Client-Oriented Energy Optimization in Clustered Federated Learning with Model Partition

Sinan Pan[1,2], Lei Shi[1,2(✉)], Han Wu[1,2], Yingying Chen[1,2], Huaili Liu[3], and Hao Xu[3]

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
shilei@hfut.edu.cn
[2] Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei 230009, China
[3] Anhui and Huaihe River Institute of Hydraulic Research (Anhui Provincial Water Conservancy Engineering Quality Testing Center Station), Hefei, China

Abstract. Clustered Federated Learning (CFL) based on the edge computing (EC) environment has great potential to promote the realization of artificial intelligence at the network edge. However, due to the large network area and the limited energy of edge devices, excessive energy consumption may occur if all devices participate in parameter transmission after training. This can result in insufficient energy for some devices, making it difficult to complete the training tasks. In this paper, we aim to reduce client energy consumption and total transmission energy consumption by the model partition technique and the adjustment of transmission modes among clients. Firstly, we establish the mathematical model, and show that the transmission energy consumption can be effectively reduced by optimizing the transmission distance. Secondly, we design the Distance Best K-Means (DBKM) algorithm to obtain the optimal transmission distance among clients. Finally, we propose a method to dynamically adjust the client training mode based on the client's energy. Simulation results show that compared with the other three methods, our algorithm can reduce the transmission energy consumption by 27%–82%.

Keywords: Clustered federated learning · Model partition · Limited energy

1 Introduction

Federated Learning (FL) is a novel distributed deep learning framework [1]. The main idea of FL is to use a server to aggregate local model parameters from participating clients to obtain a global model [2,3]. This approach can not only make full use of the computing resources of clients, but also better protect

H. Gao and X. Wang (Eds.): CollaborateCom 2024, LNICST 626, pp. 195–213, 2026.
https://doi.org/10.1007/978-3-031-93257-1_11

the privacy of each client's data during the training process [4,5]. Nowadays researchers have done lots of work on FL. For example, in [6,7], authors proposed a FL framework that can effectively reduce communication energy consumption. In [8], authors developed a federated optimization method named Cooperative Edge-Based Federated Average, which can greatly reduce the training time once the target model accuracy is achieved. In [9,10], authors optimized the problem of decreasing training accuracy due to limited resources.

However, in the actual training environment, since clients usually have the characteristics of heterogeneity, using the traditional FL training method directly may result in poor training outcomes. In order to solve this problem, the Clustered Federated Learning (CFL) framework is proposed [11]. To be specific, the main idea of CFL is to gather clients into different clusters by their heterogeneity characteristics, and train different in-cluster models within each cluster. The CFL framework can improve the accuracy of each in-cluster model and the training efficiency, and solve the problem of data heterogeneity in FL. Apparently the clustering method is the key point in CFL framework [12–14], and many researchers have focused their work on this point. For example, in [15,16], authors proposed a CFL framework, which can improve the performance and efficiency compared to the conventional CFL. In [17], authors proposed a collaborative clustering algorithm for energy and distribution in CFL, which can achieve the expected accuracy with the lowest energy consumption. In [18], authors proposed an iterative solution that can enable accurate learning with low energy cost. In [19], authors proposed a more flexible dynamic adaptive CFL scheme, which provided the most reasonable cluster partitioning results in all cases. In [20], authors proposed a new clustering method, FL with soft clustering, combining the advantages of soft clustering and iterative federated clustering. In [21], authors proposed a Variational AutoEncoder Gaussian Mixture Model Clustering Vertical Federated Learning Model, and its clustering results are better than those of other clustering methods in precision and other metrics. In [22], authors proposed an energy-efficiency clustering method using parallel Gibbs sampling to find the optimal client base, achieving improved accuracy.

CFL solves the problem of data heterogeneity in FL, but in the FL environment, clients are often small devices which are energy limited. To reduce energy consumption, some researchers propose to use the model partition technique [23]. By using the model partition technique, layers of the deep learning model are divided into two or more parts and deployed in different clients or servers for training [24]. For example, in [25], authors formulated a joint optimization problem of device scheduling and resource allocation by using the model partition technique.

Similarly, clients' energy limitations are an important problem in CFL. So the trade-off between client clustering and energy consumption has become a new hot research topic. In [26], authors investigated the training of machine learning models on a resource-constrained cluster of devices via a swarm of unmanned aerial vehicles. Hierarchical nested personalized federated learning is utilized to achieve improvements in machine learning performance, network resource

savings, and group trajectory efficiency. In [27], authors proposed a joint problem of resource allocation and device clustering. It can solve the problem of training performance degradation and optimize the system utility. In [28], authors introduced node collaboration via AirComp to hierarchical federated learning to solve the problem of limited communication and storage costs. In [29], authors proposed a dynamic federated optimization strategy to solve the problem of using federated deep reinforcement learning to adjust cache consumption in cloud-edge. In [30], authors proposed a decentralized FL scheme: federated learning empowered overlapped clustering for decentralized aggregation. This scheme can enable direct model transmission between clusters and reduces the energy consumption of the equipment.

Previous researchers have done a great deal of work on CFL from energy consumption and clustering. However, in CFL, few researchers have applied model partition technique to reduce client energy consumption and optimize transmission energy consumption by adjusting the clustering. Also, in many CFL scenarios, the non-independent and identically distributed (non-IID) data among clients affects the training accuracy. Additionally, the range of clients is wide, clients' energy is limited, and clients take on a lot of training tasks. To solve the above problems in the mentioned scenarios, we will apply model partition techniques in CFL to reduce the energy consumption of clients and decrease the total transmission energy consumption by adjusting the transmission path between clients. The main contributions of our work are summarized as follows.

1) In this paper, we design an energy dynamic balance method that comprehensively considers the clustering of non-IID data on the clients and the clustering based on physical distance. In each iteration, we will re-cluster and reselect client heads (CHs) based on physical distance to maintain the overall energy balance.
2) We propose a dynamic model partition method to ensure the energy balance. For any client whose energy is lower than the energy threshold E', the CH of the cluster to which this client belongs will undertake all the backward propagation training processes. Otherwise, the backward propagation process is performed by both the CH and the client.
3) We have simulated the proposed scheme and compared it with other schemes. Simulation results show that our scheme can achieve a reduction in transmission energy consumption compared to other schemes.

The rest of this paper is organized as follows. In Sec.2, we present the system model and define the problem. In Sec.3, we analyze the problem and give our algorithm. In Sec.4, we give the simulation results and analyze them. In Sec.5, we summarize this paper.

2 System Model and Problem Formulation

In this section, we first introduce the system model, and then give the problem definition. Consider a CFL scenario with one parameter server and many clients. Suppose the data in different clients is non-IID, and the clients' distribution

area is physically large while each client is limited in energy. Based on these two assumptions, we will process the clustering in two steps. For the first step, we group these clients into different clusters based on their data characteristics. For the second step, in each cluster, we group clients into some sub-clusters based on their physical distances. After finishing clustering, the whole system will try to train learning models for all clusters. Notice that in this scenario, each cluster (not each sub-cluster) has one training model. Since clients have limited energy, we use a model partition technique to help balance the clients' energy. We will identify some CHs to train part of the models. Considering that CHs are in sub-clusters, and client energy is constantly changing throughout the entire training period, which means that how to set sub-clusters and how to select CHs are important and difficult. Our goal is to complete the entire training process with minimal transmission energy consumption without depleting the energy of any client.

2.1 System Model

Consider a CFL scenario consisting of a server and n clients in a two-dimensional region, as shown in Fig. 1. Assume the server is located at the center of this area. Define D as the set of clients and $d_i (d_i \in D, 1 \leq i \leq n)$ as one client. Suppose that the number of clusters after the first step of clustering is Q. Define that each client has a certain energy $E_i (1 \leq i \leq n)$. Define the energy threshold as E', which means that by using the model partition technique, if the energy of the client exceeds E', the CH can transmit backward propagation parameters to the client during the training process, and the training parameters are finally stored on the client. Otherwise, the backward propagation is carried out on the CH, and the training parameters are ultimately stored on the CH. Apparently, in each sub-cluster, the client with the highest energy is selected as the CH. To simplify the problem, we suppose that in each training round, the total number of sub-clusters will not change (and denote it as m), and for a cluster, the number of sub-clusters will not change either, but the grouped clients may be different. Define C as the set of CHs and $c_j (c_j \in C, 1 \leq j \leq m)$ as one CH. Suppose the trained DNN model has V layers and denote $l_r (r = 1...V)$ as one layer. By using the model partition technique, some layers will be trained on clients, while others will be trained on the CH. To simplify the problem, we assume that the model partition point is $l_v (1 \leq v \leq V)$ for all clients. This means that each task will first train the initial l_v layers on the client. Then, the intermediate parameters are transmitted to the CH of the sub-cluster where the client belongs, and the training of the subsequent layers will be completed on the CH.

Since all model partition points are on the same layer, we suppose the size of transferred parameters between clients and CHs is the same. Denote the transferred data between d_i and c_j as γ. The data transfer size between each CH and the server is also same, and denote it as α. Denote the channel bandwidth as B. Denote the channel transmission power as P. Denote the reference distance as D_0. Denote the transmission distance between d_i and c_j as D_{ij}. Denote the

total transmission energy consumption of a round of global iteration as E_t^{comm}, and denote the total transmission distance as D_t^{comm}.

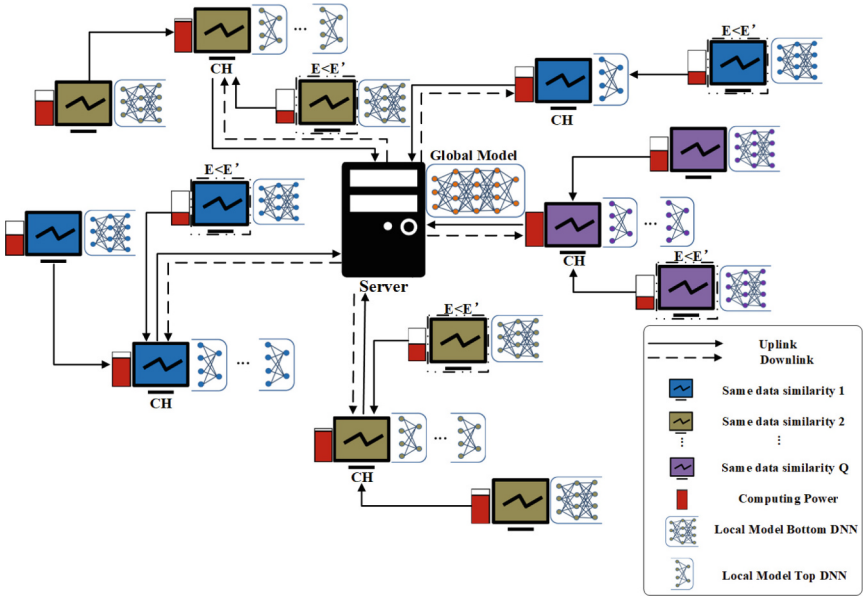


Fig. 1. Clustered federated learning with model partition.

In the t-th round of training, we assume the parameter in the CH is $w_{j,t}$, and the parameter in the client is $w_{i,t}$. The trained model parameters are aggregated in the CH, and the parameters of multiple aggregated sub-clusters are sent to the server for classification aggregation. The aggregation parameter of the server is $W_{q,t}(1 \leq q \leq Q)$. In the global iteration of round t, the specific steps for $w_{j,t}$, $w_{i,t}$ and $W_{q,t}$ are as follows.

1) The server broadcasts the global model parameter $W_{q,t}$ to each CH, and then CHs broadcast $W_{q,t}$ to clients in sub-clusters.
2) After receiving $W_{q,t}$, clients and CHs cooperate to conduct local training of e rounds according to the model partition technique. By using the model partition technique, clients will decide whether to perform the backward propagation process based on their energy.
3) After completing the local model training, clients in each sub-cluster send parameter $w_{i,t}$ to their CHs for updating $w_{j,t}$ with the CH parameter aggregation algorithm. Then, CHs send $w_{j,t}$ to the server for classification aggregation and update $W_{q,t}$.

We hope to design a scheme to minimize the total transmission energy consumption in one iteration, while ensuring the accuracy of the training.

2.2 Problem Formulation

Denote E_t^{comm} as the total transmission energy consumption in one global iteration. E_t^{comm} consists of the transmission energy consumption of clients to the CH (denoted as $E_{t,i,j}^{comm}$), and the transmission energy consumption of the CH to the server (denoted as $E_{t,j,s}^{comm}$). Denote T as the total number of training iterations. Then we have(Table 1)

Table 1. Notations

Variables	Meaning
d_i	One of the client $i \in \{1, 2, 3, ..., n\}$;
Q	The number of clusters after the first step of clustering;
E_i	One of the energy of client $i \in \{1, 2, 3, ..., n\}$;
E'	The energy threshold;
c_j	One of the CH $j \in \{1, 2, 3, ..., m\}$;
l_r	One of the layer of the training model $i \in \{1, 2, 3, ..., V\}$;
l_v	The model partition point;
γ	The size of the parameter transmitted by the client to the CH;
α	The size of the parameter transmitted by the CH to the server;
B	The channel bandwidth;
P	The channel transmission power;
D_0	The reference distance;
$D_{i,j}$	The transmission distance between client i and CH j;
E_t^{comm}	The total transmission energy consumption in round t training $t \in \{1, 2, 3, ..., T\}$;
D_t^{comm}	The total transmission distance in round t training $t \in \{1, 2, 3, ..., T\}$;
$w_{j,t}$	The training parameters of CH j in round t;
$w_{i,t}$	The training parameters of client i in round t;
$W_{q,t}$	The aggregation parameters of cluster Q in round t;

$$E_t^{comm} = \sum_{j=1}^{m} \sum_{i=1}^{n} (a_{i,j} \cdot E_{t,i,j}^{comm} + E_{t,j,s}^{comm}), \tag{1}$$

where $a_{i,j}$ represents the number of propagations of client d_i in cluster j during one global iteration. We have

$$a_{i,j} = \begin{cases} 2e & : E_i > E'; \\ e & : E_i < E'. \end{cases} \tag{2}$$

The e represents the number of local iterations.
$E_{t,i,j}^{comm}$, $E_{t,j,s}^{comm}$ can be represented as follows.

$$E_{t,i,j}^{comm} = P\tau_{t,i,j}, \tag{3}$$

$$E_{t,j,s}^{comm} = P\tau_{t,j,s}, \tag{4}$$

where P represents the initial power transmitted by the client to the CH and by the CH to the server. $\tau_{t,i,j}$ and $\tau_{t,j,s}$ are their respective transmission times. The specific calculation formula is as follows.

$$\tau_{t,i,j} = \frac{\gamma}{B\log_2\left(1+\frac{Ph_{t,i,j}}{BN_0}\right)}, \tag{5}$$

$$\tau_{t,j,s} = \frac{\alpha}{B\log_2\left(1+\frac{Ph_{t,j,s}}{BN_0}\right)}, \tag{6}$$

where $h_{t,i,j}$ and $h_{t,j,s}$ represent the power gain from the client to the CH and from the CH to the server respectively. We have

$$h_{t,i,j} = h_0\rho\left(\frac{D_0}{D_{t,i,j}}\right)^v, \tag{7}$$

$$h_{t,j,s} = h_0\rho\left(\frac{D_0}{D_{t,j,s}}\right)^v, \tag{8}$$

where v represents the channel loss factor, D_0 represents the reference distance, $D_{t,i,j}$ and $D_{t,j,s}$ represent the distance from the client to the CH and the CH to the server in the t iteration, respectively.

The optimization problem is expressed as follows.

$$\begin{aligned} &\min \quad E_t^{comm}, \\ &s.t.(2)(3)(4)(5)(6)(7)(8). \end{aligned} \tag{9}$$

In (9), $a_{i,j}$ is the variable, while all other notations are constants or predetermined values. By combining (3), (4), (5), (6), (7), and (8), we can derive the specific expression for $E_t^{comm} = \sum_{j=1}^{m}\sum_{i=1}^{n}(a_{i,j}\cdot\frac{P\gamma}{B\log_2\left(1+\frac{Ph_0\rho\left(\frac{D_0}{D_{t,i,j}}\right)^v}{BN_0}\right)}+\frac{P\alpha}{B\log_2\left(1+\frac{Ph_0\rho\left(\frac{D_0}{D_{t,j,s}}\right)^v}{BN_0}\right)})$. From this expression, it can be seen that when all other variables are constant, E_t^{comm} is linearly related to the transmission distance D_t^{comm}. Therefore, we can transform the problem of optimizing transmission energy consumption into the problem of optimizing transmission distance.

2.3 Problem Deformation

As discussed in the last section, we can convert the original problem for minimizing the total transmission energy consumption into the problem for minimizing the total transmission distance. Denote D_t^{total} as the total transmission distance in the t iteration. Denote $D_{t,i,j}^{total}$ as the transmission distance from d_i to c_j during

the t iteration, and denote $D_{t,j,s}^{total}$ as the transmission distance from c_j to the server during the t iteration. We have

$$D_t^{total} = \min_{i \in n, j \in m} \{D_{t,i,j}^{total} + D_{t,j,s}^{total}\}. \tag{10}$$

For $D_{t,i,j}^{total}$, we have

$$D_{t,i,j}^{total} = \sum_{j=1}^{m} \sum_{i=1}^{n} (a_{i,j} \cdot D_{t,i,j}). \tag{11}$$

For $D_{t,j,s}^{total}$, we have

$$D_{t,j,s}^{total} = \sum_{j=1}^{m} D_{t,j,s}. \tag{12}$$

Then the optimization problem can be expressed as follows.

$$\begin{aligned} &\min \quad D_t^{total}. \\ &s.t. (10)(11)(12) \\ &0 \leq D_{t,i,j} \leq d; \\ &0 \leq D_{t,j,s} \leq \frac{\sqrt{2}}{2} d. \end{aligned} \tag{13}$$

In (13), d represents the length of the region side, which is a determined value. So there is only one variable. However, in different networks, the distribution of clients is often variable and complex, which makes the original problem complex and difficult to solve directly. Therefore, we need to further design algorithm to obtain the minimum transmission energy consumption under various distribution scenarios.

3 Algorithm

In this section, we will introduce the proposed algorithm for solving the above optimization problem. Firstly, in order to solve the problem where the distance between clients in the clusters is too far after the initial clustering, we propose the Distance Best K-Means (DBKM) algorithm. The core idea of the DBKM algorithm is to partition each cluster into multiple sub-clusters, thereby reducing the transmission distance from the client to the CH. Secondly, within each sub-cluster, we train the model using the model partition technique, while selecting the client with the highest energy in each sub-cluster as the CH. Finally, in order to reduce transmission energy consumption by decreasing the number of parameter transfers between the client and the CH, we propose an algorithm to dynamically adjust the client training mode according to the client's energy.

3.1 Problem Analysis

In this subsection, we will analyze the specific optimization algorithm. We found that data heterogeneity in CFL affects the training accuracy. Data heterogeneity is usually manifested as a non-IID data distribution, which means that different clients have data with significant statistical differences. These differences often lead to the global model and local model objectives being inconsistent, causing the client model to deviate from the ideal global optimization point. This results in the model performing well on local data but having poor generalization ability on global data, leading to overfitting. At the same time, updates from different client models may also conflict with each other, causing instability in the training process, which in turn affects the training accuracy. So all clients are divided into several clusters by hierarchical clustering based on data similarity. This ensures the accuracy of training. In a large-area scenario, the physical distance between clients in a cluster is relatively large after the hierarchical clustering. This consumes a lot of energy during parameter transfer. To reduce transmission energy consumption, we propose the DBKM algorithm to divide each cluster into multiple sub-clusters. The purpose is to divide clients that are far apart into different sub-clusters and reduce the transmission distance from the client to the CH, thus reducing the total transmission energy consumption. Suppose we have m sub-clusters. The specific DBKM algorithm is shown in Algorithm 1.

Algorithm 1. DBKM Algorithm

Input: The distribution of Q clusters after hierarchical clustering;
Output: The distribution of m sub-clusters;

for cluster q from 1 to Q **do**
  Use *k-means* for intra-cluster clustering to get m sub-clusters, $m > Q$;
  Select the client with the highest energy in the sub-cluster as CH;
  repeat
    for cluster j from 1 to m **do**
      Find the client d_i that is farthest away from CH c_j in sub-cluster j and calculate its distance $D_{i,j}$ from c_j;
      The distance between d_i and CH(c_{j1},c_{j2},...) is calculated and expressed as $D_{i,j1}$,$D_{i,j2}$, $D_{i,j3}$..., the condition of CHs is that the data in the CH and the client d_i are IID, assuming that the minimum distance is $D_{i,j1}$;
      if Distance $D_{i,j1} < D_{i,j}$ **then**
        Remove client i from sub-cluster j and add client i to the sub-cluster $j1$;
      else
        Don't make any changes;
      end if
      Reselect the CH in sub-cluster j;
    end for
  until the members of each sub-cluster no longer change;
end for

By observing the pseudo-code, we can see that the DBKM algorithm consists of three key steps. Firstly, after using hierarchical clustering, there are Q clusters.

In each cluster, multiple sub-clusters are divided based on physical distance. The client with the highest energy in each sub-cluster is selected as the CH. Secondly, after the first iteration of the DBKM algorithm, there will be m initial distribution sub-clusters. For the sub-cluster j, select the client i farthest from CH c_j, and calculate its distance $D_{i,j}$ to CH c_j and its distances $D_{i,j1}$, $D_{i,j2}$, etc., to CHs c_{j1}, c_{j2}, etc., which have the same data similarity as CH c_j. Compare $D_{i,j}$, $D_{i,j1}$, $D_{i,j2}$, etc., and add client i to the sub-cluster with the smallest distance. The client member within each sub-cluster is updated in each DBKM iteration with the goal of minimizing D_t^{Total}, until the members of each sub-cluster no longer change. Finally, after each subsequent DBKM iteration, the CH for each sub-cluster is updated once. Clients whose data distribution is IID may be distributed across multiple sub-clusters, but the data distribution of all clients within a sub-cluster must be IID.

There are m sub-clusters with identified members. In each sub-cluster, the training is based on model partition technique and the model is partitioned into two parts, the client undertakes the training of the bottom part of the model and the CH undertakes the training of the top part of the model. Simultaneously, based on the client's energy, the training mode is determined and the energy threshold E' is set. In order to more evenly utilize each client's energy to complete the training, we set the value of E' to the average energy of all clients. If the client's energy exceeds E', backward propagation occurs on the client during model partition training, and the final parameters are updated on the client. If the client's energy does not exceed E', backward propagation is carried out entirely at the CH during model partition training, and the final parameters are updated on the CH. Meanwhile, considering the varying energy of clients and the possibility that some clients may not be able to complete all training rounds due to energy issues, we employ dynamic model partitioning technology to achieve energy balance among clients. The specific situation is as follows. Clients with energy below E' perform backward propagation training at the CH, which does not consume the client's energy, clients with energy above E' perform backward propagation training on the client, which consumes the client's energy. During the training process, clients with high energy will consume more energy than those with low energy. Therefore, the gap in energy between clients will gradually narrow during the training process, allowing clients with low energy to complete more training rounds, achieving a kind of energy balance.

After the clients in the sub-clusters complete their training, the parameters are transmitted to the CH for aggregation. Then, the CH transmits the aggregated parameters to the server. The relationship between energy consumption, transmission distance, and the amount of data transmitted can be expressed as follows.

$$E_{t,i,j}^{comm} = \frac{P\gamma}{B\log_2\left(1 + \frac{Ph_0\rho\left(\frac{D_0}{D_{i,j}}\right)^v}{BN_0}\right)}, \tag{14}$$

where $E_{t,i,j}^{comm}$ is the transmission energy consumption, γ is the data amount, and $D_{i,j}$ is the transmission distance.

Based on this, we design the Energy Consumption Optimization(ECO) algorithm to dynamically adjust the transmission path between clients to reduce the total transmission distance and minimize the total transmission energy consumption.

3.2 ECO Algorithm

According to the analysis in the previous section, the ECO algorithm can be summarized as follows.

1) Firstly, n clients are divided into multiple clusters using hierarchical clustering. In each cluster, the clients are distributed across different regions.
2) In each cluster partitioned in Step 1, the DBKM algorithm is used again to further divide each cluster into multiple sub-clusters.
3) Select the client with the highest energy in each sub-cluster as the CH.
4) The model partition technique is used to train the client, and the client's energy determines whether it conducts backward propagation training.

The algorithm is shown in Algorithm 2.

Time Complexity Analysis: Let the number of iterations of *k-means* and the repeated adjustment of each sub-cluster member processes in the DBKM algorithm be T_1 and T_2, respectively. Therefore, the time complexity of the DBKM algorithm can be expressed as $O(T_1 + T_2)$. Let the total number of iterations be T, the number of local iterations be e, and the number of iterations for clustering n clients according to data similarity be T_3. The total time complexity of the ECO algorithm can be expressed as $O(T(e + T_1 + T_2 + T_3))$.

4 Simulation and Experiment

In this section, we will introduce experiments and simulations. The DNN model used in our experiments is VGG16. The model is trained using the EMNIST dataset, which consists of 814255 28×28 pixel grayscale images in 62 classes, and the framework used is pytorch. We use GPU to simulate the computing power of server and use CPUs of different computers as clients. We set the scene area size to 10 km × 10 km. To get the constants in Sect. 2, such as E_i, $D_{i,j}$, γ, α and so on, we first assign each client a random number between 50 and 100 as the client's energy consumption E_i. Then, we obtain $D_{i,j}$ by calculating the Euclidean distance between clients and CHs. Finally, perform forward and backward propagation of the model 50 times, then calculate the average output data amount of the partition layers, i.e. the value of γ. The values of α are obtained by the same way. We set the other parameters in the experiment as shown in Table 2.

Algorithm 2. Energy Consumption Optimization Algorithm

Input: Number of clients n, client energy(E_1, E_2...E_i), total number of rounds T;
Output: The minimum transmission energy consumption E_t^{comm};

for round t from 1 to T **do**
  // For *Client*
  // *Clustering process*
  The n clients are divided into Q clusters according to data similarity, $n > Q$;
  Using DBKM algorithm, Q clusters are divided into m sub-clusters;
  // *Training process*
  for round e from 1 to E **do**
    for $j \in m$ **do**
      for $d_i \in j$ **do**
        Model partition is introduced in the training process;
        if $E_i < E'$ **then**
          One last round of backward propagation takes place at CH of the sub-cluster where the client resides;
        else
          Backward propagation to the client;
        end if
      end for
    end for
  end for
  // For *Server*
  Accept each CH parameter $w_{j,t}(j = 1, 2, ..., m)$ and classify aggregate;
end for

In Sect. 4.1, based on the DBKM algorithm, we provide the transmission distribution for different numbers of clients, which achieves the minimum transmission distance. In Sect. 4.2, we simulate different numbers of clients to verify the proposed system model and algorithm. We use the following algorithms in terms of accuracy and transmission energy consumption for comparison with our proposed algorithms: 1) **Hierarchical Clustering (HC)**, in which clients are clustered based on their data similarity; 2) **Federated Learning (FL)**, in which clustering is not performed, and each client transmits parameters directly to the server after training; 3) **K-Means (KM)**, in which clients are clustered based on their distance.

4.1 Clustering Effects for Two Special Networks

Firstly, in the case of the initial client distribution, we cluster according to client data similarity. Figure 2(a) represents the initial distribution of clients when the number of clients n is 20 and the number of sub-clusters m is 8. The dots of the same color indicate that the data is IID for the clients. Then according to the DBKM algorithm, the clients in each cluster are divided into several sub-clusters. After clustering, the client with the highest energy in each sub-cluster is selected as the CH. Figure 2(b) shows the transmission distribution of the client after the CH is selected.

Table 2. Simulation parameters

Simulation parameters	Values
Global iteration number T	80
Channel bandwidth B	1 MHZ
Channel transmission power P	1 W
Reference distance D_0	1 m
Channel loss factor v	2
local iteration number e	5
Noise power spectral density N_0	−174 dBm/Hz
Channel power gain h_0	−30 dB
Channel fading factor ρ	3
The size of data transferred from client i to CH j γ	9 MB
The size of data transferred from CH j to server α	552 MB
Model partition point l_r	7

Graphs with the same color and shape represent clients of the same sub-cluster. A larger graph of the same color and shape represents the CH. Figure 2(c) and Fig. 2(d) represent the initial distribution of clients and the distribution of clients after clustering when the number of clients n is 40 and the number of sub-clusters m is 8.

Table 3. Comparison of total transmission distances(km) when the number of clients is 20

Scheme	The distances before clustering	The distances after clustering
ECO	75.01	42.25
HC	72.11	106.89
KM	76.59	47.55
FL	74.10	74.52

We conducted distribution experiments with varying numbers of clients and statistically compared the transmission distances between clients before and after clustering, as shown in Table 3 and Table 4. The data in Table 3 represents the distance comparison before and after clustering for various schemes when the number of clients is 20. In Table 3, we can see that after clustering, our scheme ECO has a transmission distance lower than that of other schemes. The HC scheme, however, has the longest transmission distance, as it prioritizes data similarity over the geographical locations of the clients. The transmission distances before clustering for all schemes are essentially the same as those in the FL scheme because, before clustering, the clients directly transmit parameters to

Table 4. Comparison of total transmission distances(km) when the number of clients is 40

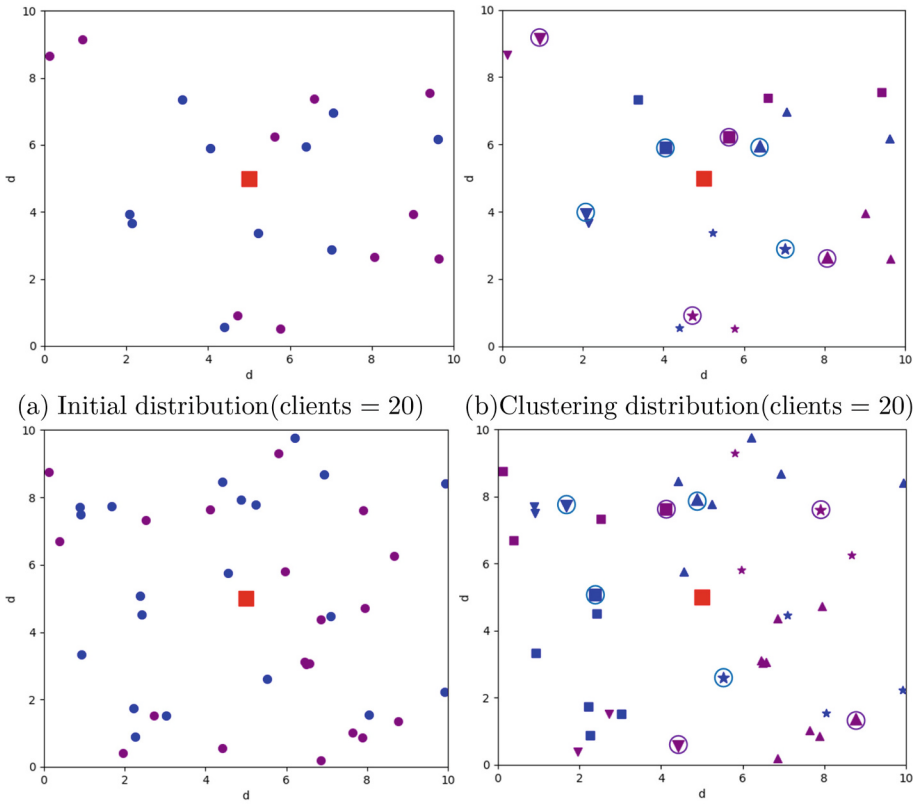
Scheme	The distances before clustering	The distances after clustering
ECO	146.88	65.54
HC	149.11	215.73
KM	153.36	77.66
FL	149.12	150.91

the server after training, which is the procedure followed in the FL scheme. The data in Table 4 represents the distance comparison before and after clustering for various schemes when the number of clients is 40. In Table 4, we can also see that our scheme ECO achieves the smallest transmission distance compared to other schemes, while the HC scheme has the largest distance. Similarly, the transmission distances before clustering for all schemes are the same as the FL scheme. The specific reason is the same as mentioned above.

4.2 Simulation Results for More Experiments

To demonstrate the effectiveness of our algorithm, we conducted additional experiments. Firstly, in Fig. 3, we increase the number of clients from 20 to 60 in steps of 10. For each type, we conducted 50 sets of experiments. As we can see, in Fig. 3(a), among all algorithms, the total transmission energy consumption increases with the number of clients. Compared to other algorithms, our algorithm achieves the minimum total transmission energy consumption for every number of clients. The experimental results show that compared with the HC algorithm, FL algorithm, and KM algorithm, our algorithm reduces the transmission energy consumption by 27%, 61%, and 82% respectively. Figure 3(b), Fig. 3(c), and Fig. 3(d) respectively show the changes in total transmission energy consumption of each algorithm as the number of clients increases when the number of sub-clusters is 10, 12, and 14. Secondly, in Fig. 3, we can see that as the number of sub-clusters increases, our scheme achieves more significant results compared to other schemes. Because as the number of sub-clusters increases, more clients can transmit parameters to closer CHs during training. At the same time, the total transmission energy consumption of the FL algorithm does not vary significantly with different numbers of sub-clusters. Because the FL algorithm does not use clustering, clients directly transmit parameters to the server after training, so it is not affected by the number of sub-clusters. This experiment validates the effectiveness of our algorithm.

In Fig. 4, we compare the accuracy of the four algorithms. We set the total number of iterations T from 0 to 80 in steps of 10. In Fig. 4(a) and Fig. 4(b), we set the number of sub-clusters to the same value of 8, and the number of clients to 20 and 40, respectively. We can see that in both cases, the ECO algorithm achieves the highest accuracy, which is comparable to the HC algorithm. Because our algorithm takes into account the non-IID nature of the data when optimizing transmission energy consumption. The KM scheme and FL scheme fail to consider the non-IID during training, resulting in a decrease in accuracy.



(a) Initial distribution(clients = 20) (b)Clustering distribution(clients = 20)

(c) Initial distribution(clients = 40) (d)Clustering distribution(clients = 40)

Fig. 2. Distribution results.(a)Initial client distribution when the number of clients is 20. (b)Clustering client distribution when the number of clients is 20. (c)Initial client distribution when the number of clients is 40. (d)Clustering client distribution when the number of clients is 40.

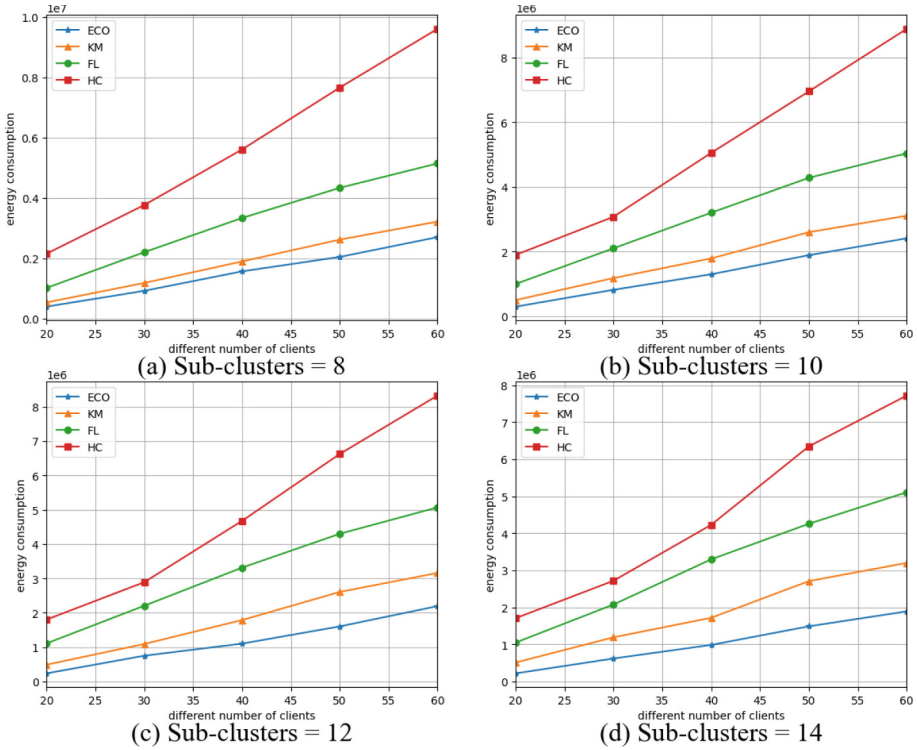


Fig. 3. The variation in total transmission energy consumption with the number of clients. (a)The number of sub-clusters is 8. (b)The number of sub-clusters is 10. (c)The number of sub-clusters is 12. (d)The number of sub-clusters is 14.

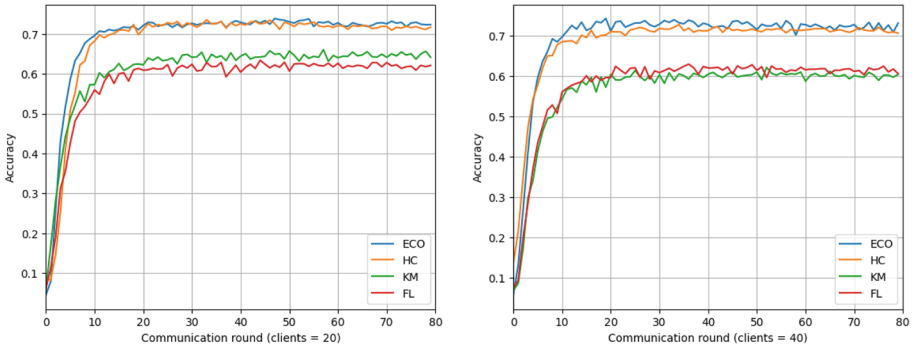


Fig. 4. Accuracy comparison of different schemes. (a)clients = 20, sub-clusters = 8. (b)clients = 40, sub-clusters=8.

5 Conclusion and Future Work

In this paper, we study the optimization problem of transmission energy consumption in CFL. Firstly, we establish a mathematical model for transmission energy consumption in a CFL scenario and find that the problem of transmission energy consumption can be transformed into the problem of transmission distance. Then, we propose the DBKM algorithm to divide multiple sub-clusters within each cluster. Finally, by dynamically adjusting the training mode of the clients to reduce the number of transmissions and by partitioning the model to lower the client's energy usage, we ensure that the client has enough energy to complete the training. The experimental and simulation results show that the transmission energy consumption can be reduced effectively by using model partition, the DBKM algorithm, and dynamically adjusting the training mode of the clients.

This article attempts to reduce transmission energy consumption through techniques such as intra-cluster sub-clustering and model partition. In the future, we will consider dynamic changes in the partition points of the client model and the number of sub-clusters.

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, vol. 54, pp. 1273–1282 (2017)
2. Li, Q., et al.: A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE Trans. Knowl. Data Eng. **35**(4), 3347–3366 (2023)
3. Zhu, J., Cao, J., Saxena, D., Jiang, S., Ferradi, H.: Blockchain-empowered federated learning: challenges, solutions, and future directions. ACM Comput. Surv. **55**(11), 1–31 (2023)
4. Zheng, Y., Lai, S., Liu, Y., Yuan, X., Yi, X., Wang, C.: Aggregation service for federated learning: an efficient, secure, and more resilient realization. IEEE Trans. Depend. Sec Comput. **20**(2), 988–1001 (2023)
5. Qu, Y., Uddin, M., Gan, C., Xiang, Y., Gao, L., Yearwood, J.: Blockchain-enabled federated learning: a survey. ACM Comput. Surv. **55**(4), 1–31 (2023)
6. Pei, J., Yu, Z., Li, J., Jan, M.A., Lakshmanna, K.: TKAGFL: a federated communication framework under data heterogeneity. IEEE Trans. Netw. Sci. Eng. **10**(5), 2651–2661 (2023)
7. Aouedi, O., Piamrat, K., Muller, G., Singh, K.: Federated semisupervised learning for attack detection in industrial Internet of Things. IEEE Trans. Ind. Inf. **19**(1), 286–295 (2022)
8. Zhang, Z., Gao, Z., Guo, Y., Gong, Y.: Scalable and low-latency federated learning with cooperative mobile edge networking. IEEE Trans. Mobile Comput. **23**(1), 812–822 (2024)
9. Ma, Z., Xu, Y., Xu, H., Meng, Z., Huang, L., Xue, Y.: Adaptive batch size for federated learning in resource-constrained edge computing. IEEE Trans. Mobile Comput. **22**(1), 37–53 (2023)

10. Wang, Z., Xu, H., Liu, J., Xu, Y., Huang, H., Zhao, Y.: Accelerating federated learning with cluster construction and hierarchical aggregation. IEEE Trans. Mobile Comput. **22**(7), 3805–3822 (2023)
11. Sattler, F., Muller, K., Samek, W.: Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints. IEEE Trans. Neural Netw. Lear Syst. **32**(8), 3710–3722 (2021)
12. Long, G., Xie, M., Shen, T., Zhou, T., Wang, X., Jiang, J.: Multi-center federated learning: clients clustering for better personalization. World Wide Web J. **26**(1), 481–500 (2023)
13. Ngo, H., Fang, H., Rumbut, J., Wang, H.: Federated fuzzy clustering for decentralized incomplete longitudinal behavioral data. IEEE Internet Things J. **11**(8), 14657–14670 (2021)
14. Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. IEEE Trans. Inf. Theor. **68**(12), 8076–8091 (2022)
15. Hu, X., Qin, J., Shen, Y., Pedrycz, W., Liu, X., Liu, J.: An efficient federated multiview fuzzy c-means clustering method. IEEE Trans. Fuzz Syst. **32**(4), 1886–1899 (2024)
16. Wei, X., Huang, H.: Edge devices clustering for federated visual classification: a feature norm based framework. IEEE Trans. Imag Proc. **32**, 995–1010 (2023)
17. Lee, J., Ko, H.: Energy and distribution-aware cooperative clustering algorithm in internet of things (IoT)-based federated learning. IEEE Trans. Veh. Technol. **72**(10), 13799–13804 (2023)
18. Li, Y., Qin, X., Chen, H., Han, K., Zhang, P.: Energy-aware edge association for cluster-based personalized federated learning. IEEE Trans. Veh. Technol. **71**(6), 6756–6761 (2022)
19. Du, R., Xu, S., Zhang, R., Xu, L., Xia, H.: A dynamic adaptive iterative clustered federated learning scheme. Knowl.-Based Syst. **276**, 110741 (2023)
20. Li, C., Li, G., Varshney, P.: Federated learning with soft clustering. IEEE Internet Things J. **9**(10), 7773–7782 (2022)
21. Zhao, Z., Liang, X., Huang, H., Wang, K.: Deep federated learning hybrid optimization model based on encrypted aligned data. Pat Recogn. **148**, 110193 (2024)
22. Bian, J., Xu, J.: Client clustering for energy-efficient clustered federated learning in wireless networks. In: The 2023 ACM International Symposium on Wearable Computing, pp. 718–723. Association for Computing Machinery, Cancun (2023)
23. Wang, X., Han, Y., Leung, V., Niyato, D., Yan, X., Chen, X.: Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun. Surv. Tutorials. **22**(2), 869–904 (2020)
24. Gao, M., Shen, R., Shi, L., Qi, W., Li, J., Li, Y.: Task partitioning and offloading in DNN-task enabled mobile edge computing networks. IEEE Trans. Mobile Comput. **22**(4), 2435–2445 (2023)
25. Deng, X., et al.: Low-latency federated learning with DNN partition in distributed industrial IoT networks. IEEE J. Sel. Areas Commun. **41**(3), 755–775 (2023)
26. Wang, S., Hosseinalipour, S., Gorlatova, M., Brinton, C., Chiang, M.: UAV-assisted online machine learning over multi-tiered networks: a hierarchical nested personalized federated learning approach. IEEE Trans. Netw. Serv Mag. **20**(2), 1847–1865 (2023)
27. Xu, B., Xia, W., Zhao, H., Zhu, Y., Sun, X., Quek, T.: Clustered federated learning in internet of things: convergence analysis and resource optimization. IEEE Internet Things J. **11**(2), 3217–3232 (2024)

28. Zheng, K., Li, Z.: Energy optimization for hierarchical federated learning based on over-the-air computing. In:2023 IEEE Smart World Congress (SWC), pp. 1–8. IEEE, Portsmouth (2023)
29. Zhang, X., et al.: A federated deep reinforcement learning-based low-power caching strategy for cloud-edge collaboration. J. Grid Comput. **22**(1) (2024)
30. Al-Abiad, M., Obeed, M., Hossain, M., Chaaban, A.: Decentralized aggregation for energy-efficient federated learning via D2D communications. IEEE Trans. Commun. **71**(6), 3333–3351 (2023)