LEAP: Optimization Hierarchical Federated Learning on Non-IID Data with Coalition Formation Game*

Jianfeng Lu[1,2], **Yue Chen**[1], **Shuqin Cao**[1,2 †], **Longbiao Chen**[3], **Wei Wang**[1,2] and **Yun Xin**[1]
[1]School of Computer Science and Technology, Wuhan University of Science and Technology, China
[2]Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, China
[3]School of Informatics, Xiamen University, China
{lujianfeng, chenyue, shuqincao}@wust.edu.cn, longbiaochen@xmu.edu.cn, wangwei8@wust.edu.cn
yunxin.wust@gmail.com

Abstract

Although Hierarchical Federated Learning (HFL) utilizes edge servers (ESs) to alleviate communication burdens, its model performance will be degraded by non-IID data and limited communication resources. Current works often assume that data is uniformly distributed, which however contradicts the heterogeneity of IoT. Solutions involving additional model training to check the data distribution inevitably increase computational costs and the risk of privacy leakage. The challenges in solving these issues are how to reduce the impact of non-IID data without involving raw data, and how to rationalize the communication resource allocation for addressing straggler problem. To tackle these challenges, we propose a novel optimization method based on coaLition formation gamE and grAdient Projection, called LEAP. Specifically, we combine edge data distribution with coalition formation game innovatively to adjust the correlations between clients and ESs dynamically, ensuring optimal correlations. We further capture the client heterogeneity to achieve the rational bandwidth allocation from coalition perception and determine the optimal transmission power within specified delay constraints at the client level. Experimental results on four real datasets show that LEAP is able to achieve 20.62% improvement in model accuracy compared to the state-of-the-art baselines. Moreover, LEAP effectively reduces transmission energy consumption by at least about 2.24 times.

1 Introduction

As a novel distributed machine learning paradigm, FL [McMahan *et al.*, 2017] has gained the attention of many fields, such as the Internet of Things (IoT) [Rahman *et al.*, 2023], smart transportation [Pandya *et al.*, 2023], and healthcare [Zhang *et al.*, 2023], to break down the information silos while enabling privacy preservation. With the power of FL, Artificial Intelligence (AI) can effectively handle machine learning tasks involving decentralized data, which draws upon the advantages of distribution machine learning, while also significantly eliminating the privacy risks. During the training process, only model parameters are transmitted without involving local data, breaking information silos and greatly improving training efficiency [Wang *et al.*, 2023]. However, FL performance is affected by various factors, both in the training and transmission phases [Xu *et al.*, 2018]. During the training phase, FL involves hundreds and thousands of clients, and the data distribution of each client is severely different due to diverse user behavior patterns and data collection methods. Consequently, the local data of an individual client fails to represent the overall data distribution of the environment, leading to significant reductions in model performance and compromising the model's generalization capability. In the transmission phase, there is high communication latency and instability between clients and central server (CS) by mass and frequent data transfers, reducing model training efficiency and even increasing the risk of data leakage.

In recent years, numerous solutions have been proposed to address data heterogeneity and communication bottlenecks. For a FL task, the client's data distribution is very important to the performance of the FL model. Significant deviations in the data distribution among clients can severely impair FL model learning and performance. For example, non-IID data causes a number of FL model performance issues, including decreased accuracy, sluggish model convergence, and model communication delays. Reinforcement learning [Xia *et al.*, 2020] and data augmentation [Chen *et al.*, 2023] have been proposed to address the non-IID challenges. These studies overemphasize the importance of individual clients, ignoring the performance improvement of the model benefited from aggregating local updates. In addition, despite their significance, most of these approaches require auxiliary models or extra data transmissions in FL, potentially introducing additional complexities. To alleviate communication pressures, various techniques such as model compression [Zhu *et al.*, 2023], gradient sparsity [Lin *et al.*, 2023], and over-the-air computation [An *et al.*, 2023] were proposed. Although these methods can effectively reduce communication over-

*The full version is at https://arxiv.org/abs/2405.00579.
†Corresponding Author.

head, they may still result in bottlenecks in communication at the CS each communication round. This is due to the fact that during some training epochs, the CS still receives large model weight updates.

Inspired by [Liu *et al.*, 2020], our aim is reduce the impact of non-IID distribution on HFL training by increasing the degree of IID of data distribution between ESs. Additionally, we seek to further optimize the resource allocation scheme to reduce the communication latency. The combination of these two objectives gives rise to an extremely complex and difficult problem, which presents the following three challenges: First, *different edge association relationships represent different edge data distributions.* Once edge association changes, the data distribution will evolve in an unpredictable direction, potentially reducing or increasing the degree of edge IID. Therefore, the impact of changes in association relationships on changes in data distribution is vague and uncertain. Second, *straggler problem caused by worst-performing client.* Edge aggregation latency is susceptible to the communication performance of the worst-performing client in synchronized FL. Dynamic edge association relationships make it difficult to capture communication performance information for each edge coalition. As a result, targeted resource optimization is impossible. Third, *contradiction between task execution latency requirements and clients' energy consumption.* Sufficient resource investment can meet the task requirements but may cause excessive overhead on clients, which is impractical. Consequently, striking a balance that satisfies the needs of both parties simultaneously poses a formidable challenge.

To tackle the abovementioned challenges, we propose a novel optimization method for HFL based on coaLition formation gamE and grAdient Projection method, named LEAP, which not only effectively reduces the impact of cross-edge non-IID, but also improves the communication efficiency. The main contributions of our work are as follows:

- Theoretically, we frame a complex optimization problem that focuses on the effect of multi-dimensional properties (i.e., time delay, energy consumption, and data distribution) on the performance of HFL. We strategically decouple this problem by transforming data distribution optimization into edge correlation analysis and further optimizing heterogeneous resource allocation.
- Methodologically, we construct a coalition formation game by analyzing the relationship between edge association and edge data distribution similarity, and prove the existence of stable coalitions. Moreover, we utilize the gradient projection method to calculate the optimal bandwidth allocation for each coalition, and further determine the transmission performance of heterogeneous clients to ensure that the latency requirements of the tasks are met.
- Experimentally, we validate the effectiveness of LEAP on four real datasets and baselines, it is able to achieve 20.62% improvement in accuracy compared to Meanshift algorithm. Moreover, experiments demonstrate that our optimization method is able to reduce the transmission energy consumption by at least 2.4 times while ensuring that the maximum latency requirement is met.

2 Related Work

In this section, we briefly discuss related work on non-IID data and communication bottleneck in FL.

Non-IID Data. In FL, the non-IID data, caused by the heterogeneity among clients, poses a challenge for training robust FL models due to its impact on slowing down the convergence of the global model [Yao *et al.*, 2019]. Many efforts have been made to address this issue. For instance, Arisdakessian *et al.* [Arisdakessian *et al.*, 2023] proposed a trust-based coalitional FL approach, which mitigates non-IID problems by sharing data of coalition masters. Lu *et al.* [Lu *et al.*, 2023] utilized the Mean-shift algorithm to cluster clients according to data distribution and then selected clients from different clusters to participate in training. Shin *et al.* [Shin *et al.*, 2020] proposed a novel approach that uses XorMixup hybrid data enhancement technology. This approach generates synthetic yet realistic sample data on the server, aiming to solve the issue of unbalanced training datasets in one-shot FL. However, it will bring a large computational burden.

Data-sharing operation raises privacy concerns for clients, thus limiting its application scenarios and it is difficult to operate under privacy-preserving FL. The method of selecting clients through clustering does mitigate the non-IID problem, but it does not guarantee that the final selection result is optimal. In contrast, our work can find optimal edge association relationship in no additional model training without considering the raw data leakage problem.

Communication Bottleneck. In FL, communication cost is a significant factor that affects overall efficiency and effectiveness, while the uplink transmission rate of the underlying client is a major bottleneck in the training process. Many researchers proposed related solutions. For example, Mills *et al.* [Mills *et al.*, 2019] focused on enhancing communication efficiency in FL by combining a distributed Adam optimizer with a compression technique. They emphasized reducing the uploaded data size during training rounds to mitigate communication costs. Building upon of model compression, Liu *et al.* [Liu *et al.*, 2021] applied it to wireless FL to alleviate local computation and communication bottlenecks.

The abovementioned studies were conducted on cloud-based FL systems, whereby the CS receives the local model from the clients. However, in cloud-based FL, the transmission distance can often be considerable, resulting in unstable and undependable communication among the clients and CS. In our study, we make full use of abundant bandwidth resources of ESs and design communication optimization method for heterogeneous clients to solve the resource allocation problem in HFL, as well as to improve the effectiveness of HFL in heterogeneous client environments.

3 System Model and Problem Formulation

In this section, we introduce the workflow of HFL, refine its multidimensional properties, and give an explicit definition of the optimization problem.

3.1 HFL Framework

We consider a HFL framework that consists of a set $\mathcal{N} = \{1, \cdots, N\}$ of clients, a set $\mathcal{M} = \{1, \cdots, M\}$ of ESs, and a

CS. The data set of the client n is denoted as $\mathcal{D}_n = \{\mathcal{X}_n, \mathcal{Y}_n\}$, where $\mathcal{X}_n = \{x_{n,1}, \cdots, x_{n,D_n}\}$ are the training dataset, $\mathcal{Y}_n = \{y_{n,1}, \cdots, y_{n,D_n}\}$ are the corresponding label set, and $|\mathcal{D}_n|$ is the number of training data owned by the client n. The CS aims to train a model, with parameters denoted by a vector ω, over K iterations to minimize the global loss $L^K(\omega)$. The ESs are employed to facilitate the uplink transmission of parameter updates by distributing orthogonal resource blocks to their clients. Then, each client can only associate with one ES to perform the model training. We define $\mathcal{G}_m$ as the set of clients that associate with ES m, i.e., $\mathcal{G}_m = \{n \in \mathcal{N} : a_{m,n} = 1\}$, where $\mathbf{A} = [a]_{M\times N}$ is the edge association matrix and $\mathcal{G}_m \cap \mathcal{G}_{m'} = \emptyset$ for $m \neq m'$. The HFL iteration i consists of four main steps as follows [Ng *et al.*, 2022]:

- *Local Training:* Each client receives the intermediate model from ES m denoted by ω^i, to train a local model using its dataset.
- *Local Model Parameter Transmission:* After every τ_c rounds of local updates, clients transmit the updated local model $\omega_{n,m}^{i,\tau_c}$ to the associated ES m.
- *Edge Aggregation:* ES m aggregates the local model parameters from its associated clients to derive the intermediate model $\omega_m^{i,1}$, which is transmitted back to the clients for the next edge iteration.
- *Global Aggregation:* At the end of predefined intervals τ_e, each ES transmits the intermediate model ω_m^{i,τ_e} to CS for aggregation to derive the updated global model ω^{i+1} and transmits a new global model back to clients for the next global iteration.

The entire process described above will continue until a predetermined number of global training rounds τ_g is reached.

3.2 Multi-Dimensional Properties in HFL

The efficiency and sustainability of HFL are affected by execution time, energy cost and data quality, and these three comprehensive properties consider the impacts of different aspects on FL systems. We hence give a formal definition of each property as follows:

Definition 1. *The muti-dimensional properties of FL $\mathcal{V}$ are represented as a 3-tuple $(\mathcal{T}, \mathcal{E}, \mathcal{J})$, i.e., execution time $\mathcal{T}$, energy consumption $\mathcal{E}$, and data distribution similarity $\mathcal{J}$.*

- *$\mathcal{T}$ is the execution time of a task, which includes computation latency $\mathcal{T}^{\mathcal{C}}$ and communication latency $\mathcal{T}^{\mathcal{U}}$, i.e.,*

$$\begin{cases} \mathcal{T}_{n,t}^{\mathcal{C}} = \tau_c \dfrac{c_n|\mathcal{D}_n|}{f_n}, & (1)\\ \mathcal{T}_{n,t}^{\mathcal{U}} = \dfrac{\mathbb{Z}}{\mathbb{R}_{n,m}}, & (2)\\ \mathcal{T}_{n,t} = \mathcal{T}_{n,t}^{\mathcal{C}} + \mathcal{T}_{n,t}^{\mathcal{U}}, & (3)\\ \mathcal{T}_m = \sum_{t=1}^{\tau_e} \left(\max_{n\in\mathcal{G}_m} \mathcal{T}_{n,t} \right), & (4)\\ \mathcal{T} = \tau_g \max_{m\in\mathcal{M}} \mathcal{T}_m, & (5) \end{cases}$$

  where t is an edge iteration, c_n is the number of CPU cycles for training unit data, f_n is the CPU cycle frequency that determines the computational power, and $\mathbb{Z}$ is the model size. Clients upload local models to ESs via frequency domain multiple access (FDMA). The bandwidth allocation matrix is defined as $\mathbf{B} = [B]_{1\times M}$. B_m is the bandwidth allocated for ES m and $B_{m,n}^U$ is the bandwidth allocated for client n to upload local model. $\mathbb{R}_{n,m}$ represents the uplink transmission rate of client n:

$$\mathbb{R}_{n,m} = B_{n,m}^U \log_2 \left(1 + \frac{p_{n,m} h_{n,m}}{B_{n,m}^U \mathbb{N}_0} \right), \quad (6)$$

  where $p_{n,m}$ denotes the transmission power of the client n, $h_{n,m}$ is the channel gain between client n and ES m, and $\mathbb{N}_0$ is the power of additive white Gaussian noise.

- *$\mathcal{E}$ defines the energy consumption, which contains the training energy $\mathcal{E}^{\mathcal{C}}$ and transmission energy $\mathcal{E}^{\mathcal{U}}$.*

$$\begin{cases} \mathcal{E}_{n,t}^{\mathcal{C}} = \tau_c \varphi c_n |\mathcal{D}_n| f_n^2, & (7)\\ \mathcal{E}_{n,t}^{\mathcal{U}} = \mathcal{T}_{n,t}^{\mathcal{U}} p_n, & (8)\\ \mathcal{E}_{n,t} = \mathcal{E}_{n,t}^{\mathcal{C}} + \mathcal{E}_{n,t}^{\mathcal{U}}, & (9)\\ \mathcal{E}_m = \sum_{n\in\mathcal{G}_m} \tau_g \tau_e \mathcal{E}_{n,t}, & (10)\\ \mathcal{E} = \sum_{m=1}^{M} \mathcal{E}_m, & (11) \end{cases}$$

  where φ is the effective capacitance parameter of the computing chipset.

- *$\mathcal{J}$ denotes the data distribution similarity cross-edge that measured by Jensen Shannon Divergence (JSD) [Menéndez* et al.*, 1997] . JSD and data distribution similarity are negatively correlated. A lower JSD implies that the two sets of data are more likely to fulfill the assumption of being IID. We only compute the JSD value once between the two ESs because of the symmetry of JSD, i.e., $\mathcal{JS}(Q_a, Q_b) = \mathcal{JS}(Q_b, Q_a)$ and $JSD \in [0, 1]$.*

$$\begin{cases} \mathcal{JS}(Q_a, Q_b) = \dfrac{1}{2} \sum_{i\in\{a,b\}} \mathcal{KL}(Q_i, \mathbb{M}_{ab}), & (12)\\ \overline{\mathcal{JS}} = \dfrac{\sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \mathcal{JS}(Q_i, Q_j)}{M}, & (13) \end{cases}$$

  where Q_a, Q_b are the probability distributions of the data under ES a and ES b, $\mathcal{KL}(\cdot)$ denotes the KLD (Kullback-Leibler Divergence) [van Erven and Harremos, 2014]. $\mathbb{M}_{ab}$ denotes the mean distribution of Q_a and Q_b:

$$\mathbb{M}_{ab} = \frac{Q_a + Q_b}{2}. \quad (14)$$

Remark: We mainly consider the case of equal number of local data in this paper. And we are based on a synchronized FL scenario, so the execution latency in a round of global iteration under a ES depends on the last completed client, as

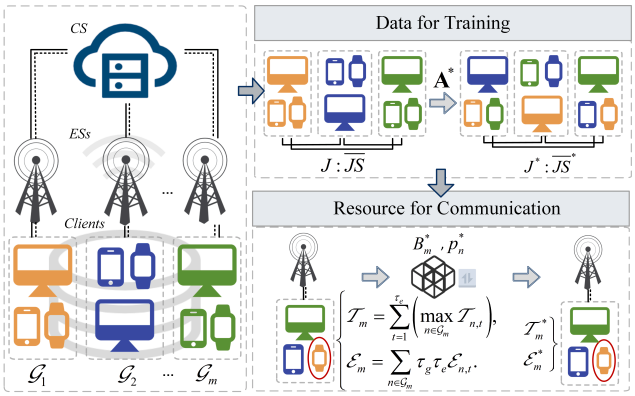


Figure 1: An overview of LEAP

shown in Eq. (4). Due to the high transmission power of ES, the aggregation time and downlink transmission time are ignored compared to the local training and upload time. The clients under the same ES share bandwidth resources equally, which means that $B^U_{1,m} = \cdots = B^U_{n,m}, n \in \mathcal{G}_m$.

3.3 Problem Formulation

For a FL task, high-quality data can enhance model performance, while longer time delays and higher energy consumption bring negative impacts. For ease of representation, we define the utility of the network as a function of $\mathcal{J}$ and $\mathcal{E}$.

$$\mathcal{U} = \lambda_1(1 - \overline{\mathcal{JS}}) - \lambda_2\mathcal{E}, \tag{15}$$

where λ_1 and λ_2 are weighting parameters. Based on the multi-dimensional properties, we model the problem as follows:

$$\mathbb{P}_1: \max_{\mathbf{A},\mathbf{B},\mathbf{p}} \mathcal{U} \quad \& \quad \min_{\mathbf{A},\mathbf{B},\mathbf{p}} \mathcal{T}, \tag{16a}$$

$$\text{s.t.} \quad a_{n,m} \in \{0,1\}, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}, \tag{16b}$$

$$\sum_{m=1}^{M} B_m = B, \forall m \in \mathcal{M}, \tag{16c}$$

$$B_m > 0, \forall m \in \mathcal{M}, \tag{16d}$$

$$p_n \in (0, p_n^{max}], \forall n \in \mathcal{N}, \tag{16e}$$

$$\mathcal{T}_{n,t} \leq \frac{\mathbb{I}}{\tau_e \tau_g}, \forall n \in \mathcal{N}, \tag{16f}$$

where Eq. (16b) indicates that each client can only associate with one ES at a time. Eq. (16c) and Eq. (16d) are the bandwidth constraints of uplink channels. The transmission power constraint is given by Eq. (16e) and Eq. (16f). $\mathbb{I}$ in Eq. (16f) is the maximum execution latency of the currently executed FL task.

LEAP decomposes the original problem $\mathbb{P}_1$ into several subproblems, as shown in Figure.1, which can be solved one by one by combining coalition formation game and gradient projection method. In the coalition formation game, we design a coalition-friendly preference rule $\mathcal{F}$ to determine the optimal edge association relationship $\mathbf{A}^*$. Based on a stable coalition structure, to reduce the communication delay and energy consumption, LEAP optimizes the bandwidth allocation $\mathbf{B}^*$ using the gradient projection method. In addition to this, LEAP captures the heterogeneous client communication resource conditions and determines the optimal transmission power $\mathbf{p}^*$ of the clients based on satisfying the maximum delay of the task.

4 Optimal Solution Based on LEAP

In this section, we address the previously formulated problem $\mathbb{P}_1$. We start by identifying a stable coalition partition, and then optimize bandwidth allocation and transmission power based on this result.

4.1 Optimal Data Distribution

To achieve the goal of optimal network utility, we need to enhance the similarity of data distribution between ESs. Coalition game possesses an excellent tool for revealing the coalition formulation process. We model the problem of minimizing the JSD value of data distribution among ESs as a coalition formation game.

Definition 2. *A coalition formation game $\mathcal{C}$ is represented as a 4-tuple ($\mathcal{N}$, $\mathcal{O}$, $\mathcal{F}$, $\mathbf{A}$), i.e., a set $\mathcal{N}$ of clients, a coalition partition $\mathcal{O}$, a preference relation $\mathcal{F}$, and a game strategy profile $\mathbf{A}$.*

- *$\mathcal{N}$: A set $\mathcal{N} = \{1, \cdots, N\}$ of players.*
- *$\mathcal{O}$: A coalition partition $\mathcal{O} = \{\mathcal{G}\}_1^M$, where $\mathcal{G}_m \subseteq \mathcal{O}$, $\cup_{m=1}^{M} \mathcal{G}_m = \mathcal{N}$ and m denotes the index of the coalition or ES.*
- *$\mathcal{F}$: A preference relation $\succ_n$ is a complete, reflexive, and transitive binary relation over the set of all coalitions that client n may join in, i.e., $\mathcal{G}_1 \succ_n \mathcal{G}_2$ indicates that client n strictly prefers joining coalition $\mathcal{G}_1$ over coalition $\mathcal{G}_2$.*
- *$\mathbf{A}$: A strategy profile of edge association of clients, i.e., $a_{m,n} = 1$ means client n associates with ES $m \in \mathcal{M}$.*

When determining client preferences for multiple coalitions, the order of coalition preferences can be determined according to different rules. For example, the "Selfishness Rule", which only considers individual's choices, and the "Pareto Rule", which never harms the choices of any member in original coalition and new coalition [Zhang *et al.*, 2018]. The former completely ignores the development of other clients in same coalition, posing a risk of harming coalition partition. The latter is too strict for the development of clients and coalitions. To minimize $\overline{\mathcal{JS}}$, we use a preference rule that places greater emphasis on the collective welfare of the entire coalition, called coalition-friendly preference rule, which the definition is as follows,

Definition 3. *If there are two potential coalitions that client n can join, i.e., $\mathcal{G}_a, \mathcal{G}_b \in \mathcal{O}$, then the preference relation is*

$$\mathcal{G}_a \succ_n \mathcal{G}_b \Leftrightarrow \overline{\mathcal{JS}}^n_{\mathcal{G}_b \to \mathcal{G}_a} < \overline{\mathcal{JS}}^n_{\mathcal{G}_b}, \tag{17}$$

where $\overline{\mathcal{JS}}^n_{\mathcal{G}_b \to \mathcal{G}_a}$ means the $\overline{\mathcal{JS}}$ value after client n leaves original coalition $\mathcal{G}_b$ to join the new coalition $\mathcal{G}_a$, and $\overline{\mathcal{JS}}^n_{\mathcal{G}_b}$ means the $\overline{\mathcal{JS}}$ value before client n leaves.

Algorithm 1: Coalition Formation Game for Data Distributions Adjustment

Input: $\mathcal{N} = \{1, \cdots, N\}$, $\mathcal{M} = \{1, \cdots, M\}$, $\mathcal{O}_{cr}$, and L^{max}
Output: Final partition $\mathcal{O}^* = \{\mathcal{G}^*\}_1^M$

1 $\mathcal{O}^* = \emptyset, l = 0$;
2 **repeat**
3 $n = random\{1, \cdots, N\}, n \in \mathcal{G}_m$;
4 **foreach** $\mathcal{G}_{m'} \in \mathcal{O}_{cr}, m \neq m'$ **do**
5 Calculate $\overline{\mathcal{JS}}^n_{\mathcal{G}_m \to \mathcal{G}_{m'}}$;
6 $m' = \min_m \left\{ \overline{\mathcal{JS}}^n_{\mathcal{G}_m \to \mathcal{G}_1}, \cdots, \overline{\mathcal{JS}}^n_{\mathcal{G}_m \to \mathcal{G}_M} \right\}$;
7 **if** $m'! = m$ **then**
8 $\mathcal{G}_m = \mathcal{G}_m \backslash \{n\}, \mathcal{G}_{m'} = \mathcal{G}_{m'} \cup \{n\}$;
9 $\mathcal{O}_{cr} = (\mathcal{O}_{cr} \backslash \{\mathcal{G}_m, \mathcal{G}_{m'}\}) \cup (\mathcal{G}_m \backslash \{n\}) \cup (\mathcal{G}_{m'} \cup \{n\})$;
10 $l = l + 1$;
11 **until** *coalition partition converges or* $l = L^{max}$;

Under the coalition-friendly preference rule, clients aim towards a globally optimal solution by considering the reduction of $\overline{\mathcal{JS}}$ before and after the switch. Based on the preference relations given in Eq. (17), we define the switch rule:

Definition 4. *Given a partition* $\mathcal{O} = \{\mathcal{G}\}_1^M$*, the client* $n \in \mathcal{G}_a$ *decides to leave the original coalition* $\mathcal{G}_a$ *and move to another coalition* $\mathcal{G}_b$*,* $b \neq a$ *, if and only if* $\mathcal{G}_b \cup \{n\} \succeq_n \mathcal{G}_a$*. The new coalition partition can be described as* $\widetilde{\mathcal{O}} \to \{(\mathcal{O} \backslash \{\mathcal{G}_a, \mathcal{G}_b\}) \cup (\mathcal{G}_a \backslash \{n\}) \cup (\mathcal{G}_b \cup \{n\})\}$.

The coalition-friendly preference rule is considered from a coalition standpoint, which can be viewed as a partially collaboration. It is critical to investigate the stability under it.

Definition 5. *If there exists a potential function* ϕ *such that the difference between potential function and utility function remains constant when the client's association relationship changes, the game is an exact potential game.*

$$\phi\left(\widetilde{a_n}, a_{-n}\right) - \phi\left(a_n, a_{-n}\right) = \mathcal{U}_n(\widetilde{a_n}, a_{-n}) - \mathcal{U}_n(a_n, a_{-n}). \tag{18}$$

Theorem 1. *The coalition formation game* $\mathcal{C}$ *is an exact potential game.*

According to Theorem 1, the coalition formation game $\mathcal{C}$ has at least a stable coalition partition. To obtain the solution of the game, we will focus on the algorithm for forming an effective coalition partition, which shown in Algorithm 1. In the coalition formation algorithm, a client n is selected to undergo a comparative update based on the switch rule defined in Definition 4. This rule determines whether the client should leave its current coalition or join another coalition (lines 3-7). We assume that client n leaves current coalition and compute $\overline{\mathcal{JS}}$ of each situation that client n joins in other coalition respectively based on Eq. (13). Therefore, according to the result of assumption, the prioritization of each situation or coalition can be determined. We choose the case that yields the lowest $\overline{\mathcal{JS}}$, and then client n leaves the current coalition $\mathcal{G}_m$, joins the new coalition $\mathcal{G}_{m'}$ if the two coalition are not the same (lines 8-9). Coalition partition will be updated due to this switching (line 10). However, if $\overline{\mathcal{JS}}$ increases after the switch operation, client will remain in the current coalition. The iterative process described above repeats until form a stable partition of coalitions $\mathcal{O}^* = \{\mathcal{G}^*\}_1^M$ where no exchange exists that can bring down $\overline{\mathcal{JS}}$ in current partition $\mathcal{O}_{cr}$ or reach the maximum iteration rounds. We need to perform $\frac{(m-1)m}{2}$ calculations for $\overline{\mathcal{JS}}$. However, in reality, the computation is equally distributed to each ES, so the final time complexity is $O(M)$. This is hardly a burden for a high-performance ES that can respond quickly to clients.

4.2 Optimal Bandwidth Allocation

Based on the final coalition partition, the delay of local training is determined. According to Eqs. (7) to (11), energy consumption is proportional to the execution time of training. From Eq. (2) and Eq. (16c), we can observe that the transmission delay is minimized when $p_n^* = p_n^{max}, \forall n \in \mathcal{N}$. Hence, the joint optimization problem Eq. (16a) is distilled into a single-object optimization problem:

$$\mathbb{P}_2: \quad \min_{\mathbf{B}} \ \mathcal{T}^{\mathcal{U}}, \tag{19a}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} B_m = B, \forall m \in \mathcal{M}, \tag{19b}$$

$$B_m > 0, \forall m \in \mathcal{M}. \tag{19c}$$

It can be observed that $\mathcal{T}_n^{\mathcal{U}}(B_m)$ is a convex function with respect to B_m from Eq. (2). We assume that the worst transmission case in each coalition is n_m^0 and the clients in each coalition have the same status as the worst case. The solution of the communication minimization problem $\mathbb{P}_2$, i.e., optimal bandwidth allocation for coalitions, is denoted as $\mathbf{B}^*$. Then, the $\lambda_2 E$ can reach the minimum value when $\mathbf{B} = \mathbf{B}^*$. Because a strictly convex function has at most one minimum, by setting $n_m = n_m^0, \forall m \in \mathcal{M}$ and $p_{n,m}^* = p_{n,m}^{max}, \forall n \in \mathcal{N}$, the optimization problem is transformed as:

$$\mathbb{P}_3: \quad \min_{\mathbf{B}} \ \sum_{m=1}^{M} \lambda_2 \left|\mathcal{G}_m\right| \tau_g \tau_e \frac{p_{n_m^0}^{max} \mathbb{Z}}{\frac{B_m}{|\mathcal{G}_m|} log_2 \left(1 + \frac{p_{n_m^0}^{max} h_{n,m}}{\frac{B_m}{|\mathcal{G}_m|} \mathbb{N}_0}\right)}, \tag{20a}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} B_m = B, \forall m \in \mathcal{M}, \tag{20b}$$

$$B_m > 0, \forall m \in \mathcal{M}. \tag{20c}$$

Lemma 1. *If* $g_i(x)$ *is convex function,* $\max(\min) g_i(x)$*,* $\sum g_i(b_i x)$ *and* $\sum b_i g_i(x)$ *are also convex functions.*

Since the objective function of the optimization problem is the sum of a convex function, according to Lemma 1, the objective function is also convex with respect to $\mathbf{B}$. We can apply the gradient projection method (GP) [Chu *et al.*, 2023] to allocate bandwidth. The GP method is summarized in Algorithm 2. Through several iterations (lines 3-6), we can obtain the optimal bandwidth allocation in $\mathbb{P}_2$ and $\mathbb{P}_3$.

Algorithm 2: Bandwidth Allocation

Input: $\mathbf{B}(0)$, step size η, accuracy tolerance ϵ, and iteration number J^{max}

$j = 0$;
repeat
  Gradient: $\nabla G(\mathbf{B}_j)$;
  Projection: $P_{\Omega_{\mathbf{B}}}$;
  Update $\mathbf{B}$: $\mathbf{B}_{j+1} \longleftarrow P_{\Omega_{\mathbf{B}}}(\mathbf{B}_j - \eta \nabla G(\mathbf{B}_j))$;
  $j = j + 1$;
until *objective value converges or* $j = J^{max}$;

4.3 Optimal Transmit Power

Once the bandwidth allocation matrix $\mathbf{B}^*$ and stable coalition partition $\mathcal{G}^*$ are determined, the optimization of transmit power for each client can be formulated as follows:

$$\mathbb{P}_4: \quad \min_{\mathbf{p}} \ \lambda_2 \mathcal{E}^{\mathcal{U}}, \tag{21a}$$

$$\text{s.t.} \quad p_n \in (0, p_n^{max}], \forall n \in \mathcal{N}, \tag{21b}$$

$$\mathcal{T}_{n,t} \leq \frac{\mathbb{I}}{\tau_e \tau_g}, \forall n \in \mathcal{N}. \tag{21c}$$

Eq. (21b) gives the clients' transmission power range and Eq. (21c) emphasizes the constraint of maximum execution latency, so the solution needs to satisfy both of them.

Theorem 2. *There exists an optimal solution $p^*_{n,m}$ for problem $\mathbb{P}_4$, i.e.,*

$$p^*_{n,m} = \min\{p_n^{max}, p_{n,\mathbb{I}}\}, \tag{22}$$

where

$$p_{n,\mathbb{I}} = \frac{B^U_{n,m}\mathbb{N}_0\left(2^{\frac{\mathbb{Z}}{B^U_{n,m}\left(\frac{\mathbb{I}}{\tau_e\tau_g} - \tau_c \mathcal{T}^C_{n,t}\right)}} - 1\right)}{h_{n,m}}. \tag{23}$$

5 Experiments

In this section, we conduct extensive experiments to assess the performance of LEAP. We first introduce the experimental setups, and then compare and analyze the effectiveness of our approach to SOTAs.

5.1 Experimental Setup

Datasets and Models. We evaluate the performances of LEAP on two commonly adopted learning models and four real datasets: a LR (Logistic Regression) model on MNIST dataset [Lecun *et al.*, 1998] and a CNN (Convolutional Neural Network) with two convolution layers and three fully connected layers on CIFAR-10 [Krizhevsky, 2009], SVHN [Netzer *et al.*, 2011] and CINIC-10 [Darlow *et al.*, 2018].

Parameter Settings. We set two scale settings on each dataset, 5 ESs with 50 clients or 3 ESs with 10 clients. For each dataset, 5 rounds of local training and 12 rounds of edge iterations are conducted. Among these setting, 100 rounds of global iterations are performed for MNIST and CINIC-10 datasets, while 200 rounds of global iterations for CIFAR-10 and SVHN. We set learning rate to 0.01 for MNIST, CIFAR-10 and CINIC-10, and 0.005 for SVHN. The momentum is set within the range of $[0, 0.9]$ and weight decay is 0.005.

Baselines. Four baselines are considered for comparison with LEAP, consist of two clustering algorithms, a coalition formation method and a model aggregation method.

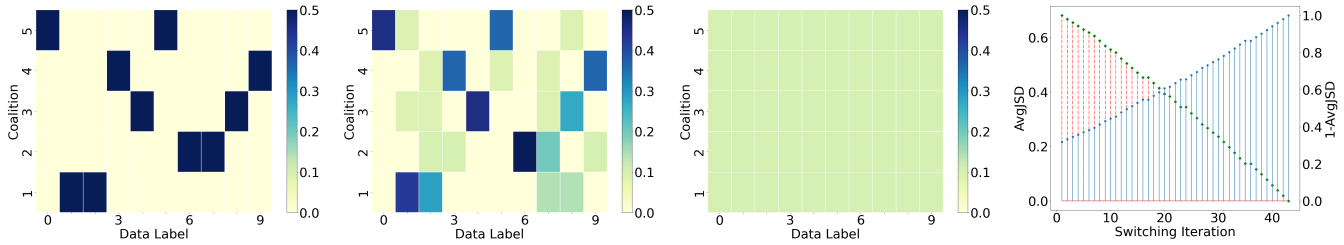
- **Mean-Shift** [Lu *et al.*, 2023]: It is a density-based non-parametric clustering algorithm. One advantage is that it does not require specifying the number of clusters in advance, as it automatically determines this in the process.
- **K-Means** [Lim *et al.*, 2022]: It is an iterative clustering algorithm that partitions data points into K clusters (K is pre-specified) and assigns each data point to the nearest cluster center based on distance.
- **RH** [Ng *et al.*, 2022]: It is a reputation-aware hedonic coalition formation algorithm, in which clients form stable coalition partitions with selfish preferences based on the reputation of cluster heads and their own utility.
- **MA** [Zhang *et al.*, 2021] [Shi *et al.*, 2022]: It is a model aggregation method based on marginal losses. By setting marginal loss thresholds, it becomes possible to identify and reduce the impact of low-quality models on the aggregation process.

5.2 Experimental Results

Validating the effectiveness of mitigating the degree of cross-edge non-IID. Figures 2(a) to 2(c) show the distribution of data for each coalition during the coalition formation process, with the color of each cell indicating the percentage of such data under that coalition. The initial $\overline{\mathcal{JS}}$ is 0.69 with two label categories of each coalition. As the client switching process progresses, the data distributions under each coalition become increasingly similar, with the final $\overline{\mathcal{JS}}$ reaching 0, which means that the distribution in each coalition is same. Figure 2(d) shows the complete variation of $\overline{\mathcal{JS}}$ during the client switching process. Each switching operation demonstrates a consistent decreasing trend of $\overline{\mathcal{JS}}$.

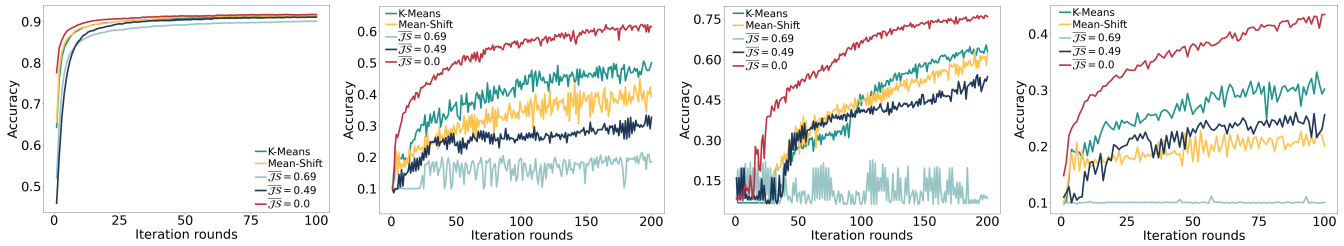
Comparing with K-means and Mean-shift algorithms. Figure. 3 shows the accuracy under different methods and different data distributions. Comparing the initial state, the average accuracy based on the final distribution is improved by 2.9%, 33.3%, 47.6%, and 26.2% in the four datasets, respectively. Based on the same initial conditions, Mean-shift algorithm divides the data into five clusters. While randomly assigning the clients to ESs, the optimal client combination cannot be ensured because of the duplicate labels within the clusters. A similar issue arises when using the K-means algorithm. In addition, the K-means algorithm requires specifying the number of clusters in advance, which further hampers its applicability. It is clear that the final result after optimization based on our method is significantly improved compared to the other two methods, because our method is always in the direction of better when adjusting the combination of data distribution.

Comparing with RH and MA. The the initial correlation relation of Table 1 is based on the experimental results in [Ng *et al.*, 2022] that presented RH, with 10 clients and 3 ESs. Compared to RH and MA, our approach still performs well.



(a) $\overline{\mathcal{JS}} = 0.69$ (b) $\overline{\mathcal{JS}} = 0.49$ (c) $\overline{\mathcal{JS}} = 0.0$ (d) $\overline{\mathcal{JS}}$'s change process

Figure 2: Changes of data distribution and $\overline{\mathcal{JS}}$ during coalition formation.

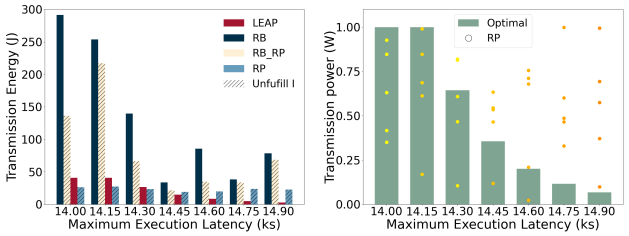


(a) MNIST (b) CIFAR-10 (c) SVHN (d) CINIC-10

Figure 3: Global model performance comparison of different data distributions and methods on four datasets.

Methods	Datasets			
	MNIST	CIFAR-10	SVHN	CINIC-10
RH	88.65%	51.19%	68.93%	32.95%
MA	80.00%	36.31%	53.07%	24.29%
Our	**90.75%**	**58.63%**	**73.77%**	**40.14%**

Table 1: Average model accuracy of different methods based on the finial coalition partition in RH.



(a) Transmission Energy (b) Transmission Power

Figure 4: Transmission energy consumption and transmission power under different optimization schemes.

This is because RA performs association formation with selfish client preference rule without considering the impact on the coalition partition. MA discards some model parameters below the loss threshold when aggregating based on marginal losses, resulting in data wasting.

Verifing the effectiveness on resource allocation. We calculate average transmission energy consumed per round of edge aggregation with random bandwidth allocation (RB), random transmission power (RP), and a combination of RB and RP (RB_RP). From Figure. 4(a), we can observe that LEAP achieves a significant reduction in transmission energy consumption. We notice that in some cases RP is lower, but it fails to satisfy the maximum execution delay. From Figure. 4(b), the randomly determined transmission power is below the optimal value several times, so it fails to satisfy the delay requirement despite producing lower energy consumption.

6 Conclusion

In this paper, a novel optimization method LEAP, which has a lightweight implementation, was proposed to address the impact of muti-dimentional properties on HFL, i.e., data contribution, consumption of time and energy. Due to the stochastic characteristics of data distribution under ESs, edge association was combined with LEAP and a coalition formation game was built to model data distribution under different associations. To reduce the degree of non-IID of cross-edge data, a coalition-friendly preference rule was employed, and the existence of stable coalition partitions was proved. Further, the gradient projection method (GP) was utilized to reduce task execution time in heterogeneous resources within stable coalitions, improving the communication efficiency. Finally, extensive experiments were conducted on various real datasets to validate the effectiveness of LEAP.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No. 62372343, 62072411), in

part by the Zhejiang Provincial Natural Science Foundation of China (No. LR21F020001), and in part by the Key Research and Development Program of Hubei Province (No. 2023BEB024).

References

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, Fort Lauderdale, USA, April 2017. Proceedings of Machine Learning Research.

[Rahman *et al.*, 2023] Anichur Rahman, Kamrul Hasan, Dipanjali Kundu, Md. Jahidul Islam, Tanoy Debnath, Shahab S. Band, and Neeraj Kumar. On the ICN-IoT with federated learning integration of communication: Concepts, security-privacy issues, applications, and future perspectives. *Future Generation Computer Systems*, 138:61–88, 2023.

[Pandya *et al.*, 2023] Sharnil Pandya, Gautam Srivastava, Rutvij Jhaveri, M. Rajasekhara Babu, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Spyridon Mastorakis, Md. Jalil Piran, and Thippa Reddy Gadekallu. Federated learning for smart cities: A comprehensive survey. *Sustainable Energy Technologies and Assessments*, 55:102987, 2023.

[Zhang *et al.*, 2023] Li Zhang, Jianbo Xu, Pandi Vijayakumar, Pradip Kumar Sharma, and Uttam Ghosh. Homomorphic encryption-based privacy-preserving federated learning in iot-enabled healthcare system. *IEEE Transactions on Network Science and Engineering*, 10(5):2864–2880, 2023.

[Wang *et al.*, 2023] Zhilin Wang, Qin Hu, Ruinian Li, Minghui Xu, and Zehui Xiong. Incentive mechanism design for joint resource allocation in blockchain-based federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 34(5):1536–1547, 2023.

[Xu *et al.*, 2018] Wenchao Xu, Haibo Zhou, Nan Cheng, Feng Lyu, Weisen Shi, Jiayin Chen, and Xuemin Shen. Internet of vehicles in big data era. *IEEE/CAA Journal of Automatica Sinica*, 5(1):19–35, 2018.

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

[Chen *et al.*, 2023] Haokun Chen, Ahmed Frikha, Denis Krompass, Jindong Gu, and Volker Tresp. Fraug: Tackling federated learning with non-iid features via representation augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4849–4859, Paris, France, October, 2023. Institute of Electrical and Electronics Engineers.

[Zhu *et al.*, 2023] Xi Zhu, Junbo Wang, Wuhui Chen, and Kento Sato. Model compression and privacy preserving framework for federated learning. *Future Generation Computer Systems*, 140:376–389, 2023.

[Lin *et al.*, 2023] Xiaohan Lin, Yuan Liu, Fangjiong Chen, Xiaohu Ge, and Yang Huang. Joint gradient sparsification and device scheduling for federated learning. *IEEE Transactions on Green Communications and Networking*, 7(3):1407–1419, 2023.

[An *et al.*, 2023] Qiaochu An, Yong Zhou, Zhibin Wang, Hangguan Shan, Yuanming Shi, and Mehdi Bennis. Online optimization for over-the-air federated learning with energy harvesting. *IEEE Transactions on Wireless Communications*, 1-1, 2023.

[Liu *et al.*, 2020] Lumin Liu, Jun Zhang, Shenghui Song, and Khaled B. Letaief. Client-edge-cloud hierarchical federated learning. *ICC 2020 - 2020 IEEE International Conference on Communications*, pages 1–6, Dublin, Ireland, June 2020. Institute of Electrical and Electronics Engineers.

[Yao *et al.*, 2019] Xin Yao, Tianchi Huang, Chenglei Wu, Ruixiao Zhang, and Lifeng Sun. Federated learning with additional mechanisms on clients to reduce communication costs. *ArXiv Preprint*, arXiv:1908.05891, 2019.

[Arisdakessian *et al.*, 2023] Sarhad Arisdakessian, Omar Abdel Wahab, Azzam Mourad, and Hadi Otrok. Coalitional federated learning: Improving communication and training on non-iid data with selfish clients. *IEEE Transactions on Services Computing*, 16(4):2462–2476, 2023.

[Lu *et al.*, 2023] Renhao Lu, Weizhe Zhang, Yan Wang, Qiong Li, Xiaoxiong Zhong, Hongwei Yang, and Desheng Wang. Auction-based cluster federated learning in mobile edge computing systems. *IEEE Transactions on Parallel and Distributed Systems*, 34(4):1145–1158, 2023.

[Shin *et al.*, 2020] Myungjae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. *ArXiv Preprint*, arXiv:2006.05148, 2020.

[Mills *et al.*, 2019] Jed Mills, Jia Hu, and Geyong Min. Communication-efficient federated learning for wireless edge intelligence in iot. *IEEE Internet of Things Journal*, 7(7):5986–5994, 2019.

[Liu *et al.*, 2021] Shengli Liu, Guanding Yu, Rui Yin, and Jiantao Yuan. Adaptive network pruning for wireless federated learning. *IEEE Wireless Communications Letters*, 10(7):1572–1576, 2021.

[Ng *et al.*, 2022] Jer Shyuan Ng, Wei Yang Bryan Lim, Zehui Xiong, Xianbin Cao, Jiangming Jin, Dusit Niyato, Cyril Leung, and Chunyan Miao. Reputation-aware hedonic coalition formation for efficient serverless hierarchical federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2675–2686, 2022.

[van Erven and Harremos, 2014] Tim van Erven, and Peter Harremoës. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[Menéndez *et al.*, 1997] María Luisa Menéndez, Julio Angel Pardo, Leandro Pardo, and María Del Carmen Pardo. The jensen-shannon divergence. *Journal of The Franklin Institute-engineering and Applied Mathematics*, 334(2):307–318, 1997.

[Zhang *et al.*, 2018] Yuli Zhang, Yuhua Xu, Qihui Wu, Yunpeng Luo, Yitao Xu, Xueqiang Chen, Alagan Anpalagan, and Daoqiang Zhang. Context awareness group buying in d2d networks: A coalition formation game-theoretic approach. *IEEE Transactions on Vehicular Technology*, 67(12):12259–12272, 2018.

[Chu *et al.*, 2023] Shunfeng Chu, Jun Li, Kang Wei, Yuwen Qian, Kunlun Wang, Feng Shu, and Wen Chen. Design of two-level incentive mechanisms for hierarchical federated learning. *ArXiv Preprint*, arXiv:2304.04162, 2023.

[Lecun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, A. Bissacco, Bo Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. *Nips Workshop on Deep Learning & Unsupervised Feature Learning*, 2011.

[Darlow *et al.*, 2018] Luke Nicholas Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. Cinic-10 is not imagenet or cifar-10. *ArXiv Preprint*, arXiv:1810.03505, 2018.

[Lim *et al.*, 2022] Wei Yang Bryan Lim, Jer Shyuan Ng, Zehui Xiong, Jiangming Jin, Yang Zhang, Dusit Niyato, Cyril Leung, and Chunyan Miao. Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(3):536–550, 2022.

[Zhang *et al.*, 2021] Jingwen Zhang, Yuezhou Wu, and Rong Pan. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In *Proceedings of the Web Conference 2021*, pages 947–956, New York, United States, April 2021. Association for Computing Machinery.

[Shi *et al.*, 2022] Zhuan Shi, Lan Zhang, Zhenyu Yao, Lingjuan Lyu, Cen Chen, Li Wang, Junhao Wang, and Xiang-Yang Li. Fedfaim: A model performance-based fair incentive mechanism for federated learning. *IEEE Transactions on Big Data*, 1-13, 2022.