

# FedCure: Mitigating Participation Bias in Semi-Asynchronous Federated Learning with Non-IID Data

Yue Chen<sup>1</sup>, Jianfeng Lu<sup>2,3\*</sup>, Shuqin Cao<sup>2</sup>, Wei Wang<sup>1</sup>, Gang Li<sup>4</sup>, Guanghui Wen<sup>5\*</sup>

<sup>1</sup>School of Computer Science and Technology, Wuhan University of Science and Technology, China

<sup>2</sup>Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, China

<sup>3</sup>Key Laboratory of Social Computing and Cognitive Intelligence (Dalian University of Technology), Ministry of Education, China

<sup>4</sup>College of Computer Science, Inner Mongolia University, China

<sup>5</sup>School of Automation, Southeast University, China

{chenyue, lujianfeng, shuqincao, wangwei8}@wust.edu.cn, gli@imu.edu.cn, ghwen@seu.edu.cn

## Abstract

While semi-asynchronous federated learning (SAFL) combines the efficiency of synchronous training with the flexibility of asynchronous updates, it inherently suffers from participation bias, which is further exacerbated by non-IID data distributions. More importantly, hierarchical architecture shifts participation from individual clients to client groups, thereby further intensifying this issue. Despite notable advancements in SAFL research, most existing works still focus on conventional cloud-end architectures while largely overlooking the critical impact of non-IID data on scheduling across the cloud-edge-client hierarchy. To tackle these challenges, we propose FedCure, an innovative semi-asynchronous Federated learning framework that leverages Coalition construction and participation-aware scheduling to mitigate participation bias with non-IID data. Specifically, FedCure operates through three key rules: (1) a preference rule that optimizes coalition formation by maximizing collective benefits and establishing theoretically stable partitions to reduce non-IID-induced performance degradation; (2) a scheduling rule that integrates the virtual queue technique with Bayesian-estimated coalition dynamics, mitigating efficiency loss while ensuring mean rate stability; and (3) a resource allocation rule that enhances computational efficiency by optimizing client CPU frequencies based on estimated coalition dynamics while satisfying delay requirements. Comprehensive experiments on four real-world datasets demonstrate that FedCure improves accuracy by up to 5.1x compared with four state-of-the-art baselines, while significantly enhancing efficiency with the lowest coefficient of variation 0.0223 for per-round latency and maintaining long-term balance across diverse scenarios.

## Introduction

According to Statista, a leading market data provider, the number of IoT devices worldwide is projected to approach 30 billion by 2030 (Vailshery 2024). The rapid proliferation of devices is accelerating the development of edge intelligence services such as smart healthcare (Nguyen et al.

2022), Internet of Vehicles (Devarajan et al. 2025), and smart manufacturing (Leng et al. 2025), but faces data privacy leakage and communication overload. Federated Learning (FL) (Wu et al. 2023c), a distributed machine learning paradigm, addresses privacy concerns by enabling local training and exchanging only model parameters. To further reduce communication latency and improve system efficiency, the semi-asynchronous FL (SAFL) framework with hierarchical architectures (Wang et al. 2023) has gained increasing attention. By combining synchronous aggregation at the client-edge layer with asynchronous updates at the edge-cloud layer (Deng et al. 2024), SAFL alleviates the traditional cloud-end communication bottleneck and significantly improves training efficiency, demonstrating strong potential across diverse distributed intelligent applications.

However, the semi-asynchronous nature of SAFL, combined with heterogeneous client responsiveness in real-world deployments, gives rise to a critical issue of participation bias (Sun et al. 2022). Under asynchronous training, clients' participation frequency heavily depends on its responsiveness. Models from faster-responding clients are frequently aggregated, while the slower clients are marginalized over time. This skewed participation leads to model updates being dominated by a few clients, thereby degrading model generalization. This issue not only compromises system performance but also contradicts the core of AI development. The updated "AI Principles" of OECD in 2024 (OECD 2024) and the 2025 Paris AI Action Summit (Artificial Intelligence Action Summit 2025) all emphasize "Inclusive Growth".

Although some studies have attempted to tackle participation bias by adding restrictions based on device heterogeneity (Zhang et al. 2023), the asynchronous nature of SAFL causes highly dynamic participation, making static modeling insufficient to capture clients' real-time capabilities. Moreover, participation bias often intertwines with additional challenges, such as non-IID data (He et al. 2025), which warrant further investigation. Current researches mainly focus on the cloud-end structure, while limited exploration of edge-end and edge-cloud collaboration, hinder-

\*Corresponding Authors are Jianfeng Lu and Guanghui Wen. Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ing the practical deployment of SAFL (Wu et al. 2023a).

The opposite of participation bias is balanced scheduling, which can improve model comprehensiveness and generalization (Zhu et al. 2023; Mohri, Sivek, and Suresh 2019). However, achieving this in SAFL faces three key challenges: 1) *Non-IID data exacerbates participation bias*. Non-IID data further amplifies certain data features while neglecting other critical information during participation biased training (Li et al. 2024). 2) *Hierarchical architecture complicates scheduling*. Cloud Server (CS) aggregates edge models from groups of clients associated with the same Edge Server (ES), shifting the scheduling focus from individual clients to coalitions (Liu et al. 2020). 3) *Efficiency degradation under fully balanced scheduling*. Balanced scheduling inevitably reduces efficiency due to the participation of high-latency clients, which necessitates a trade-off (Paragliola 2022).

To address these challenges, we propose FedCure, a novel SAFL framework that mitigates participation bias under non-IID data while ensuring model performance and training efficiency. Specifically, we first formalize the challenges as three constraints: the edge association constraint affects data distribution by restricting the client-ES association (Challenge 1), the scheduling constraint imposes long-term participation requirements on coalitions (Challenges 2 & 3), the resource constraint promotes feasible client contribution for efficiency (Challenge 3). FedCure addresses these constraints via three key rules: (i) We construct a coalition formation game to model the client-ES association and design a *preference rule* to minimize the Jensen-Shannon Divergence, thereby reducing non-IID level and satisfying the edge association constraint. (ii) To balance participation and ensure efficiency, we design a *scheduling rule* by modeling the participation frequency of coalitions and combining coalition dynamics (i.e., execution delay). (iii) To further improve training efficiency, we propose a *resource allocation rule* that sets the optimal CPU frequency for clients based on coalition dynamics, ensuring compliance with delay requirements. The specific contributions are as follows:

- Methodologically, prior to training, we propose a preference rule for coalition formation that prioritizes collective benefit, avoiding inefficiencies of selfish and Pareto preferences and mitigating model degradation under non-IID data. During training, to counter efficiency loss from fully balanced participation, we employ Bayesian estimation to capture coalition dynamics and integrate a virtual queue to guide scheduling for more efficient training. Building on the above, we introduce a resource allocation rule to dynamically set optimal CPU frequencies within feasible bounds, further improving efficiency.
- Theoretically, we prove that FedCure guarantees the existence of a stable coalition partition, as the coalition formation game based on the proposed preference rule is proven to be an exact potential game. Moreover, we show that FedCure guarantees the mean rate stability of the virtual queue, thereby achieving long-term balance under SAFL. We also derive an upper bound on the error induced by the trade-off between efficiency and balance.
- Experimentally, we verify the effectiveness of FedCure

on four real-world datasets, comparing it with two clustering and two scheduling methods, and observe a remarkable 5.1x improvement in accuracy. FedCure also attains the lowest coefficient of variation for per-round training latency, with a value of 0.0223. Furthermore, we confirm that FedCure maintains long-term balance under varying weight settings.

## Related Work

**Non-IID Data in SAFL.** In FL, data heterogeneity across clients leads to non-IID distributions, causing local models to drift from the global optimum and ultimately degrading model performance (Mu et al. 2023). Lim et al. (2022) and Lu et al. (2023) utilized the K-Means and Mean-Shift algorithms, respectively, to cluster clients and selectively sample participants from different clusters for training. Distinguishing from the clustering approach, Arisdakessian et al. (2023) mitigated the effects of non-IID by sharing a portion of the coalition master’s data. Wu et al. (2023a) explored the non-IID challenge in SAFL and proposed a heterogeneity-aware client-edge association mechanism, which enable fast and accurate cloud model learning by measuring response latency and collecting clients’ data distributions. However, the IID assumption of global data distribution presents limitations in real-world scenarios.

While the client selection strategy after clustering mitigates the non-IID problem to some extent, it still excludes clients with rare or unique data from training. In contrast, our work focuses on client coalitions, which naturally aligning with the hierarchical structure of SAFL. Moreover, our work continuously guides the system toward a more favorable data distribution without relying on additional assumptions or introducing the uncertainty of random selection, ultimately achieving a stable and efficient coalition structure.

**Scheduling Strategy in SAFL.** Given the limited resources and heterogeneous capability, effective scheduling algorithms have been increasingly studied to enhance model performance and training efficiency. In synchronous FL, Cho, Wang, and Joshi (2020) prioritized efficient clients to accelerate convergence, while Huang et al. (2022) maximized expected cumulative effective participation to improve convergence speed. In asynchronous FL, Zhu et al. (2023) proposed a dynamic client selection approach to reduce the overall training duration. With the advent of SAFL, related works emerged, such as Zhang et al. (2023) proposed a clustered SAFL that groups devices with similar training conditions to mitigate the impact of stragglers. He et al. (2025) proposed Libra, which addresses unfair client scheduling by alleviating the limitations of fast-training devices due to excessive training and upload demands.

Favoring specific clients in scheduling may exhaust their resources, leaving out others that possess important data. Moreover, the impact of non-IID data underlying participation bias also deserves serious attention. In our study, scheduling is based on coalitions with data that exhibits a high level of IID. The scheduling rule jointly considers participation balance and training efficiency, with further efficiency gains achieved via resource optimization.

## System Model

### Workflow of SAFL

A typical SAFL system consists of a set  $\mathcal{N} = \{1, \dots, N\}$  of clients, a set  $\mathcal{M} = \{1, \dots, M\}$  of ESs, and a CS. The dataset of the client  $n$  is denoted as  $\mathcal{D}_n = \{\mathcal{X}_n, \mathcal{Y}_n\}$ , where  $\mathcal{X}_n = \{x_{n,1}, \dots, x_{n,|\mathcal{D}_n|}\}$  is the training dataset,  $\mathcal{Y}_n = \{y_{n,1}, \dots, y_{n,|\mathcal{D}_n|}\}$  is the corresponding label set. We define  $\mathcal{G}_m$  as the set of clients that are associated with ES  $m$ , which form a coalition. Since each client can only be associated with one ES, the coalitions are disjoint, i.e.,  $\mathcal{G}_m \cap \mathcal{G}_{m'} = \emptyset$  for  $m \neq m'$ . The training of SAFL occurs in the following two layers:

1) **Client-Edge Layer:** Each client in coalition  $\mathcal{G}_m$  receives the interim model from ES  $m$  to train a local model using its dataset (Zhou et al. 2023), i.e.,  $F_n(\omega_n) = \frac{1}{|\mathcal{D}_n|} \sum_{i \in \mathcal{D}_n} f_i(\omega_n)$ . After every  $\tau_c$  rounds of local updates, each client transmits its updated model  $\omega_{n,m}^{t_e}$  to ES  $m$  during the  $t_e$ -th edge training round. Upon receiving the local models from all clients in  $\mathcal{G}_m$ , the ES  $m$  proceeds with the edge aggregation process. The updated edge model is defined as

$$\omega_m^{t_e+1} = \frac{\sum_{n \in \mathcal{G}_m} |\mathcal{D}_n| \omega_{n,m}^{t_e}}{|\mathcal{D}_m|}, \quad (1)$$

where  $|\mathcal{D}_m| = \sum_{n \in \mathcal{G}_m} |\mathcal{D}_n|$  is the sample data size of all clients in  $\mathcal{G}_m$ . After  $\tau_e$  rounds of edge updates, the edge model will be forwarded to the CS for global aggregation.

2) **Edge-Cloud Layer:** The CS performs global aggregation as soon as it receives a model from any ES without waiting. However, asynchronous aggregation introduces the challenge of staleness, which may degrade the convergence rate and in turn increase the training time. For coalition  $\mathcal{G}_m$ , its staleness is the number of experienced epochs since its last global update, denoted as  $\varphi$ . For example, when the CS receives a stale model at epoch  $t$  from ES  $m$ , the updated global model will be determined by  $\varphi$  (Liu et al. 2024):

$$\omega^t = (1 - \xi_\varphi) \omega^{t-1} + \xi_\varphi \omega_m, \quad (2)$$

where  $\xi_\varphi = \ell \cdot \mathbb{k}^\varphi$  is the weight of  $\omega_m$  with staleness  $\varphi$ ,  $\ell \in (0, 1)$  is initial weight, and  $\mathbb{k} \in (0, 1)$  is penalty coefficient. Intuitively, the model with smaller staleness has a larger  $\xi_\varphi$  and contributes more to model aggregation.

For a global round  $t$ , a coalition will be selected before the CS sends the latest model. The coalitions that have not uploaded their trained models are considered unavailable in our work. In particular, we define  $\theta_m(t) = 1$  if coalition is available, and  $\theta_m(t) = 0$ , otherwise. Hence, the optional set of coalitions in global round  $t$  is  $\Theta(t) = [\theta_m(t)]_{1 \times M}$ .

### Three Constraints in SAFL

**Edge Association Constraint (EAC)** In the client-edge layer, association relationships affect the IID degree of data. A lower Jensen-Shannon Divergence (JSD) (Menéndez et al. 1997) indicates greater similarity between two data distributions, which means data is more evenly distributed.

**Definition 1.** The average JSD value across the coalitions is calculated as follows:

$$\overline{\mathcal{J}\mathcal{S}} = \frac{1}{\binom{M}{2}} \sum_{i=1}^{M-1} \sum_{j=i+1}^M \mathcal{J}\mathcal{S}(Q_i, Q_j), \quad (3)$$

where  $\mathcal{J}\mathcal{S}(Q_i, Q_j) = \frac{1}{2} \{\mathcal{K}\mathcal{L}(Q_i, \mathbb{M}_{ij}) + \mathcal{K}\mathcal{L}(Q_j, \mathbb{M}_{ij})\}$ ,  $Q_i$  is the probability distribution of data within coalition  $\mathcal{G}_i$ ,  $\mathcal{K}\mathcal{L}(\cdot)$  is Kullback-Leibler Divergence (van Erven and Harremoos 2014), and  $\mathbb{M}_{ij} = \frac{Q_i + Q_j}{2}$  is the mean distribution.

To achieve a lower level of non-IID distribution, the formed coalitions need to satisfy the following constraint, ensuring that no client's switching can reduce  $\overline{\mathcal{J}\mathcal{S}}$ :

$$\overline{\mathcal{J}\mathcal{S}} < \overline{\mathcal{J}\mathcal{S}}_{\mathcal{G}_b \rightarrow \mathcal{G}_a}, \forall n \in \mathcal{N}, \quad (4)$$

where  $\overline{\mathcal{J}\mathcal{S}}_{\mathcal{G}_b \rightarrow \mathcal{G}_a}$  is the average JSD value after client  $n$  leaves original coalition  $\mathcal{G}_b$  to join the new coalition  $\mathcal{G}_a$ .

**Scheduling Constraint (SC)** To avoid local optima in SAFL, we introduce a long-term balance constraint on the participation requirement (Lu et al. 2024), thereby achieving relative balance among coalitions and guaranteeing a minimum number of participation rounds for each coalition,

$$\lim_{\tau_g \rightarrow \infty} \frac{1}{\tau_g} \sum_{t=0}^{\tau_g-1} \mathbb{E}[\chi_m(t)] \geq \delta_m, \delta_m \in (0, 1], \forall m \in \mathcal{M}, \quad (5)$$

where  $\delta_m$  is the minimum fraction that coalition  $m$  should participate in model training,  $\mathcal{X}(t) = [\chi_1(t), \dots, \chi_M(t)]^T$ , and  $\chi_m(t)$  is a binary variable indicating whether coalition  $m$  is scheduled in round  $t$ . Specifically, if CS schedules coalition  $m$ , then  $\chi_m(t) = 1$ ,  $\pi(t) = m$ , and  $\chi_{m'}(t) = 0$  otherwise.  $\pi(t)$  is the index of the scheduled coalition in round  $t$  and  $\Pi$  is the set of  $\pi(t)$  of all rounds. In addition,  $\sum_{m=1}^M \chi_m(t) = 1, t \geq 1$ , which means only one coalition can be scheduled in each round. The CS schedules  $M$  coalitions in the 0-th round for initialization. We set the boundary of the expected scheduling probability as  $\delta_m = \frac{\kappa |\mathcal{D}_m|}{|\mathcal{D}|}$ ,  $\kappa \in [0, 1]$ , where  $|\mathcal{D}|$  is the data size of all coalitions.

**Resource Constraint (RC)** Clients in the scheduled coalition  $\mathcal{G}_{\pi(t)}$  perform synchronized training, facing a trade-off between system efficiency and resource consumption.

$$\mathcal{Z} = \mathbb{E} \left\{ \alpha \left( 1 - \frac{t_n}{\mathcal{T}_m(t)} \right) - C_n(f_n) \right\}, \quad (6)$$

where  $\mathcal{T}_m(t)$  is the training latency,  $t_n = c_n / f_n$  is the computation time,  $c_n$  is the computation load,  $f_n \in (0, f_n^{max}]$  is the computation resource allocated by client  $n$ ,  $C_n(f_n) = \gamma (f_n)^\varsigma$  is the energy consumption of client  $n$  in model training (Yang et al. 2021), and  $\alpha, \gamma, \varsigma$  are weight coefficients.

### Problem Formulation

High-quality data improve model performance, whereas prolonged delays and high energy consumption bring negative impacts. Based on EAC, SC and RC, we model the

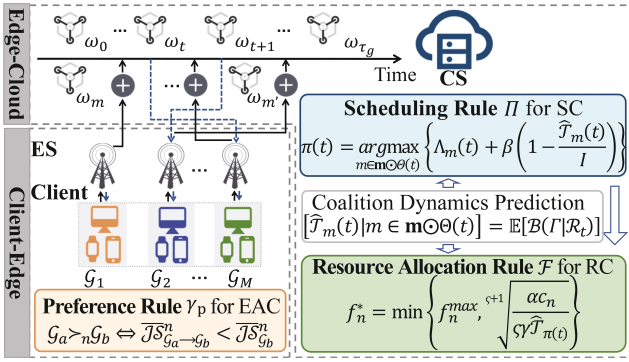


Figure 1: An overview of FedCure.

problem to optimize efficiency and scheduling in SAFL:

$$\mathbb{P}: \quad \min_{\{\mathcal{G}\}_1^M, \pi(t)} \sum_{t=0}^{\tau_g} \frac{\mathcal{J}_{\pi(t)}(t)}{\mathcal{I}} \quad \& \quad \max_f \mathcal{Z}, \quad (7a)$$

$$\text{s.t.} \quad (4), \quad (5) \quad \text{and} \quad (6), \quad (7b)$$

$$\pi(t) \in \mathbf{m} \odot \Theta(t), \quad (7c)$$

$$f_n \in (0, f_n^{\max}], \quad (7d)$$

$$m = \pi(t), \quad n \in \mathcal{G}_{\pi(t)}, \quad (7e)$$

where  $\mathcal{G}_{\pi(t)}$  is the scheduled coalition at round  $t$ ,  $\mathbf{m} = [1, \dots, M]$ , and  $\mathcal{I}$  is the average max training latency.

This problem shows a profound interdependence between constraints, presenting two-fold challenges: (i) SC and RC must be solved jointly under the satisfied EAC, with RC depending on the coalition scheduled by SC, forming a tightly coupled process; (ii) the time-coupled SC further increases the complexity.

## Design of FedCure

In this section, we address problem  $\mathbb{P}$  by designing FedCure with three strategic rules.

**Definition 2.** FedCure is defined as a 3-tuple  $(\Upsilon_p, \Pi, \mathcal{F})$ :

- $\Upsilon_p \Rightarrow \mathcal{O}$ : a preference rule that guides a coalition partition  $\mathcal{O}$  formulating by determining client preferences over ESs, aiming to mitigate non-IID issue.
- $\Pi = \bigcup_{t=1}^{\tau_g} \{\pi(t) | \pi(t) \in \mathbf{m} \odot \Theta(t)\}$ : a scheduling rule  $\Pi$  governing coalition participation  $\mathcal{G}_{\pi(t)} \in \mathcal{O}$  per global round  $t$ , considering the trade-off between balanced scheduling and training efficiency.
- $\mathcal{F} = \bigcup_{t=1}^{\tau_g} \{f_n | n \in \pi(t)\}$ : a resource allocation rule  $\mathcal{F}$  sets the CPU frequency  $f_n$  for clients in scheduled coalition  $\mathcal{G}_{\pi(t)}$  to optimize computational capacity.

**Remark:** As illustrated in Figure 1, the preference rule  $\Upsilon_p$  guides clients to associate with ESs, forming stable coalition partitions that satisfy the EAC by reducing the average JSD value. During asynchronous training, the CS applies the scheduling rule to select participating coalitions, ensuring long-term balance as required by the SC, while maintaining training efficiency. Furthermore, clients in selected coalitions train at CPU frequencies determined by the resource

allocation rule, further improving efficiency in compliance with the RC.

## Optimal Design of Preference Rule

To meet EAC, we construct a coalition formation game (Zhang et al. 2024) where clients dynamically associate with ESs under the preference rule  $\Upsilon_p$  that minimizes  $\overline{\mathcal{J}}\mathcal{S}$  across ESs, forming a stable partition with high IID degree.

**Definition 3.** A coalition formation game  $\mathbb{C}$  is represented as a 4-tuple  $(\mathcal{N}, \mathcal{O}, \Upsilon_p, \Upsilon_s)$ , i.e.,

- $\mathcal{N}$ : A set  $\mathcal{N} = \{1, \dots, N\}$  of players.
- $\mathcal{O}$ : A coalition partition  $\mathcal{O} = \{\mathcal{G}\}_1^M$ , where  $\mathcal{G}_m \subseteq \mathcal{O}$ ,  $\bigcup_{m=1}^M \mathcal{G}_m = \mathcal{N}$ .
- $\Upsilon_p$ : 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.,

$$\mathcal{G}_a \succ_n \mathcal{G}_b \Leftrightarrow \overline{\mathcal{J}}_{\mathcal{G}_b \rightarrow \mathcal{G}_a}^n < \overline{\mathcal{J}}_{\mathcal{G}_b}^n, \quad (8)$$

where  $\mathcal{G}_a \succ_n \mathcal{G}_b$  indicates that client  $n$  strictly prefers to join coalition  $\mathcal{G}_a$  over coalition  $\mathcal{G}_b$ ,  $\overline{\mathcal{J}}_{\mathcal{G}_b}^n$  and  $\overline{\mathcal{J}}_{\mathcal{G}_b \rightarrow \mathcal{G}_a}^n$  represent the  $\overline{\mathcal{J}}\mathcal{S}$  values before and after client  $n$  leaves the coalition  $\mathcal{G}_b$ , respectively.

- $\Upsilon_s$ : A switching rule states that a client  $n \in \mathcal{G}_a$  decides to leave  $\mathcal{G}_a$  and join  $\mathcal{G}_b$  ( $b \neq a$ ) if and only if  $\mathcal{G}_b \cup \{n\} \succeq_n \mathcal{G}_a$  holds, and hence the new coalition partition is updated to

$$\tilde{\mathcal{O}} \rightarrow \{(\mathcal{O} \setminus \{\mathcal{G}_a, \mathcal{G}_b\}) \cup (\mathcal{G}_a \setminus \{n\}) \cup (\mathcal{G}_b \cup \{n\})\}. \quad (9)$$

There are two widely used preference rules: the ‘‘Self-Centered Rule’’ (Ng et al. 2021) prioritizes individual choices but risks coalition cohesion, while the ‘‘Pareto Rule’’ (Zhang et al. 2018) safeguards against harm but may constrain growth. In contrast, we propose a coalition-friendly preference rule  $\Upsilon_p$  that emphasizes the welfare of coalition partition. Clients seek associations by calculating the reduction of  $\overline{\mathcal{J}}\mathcal{S}$  before and after the switch.  $\Upsilon_p$  is assessed from a coalition perspective, similar to partial collaboration. Consequently, examining its stability becomes paramount.

**Definition 4.** An exact potential game exists when a potential function  $\phi$  maintains a constant difference from the utility function, regardless of client association changes, i.e.,

$$\phi(\tilde{a}_n, a_{-n}) - \phi(a_n, a_{-n}) = \mathcal{U}_n(\tilde{a}_n, a_{-n}) - \mathcal{U}_n(a_n, a_{-n}), \quad (10)$$

where  $a_n$  denotes the coalition that client  $n$  joined in, and  $a_{-n}$  represents the associations of all other clients except  $n$ .

**Theorem 1.** The coalition formation game  $\mathbb{C}$ , guided by the preference rule  $\Upsilon_p$ , is an exact potential game that can form a stable partition to satisfy EAC.

Algorithm 1 illustrates the process of constructing a stable partition. Specifically,  $\Upsilon_p$  evaluates the highest preference of the selected client  $n$  ( $n \in \mathcal{G}_m$ ) among coalitions (lines 3-6). We assume client  $n$  leaves its current coalition and calculates  $\overline{\mathcal{J}}\mathcal{S}$  for each potential scenario of joining a different coalition using Eq. (3) (line 5), allowing prioritization based on the resulting values (line 6). Lines 7-9 show the

---

**Algorithm 1: Data Distribution Adjustment**

---

**Input:** Client set  $\mathcal{N}$ , ES set  $\mathcal{M}$ , current partition  $\mathcal{O}_{cr}$ , and maximum iteration round  $L$

**Output:** Final partition  $\mathcal{O}^*$

```
1 Initialize  $\mathcal{O}^* = \emptyset, l = 0$ ;  
2 repeat  
3    $n = \text{Random} \{1, \dots, N\}, n \in \mathcal{G}_m$ ;  
4   foreach  $\mathcal{G}_{m'} \in \mathcal{O}_{cr}, m' \neq m$  do  
5     Calculate  $\overline{\mathcal{J}\mathcal{S}}_{\mathcal{G}_m \rightarrow \mathcal{G}_{m'}}$  according to Eq. (3);  
6    $m' = \min_m \left\{ \overline{\mathcal{J}\mathcal{S}}_{\mathcal{G}_m \rightarrow \mathcal{G}_1}, \dots, \overline{\mathcal{J}\mathcal{S}}_{\mathcal{G}_m \rightarrow \mathcal{G}_M} \right\}$ ;  
7   if  $m' \neq m$  then  
8      $\mathcal{G}_m = \mathcal{G}_m \setminus \{n\}, \mathcal{G}_{m'} = \mathcal{G}_{m'} \cup \{n\}$ ;  
9      $\mathcal{O}_{cr} = (\mathcal{O}_{cr} \setminus \{\mathcal{G}_m, \mathcal{G}_{m'}\}) \cup (\mathcal{G}_m \setminus \{n\}) \cup$   
        $(\mathcal{G}_{m'} \cup \{n\})$ ;  
10   $l = l + 1, \mathcal{O}^* = \mathcal{O}_{cr}$ ;  
11 until coalition partition converges or  $l = L$ ;
```

---

switch operation of clients between coalitions when  $\overline{\mathcal{J}\mathcal{S}}$  increased, otherwise, the client remains in its current coalition. This iterative process continues until a stable partition  $\mathcal{O}^* = \{\mathcal{G}^*\}_1^M$  is formed, where no exchange can further reduce  $\overline{\mathcal{J}\mathcal{S}}$  or the maximum number of iteration rounds is reached. Calculating  $\overline{\mathcal{J}\mathcal{S}}$  requires  $\frac{(M-1)M}{2}$  steps, yielding a total time complexity  $\mathcal{O}(M^2)$ . It's crucial to note that the computation is distributed across all ESs in reality, so each ES only computes  $(M - m)$  times for  $\sum_{j=m+1}^M \mathcal{J}\mathcal{S}(Q_m, Q_j)$ , reducing the time complexity to  $\mathcal{O}(M)$ . This workload is a negligible load for high-performance ESs.

### Near-Optimal Scheduling Rule Design and Optimal Resource Allocation Rule Design

**Coalition Dynamic Estimation** The solution of  $\mathbb{P}$  depends on current delay  $\mathcal{T}_m(t)$ , requiring dynamic prediction. While the law of large numbers states event frequency approximates probability as the number of occurrences increases (Sahu 2024), this fails with limited rounds and scarce data. To address this, we employ Bayesian estimation (Díaz, Batanero, and Contreras 2010) to estimate the latency vector  $\Gamma = [\mathcal{T}_m(t) | m \in \mathbf{m} \odot \Theta(t)]$  as follows:

$$\left\{ \begin{aligned} \mathcal{H}(\mathcal{R}_t | \Gamma) &= \prod_{m \in \mathbf{m} \odot \Theta(t)} \mathcal{P}(r_{t,m} | \Gamma), \end{aligned} \right. \quad (11)$$

$$\left\{ \begin{aligned} \mathcal{B}(\Gamma | \mathcal{R}_t) &= \frac{\mathcal{P}(\Gamma)\mathcal{H}(\mathcal{R}_t | \Gamma)}{\int_{\Gamma} \mathcal{P}(\Gamma)\mathcal{H}(\mathcal{R}_t | \Gamma)d\Gamma}, \end{aligned} \right. \quad (12)$$

where  $\Gamma$  denotes the parameters that need to be estimated,  $\mathcal{P}(\Gamma)$  is a prior distribution, and  $\mathcal{R}_t = \{r_{t,m} | m \in \mathbf{m} \odot \Theta(t)\}$  is the latency vector from  $\mathcal{P}(\Gamma)$ . Based on the observed samples, we construct the joint probability distribution function  $\mathcal{H}(\mathcal{R}_t | \Gamma)$ , calculate the marginal probability  $\int_{\Gamma} \mathcal{P}(\Gamma)\mathcal{H}(\mathcal{R}_t | \Gamma)d\Gamma$ , and use the Bayesian formula to get the posterior distribution for the parameters  $\mathcal{B}(\Gamma | \mathcal{R}_t)$ .

Based on these steps, we obtain the parameter estimation  $[\mathcal{T}_m(t) | m \in \mathbf{m} \odot \Theta(t)] = \mathbb{E}[\mathcal{B}(\Gamma | \mathcal{R}_t)]$ . Since the CS can estimate the successful probability of model upload of each available coalition, we gain a clearer scheduling direction.

**Coalition Scheduling** As long-term SC poses difficulties for conventional methods, we adopt the virtual queue approach from Lyapunov optimization, defined as:

$$\Lambda_m(t) = \begin{cases} -\delta_m, & \text{if } t = -1, \\ \max\{\Lambda_m(t-1) + \delta_m - \chi_m(t), 0\}, & \text{otherwise.} \end{cases} \quad (13)$$

The CS is more intuitively referenced for coalition selection based on the estimated execution time  $\mathbb{E}[\mathcal{B}(\Gamma | \mathcal{R}_t)]$  and the virtual queue  $\Lambda_m(t)$ . Consequently,  $\pi(t)$  in  $\mathbb{P}$  can be converted into a solution that satisfies the following scheduling rule  $\pi(t) \in \Pi, t > 0$ :

$$\pi(t) = \arg \max_{m \in \mathbf{m} \odot \Theta(t)} \left\{ \Lambda_m(t) + \beta \left( 1 - \frac{\hat{\mathcal{T}}_m(t)}{\mathcal{I}} \right) \right\}, \quad (14)$$

where  $\beta > 0$  is a weight parameter.

If the long-term balance of scheduling constrain in Eq. (5) is violated, the virtual queue grows unbounded. To ensure long-term balance, each coalition's queue length must satisfy the following defined mean rate stability (Neely 2010).

**Definition 5.** Discrete-time queue  $\Lambda(t)$  is mean rate stable if  $\lim_{t \rightarrow \infty} \frac{\mathbb{E}[\Lambda(t)]}{t} = 0$ .

**Theorem 2.** FedCure ensures the mean rate stability of the virtual queues, i.e.,

$$\lim_{\tau_g \rightarrow \infty} \frac{\mathbb{E}[\Lambda(\tau_g)]}{\tau_g} = 0, \quad (15)$$

and satisfies the long-term balance constraint.

According to Theorem 2, long-term balance is consistently preserved under the proposed scheduling rule. Notably, this balance holds regardless of the value of the weight parameter  $\beta$  in Eq. (14). Moreover, Eq. (14) reflects the trade-off between efficiency and virtual queue length in each rounds, and a near-optimal upper bound is analyzed in the next section.

**Resource Allocation Optimization** The resource allocation rule  $\mathcal{F}$  optimizes CPU frequency for clients in  $\mathcal{G}_{\pi(t)}$  based on the estimated latency.

**Theorem 3.** There exists an optimal solution  $f_n^* \in \mathcal{F}, \forall n \in \mathcal{G}_{\pi(t)}$  for  $\mathbb{P}$ , i.e.,

$$f_n^* = \min \left\{ f_n^{max}, \varsigma^{+1} \sqrt{\frac{\alpha c_n}{\varsigma \gamma \hat{\mathcal{T}}_{\pi(t)}(t)}} \right\}, n \in \mathcal{G}_{\pi(t)}. \quad (16)$$

Algorithm 2 outlines the scheduling and optimization process. After initialization, coalition scheduling is executed (lines 3-9), including dynamic estimation (line 4) and virtual queue computation (line 8). Based on these steps, the clients in  $\pi(t)$  calculate  $f_n^*$  to decide the computation resource (line 9). The primary complexity of Algorithm 2 stems from the sorting operations during coalition selection, with the time complexity of quicksort being  $\mathcal{O}(M \log M)$ .

---

**Algorithm 2:** Scheduling and Resource Allocation

---

**Input:** Latency vector  $\mathcal{R}_t$ , Available coalitions  $\Theta(t)$ **Output:** Scheduled coalition  $\pi(t)$ , allocated computation power  $\{f_n^*\}$ ,  $n \in \mathcal{G}_{\pi(t)}$ 

- 1 Initialize  $\Lambda_m(-1) = -\delta_m$ ,  $\chi_m(-1) = 0$ ;
  - 2 **foreach**  $t < \tau_g$  **do**
  - 3     **foreach**  $m \in \Theta(t)$  **do**
  - 4         Estimate  $[\hat{\mathcal{T}}_m(t)]$  and update  $\Lambda_m(t)$  with Eq. (12) and Eq. (13), respectively;
  - 5     **if**  $t=0$  **then**
  - 6         Select all the ESs in  $\mathcal{M}$  for initialization;
  - 7     **else**
  - 8          $\pi(t) = \arg \max_{m \in \Theta(t)} \left\{ \Lambda_m(t) + \beta \left( 1 - \frac{\hat{\mathcal{T}}_m(t)}{\mathcal{I}} \right) \right\}$ ;
  - 9     Calculate  $\{f_n^*\}$ ,  $n \in \mathcal{G}_{\pi(t)}$  according to Eq. (16);
- 

## Theoretical Analysis

**Boundary Analysis.** To satisfy participation balance, the scheduling rule must sometimes sacrifice the optimal choice of  $g(t) := \left( 1 - \frac{\hat{\mathcal{T}}_m(t)}{\mathcal{I}} \right)$  to maintain virtual queue stability. We use the Drift-plus-Penalty method to analyze this trade-off and derive bounds on the optimality gap.

**Theorem 4.** *The time-average expected efficiency under the scheduling rule satisfies:*

$$\frac{1}{\tau_g} \sum_{t=0}^{\tau_g-1} \mathbb{E}[g(t)] \geq g^* - \mathcal{O}(1/\beta), \quad (17)$$

where  $g^*$  is the optimal solution over all feasible options.

The result implies that a larger  $\beta$  yields more near-optimal performance, but slower convergence and larger queue.

**Convergence Analysis.** General assumptions are introduced to analysis:

**Assumption 1.** *The assumption parameters for convergence analysis are represented as a 4-tuple  $(\sigma, \mu, \rho, [\vartheta_1, \vartheta_2, \vartheta_3])$ , where the function  $\mathcal{F}_n(\omega)$  is  $\sigma$ -smooth,  $\mu$ -strong convexity,  $\rho$ -lipschitz (Wu et al. 2023b) and gradient difference are bounded by  $[\vartheta_1, \vartheta_2, \vartheta_3]$  (Wang et al. 2023; Wu et al. 2023a).*

- $\sigma$ -smooth,  $\mu$ -strong convexity,  $\rho$ -lipschitz: For  $\forall \omega, \omega'$ ,  $\|\nabla \mathcal{F}_n(\omega) - \nabla \mathcal{F}_n(\omega')\| \leq \sigma \|\omega - \omega'\|$  ( $\sigma > 0$ ),  $\langle \nabla \mathcal{F}_n(\omega'), \omega - \omega' \rangle + \frac{\mu}{2} \|\omega - \omega'\|^2 \leq \mathcal{F}_n(\omega) - \mathcal{F}_n(\omega')$  ( $\mu \geq 0$ ),  $\|\mathcal{F}_n(\omega) - \mathcal{F}_n(\omega')\| \leq \rho \|\omega - \omega'\|$ .
- $\vartheta_1$ -bounded of gradient difference: The gradient difference of loss function between client  $n$  and the associated ES  $m$  is denoted as  $\|\nabla \mathcal{F}_n(\omega) - \nabla \mathcal{F}_m(\omega)\| \leq \vartheta_1$ .
- $\vartheta_2$ -bounded of gradient difference: The gradient difference of loss function between ES  $m$  and CS is denoted as  $\|\nabla \mathcal{F}_m(\omega) - \nabla \mathcal{F}(\omega)\|^2 \leq \vartheta_2$ , where  $\vartheta_2 = c_0 + c_1 \frac{M}{2} \overline{JS}$ .
- $\vartheta_3$ -bounded of stochastic gradient: For  $\forall n$ , the expected squared norm of stochastic gradients is bounded by  $\mathbb{E} \|\nabla F_n(\omega)\|^2 \leq \vartheta_3$ .

The convergence of SAFL is as follows:

**Theorem 5.** *After  $\tau_g$  global aggregations, SAFL converges to a critical point, if the learning rate meets  $\eta < \frac{1}{\sigma}$ ,*

$$\mathbb{E}[\mathcal{F}(\omega(\tau_g) - \mathcal{F}(\omega^*))] \leq \bar{\delta} [\mathcal{F}(\omega(0) - \mathcal{F}(\omega^*))] + (1 - \bar{\delta}) \frac{O_1 \vartheta_3 + O_2 \vartheta_1 + O_3 \vartheta_2}{O_4}, \quad (18)$$

where  $\bar{\delta} = (1 - \xi_\varphi + \xi_\varphi(1 - \eta\mu)^{\tau_c \tau_e})^{\tau_g}$ ,  $O_1 = \frac{1}{2\mu}$ ,  $O_2 = \rho \tau_e \frac{(\eta\sigma+1)^\tau - \sigma\eta\tau_c - 1}{\sigma}$ ,  $O_3 = \frac{\tau_c \tau_e \eta}{2}$ , and  $O_4 = 1 - (1 - \eta\mu)^{\tau_c \tau_e}$ .

**Remark:** As  $\tau_g \rightarrow \infty$ ,  $\bar{\delta} \rightarrow 0$ , for strongly convex functions, the convergence bound simplifies to  $\frac{O_1 \vartheta_3 + O_2 \vartheta_1 + O_3 \vartheta_2}{O_4}$ , which is mainly influenced by  $\vartheta_1$ ,  $\vartheta_2$  and  $\vartheta_3$ . The values of  $\vartheta_1$  and  $\vartheta_2$  depend on the edge association, highlighting the importance of adjusting the edge relationships to achieve a low  $\overline{JS}$  in alignment with the intent of FedCure.

## Experiments

### Experimental Configurations

**Datasets and Models.** We conduct experiments on four real-world datasets commonly used in FL: a CNN with 2 convolutional layers, 2 pooling layers and a fully connected layer on MNIST (Lecun et al. 1998); a CNN with 2 convolutional layers, one pooling layer and 3 fully connected layers on CIFAR-10 (Krizhevsky and Hinton 2009), SVHN (Netzer et al. 2011) and CINIC-10 (Darlow et al. 2018).

**Parameter Configurations.** The experiment involves 5 ESs and 50 clients. The initial clients' data distributions are configured as edge non-IID following (Liu et al. 2020). The training process includes 5 rounds of local training, 12 rounds of edge iterations, and either 100 or 200 rounds of global iterations. The learning rate is set to 0.01 for MNIST, CIFAR-10 and CINIC-10, and 0.005 for SVHN. The other parameters are:  $\ell = 0.2$ ,  $\mathbb{k} = [0.9, 0.99]$ , and  $\beta = 0.5$ .

**Baselines.** Two clustering and two scheduling methods: **K-Means** (Lim et al. 2022) is a clustering algorithm that segregates data into K clusters (K is pre-specified) and assigns data to the nearest cluster based on distance. **Mean-Shift** (Lu et al. 2023) is a density-based non-parametric clustering algorithm that automatically determines the number of clusters. **Greedy/FedG** (Albaseer et al. 2021) schedule coalitions by selecting options that maximize training efficiency. **Fair/FedFair** (Zhu et al. 2023) leverages virtual queue to ensure balanced participation.

### Experimental Results

**Validating the effectiveness of FedCure on non-IID data.** Figures 2(a)-2(c) show the data distribution during the coalition formation process, with the width of the violin plots indicating the density of different data labels. The initial  $\overline{JS}$  value is 0.69 and each coalition contains two label categories. Under the preference rule, the data distribution gradually becomes homogeneous, and the similarity of each coalition reaches its maximum in the final state where the  $\overline{JS}$  value is 0. Figure 2(d) shows a consistent decline in  $\overline{JS}$

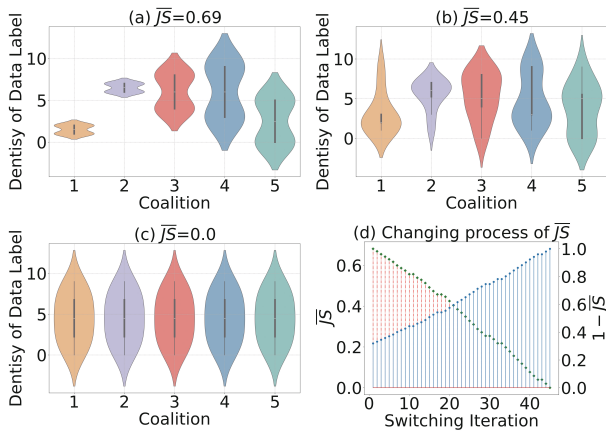


Figure 2: Data distribution variation and  $\overline{JS}$ 's changing.

Meth.	Datasets			
	MNIST	CIFAR-10	SVHN	CINIC-10
Greedy	37.97±0.03	18.25±0.06	9.36 ± 1.08	14.56±1.84
Fair	76.19±0.90	18.65±0.85	29.20±7.93	15.99±2.45
FedG	96.09±0.12	50.39±1.33	56.63±2.09	33.75±1.73
FedFair	96.37±0.05	53.15±0.62	58.46±1.13	34.83±1.72
FedCure	96.37±0.05	52.90±0.92	57.51±1.30	34.94±1.66
Cohen's <i>d</i>	0.00	0.32	0.78	0.07

Table 1: Accuracy comparison with baselines.

value during client switching operations. Figure 3 shows the accuracy achieved using the two clustering algorithms (K-means and Mean-shift) and different data distributions. The maximum performance achieved based on the final data distributions is 1.40x, 3.09x, 2.95x, and 3.15x that of the initial state on four real-world datasets, respectively. The Mean-Shift algorithm includes five clusters, and we apply the same setting to K-means with  $K = 5$ . However, the repetition of labels in clusters limits the diversity of data distributions, and predefining  $K$  will further hinder the applicability. Figures 2 and 3 validate that FedCure can achieve highly effective data distribution adjustments in terms of both accuracy and model convergence speed. This is evidenced by each shift towards lower  $\overline{JS}$  values during coalition formation.

**Validating the efficiency of scheduling and resource allocation.** Table 1 summarizes the accuracy achieved by various scheduling methods. Greedy and Fair perform worse, mainly due to unadjusted data distribution. Greedy prioritizes coalitions with shorter training latency, exacerbating the issue of local optima. FedG and FedFair, which rely on the stable coalitions provided by FedCure, achieve relatively higher accuracy. FedCure delivers accuracy comparable to FedFair, with the range of Cohen's  $d$  (0.00–0.78) indicating no substantial performance gaps, confirming that FedCure successfully sustains competitive accuracy through balance-based scheduling. Figure 4 presents the coefficient of variation (COV) of training latency per round, serving as

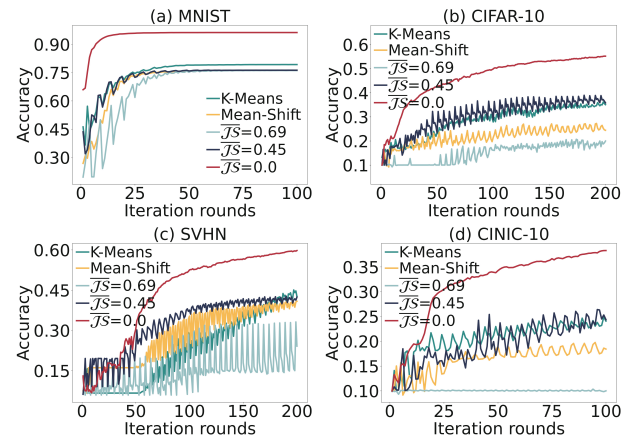


Figure 3: Comparison across clustering methods.

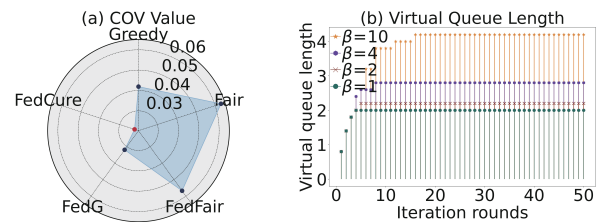


Figure 4: COV of training latency and queue length.

a relative metric for comparing different scheduling methods. In Figure 4(a), FedCure and FedG exhibit lower COVs, indicating comparable scheduling performance when prioritizing faster coalitions. Conversely, FedFair and Fair have higher COVs due to equal treatment of the fastest and slowest coalitions, resulting in greater relative variability. Combining Table 1 and Figure 4(a), FedCure effectively balances performance and efficiency. Figure 4(b) shows the impact of  $\beta$  on virtual queue length. All curves stabilize after several rounds, verifying that FedCure maintains queue stability. Larger  $\beta$  slows convergence and lengthens virtual queue, as they favor faster coalitions. While higher  $\beta$  may compromise early balance, they still ensure long-term balance.

## Conclusion

In this paper, a novel framework, FedCure, was proposed to address the negative impacts of participation bias with non-IID data. Firstly, the data distribution in client-edge layer was transformed into a correlation problem, and a coalition formation game was established with a preference rule, reducing the non-IID degree among ESs. Secondly, the coalition scheduling in edge-cloud layer was guided by a scheduling rule that leverages Bayesian-estimated coalition dynamics and virtual queue, enhancing balance and efficiency. Based on this, a resource allocation rule was introduced to align with the estimated coalition dynamics by optimizing computation settings. Finally, extensive experiments on four real-world datasets validated FedCure's effectiveness.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62325304, 62372343, 62402352, 62562050, U22B2046, 62072411), in part by the Natural Science Foundation of Jiangsu Province of China (No. BK20253020), and in part by the Key Research and Development Program of Zhejiang Province (No. 2025C01055), and in part by the Open Fund of Key Laboratory of Social Computing and Cognitive Intelligence (Dalian University of Technology), Ministry of Education (No. SCCI2024TB02).

## References

- Albaseer, A.; Abdallah, M.; Al-Fuqaha, A.; and Erbad, A. 2021. Client Selection Approach in Support of Clustered Federated Learning over Wireless Edge Networks. In *2021 IEEE Global Communications Conference*, 1–6. Madrid, Spain: IEEE Press.
- Arisdakessian, S.; Wahab, O. A.; Mourad, A.; and Otrok, H. 2023. Coalitional Federated Learning: Improving Communication and Training on Non-IID Data With Selfish Clients. *IEEE Transactions on Services Computing*, 16(4): 2462–2476.
- Artificial Intelligence Action Summit. 2025. AI Action Summit. <https://www.elysee.fr/en/sommet-pour-l-action-sur-l-ia>. Accessed: 2025-02.
- Cho, Y. J.; Wang, J.; and Joshi, G. 2020. Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies. arXiv:2010.01243.
- Darlow, L. N.; Crowley, E. J.; Antoniou, A.; and Storkey, A. J. 2018. CINIC-10 is not ImageNet or CIFAR-10. arXiv:1810.03505.
- Deng, Y.; Li, X.; Sun, C.; Fan, Q.; Wang, X.; and Leung, V. C. M. 2024. Semi-Asynchronous Federated Learning with Trajectory Prediction for Vehicular Edge Computing. In *2024 IEEE/ACM 32nd International Symposium on Quality of Service*, 1–10. Guangzhou, China: IEEE Press.
- Devarajan, G. G.; Thangam, S.; Alenazi, M. J.; Kumaran, U.; Chandran, G.; and Bashir, A. K. 2025. Federated Learning and Blockchain-Enabled Framework for Traffic Rerouting and Task Offloading in the Internet of Vehicles (IoV). *IEEE Transactions on Consumer Electronics*, 1–1.
- Díaz, C.; Batanero, C.; and Contreras, J. 2010. Teaching independence and conditional probability. *Boletín de Estadística e Investigación Operativa*, 26(2): 149–162.
- He, X.; Huang, H.; Wang, C.; Hu, F.; Cai, T.; and Zheng, Z. 2025. A Fairness-guaranteed Framework for Semi-asynchronous Federated Learning. *IEEE Transactions on Network Science and Engineering*, 1–18.
- Huang, T.; Lin, W.; Shen, L.; Li, K.; and Zomaya, A. Y. 2022. Stochastic Client Selection for Federated Learning With Volatile Clients. *IEEE Internet of Things Journal*, 9(20): 20055–20070.
- Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4).
- Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.
- Leng, J.; Li, R.; Xie, J.; Zhou, X.; Li, X.; Liu, Q.; Chen, X.; Shen, W.; and Wang, L. 2025. Federated learning-empowered smart manufacturing and product lifecycle management: A review. *Advanced Engineering Informatics*, 65: 103179.
- Li, Z.; Sun, Y.; Shao, J.; Mao, Y.; Wang, J. H.; and Zhang, J. 2024. Feature Matching Data Synthesis for Non-IID Federated Learning. *IEEE Transactions on Mobile Computing*, 23(10): 9352–9367.
- Lim, W. Y. B.; Ng, J. S.; Xiong, Z.; Jin, J.; Zhang, Y.; Niyato, D.; Leung, C.; and Miao, C. 2022. Decentralized Edge Intelligence: A Dynamic Resource Allocation Framework for Hierarchical Federated Learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(3): 536–550.
- Liu, J.; Jia, J.; Che, T.; Huo, C.; Ren, J.; Zhou, Y.; Dai, H.; and Dou, D. 2024. Fedasmu: Efficient asynchronous federated learning with dynamic staleness-aware model update. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13900–13908. Vancouver, Canada: AAAI.
- Liu, L.; Zhang, J.; Song, S.; and Letaief, K. B. 2020. Client-Edge-Cloud Hierarchical Federated Learning. In *ICC 2020 - 2020 IEEE International Conference on Communications*, 1–6. Dublin, Ireland: IEEE Press.
- Lu, J.; Sheng, Y.; Cao, S.; Elnaffar, S.; Muhammad Saad, M.; Mohammed Seid, A.; and Erbad, A. 2024. Lyapunov-Guided Long-Term Fairness-Aware Federated Learning for Collaborative TinyML on Edge Devices. *IEEE Transactions on Consumer Electronics*, 70(4): 7334–7345.
- Lu, R.; Zhang, W.; Wang, Y.; Li, Q.; Zhong, X.; Yang, H.; and Wang, D. 2023. Auction-Based Cluster Federated Learning in Mobile Edge Computing Systems. *IEEE Transactions on Parallel and Distributed Systems*, 34(4): 1145–1158.
- Menéndez, M.; Pardo, J.; Pardo, L.; and Pardo, M. 1997. The Jensen-Shannon divergence. *Journal of the Franklin Institute*, 334(2): 307–318.
- Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic Federated Learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 4615–4625. California, USA: ACM.
- Mu, X.; Shen, Y.; Cheng, K.; Geng, X.; Fu, J.; Zhang, T.; and Zhang, Z. 2023. FedProc: Prototypical contrastive federated learning on non-IID data. *Future Generation Computer Systems*, 143: 93–104.
- Neely, M. 2010. *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool Publishers.
- Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 7. Granada, Spain: MIT Press.

- Ng, J. S.; Lim, W. Y. B.; Xiong, Z.; Cao, X.; Jin, J.; Niyato, D.; Leung, C.; and Miao, C. 2021. Reputation-aware hedonic coalition formation for efficient serverless hierarchical federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(11): 2675–2686.
- Nguyen, D. C.; Pham, Q.-V.; Pathirana, P. N.; Ding, M.; Seneviratne, A.; Lin, Z.; Dobre, O.; and Hwang, W.-J. 2022. Federated learning for smart healthcare: A survey. *ACM Computing Surveys*, 55(3): 1–37.
- OECD. 2024. Principles for trustworthy AI. <https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449>. Accessed: 2024-05.
- Paragliola, G. 2022. Evaluation of the trade-off between performance and communication costs in federated learning scenario. *Future Generation Computer Systems*, 136: 282–293.
- Sahu, S. K. 2024. *Conditional Probability and Independence*, 63–80. Cham: Springer International Publishing.
- Sun, J.; Li, A.; Duan, L.; Alam, S.; Deng, X.; Guo, X.; Wang, H.; Gorlatova, M.; Zhang, M.; Li, H.; et al. 2022. Fedsea: A semi-asynchronous federated learning framework for extremely heterogeneous devices. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 106–119. Boston Massachusetts, USA: Association for Computing Machinery.
- Vailshery, L. S. 2024. Internet of Things - statistics & facts. Statistica.
- van Erven, T.; and Harremos, P. 2014. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory*, 60(7): 3797–3820.
- Wang, Z.; Xu, H.; Liu, J.; Xu, Y.; Huang, H.; and Zhao, Y. 2023. Accelerating Federated Learning With Cluster Construction and Hierarchical Aggregation. *IEEE Transactions on Mobile Computing*, 22(7): 3805–3822.
- Wu, Q.; Chen, X.; Ouyang, T.; Zhou, Z.; Zhang, X.; Yang, S.; and Zhang, J. 2023a. HiFlash: Communication-Efficient Hierarchical Federated Learning With Adaptive Staleness Control and Heterogeneity-Aware Client-Edge Association. *IEEE Transactions on Parallel and Distributed Systems*, 34(5): 1560–1579.
- Wu, X.; Huang, F.; Hu, Z.; and Huang, H. 2023b. Faster adaptive federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 10379–10387. DC, USA: AAAI Press.
- Wu, X.; Sun, J.; Hu, Z.; Zhang, A.; and Huang, H. 2023c. Solving a Class of Non-Convex Minimax Optimization in Federated Learning. In *Advances in Neural Information Processing Systems*, volume 36, 11232–11245. LA, USA: MIT Press.
- Yang, Z.; Chen, M.; Saad, W.; Hong, C. S.; and Shikh-Bahaei, M. 2021. Energy Efficient Federated Learning Over Wireless Communication Networks. *IEEE Transactions on Wireless Communications*, 20(3): 1935–1949.
- Zhang, N.; Ma, Q.; Mao, W.; and Chen, X. 2024. Coalitional FL: Coalition Formation and Selection in Federated Learning With Heterogeneous Data. *IEEE Transactions on Mobile Computing*, 23(11): 10494–10508.
- Zhang, Y.; Liu, D.; Duan, M.; Li, L.; Chen, X.; Ren, A.; Tan, Y.; and Wang, C. 2023. FedMDS: An Efficient Model Discrepancy-Aware Semi-Asynchronous Clustered Federated Learning Framework. *IEEE Transactions on Parallel and Distributed Systems*, 34(3): 1007–1019.
- Zhang, Y.; Xu, Y.; Wu, Q.; Luo, Y.; Xu, Y.; Chen, X.; Anpalagan, A.; and Zhang, D. 2018. Context Awareness Group Buying in D2D Networks: A Coalition Formation Game-Theoretic Approach. *IEEE Transactions on Vehicular Technology*, 67(12): 12259–12272.
- Zhou, H.; Zheng, Y.; Huang, H.; Shu, J.; and Jia, X. 2023. Toward Robust Hierarchical Federated Learning in Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(5): 5600–5614.
- Zhu, H.; Zhou, Y.; Qian, H.; Shi, Y.; Chen, X.; and Yang, Y. 2023. Online Client Selection for Asynchronous Federated Learning With Fairness Consideration. *IEEE Transactions on Wireless Communications*, 22(4): 2493–2506.