Edge Federated Learning for Social Profit Optimality: A Cooperative Game Approach

Wenyuan Zhang[1,2], Guangjun Wu[1(✉)], Yongfei Liu[1,2], Binbin Li[1], and Jiawei Sun[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{zhangwenyuan,wuguangjun,liuyongfei,libinbin,sunjiawei}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. As an emerging machine learning paradigm, federated learning satisfies people's privacy protection for private data properties to a certain extent. Especially in the field of Internet of Things (IoT), edge federated learning (EFL) allows edge devices to save private data locally when collaborative training. Most studies regard edge devices as rational and independent individuals, and use non-cooperative game methods to motivate devices to participate in training and maximize individual interests. But few studies have considered the fact that devices belonging to government or social construction agencies are more concerned with overall benefits than individual ones. In this paper, we propose an incentive mechanism for edge cooperative federated learning: ECFL. From the perspective of overall benefit, ECFL will fairly identify the contribution of edge devices and ensure the overall social benefit of the cooperative system. First, we propose a method of Shapley Value contribution degree estimated by Integrated Gradients based on the method of cooperative game (IG-Shapley Value). This method can quantitatively evaluate the contribution that edge devices provide to the model at each round of training in a fine-grained manner. At the same time, based on IG-Shapley Value, we design a collaborative contribution-aware aggregation algorithm IG-Fedavg. In order to maximize the overall social benefit, we consider the communication, storage and computing overhead of edge devices, and make joint optimization with reference to the contribution of edge devices. Extensive experimental results show that our proposed method can still make the model converge faster and achieve better performance when the overall profit is improved by more than 30%.

Keywords: Federated learning · Cooperative game · Internet of Things · Contribution evaluation

1 Introduction

With the rapid development of the Internet of Things (IoT), a large amount of data is collected by IoT devices, which contains a wealth of information valuable

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 460, pp. 135–153, 2022.
https://doi.org/10.1007/978-3-031-24383-7_8

to society. The emerging federated learning [2,11] paradigm allows collaborative training of distributed clients to extract value from edge data. This training mode avoids the client transmitting its own data to the cloud, thereby reducing the data security problems caused by centralized learning [12,20]. In edge scenarios where a large number of IoT devices are deployed, this training paradigm is called Edge Federated Learning (EFL). The edge devices save their own private data (such as facial recognition images), and only need to upload the training parameters to the edge server for collaborative training.

A great deal of works [4,8,10,15] treat edge devices as independent and rational individuals who do not cooperate but consider how to maximize their own benefits in federated learning. The quality of an individual dataset often determines the benchmark for edge device revenue, as this quantifies the device's contribution to overall training. However, since the incentive goals are selfish and independent, the contribution of equipment is often jointly modeled with heterogeneous resources in order to maximize the individual benefits without considering the overall benefits. [23] considers the multi-dimensional resource differences of mobile edge devices, a lightweight incentive framework FMore is designed. But the incentive of this paper is based on the multi-dimensional auction of individual selfishness. The federated learning incentive mechanism based on reputation and reverse auction theory mentioned by [24] also treats devices as rational and independent individuals.

The reality is that many social infrastructures with unified ownership, such as edge devices deployed by government departments or social service-oriented groups, are not independent and selfish. At this time, a cooperative group is formed between edge devices for collaborative learning. This kind of research based on collective interests is relatively few and incomplete. [14] designed the S-Fedavg aggregation algorithm to optimize the selection of clients, considering the cooperative game between clients. But this work ignores the overall social benefits of participating clients. [13] designed a Contribution-Aware Federated Learning (CAreFL) framework to provide a fair and interpretable quantification of FL participants' contributions to the problem of collaborative learning by collaborating medical institutions in the healthcare domain. However, in resource-constrained edge scenarios, the multi-faceted overhead of edge devices cannot be ignored. [3] proposes a collaborative learning framework for CFL in order to optimize the overall communication performance. But this work does not take into account the overall social profit, nor does it provide a fair contribution quantification mechanism for edge devices with Non-IID data.

Shapley Value is a classic cooperative game contribution evaluation method, which is widely used in machine learning feature selection [22] and model interpretation [1,5,7,18]. At the same time, the cooperating edge devices are actually playing a static cooperative game, which is relative to the non-cooperative game of similar work. In order to solve the problem of fair quantification of the contribution of cooperative devices, we consider using the powerful properties of Shapley Value for data quality evaluation of collaborative learning of edge devices. However, due to its exponential computational complexity [17], when

the number of edge devices grows linearly, the computational load of Shapley Value will not be tolerated. We propose the fringe federation cooperation incentive mechanism ECFL to solve the above problems. First, we innovatively introduced the Integrated Gradients method to approximate Shapley Value. The Shapley Value method based on Integrated Gradients (IG-Shapley Value) can not only reduce the computational complexity of Shapley Value, but also fairly determine the contribution of each client in each round of training in the deep learning process. At the same time, we design a quality-aware federated learning aggregation algorithm IG-Fedavg based on IG-Shapley Value to achieve fair selection of cooperative edge devices. In addition, we fully consider the resource consumption of heterogeneous edge devices, including storage cost, computing overhead and communication overhead. In order to maximize the overall social benefit of co-trained edge devices, we build a joint revenue maximization model based on the IG-Shapley Value method.

The main contributions of this paper are summarized as follows:

- We propose a edge federation cooperation incentive mechanism ECFL. It ensures that in edge collaborative training, the contribution of edge devices in each round of training can be quantified fairly. In addition, ECFL can ensure that the overall social benefit of collaborative training is maximized.
- We model the contribution evaluation problem of edge devices as a static cooperative game problem, and propose the Shapley Value estimated by the method based on Integrated Gradients (IG-Shapley Value) to reduce the computational complexity and ensure fairness. At the same time, based on IG-Shapley Value, we design a collaborative contribution-aware aggregation algorithm IG-Fedavg.
- We conduct extensive experiments with a two-layer CNN model as an example. The experimental results show that the selection of edge devices based on IG-Shapley Value can make the model converge faster and obtain better performance while increasing the overall profit by more than 30%.

The rest of this paper is organized as follows. Section 2 gives an overview of the system and gives some background. In Sect. 3, we propose IG-Shapley Value and IG-Fedavg. In Sect. 4, we propose a social benefit optimization model based on IG-Shapley Value, and then give the experimental analysis results in Sect. 5. Section 6 presents related work, and Sect. 7 summarizes the entire paper.

2 Preliminary

2.1 System Overview

The Edge Federated Learning (EFL) architecture is composed of edge devices and edge parameter servers. We denote edge devices as $D = \{d_1, d_2, \ldots, d_n\}$ and edge parameter servers as S. As social infrastructure, edge devices and edge parameter servers jointly train valuable global models, such as face recognition,

etc. This model has commercial value and social value, and also provides better basic services for the people. We assume that the edge devices participating in the training belong to the same or cooperative institutions, such as government functions or a consortium of enterprises that provide basic urban services. Specifically, edge devices use local datasets to collaboratively train specific tasks specified by the edge parameter server. In the communication round t, the set of participating edge devices is denoted as D^t, and the edge device d_i^t will send the local parameter ω_i^t to the edge server. After the parameter server performs model aggregation, it returns the aggregation parameter ω^{t+1} to the edge devices.

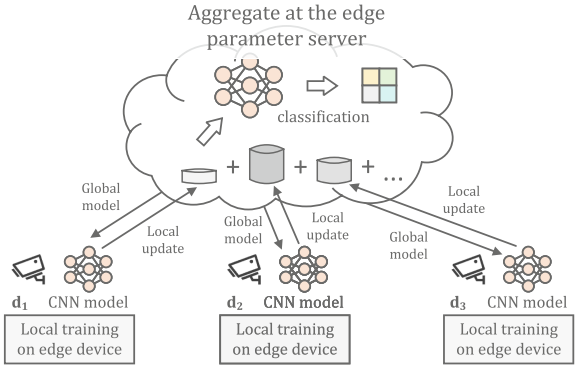

Fig. 1. The process of edge federated learning.

However, as shown in Fig. 1, datasets in edge devices are often Non-IID, and the training results of each round are skewed with different data distributions. This results in slower model convergence and lower accuracy. Considering the Non-IID data of heterogeneous devices, we abstract this problem as the contribution of edge devices. The local parameters provided by each edge device in one round of training have different effects on the overall model, that is, they have different contributions in each round. Because the edge server cannot directly collect the raw data of the edge device, it can only infer different contribution values through parameters. We use the Shapley Value to describe the contribution of edge device training to be able to quantify the cost and benefit of each round of the model.

In addition, heterogeneous edge devices have great differences in communication overhead, storage costs, and computing costs. In order to maximize social benefits, edge devices need to be selected during each round of training. This enables the model with the greatest social value to be trained with the least cost.

For better understanding, Table 1 shows the symbols that appear frequently below.

Table 1. Notation and their explanation

Notation	Explanation
$\phi(\cdot)$	The profit function
$\varepsilon(\cdot)$	Model Approximation Error
δ_i	Contribution ratio of edge device d_i
ω_k	Global aggregation parameters composed of k devices
R_k	Data transfer rate between edge device d_k and edge parameter server
T_{tr}	Transmission delay between edge device d_k and edge parameter server
T_s, T_k	Edge parameter server and edge device d_k training delay
W_{test}, W_{Sh}, W_k	Calculations required for testing, computing Shapley Value and d_k
E_k^s	Energy consumption per unit of communication between d_k and the edge parameter server
E_k^{store}	Storage cost of d_k
E_k^W	Local computation cost of d_k
E_s^W	Edge parameter server computing cost
$\sigma, r, \epsilon, \varphi, \gamma, \varpi_1, \varpi_2$	Positive scalar parameter

2.2 Cooperative Games

Cooperative games are an independent research system relative to non-cooperative games. The players involved in the game coordinate with each other to advance their own interests by forming alliances. This is a game method that emphasizes overall rationality.

We denote the cooperative game as $[N, val]$, where $N = \{N_1, N_2, \ldots, N_n\}$ is a set of n players. $C \subseteq N$ is any subset of N, represented as a alliance. $val(C)$ is expressed as the characteristic function of n-players cooperative game, which reflects the benefit of the alliance.

2.3 Shapley Value

The Shapley Value is derived from cooperative game theory and is a method created by Shapley in 1953 [16] to describe the players' contribution to the total payout to allocate payouts. The cooperating players can earn a payoff based on the sum of their contribution margins after the game is over. In recent work, the Shapley Value has been deeply used to explain machine learning. Usually the total payout represents the predicted value of an instance, and the player represents the feature value. The payoff is the difference between the actual prediction for that instance and the average prediction for all instances. Shapley Value is usually defined as follows:

$$Sh_i(val) = \sum_{\mathcal{C} \subseteq \mathcal{N} \setminus \{x_i\}} \frac{|\mathcal{C}|!(|\mathcal{N}| - |\mathcal{C}| - 1)!}{|\mathcal{N}|!}(val(\mathcal{C} \cup \{x_i\}) - val(\mathcal{C})), \tag{1}$$

where $\mathcal{C}$ is the subset of model features, $|\mathcal{N}|$ is the number of features, $\frac{|\mathcal{C}|!(|\mathcal{N}|-|\mathcal{C}|-1)!}{|\mathcal{N}|!}$ is the weight of $\mathcal{C}$, and x is the eigenvalue vector. Shapley Value is the only attribution method that satisfies the four properties of cooperative games, which ensures the fairness of the method. The four properties are:

Efficiency: The accumulation of features contributions is equal to the difference between the predicted and predicted mean of x, where $\hat{f}(\cdot)$ is a prediction model:

$$\sum_{j=1}^{|\mathcal{N}|} Sh_j = \hat{f}(x) - E_X(\hat{f}(X)), \tag{2}$$

Symmetry: The features j and k contribute the same if and only if they contribute the same to all possible alliance:

$$\begin{aligned} if \quad & val(C \cup x_j) = val(C \cup x_k) \\ for\ all \quad & C \subseteq \{x_1, \ldots, x_{|N|}\}\{x_j, x_k\} \\ then \quad & Sh_j = Sh_k, \end{aligned} \tag{3}$$

Dummy: The feature j that does not change the predicted value, no matter it is added to any feature alliance, the Shapley Value is always 0:

$$\begin{aligned} if \quad & val(C \cup \{x_j\}) = val(C) \\ for\ all \quad & C \subseteq \{x_1, \ldots, x_{|N|}\} \\ then \quad & Sh_j = 0, \end{aligned} \tag{4}$$

Additivity: For games with combined payouts, the corresponding Shapley Value is as follows:

$$Sh_j = Sh_j^+. \tag{5}$$

2.4 Shapley Value in EFL

To maximize the overall profit of Edge Federated Learning, we need to quantify the contribution of local models provided by edge devices. We use the rigorously proven Shapley Value as a quantification method. Specifically, the edge devices participating in the t round form a cooperative game alliance, which can be defined as $[\omega_{local}^t, val]$, where $\omega_{local}^t = \{\omega_d^t\}_{d \in D^t}$, val is defined as contributions for computing D^t subset alliance.

The method for estimating Shapley Value is detailed in Sect. 3.1. After obtaining the contribution of different edge devices, joint modeling is carried out according to the contribution and the operating cost of the target device, which can ultimately maximize the overall social benefit.

3 Contribution and Aggregation Algorithms

In this section, we introduce our proposed IG-Shapley Value and the aggregation algorithm designed based on the contribution of edge devices.

3.1 Estimate of Shapley Value

Since the exact Shapley Value needs to face exponentially increasing computation, such time overhead is intolerable in Edge Federated Learning. We use the Integrated Gradients (IG) method [19] to estimate the contribution of each edge device to a training round. We extend the Shapley Value to IG-Shapley Value based on IG, predicting the gradient to reformulate the integral as the expectation of player behavior. The method is shown to conform to *Sensitivity* and *Implementation Invariance*, and satisfy:

$$\sum_{i=1}^{n} IG_i(x) = F(x) - F(x'), \tag{6}$$

The integral expression is as follows:

$$IG_i(x) ::= (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} \mathrm{d}\alpha, \tag{7}$$

In a real edge environment, the result of the integral can be approximated to reduce the computational complexity:

$$IG_i^{approx}(x) ::= (x_i - x_i') \times \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}, \tag{8}$$

where x' is the baseline input, and x' can be set to an all-zero vector according to [19]. This paper takes the validation set $V = \{x_1, x_2, \ldots, x_n\}$ in the edge server as the input x_i vector. The respective Shapley Values are approximately estimated based on the locally trained models of different edge devices. The Shapley Value can be expressed as the quantitative contribution of an edge device to the final trained global model in a certain round. $F(\cdot)$ represents a neural network model.

In this paper, we take a two-layer CNN network as an example. $F(\cdot)$ is a CNN network whose input vector is image data. m represents the samples generated within the algorithm based on the input to estimate the final Shapley Value. Eigen roots with larger absolute Shapley Values are important. Since global contribution is required, we average the absolute values of the IG-Shapley Values of all validation vectors for each model:

$$Sh_i^t = \sum_{j=1}^{|V|} \sum_{i=1}^{n} |IG_i^j|, \tag{9}$$

where $|V|$ represents the validation set size. Then Sh_i^t is the final estimated Shapley Value of device d_i in round t, that is, its contribution in this round.

3.2 IG-Shapley Value Based Federated Averaging (IG-Fedavg)

Based on IG-Shapley Value, we innovatively propose a federated learning aggregation algorithm IG-Fedavg. Details of the algorithm refer to Algorithm 1. Initially, the edge server randomly initializes the global parameter ω_0 and broadcasts it to all edge devices. After the edge device performs local computation, it returns the parameter ω_i^t of this round of training to the edge server within a certain delay limit. The edge server executes IG-Shapley Value according to the local test set, and calculates the contribution of each client in this round of aggregation. After obtaining the contribution set Sh^t, sort according to the size of the contribution from small to large, and select the appropriate edge device parameters according to the optimization result k (see Sect. 4 for the calculation method of k). At this time, the set of edge devices D^t is determined, and a signal is sent to them to update and share.

For each edge device, after receiving the update request, it performs local training according to the parameters sent by the edge server and the local data set, and fits the local model of E *epochs*. After the edge device calculates the new model parameters, it sends an update to the edge server. After the edge server collects these updated parameters, it aggregates the parameters to obtain the global parameters of a new round of communication. The aggregation process can also refer to the flow shown in Fig. 1.

4 System Model

This section models the overall architecture of federated learning at the edge. The purpose of modeling is to maximize the social benefit of the system. Our model takes into account modeling the benefits of each round of training, as well as modeling the energy costs of heterogeneous edge devices in terms of communication, storage, and computing.

4.1 Income Model Based on IG-Shapley Value

The overall income of the system comes from the revenue of model training. In this section, we design a income model based on IG-Shapley Value considering the profit changes at the model communication round level.

Let the overall profit function be:

$$\phi^t(F) = \frac{\kappa}{1 + e^{r\varepsilon^t(F) - \epsilon}}, \tag{10}$$

where κ, r, ϵ are positive scalars, $\varepsilon^t(F)$ is the approximate error of the model in round t, when $\varepsilon^t(F) \to 0$, the current round profit $\phi^t(F)$ reaches the maximum value. A large $\varepsilon^t(F)$ results in a small profit $\phi^t(F)$.

Algorithm 1: IG-Shapley Value based Federated Averaging (IG-Fedavg)

Input: k^t: optimal number of edge device selections in round t; ω: parameter of the edge server model; η_i: learning rate of edge device i; B: minibatch size; T: maximum number of communication rounds; E: number of local epochs; D: set of edge devices;

Output: parameter ω^T of the edge server model;

for *each round t=1,2,...,T* **do**
  edge server executes:
  $Sh^t = \sum_{j=1}^{|V|} \sum_{i=1}^{n} |IG_i^j|$;
  $Sh^t_{sort} \leftarrow sort(Sh^t)$;
  $D^t \leftarrow choose(Sh^t_{sort}, k^t)$;
  clients (D^t) **execute:**

  for *each client* $i \in D^t$ **do**
    $\omega_i \leftarrow \omega^t$ **for** *each local epoch* e **do**
      for *batch* $b \in B$ **do**
        $\omega_i^{t+1} \leftarrow \omega_i^t - \eta_i \nabla l(\omega_i^t; b)$;
      end
    end
    return ω_i^{t+1};
  end

  edge server executes:
  $\omega_k^{t+1} = \frac{\sum_{i=1}^{k} |\mathbb{D}_i| \omega_i^{t+1}}{\sum_{i=1}^{k} |\mathbb{D}_i|}$;
end

Through the method described in Sect. 3.2, we obtain the candidate set D^t of t rounds of training within the specified time delay, and obtain all the Shapley Values calculated by the set:

$$Sh^t = \{Sh_1^t, Sh_2^t, \ldots, Sh_K^t\}, \tag{11}$$

where $K = |D^t|$. $\varepsilon^t(F)$ is expressed as:

$$\varepsilon^t(F) = \varphi(\sum_{i=1}^{k} \delta_i^t |\mathbb{D}_i|)^\gamma, \quad k \in K, \tag{12}$$

where φ and γ are positive scalars, and $|\mathbb{D}_i|$ is the dataset size local to d_i. δ_i^t represents the proportion of parameter contribution:

$$\delta_i^t = \frac{Sh_i^t}{\sum_{j=1}^{k} Sh_j^t}, \tag{13}$$

where $\delta_i^t \in [0, 1)$.

4.2 System Latency of EFL

We take into account that edge devices and servers incur system delays in transmitting model parameters. Specifically, the latency of edge parameter servers includes data loading latency and computation latency. Assuming that the parameters of k edge devices are received in round t, their communication links are orthogonal. The data transmission rate between the edge device d_k and the edge parameter server is:

$$R_k^t = B \log_2(1 + \frac{P_k^t h_k}{\sigma^2}), \tag{14}$$

where B is the channel bandwidth and P_k is the transmission power of d_k to the edge server. h_k is the channel gain and σ^2 is the noise power.

The transmission delay between d_k^t and the edge server is:

$$T_{tr}^t = \frac{|\omega_k^t|}{R_k^t}, \tag{15}$$

where $|\omega_k^t|$ is the parameter size uploaded by d_k^t in round t. The maximum tolerated waiting time of the edge parameter server is $\mathcal{T}$, so the constraint condition is obtained:

$$T_{tr}^t \leq \mathcal{T}, \quad t \in T. \tag{16}$$

The training delay of the parameter server in round t is:

$$T_s^t = \frac{W^t}{e_s}, \tag{17}$$

where W^t is the total amount of edge server computing in round t. The amount of computation can be thought of as the number of CPU cycles. e_s is the CPU frequency of the edge server. W^t is expressed as:

$$W^t = W_{test}^t + W_{Sh}^t, \tag{18}$$

where W_{test}^t is the amount of computation required to test the accuracy of the model, which is proportional to the size of the test datasets:

$$\begin{aligned} W_{test}^t &= \tau_{test} E_n C_s E_s \\ &= \tau_{test} (\frac{|\mathbb{D}_{test}|}{E_s}) C_s E_s \\ &= \tau_{test} C_s |\mathbb{D}_{test}|, \end{aligned} \tag{19}$$

where τ_{test} is the total number of epochs tested, E_n is the number of batches in the epoch, E_s is the batch size, C_s is the number of edge server CPU, and $|\mathbb{D}_{test}|$ is the size of test dataset.

W_{Sh}^t is the amount of computation required to estimate Shapley Value for each parameter aggregation process:

$$W_{Sh}^t = C_s|\mathbb{D}_{val}| + m|\mathbb{D}_{val}|, \tag{20}$$

where m is the number of generated samples set in IG, and $|\mathbb{D}_{val}|$ is the size of validation dataset.

W^t can be simplified to:

$$W^t = \tau_{test} C_s|\mathbb{D}_{test}| + (C_s + m)|\mathbb{D}_{val}|, \tag{21}$$

$$W^t \leq \mathcal{W}, \quad t \in T, \tag{22}$$

where $\mathcal{W}$ is the maximum computing load of the edge server.

For the edge server d_k, the training delay in round t is:

$$T_k^t = \frac{W_k^t}{e_k}, \tag{23}$$

where W_k^t is the local calculation amount of d_k, and e_k is the CPU frequency of d_k.

W_k^t is denoted as:

$$\begin{aligned} W_k^t &= \tau_k E_n^k C_k E_s^k \\ &= \tau_k (\frac{|\mathbb{D}_k|}{E_s}) C_k E_s^k \\ &= \tau_k C_k |\mathbb{D}_k|, \end{aligned} \tag{24}$$

where τ_k is the number of epochs set by device d_k, E_n^k is the number of batches in the epoch of d_k, E_s^k is the size of d_k batches, C_k is the number of CPU in d_k, and $|\mathbb{D}_k|$ is the size of local training datasets in d_k.

4.3 EFL Costs and Cooperation Benefits

Since edge devices incur communication overhead when broadcasting local parameters, this brings communication costs to the system. The unit communication energy consumption between the edge server d_k and the edge parameter server in round t is:

$$E_k^{s,t} = P_k^t T_{tr}^t, \tag{25}$$

where P_k^t represents the transmission power of d_k uploading local parameters to the edge server in round t, denoted as:

$$p_k^t = \frac{\sigma^2}{h_k}(2^{\frac{R_k^t}{B}} - 1), \tag{26}$$

Assuming that the storage unit price of the edge server is e_0, the storage cost for the device d_k is:

$$E_k^{store,t} = e_0(|\mathbb{D}_k| + |\omega_k^t|). \tag{27}$$

We suppose the rated power of the edge device d_k is p_k, then its unit calculation cost in round t is:

$$E_k^{W,t} = p_k T_k^t. \tag{28}$$

Assuming that the rated power of the edge parameter server is p_S, its unit calculation cost in round t is:

$$E_s^{W,t} = p_s T_s^t. \tag{29}$$

The total cost of training in round t is:

$$E^{total,t} = \sum_{k=1}^{|D|} a_k (E_s^{W,t} + E_k^{s,t}) + E_k^{store,t} + E_k^{W,t}, \tag{30}$$

where a_k is the cooperative decision value, and $|D|$ is the number of edge devices. When $a_k = 1$, d_k transmits the parameters to the edge parameter server within the specified delay, and $a_k = 0$ indicates that the parameters are not received.

Maximizing system utility is expressed as:

$$\begin{aligned} \max \quad & \Phi^t = \sum_{t=1}^{T} \varpi_1 \phi^t(F) - \varpi_2 (1 - \varpi_1) E^{total,t} \\ s.t. \quad & \begin{cases} W^t \leq \mathcal{W}, & t \in T \\ T_{tr}^t \leq \mathcal{T}, & t \in T. \end{cases} \end{aligned} \tag{31}$$

where ϖ_1 is the weighting factor and ϖ_2 is the mapping factor.

5 Experiments

5.1 Simulation Setting

In system simulations, we validate the proposed method using the MNIST dataset and perform classification tasks based on the FedML framework. Our codes are based on the widely used pytorch-1.11.0 software environment, with Intel(R) Xeon(R) CPU E5-2609 v4 @ 1.70GHz, memory of 128G, and OS of centos7. The MNIST dataset is a ten-class dataset of handwritten digit images widely used for testing federated learning evaluations, which contains 60,000 training examples and 10,000 test examples. The edge parameter server publishes a Convolutional Neural Network (CNN) model that edge devices collaborate to train for classification tasks.

We set up 1000 edge devices distributed in an area of 3×3 km^2, and they have pre-divided Non-IID datasets to simulate the difference data collected by devices in different environments. The transmission power of the edge device is uniformly distributed in $[0.1, 2]\,W$, $e_s = e_k = 5$, $C_s = 4$, $C_k = 2$. The local dataset size is distributed in [2, 2000], the size of the IG generated sample is 50, and the validation set is sampled from the test set with a size of 1000. The epoch of edge devices and edge parameter server is 1, and batch size is 50. The maximum transmission delay is 600, and the maximum calculation amount of the edge server is 150,000. The scalar parameters κ, r, ϵ, φ, γ are 1.25, 9, 5.5, 0.39 and 0.03.

5.2 Performance Analysis of Shapley Value Based Training

In order to verify the training effect of selecting the edge devices through Shapley Value, we conduct comparative experiments. Specifically, for the same number of edge devices, we assume that parameters such as system latency, device power range, and model hyperparameters are the same. As shown in Fig. 2, we compare the overall training accuracy with Shapley Value selection and without this method. In the 200 rounds of communication, we set the Shapley Value selection to not apply for the first 20 rounds. The reason for this setting is that the datasets referenced in the initial stage of model construction are too small, and edge devices that are important to the classification results but have a small amount of data may be deleted through screening, thus affecting the training accuracy.

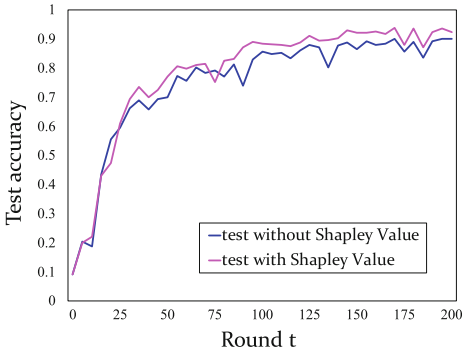

Fig. 2. Test accuracy with or without Shapley Value.

The experimental results show that the model selected by Shapley Value can not only converge, but also the performance of model accuracy and convergence speed are better than the random selection of random edge device parameters.

We choose the round 30, 80 and 120 of the 200 rounds as examples for comparative experiments. They represent the three stages of the rate of increase in model accuracy. Figure 3 shows the distribution of Shapley Value in different rounds.

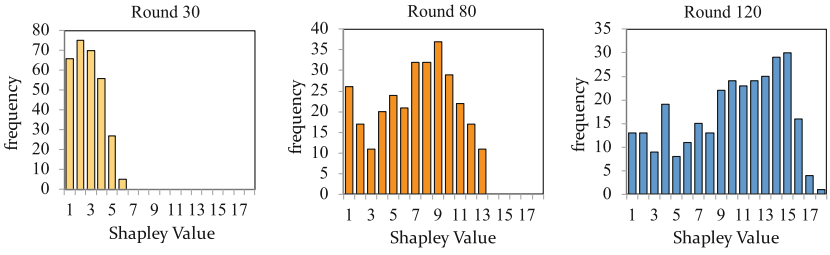

Fig. 3. Shapley Value histograms at 30, 80, 120 rounds.

The experimental results show that with the increase of rounds, the overall distribution of Shapley Value becomes larger and larger. This reflects that as the model is trained, the contribution of edge device data that meets certain characteristics to model training increases. At the same time, the minimum value range of Shapley Value does not increase, indicating that the contribution of these device data to the model is not much different from that in the early training stage.

We sort Shapley Value in ascending order, and delete the corresponding edge device parameters one by one. After testing on the test set, the training error is shown in Fig. 4.

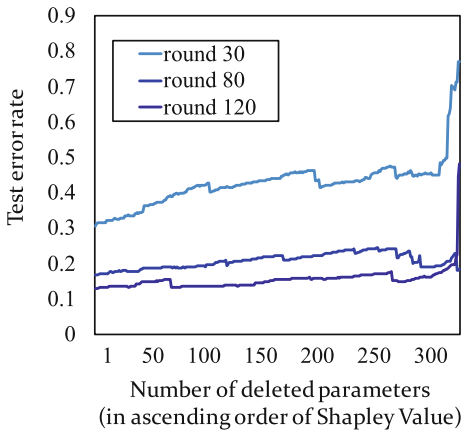

Fig. 4. The effect of deleting parameters in ascending order of Shapley Value on accuracy.

The abscissa in the figure is the Shapley Value of the first 300 edge device parameters collected within the delay range. In the round 30, the error rate increases rapidly with the reduction of edge devices participating in the training, while in the round 80 and 120, the error does not increase much when reducing a small number of device parameters, but decreases at some moments. This is also the reason why the model convergence speed and accuracy will be improved due to the introduction of Shapley Value during the training process.

5.3 Analysis of Social Benefit

Based on the Shapley Value, we calculated individual profit values for each device in each of the three comparison experiments. As shown in Fig. 5, we intercept the 50 with the smallest Shapley Value as an example.

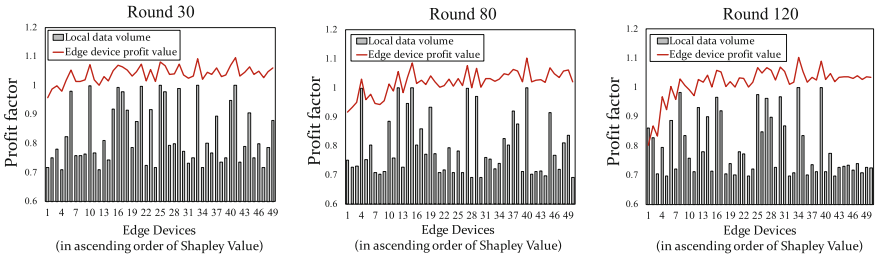

Fig. 5. The relationship between edge device profit value and Shapley Value.

In order to facilitate comparative analysis, we normalized the local data volume of edge devices. For device d_k, the amount of processed data is:

$$|\mathbb{D}_k|_{normal} = \frac{1}{1 + e^{-\frac{|\mathbb{D}_k| \sum_{i=1}^{K} |\mathbb{D}_i|}{K}}} \tag{32}$$

According to the experimental results, we can see that the profit value of edge devices with a large amount of data is relatively high, but the Shapley Value will affect the overall revenue. As the number of rounds increases, the profit of edge devices with smaller Shapley values declines as a whole, which verifies the decrease in the contribution of these nodes to the model accuracy.

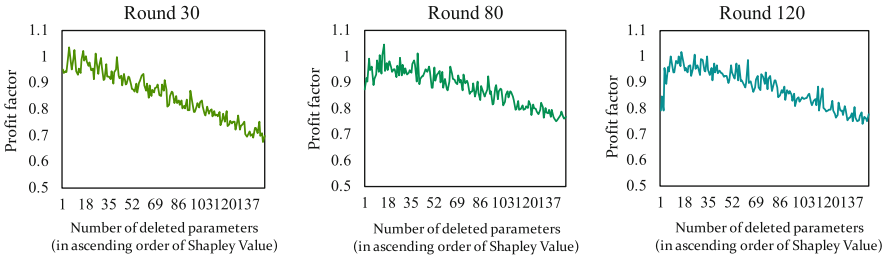

Fig. 6. The total income value for alliances.

As shown in Fig. 6, observe the change of the total revenue of the system in a single round with the deletion of the edge device parameters with a smaller value of Shapley Value.

As training progresses, the range of maximum profit points remains roughly constant, and reducing a small number of edge device parameters will improve the overall profit margin. The highest profit is increased by 12.26%, 19.05% and 29.21% respectively compared with the absence of Shapley Value. In addition, as the number of rounds increases, the profitable range becomes larger and larger, which shows that the Shapley Value in the later stage of training can better represent the contribution to the parameters of the edge devices in this training.

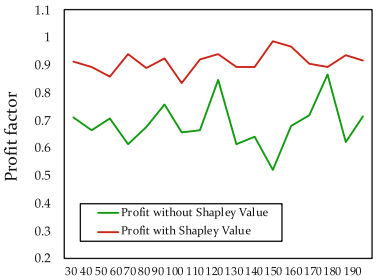

Fig. 7. Profit factor with or without Shapley Value.

Figure 7 shows the difference between the global profit of using IG-Shapley Value and the profit of normal training during a complete training process. The experimental results clearly demonstrate that our proposed method can greatly improve the social benefits. Compared with the baseline, the average profit of the overall training has increased by 32.77%, of which the profit factor of the baseline is 0.6821, and the profit factor of the optimized profit is 0.9045. This strongly validates the effectiveness of our proposed method.

6 Related Work

In recent years, edge federated learning has received extensive attention. Wang et al. [20] focused on a generic class of machine learning models that were trained using gradientdescent based approaches. They analyzed the convergence bound of distributed gradient descent from a theoretical point of view, based on which they proposed a control algorithm that determines the best trade-off between local update and global parameter aggregation to minimize the loss function under a given resource budget. Wang et al. [21] designed the "In-Edge AI" framework in order to intelligently utilize the collaboration among devices and edge nodes to exchange the learning parameters for a better training and inference of the models, and thus to carry out dynamic system-level optimization and application-level enhancement while reducing the unnecessary system communication load. To address the resource-constrained reality of edge devices, he et al. [6] reformulated FL as a group knowledge transfer training algorithm, called FedGKT. FedGKT designed a variant of the alternating minimization approach to train small CNNs on edge nodes and periodically transfer their knowledge by knowledge distillation to a large server-side CNN. Zhang et al. [25] formulated the first faithful implementation problem of federated learning and designed two faithful federated learning mechanisms which satisfy economic properties, scalability, and privacy.

There are many research results in the client incentive direction of edge federated learning. Khan et al. [9] modeled the incentive-based interaction between a global server and participating devices for federated learning via a Stackelberg

game to motivate the participation of the devices in the federated learning process. Zeng et al. [23] proposed FMore as multi-dimensional incentive framework for federated learning. FMore covered a range of scoring functions and was Pareto efficient for some specific cases. It used game theory to derive optimal policies for marginal players and used expected utility theory to guide aggregators to efficiently obtain required resources. Jiao et al. [8] proposed an auction-based market model for incentivizing data owners to participate in federated learning. They designed two auction mechanisms for the federated learning platform to maximize the social welfare of the federated learning service market.

However, in edge devices under government agencies and other departments, the assumption of rational independence of devices does not apply. [13,14] consider the client selection problem, which can solve the fairness problem of contribution to a certain extent, but do not study how to maximize the profit of cooperative equipment. In addition, [3] studied the overhead of optimizing communication and other aspects, but did not propose how to fairly calculate the contribution of heterogeneous devices.

7 Conclusions

For edge federated learning with edge devices as a community of interests in IoT scenarios, we propose an edge federation cooperation incentive mechanism ECFL. We introduce the Shapley Value estimated by Integrated Gradients to fairly compute the contribution of edge devices in each round of training. Furthermore, we design a quality-aware federated learning aggregation algorithm IG-Fedavg based on IG-Shapley Value to achieve fair selection of cooperative edge devices. Taking into account the overall profit and the storage, communication and computing costs of edge parameter servers and edge devices, we optimize social benefits. Extensive experiments show that our proposed method can still make the model converge faster and achieve better performance when the overall profit is improved by more than 30%. In the future, we can further study how to better measure contributions and how to design models when edge devices are dynamic.

Acknowledgements. This work is supported by the National Key Research and Development Program of China (Grant No. 2021YFB3101305), National Natural Science Foundation of China (Grant No. 61931019).

References

1. Aas, K., Jullum, M., Løland, A.: Explaining individual predictions when features are dependent: more accurate approximations to shapley values. Artif. Intell. **298**, 103502 (2021)
2. Bonawitz, K., et al.: Towards federated learning at scale: system design. Proc. Mach. Learn. Syst. **1**, 374–388 (2019)
3. Chen, M., Poor, H.V., Saad, W., Cui, S.: Wireless communications for collaborative federated learning. IEEE Commun. Mag. **58**(12), 48–54 (2020)

4. Ding, N., Fang, Z., Huang, J.: Optimal contract design for efficient federated learning with multi-dimensional private information. IEEE J. Sel. Areas Commun. **39**(1), 186–200 (2020)
5. Ghorbani, A., Zou, J.Y.: Neuron shapley: discovering the responsible neurons. Adv. Neural. Inf. Process. Syst. **33**, 5922–5932 (2020)
6. He, C., Annavaram, M., Avestimehr, S.: Group knowledge transfer: federated learning of large CNNs at the edge. Adv. Neural. Inf. Process. Syst. **33**, 14068–14080 (2020)
7. Heskes, T., Sijben, E., Bucur, I.G., Claassen, T.: Causal shapley values: exploiting causal knowledge to explain individual predictions of complex models. Adv. Neural. Inf. Process. Syst. **33**, 4778–4789 (2020)
8. Jiao, Y., Wang, P., Niyato, D., Lin, B., Kim, D.I.: Toward an automated auction framework for wireless federated learning services market. IEEE Trans. Mob. Comput. **20**(10), 3034–3048 (2020)
9. Khan, L.U., et al.: Federated learning for edge networks: resource optimization and incentive mechanism. IEEE Commun. Mag. **58**(10), 88–93 (2020)
10. LE, T.H.T., et al.: An incentive mechanism for federated learning in wireless cellular networks: an auction approach. IEEE Trans. Wireless Commun. **20**(8), 4874–4887 (2021)
11. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: challenges, methods, and future directions. IEEE Signal Process. Mag. **37**(3), 50–60 (2020)
12. Lim, W.Y.B., et al.: Federated learning in mobile edge networks: a comprehensive survey. IEEE Commun. Surv. Tutor. **22**(3), 2031–2063 (2020)
13. Liu, Z., et al.: Contribution-aware federated learning for smart healthcare. In: Proceedings of the 34th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-22) (2022)
14. Nagalapatti, L., Narayanam, R.: Game of gradients: mitigating irrelevant clients in federated learning. arXiv preprint arXiv:2110.12257 (2021)
15. Pandey, S.R., Tran, N.H., Bennis, M., Tun, Y.K., Manzoor, A., Hong, C.S.: A crowdsourcing framework for on-device federated learning. IEEE Trans. Wireless Commun. **19**(5), 3241–3256 (2020)
16. Shapley, L.S.: Stochastic games. Proc. Natl. Acad. Sci. **39**(10), 1095–1100 (1953)
17. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. Knowl. Inf. Syst. **41**(3), 647–665 (2014)
18. Sundararajan, M., Najmi, A.: The many shapley values for model explanation. In: International Conference on Machine Learning, pp. 9269–9278. PMLR (2020)
19. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International Conference on Machine Learning, pp. 3319–3328. PMLR (2017)
20. Wang, S., et al.: Adaptive federated learning in resource constrained edge computing systems. IEEE J. Sel. Areas Commun. **37**(6), 1205–1221 (2019)
21. Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., Chen, M.: In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Network **33**(5), 156–165 (2019)
22. Yan, T., Procaccia, A.D.: If you like shapley then you'll love the core. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 5751–5759 (2021)
23. Zeng, R., Zhang, S., Wang, J., Chu, X.: Fmore: an incentive scheme of multi-dimensional auction for federated learning in MEC. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), pp. 278–288. IEEE (2020)

24. Zhang, J., Wu, Y., Pan, R.: Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In: Proceedings of the Web Conference 2021, pp. 947–956 (2021)
25. Zhang, M., Wei, E., Berry, R.: Faithful edge federated learning: scalability and privacy. IEEE J. Sel. Areas Commun. **39**(12), 3790–3804 (2021)