., 2021. Adaptive federated learning and digital twin for Industrial Internet of Things. *IEEE Trans Ind Inform*, 17(8):5605-5614. https://doi.org/10.1109/TII.2020.3034674

Sun W, Zhao Y, Ma WQ, et al., 2024. Accelerating convergence of federated learning in MEC with dynamic community. *IEEE Trans Mob Comput*, 23(2):1769-1784. https://doi.org/10.1109/TMC.2023.3241770

Tang MX, Ning XF, Wang YT, et al., 2022. FedCor: correlation-based active client selection strategy for heterogeneous federated learning. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.10092-10101. https://doi.org/10.1109/CVPR52688.2022.00986

Tran HV, Kaddoum G, Elgala H, et al., 2020. Lightwave power transfer for federated learning-based wireless networks. *IEEE Commun Lett*, 24(7):1472-1476. https://doi.org/10.1109/LCOMM.2020.2985698

Tran NH, Bao W, Zomaya A, et al., 2019. Federated learning over wireless networks: optimization model design and analysis. Proc IEEE Conf on Computer Communications, p.1387-1395. https://doi.org/10.1109/INFOCOM.2019.8737464

Vu TT, Ngo HQ, Dao MN, et al., 2022. Energy-efficient massive MIMO for federated learning: transmission designs and resource allocations. *IEEE Open J Commun Soc*, 3:2329-2346. https://doi.org/10.1109/OJCOMS.2022.3222749

Wang H, Kaplan Z, Niu D, et al., 2020. Optimizing federated learning on non-IID data with reinforcement learning. Proc IEEE Conf on Computer Communications, p.1698-1707. https://doi.org/10.1109/INFOCOM41043.2020.9155494

Wang SQ, Tuor T, Salonidis T, et al., 2018. When edge meets learning: adaptive control for resource-constrained distributed machine learning. Proc IEEE Conf on Computer Communications, p.63-71. https://doi.org/10.1109/INFOCOM.2018.8486403

Wheeldon A, Shafik R, Rahman T, et al., 2020. Learning automata based energy-efficient AI hardware design for IoT applications. *Phil Trans R Soc A Math Phys Eng Sci*, 378(2182):20190593 https://doi.org/10.1098/rsta.2019.0593

Wu T, Qu YB, Liu CS, et al., 2023. Joint edge aggregation and association for cost-efficient multi-cell federated learning. Proc IEEE Conf on Computer Communications, p.1-10. https://doi.org/10.1109/INFOCOM53939.2023.10229060

Wu T, Fan XC, Wei H, et al., 2024. Predictive service provisioning with online learning in wireless edge networks. *IEEE Trans Mob Comput*, 23(5):4076-4091. https://doi.org/10.1109/TMC.2023.3286847

Wu Y, Song YX, Wang TS, et al., 2022. Non-orthogonal multiple access assisted federated learning via wireless power transfer: a cost-efficient approach. *IEEE Trans Commun*, 70(4):2853-2869. https://doi.org/10.1109/TCOMM.2022.3153068

Xu J, Wang HQ, 2021. Client selection and bandwidth allocation in wireless federated learning networks: a long-term perspective. *IEEE Trans Wirel Commun*, 20(2):1188-1200. https://doi.org/10.1109/TWC.2020.3031503

Yang CX, Xu MW, Wang QP, et al., 2024. FLASH: heterogeneity-aware federated learning at scale. *IEEE Trans Mob Comput*, 23(1):483-500. https://doi.org/10.1109/TMC.2022.3214234

Yang ZH, Chen MZ, Saad W, et al., 2021. Energy efficient federated learning over wireless communication networks. *IEEE Trans Wirel Commun*, 20(3):1935-1949. https://doi.org/10.1109/TWC.2020.3037554

You XH, Wang CX, Huang J, et al., 2021. Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts. *Sci China Inform Sci*, 64(1):110301. https://doi.org/10.1007/s11432-020-2955-6

Yu R, Li PC, 2021. Toward resource-efficient federated learning in mobile edge computing. *IEEE Netw*, 35(1):148-155. https://doi.org/10.1109/MNET.011.2000295

Yurtsever E, Lambert J, Carballo A, et al., 2020. A survey of autonomous driving: common practices and emerging technologies. *IEEE Access*, 8:58443-58469. https://doi.org/10.1109/ACCESS.2020.2983149

Zaman KS, Reaz MBI, Md Ali SH, et al., 2022. Custom hardware architectures for deep learning on portable devices: a review. *IEEE Trans Neur Netw Learn Syst*, 33(11):6068-6088. https://doi.org/10.1109/TNNLS.2021.3082304

Zeng QS, Du YQ, Huang KB, et al., 2020. Energy-efficient radio resource allocation for federated edge learning. Proc IEEE Int Conf on Communications Workshops, p.1-6. https://doi.org/10.1109/ICCWorkshops49005.2020.9145118

Zeng QS, Du YQ, Huang KB, et al., 2021a. Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing. *IEEE Trans Wirel Commun*, 20(12):7947-7962. https://doi.org/10.1109/TWC.2021.3088910

Zeng QS, Du YQ, Huang KB, 2021b. Wirelessly powered federated edge learning. Proc 22nd Int Workshop on Signal Processing Advances in Wireless Communications, p.286-290. https://doi.org/10.1109/SPAWC51858.2021.9593122

Zeng SG, Wu MH, 2019. Based on public health service in smart medical comprehensive service platform. Proc IEEE Int Conf on Computation, Communication and Engineering, p.48-51. https://doi.org/10.1109/ICCCE48422.2019.9010766

Zhan YF, Li P, Guo S, 2020. Experience-driven computational resource allocation of federated learning by deep reinforcement learning. Proc IEEE Int Parallel and Distributed Processing Symp, p.234-243. https://doi.org/10.1109/IPDPS47924.2020.00033

Zhang TC, Mao SW, 2022. Energy-efficient federated learning with intelligent reflecting surface. *IEEE Trans Green Commun Netw*, 6(2):845-858. https://doi.org/10.1109/TGCN.2021.3126795

Zhao BR, Cui QM, Liang SY, et al., 2022. Green concerns in federated learning over 6G. *China Commun*, 19(3):50-69. https://doi.org/10.23919/JCC.2022.03.004

Zhao JX, Feng YH, Chang XY, et al., 2022. Energy-efficient client selection in federated learning with heterogeneous data on edge. *Peer-to-Peer Netw Appl*, 15(2):1139-1151. https://doi.org/10.1007/s12083-021-01254-8

Zheng JJ, Li K, Tovar E, et al., 2021. Federated learning for energy-balanced client selection in mobile edge computing. Proc Int Wireless Communications and Mobile Computing, p.1942-1947. https://doi.org/10.1109/IWCMC51323.2021.9498853