Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

FedEem: a fairness-based asynchronous federated learning mechanism

Wei Gu[1] and Yifan Zhang[2*]

Abstract

Federated learning is a mechanism for model training in distributed systems, aiming to protect data privacy while achieving collective intelligence. In traditional synchronous federated learning, all participants must update the model synchronously, which may result in a decrease in the overall model update frequency due to lagging participants. In order to solve this problem, asynchronous federated learning introduces an asynchronous aggregation mechanism, allowing participants to update models at their own time and rate, and then aggregate each updated edge model on the cloud, thus speeding up the training process. However, under the asynchronous aggregation mechanism, federated learning faces new challenges such as convergence difficulties and unfair model accuracy. This paper first proposes a fairness-based asynchronous federated learning mechanism, which reduces the adverse effects of device and data heterogeneity on the convergence process by using outdatedness and interference-aware weight aggregation, and promotes model personalization and fairness through an early exit mechanism. Mathematical analysis derives the upper bound of convergence speed and the necessary conditions for hyperparameters. Experimental results demonstrate the advantages of the proposed method compared to baseline algorithms, indicating the effectiveness of the proposed method in promoting convergence speed and fairness in federated learning.

Keywords Federated learning, AISecurity, Edge computing

Introduction

In the context of edge computing, edge devices use local data to train local models and upload them to the cloud to aggregate and update the global model. A lot of practice has found that data is not independent and identically distributed [1–4]. Federated learning and traditional distributed machine learning share a common research objective: minimizing training time, as measured by the clock time needed to achieve the desired accuracy. It is important to mention that in most existing literature on federated learning, including the pioneering work on the FedAveraging algorithm, the assumption is made that communication between clients and the server is fully synchronous. This means that the server waits for all selected clients to finish their local training and report their trained models before aggregation takes place [5]. This straightforward and efficient design has been widely adopted by many existing studies and bears resemblance to the batch synchronous parallel mechanism used in distributed machine learning within a single cluster. However, it should be noted that in the case of heterogeneous clients, where different edge devices act as clients with varying computing abilities, there can be significant differences in their local training performance. In fact, the training time for the same amount of computation may exhibit a heavy-tailed distribution. If some clients perform their local training at a much slower pace than others, the performance of this synchronous communication mechanism may be compromised, as the server has to

*Correspondence:
Yifan Zhang
yifan_zhang_2001@163.com
[1] School of Computer Science, Nanjing University of Information Science and Technology, 210044 Nanjing, China
[2] School of Software, Nanjing University of Information Science and Technology, 210044 Nanjing, China

wait for these stragglers, significantly reducing system parallelism.

In this scenario, introducing an asynchronous communication mechanism is an effective solution [6, 7]. In asynchronous federated learning, the server is not required to wait for all selected clients to report their model updates; instead, it continues the aggregation process immediately when a client's model update arrives. The asynchronous mechanism has advantages over synchronous FL. In synchronous federated learning, the number of active clients fluctuates throughout each round as clients join and leave the queue, with a decrease towards the end due to stragglers. In contrast, asynchronous federated learning maintains a relatively stable number of active clients over time. As clients complete their training and upload their model updates, their positions are replaced by newly selected clients, increasing the parallelism of the asynchronous system.

Although the asynchronous mechanism enhances system parallelism, when the client computing speeds follow a heavy-tailed distribution, potential issues can arise. In a problematic scenario, fast clients can quickly update the global model by completing their local training, while slower clients make minimal progress based on outdated global models. In traditional distributed machine learning with parameter servers, the pathological scenario is prevented by introducing bounded staleness in the outdated synchronous parallel (SSP) mechanism [8]. In FedAsync, a staleness function is proposed to compute a mixed hyperparameter α for model aggregation. Intuitively, the weight assigned to a client's model updates when aggregating into the global model decreases as the client becomes more "stale". Through this simple design, FedAsync demonstrates its ability to address the issue of localized regularization to ensure convergence, and a similar approach is introduced in the asynchronous aggregation mechanism proposed in this paper.

Another potential issue arising from this process is fairness [9]. It is evident that fast clients are selected for local training far more frequently throughout the entire federated learning process, as they can quickly complete their local model training and enter the waiting-to-be-selected state. However, this process is undoubtedly unfair for fast clients. Specifically, fast clients may expend much more computational power throughout the process than slow clients, only to end up with the same model as slow clients. What's worse is that the models obtained by fast clients, who contribute more, may have lower accuracy on their local test sets compared to the models contributed by less contributing clients. The fairness issue arising from this situation deserves further investigation, as unfair mechanisms designed in federated systems may discourage clients from joining the federation for distributed training, especially for fast clients. This, in turn, may result in a reduction of fast clients in the entire system, thereby decreasing the overall performance of the federation.

In summary, the heterogeneity of clients and the introduction of asynchronous mechanisms make the entire system more complex and complicate the trade-off between fairness and model performance. To address this, this paper proposes an adaptive asynchronous federated learning aggregation mechanism, referred to as FedEem , with the following two main improvements.

- Propose an aggregation algorithm to judge the obsolescence degree and gradient drift degree of client models, effectively reducing the impact of system failures, and allowing clients to perform local updates in different rounds, instead of using globally synchronized rounds of local updates commonly used.
- Propose an early exit mechanism to reduce the fairness issue caused by the over-selection of fast clients while ensuring that the convergence speed of the system does not significantly decrease.

This paper first introduces and analyzes the necessity and effectiveness of these two mechanisms in detail. Then, through experiments, the superiority of the proposed method compared to Fedbuff, FedAsync, and FedAvg methods is demonstrated. Finally, detailed mathematical analysis of the convergence is provided.

Related work

The related work can be summarized into the following three points: Asynchronous Federated Learning, Personalized Federated Learning and Fairness Issues in Federated Learning.

Asynchronous federated learning

In the classical federated learning paradigm, synchronous aggregation strategies face challenges in effectively utilizing limited resources, particularly on heterogeneous devices. This is because they have to wait for slower devices to complete their computations before aggregating in each training round. Additionally, the heterogeneity of data distribution, known as data heterogeneity, in real-world mobile edge computing scenarios can significantly impact the accuracy of the model. Hence, some research works have attempted to use asynchronous model updates to improve efficiency, performance, privacy, and security. Xie et al. proposed the paradigm of asynchronous federated learning called FedAsync, which solves the regularized local problem to ensure convergence, and then updates the global model using stale-weighted averaging, demonstrating the proposed

method's near-linear convergence for both strongly convex and constrained non-convex problem families. Chen et al. introduced Asynchronous Online Federated Learning (ASO Fed) as an extension of FedAsync. They proposed online optimization policies to tackle three potential training challenges: 1) the data on local devices can increase over time, leading to changing correlations among clients in an online setting; 2) due to network constraints, mobile devices may frequently go offline or have poor communication bandwidth, making synchronous federated learning frameworks highly sluggish; 3) In the context of federated learning, edge devices may experience delays or even drop out of the training process due to various factors such as data heterogeneity or system heterogeneity. These factors can introduce inconsistencies and hinder the smooth progress of the federated learning process [10]. Nguyen introduced a model aggregation scheme called FedBuff, which aims to leverage the benefits of both synchronous federated learning (FL) and asynchronous FL. In FedBuff, the server aggregates client updates in a dedicated buffer, allowing for more flexible and efficient aggregation [11]. This approach demonstrates improved convergence speed compared to FedAsync and is compatible with existing secure aggregation and privacy techniques. It offers a promising solution for achieving efficient and secure federated learning. Su et al. enhanced FedBuff by dynamically adjusting aggregation weights considering the staleness and divergence of model updates. They carefully selected operating points in each dimension of the design space and ensured verified convergence guarantees [12].

Personalized federated learning

Data heterogeneity poses a significant challenge in current federated learning approaches.Research findings suggest that the accuracy of FedAvg experiences a significant decrease when trained on non-identically and independently distributed data [13]. Additionally, the updates of completely synchronized models result in a lack of personalized solutions. Users from diverse scenarios may exhibit varying usage patterns due to subtle distinctions in their environments and requirements [14]. In such scenarios, the need for more personalized predictions arises to provide users with more meaningful word suggestions. This challenge not only affects the training of the global model but also impacts its performance on local data of specific clients. Consequently, this may discourage the participation of affected clients in the federated learning process.

Personalized federated learning offers a promising solution to tackle the issue at hand. By training customized local models for each user, it effectively addresses the data heterogeneity among clients [15]. Presently, personalized federated learning methods primarily concentrate on optimizing from both data-based and model-based standpoints. Data-based approaches strive to minimize the statistical heterogeneity of client data distribution through techniques such as data augmentation [14] and node selection [16]. Model-based approaches, on the other hand, focus on learning a robust global model that can be further personalized for each client or enhance the adaptability of local models. Common practices include adding regularization terms [17], meta-learning [18], and transfer learning [19]. Some research attempts to enhance the robustness and generalization of federated learning through methods like clustering [20], multitask learning [21], model interpolation, and knowledge distillation [22]. In this paper, we utilize meta-learning for client initialization during training.

Fairness issues in federated learning

In the federated learning system, when clients participate in federated learning, they inevitably consume resources on their devices, including computational resources, communication resources, and power resources. Without sufficient rewards, clients may be unwilling to participate or share their trained models. Hence, creating a fair, rewarding, and secure environment for federated learning becomes imperative to encourage a substantial client participation.

Zhou et al. classify fairness in federated learning into three categories: performance fairness, collaboration fairness, and model fairness [23]. For performance fairness, most schemes aim to promote a consistent accuracy distribution among participants and achieve reasonable resource allocation in heterogeneous systems through joint optimization. In terms of collaboration fairness, current research primarily focuses on ensuring that each participant receives a fair representation of the rewards they contribute to the federated system [24], thus establishing a sound incentive mechanism. Incentive mechanisms in federated learning mainly attempt to construct a contribution model for each participant and provide corresponding rewards. Currently, contribution models are mainly based on the value of client data, which is evaluated from the perspectives of data quality and data quantity. Evaluation methods based on data quality employ metrics such as Shapley value [25], auction mechanisms [26], contractual theory, etc. Evaluation methods based on data quantity adjust the size of participating data to fully consider the rewards and energy costs obtained by each client. Furthermore, Zhan et al. introduce a novel approach that integrates game theory and deep reinforcement learning. In this approach, the parameter server functions as a deep reinforcement learning agent, enabling it to determine optimal payments without the

requirement of accurately assessing each client's contributions or obtaining their private information beforehand [27].

Regarding model fairness, Du et al. proposed reweighting the objective function under fairness constraints [28]. Liang et al. attempts to reduce the impact of variance in data distribution by locally learning representations on each client while jointly learning the global model [29]. It is important to acknowledge the trade-off between performance fairness and model fairness. Performance fairness prioritizes achieving a balanced accuracy of the global model, whereas model fairness focuses on the performance of the model on local data. Collaboration fairness relies on an executable and sound incentive mechanism. In the context of mobile edge computing, most traditional federated incentive mechanisms are ineffective because most clients (such as mobile phone users and IoT devices [30–34]) do not expect to gain economic benefits through federated learning. Their primary concern lies in determining whether federated learning can enhance the accuracy of the model on their respective local data, which is known as model fairness.

In summary, the existing work has the following shortcomings: 1) The heterogeneity of the clients and the introduction of asynchronous mechanisms make the entire system more complex. This may lead to imbalanced resource utilization and slower model convergence speed. 2) Balancing fairness and model performance in federated learning is often challenging. This paper aims to improve the fairness of the model while ensuring that the convergence speed of the model does not significantly decrease.

FedEem

In order to address the issue of slow model updates, an increasing number of federated learning approaches have adopted asynchronous aggregation patterns in recent years. FedEem also utilizes this asynchronous aggregation mechanism, which allows clients to upload models at different time points and update the global model by merging these models. However, FedEem has made certain innovations in the aggregation mechanism by introducing obsolescence discount, diversity discount, and early stopping mechanism.

Asynchronous aggregation mechanism

In an asynchronous federated learning system, clients that receive the global model from the server several rounds ago may become outdated, resulting in lower quality model updates during the aggregation process. This can disrupt the approximate consensus of the majority of other clients and impede the convergence process. It is intuitive to reduce the weight assigned to these outdated clients during the aggregation process [6]. To measure these effects, the following evaluation metrics are proposed in this section.

Obsolescence Discount refers to the concept of quantifying the obsolescence of a client in an asynchronous federated learning system. The obsolescence of a client is determined by the number of global update rounds that have passed since the client last received the global model from the server. It is reasonable to assume that the more obsolete a client is, the lower its aggregation weight should be. According to [11], the obsolescence of clients must be bounded, otherwise the convergence of the model cannot be guaranteed. Let τ represent the current training round on the server, and let τ_k represent the training round corresponding to the last time client k received the global model from the server. The obsolescence s_k of client k can be calculated as $\tau - \tau_k$. The following obsolescence function is used to calculate the obsolescence discount, which discounts the aggregation weight:

$$s_\tau^k = \alpha \cdot \frac{\Omega}{S^k + \Omega}, \tag{1}$$

Where Ω is used to represent the upper bound of obsolescence. Clearly, the upper bound of s_τ^k is 0.5α, where α is a hyperparameter that controls the importance of obsolescence discount in the aggregation process.

To measure the staleness of client updates, the difference between local accumulated gradients and global aggregated gradients can be utilized. Let $w_i - w_{i-1}$ represent the disparity between the models obtained from the most recent two rounds of server aggregation. Here, w_i denotes the parameters of the global model in i-th round. In round i, client k uploads its weight updates obtained from training, denoted as δi_k. If δi_k significantly disrupts the general consensus $w_i - w_{i-1}$, it implies that the update from client k may not contribute to global optimization and should be discounted during the aggregation process. The interference can be quantitatively assessed by calculating the cosine similarity θ_i^k between δi_k and $w_i - w_{i-1}$. A lower θ_i^k indicates less similarity between the two vectors. Consequently, the dissimilarity discount can be defined as follows:

$$\theta_i^k = \beta \cdot Similarity\left(\Delta_i^k, w_i - w_{i-1}\right) + 1, \tag{2}$$

Where $Similarity(X, Y)$ represents the cosine similarity between vectors X and Y. Similar to the obsolescence discount, a hyperparameter β is introduced here to control the magnitude of the dissimilarity discount. Taking these two influencing factors into account, the aggregation weight can be defined as follows:

$$p_i^k = \frac{|T_k|}{|T|}\left(s_i^k + \theta_i^k\right), \tag{3}$$

Where T represents the dataset of client k.

Early exit mechanism

Another mechanism introduced in this chapter is the early exit mechanism. In previous works such as FedAvg and FedAsync, clients were not allowed to stop early during the training process and were required to complete all training rounds. This requirement is reasonable in synchronous federated learning, where clients are sampled with equal probability, resulting in consistent expected rounds $\mathcal{E}(n_k) = ST/K$ for all clients sampled in total training rounds T. However, the introduction of asynchronous mechanism breaks this balanced expectation. Clearly, in an asynchronous federated learning system, if clients are still sampled with equal probability per round, the expected rounds $mathcalE(n_k)$ for fast clients and slow clients required to perform local training are not consistent, and this difference increases with the increase in computational speed differences among clients. This unfairness is problematic for fast clients because it discourages their participation in federated learning. While the concept of federated learning promotes collaboration among clients in a distributed manner, the unequal treatment of fast clients would deter users with high-performance computing devices from engaging in federated learning.

Furthermore, it is important to acknowledge that the non-independent and non-identically distributed data introduces variations in objectives among different clients. In traditional federated learning, the objective is to achieve optimal performance of the global model across all clients. However, from an individual client's perspective, the objective is to attain excellent local performance. Please note the difference between these two objectives, as the latter allows for inconsistent models across different local clients.

If a client has achieved sufficient performance in its local training after participating in several rounds of federated updates, it naturally tries to exit the federated system. However, it is important to note that allowing quick client exits may potentially cause issues. After a client is selected, it undergoes local training and uploads its model for aggregation. In return, the client receives the global model. While this global model may exhibit good performance on the client's local dataset, it may suffer from poor generalization performance. This is because the global model tends to optimize towards the client's objectives based on the previous round of updates. Therefore, it may result in the global model not having good generalization performance yet.Another more critical reason is that when the aggregation of the global model reaches a high position (e.g., 92%), there is a high probability that the client immediately achieves its local training goal (e.g., 95%). If the clients are allowed to exit at this time, a large number of clients may withdraw within a few rounds. The problem with this is that only a few clients are left and continuously selected, leading to a severe deviation between their optimization direction and the global optimum. Consequently, it may result in a global model with poor generalization performance.

As shown in Fig. 1, clients *a*,*b*,*c* are participating in asynchronous federated learning. In rounds t and $t+1$, client b is performing gradient descent with clients a and b. At this point, client b is overly involved, and the global parameter w^{t+2} is already sufficiently close to the

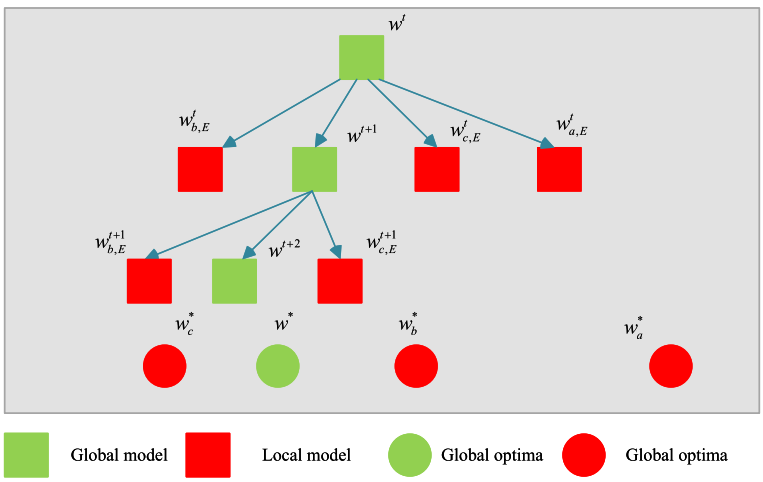

Fig. 1 Simulation results for the network

global optimum w^* and its local target w_b^*, while the local optima of clients a and c are relatively far away.If we insist on client b's participation in the federated learning process at this stage, it may lead to a significant deviation in the global gradient update direction compared to the dominant client's gradient direction, and affect the convergence process. Additionally, some clients may be unwilling to continue contributing computational resources due to being selected too many times.

This article explores a reliable early exit mechanism: (1) setting a lower bound on the number of training rounds for all clients, t_{bnd}, which requires clients to be selected at least t_{bnd} times before being allowed to exit early; (2) setting an additional number of training rounds for all clients, t_{ext}, which requires clients to be selected at least t_{ext} times after the model reaches an accuracy target before being allowed to exit early; (3) setting a lower bound on the number of remaining rounds for all clients, t_{stay}, which requires clients to remain in the client pool for at least t_{stay} rounds after achieving the target accuracy on their own dataset before being allowed to exit early.

Convergence analysis

In order to analyze the convergence performance of FedEem , in combination with previous convergence proof methods for federated learning, the following settings are considered.In each round of global update $\tau \in T$,where T represents the total number of rounds of global updates, the server selects k clients from the client pool.Each client first receives the global model $w_{\tau^k}^k$ from the server, and then performs ϵ rounds of training based on its own data. For the j-th round of local training, with a data size of B and a learning rate of η_l^j, the local model is optimized using SGD. This can be formulated as $\boldsymbol{w}_{\tau^k,j+1}^k = \boldsymbol{w}_{\tau^k,j}^k - \eta_l^j g\left(\boldsymbol{w}_{\tau^k,j}^k\right)$, where the gradient $g\left(\boldsymbol{w}_{\tau^k,j}^k\right) = \nabla f_k\left(\boldsymbol{w}_{\tau^k,j}^k, D^k\right)$. Once all selected clients have reported, the server starts the aggregation process. To provide a better explanation, we have made use of some common assumptions and listed all the parameters used in Table 1 [11].

Assumption 1 The objective function f_k of each client k is $L-$smooth, which means its derivative is $L-$Lipschitz continuous, resulting in $\left\|\nabla f_k(\boldsymbol{w}) - \nabla f_k(\boldsymbol{w})\right\| \leq L\left\|w - w'\right\|$.

Assumption 2 $E_\xi[f_k(\boldsymbol{w}, \xi)] = \nabla f_k(\boldsymbol{w})$, where w represents the trainable parameters.

Assumption 3 The expected square norm of the stochastic gradient is uniformly bounded, i.e.,$\mathbb{E}\left\|\nabla f_k(w, \xi)\right\|^2 \leq G^2$ for $k = 1, \ldots, K$.

Assumption 4 Assuming ξ is uniformly sampled from the local data of the k-th client device. The variance of the stochastic gradient in each device is bounded, i.e., for $\mathbb{E}_\xi \left\|f_k(\boldsymbol{w}, \xi) - f_k(\boldsymbol{w})\right\|^2 \leq \sigma_k^2$ for $k = 1, \ldots, K$. Then, we define $\sigma_l^2 := \sum_{k=1}^K \frac{\left|D^k\right|}{|D|}\sigma_k^2$.

Assumption 5 For any client k and parameter w, we define δ_k as the upper bound of the local objective function with the global objective function, which is $\left\|f_k(\boldsymbol{w}) - f(\boldsymbol{w})\right\|^2 \leq \delta_k^2$. Furthermore, we define $\delta_g^2 := \sum_{k=1}^K \frac{\left|D^k\right|}{|D|}\delta_k^2$.

Based on adaptive weight gradient aggregation, Lemma 1 can be obtained.

Lemma 1 *Given hyperparameters α and β for outdatedness and interference discount, the aggregation weight p_τ^k for each gradient has an upper bound $p_\tau^k \in \left[\frac{\alpha}{2}d_k, (\alpha+\beta)d_k\right]$, where $d_k = \frac{\left|D^k\right|}{|D|}$.*

Table 1 Experimental parameters

Symbol	Description
T, t	Server update frequency, server update index
$\mathcal{S}^t$	Server updates the selected subset of client in time t
Q, q, ϵ	Local step count per round, round index
w^t	The model after t updates
$g_i(w; \zeta_i) := g_i(w)$	Random gradient
$\eta_g \eta_l$	Global and local learning rate
K	The number of clients selected for a single aggregation
$\sigma_g^2, \sigma_\ell^2$	Global and Local gradient variance
$\tau_i(t)$	Outdatedness of client i's model after t rounds of global updates
$\tau_{\max}$	Obsolete upper bound

Lemma 2 $\mathbb{E}\left[\left\|g_k\right\|^2\right] \leq 3\left(\sigma_l^2 + \sigma_g^2 + G\right)$, *where the expectation* $\mathbf{E}[\cdot]$ *takes into account the random participation of clients and the client's random gradient.*

To simplify the analysis without compromising the proof of convergence, we can disregard the denominator term in Lemma 1. Based on this, we can derive the convergence rate of FedEem as follows:

Theorem 1 *Based on Assumptions 1, 2, 3 and 4 and Lemma 1, the convergence rate of FedEem can be expressed as*:

$$\frac{1}{T}\sum_{\tau=0}^{T-1}\mathbb{E}\left\|\nabla f(\boldsymbol{w}_\tau)\right\|^2 \leq 2\frac{\left(f(\boldsymbol{w}_0)-f(\boldsymbol{w}^*)\right)}{\phi(E)TK} + 6K(\alpha+\beta)^2\lambda(d)L^2E\psi(E)\left(K^2\Omega^2+1\right)\sigma^2 + L\frac{\psi(E)}{K\phi(E)}(\alpha+\beta)\sigma_l^2, \tag{4}$$

Where $\lambda(d) = \sum_{i=1}^{K} d_i^2, \phi(E) = \sum_{j=1}^{E}\eta_l^j, \psi(E) = \sum_{j=1}^{E}\left(\eta_l^j\right)^2$. In addition, to simplify the expression, let $\sigma^2 = (\alpha+\beta)\sigma_l^2 + (\alpha+\beta)\delta_g^2 + G^2$.In addition, in order to ensure the convergence upper bound, K and ηl must satisfy the following relationship:

$$\frac{4(\alpha+\beta)}{\alpha^2\lambda(d)}K\eta_l^j \leq \frac{1}{L}. \tag{5}$$

Proof

The update process of FedEem can be described as: □

$$w^{t+1} = w^t + \eta_g\tilde{\Delta}^t = w^t + \eta_g\frac{1}{K}\sum_{k\in S^t}\left(-\eta_l\sum_{q=1}^{\epsilon_k}g_k\left(y_{k,q}^{t-\tau_k(t)}\right)\right), \tag{6}$$

Where S^t represents selected clients in the t-th global update.

Specifically, unlike previous federated learning proof, due to data heterogeneity and device heterogeneity, it cannot be simply assumed that St is a unified subset because the possibility of clients participating is not the same in different rounds. Specifically, in the early rounds, fast clients are more likely to participate in more rounds due to faster updates, while in the later rounds, the situation is reversed as fast clients drop out early.

Following the common convergence proof procedure used in federated learning methods, the proof of convergence rate for the non-convex objective function proposed in [35] starts by utilizing smoothness Assumption 1. Therefore, it follows that :

$$f\left(w^{t+1}\right) \leq f\left(w^t\right) + T_1(t) + T_2(t), \tag{7}$$

$$T_1(t) = -\eta_q\sum_{k\in S_k}p_k^t\left\langle\nabla\tilde{f}\left(w^t\right),\Delta_k^{t-\tau_k}\right\rangle \tag{8}$$

$$T_2(t) = \frac{L\eta_g^2}{2}\left\|\sum_{k\in S_k}p_k^t\Delta_k^{t-\tau_k}\right\|^2 \tag{9}$$

Where $\Delta_k^{t-\tau_k}$ is the parameter update made by the client after receiving global model parameters before the $t-\tau_k$ global update. $\nabla\tilde{f}\left(w^t\right)$ is the global gradient at global update round t. Then, upper bounds are computed for T_1 and T_2

$$\begin{aligned} T_1(t) &= -\eta_g\sum_{k\in S_t}p_k^t\left\langle\nabla f\left(w^t\right)\cdot\sum_{q=1}^{\epsilon_k}\eta_t^{(q)}g_k\left(y_{k,q}^{t-\tau_k}\right)\right\rangle \\ &= -\frac{\eta_g}{K}\sum_{k\in S_t}\sum_{q=1}^{\epsilon_k}\eta_t^{(q)}p_k^t\left\langle\nabla f\left(w^t\right),g_k\left(y_{k,q}^{t-\tau_k}\right)\right\rangle. \end{aligned} \tag{10}$$

By utilizing conditional expectations, it is possible to represent the expectation operator in a more concise manner:

$$\mathbb{E}[\cdot] := \mathbb{E}_\pi\mathbb{E}_{i\sim[m_t]}\mathbb{E}_{g_i,p_i^t|i,\pi}[\cdot], \tag{11}$$

Where $\mathbb{E}_\pi$ is the expectation with respect to all client policies,$\pi = \{\pi_1,\ldots,\pi_N\}$ represents the collection of all client policies participating in federated learning, $\mathbb{E}_{i\sim[m_t]}$ is the evaluation over the randomness of selecting client $i\sim[m_t]$

from the distribution at the global round t. Please note that m_t is not a fixed value due to the presence of dropout mechanism. The inner expectation refers to one-step of random gradient descent on the client. Therefore, under the unbiased estimation assumption, we have

$$\begin{aligned}\mathbb{E}[T_1(t)] &= -\mathbb{E}\left[\frac{\eta_g}{K}\sum_{k\in S_i}\sum_{q=1}^{\epsilon_k}\eta_l p_k^t\left\langle\nabla\tilde{f}(w^t), g_k\left(y_{k,q}^{t-\tau_k}\right)\right\rangle\right]\\ &= -\eta_g\mathbb{E}_\pi\left[\frac{1}{m}\sum_{i=1}^{m}\sum_{q=1}^{\epsilon_i}\eta_l^{(q)}\mathbb{E}_{g_i|i\sim[m]}\left\langle\nabla\tilde{f}(w^t), g_i\left(y_{i,q}^{t-\tau_i}\right)\right\rangle\right]\\ &= -\frac{\eta_g}{m_t}\mathbb{E}_\pi\left[\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_k}\eta_l\left\langle\nabla\tilde{f}(w^t), p_i^t\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\rangle\right]\\ &= -\eta_g\bar{\eta}_l\mathbb{E}_\pi\left[\left\langle\nabla\tilde{f}(w^t), \frac{1}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_k}p_i^t\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\rangle\right].\end{aligned} \tag{12}$$

Furthermore,$\mathbb{E}[T_1(t)]$ can be written as:

$$\begin{aligned}\mathbb{E}[T_1(t)] = -\frac{\eta_g\overline{\eta_l}}{2}\mathbb{E}\left(\left\|\nabla\tilde{f}(w^t)\right\|^2\right) + \frac{\eta_g\bar{\eta}_l}{2}\Bigg(&-\mathbb{E}_\pi\left\|\frac{1}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_i}\eta_l^{(q)}p_i^k\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2\\ &+\underbrace{\mathbb{E}_\pi\left\|\nabla\tilde{f}(w^t) - \frac{1}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_i}p_i^k\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2}_{T_3(t)}.\end{aligned} \tag{13}$$

For T_3, it can be derived from the definition of $f(w_t)$

$$\begin{aligned}\mathbb{E}_\pi[T_3(t)] &= \mathbb{E}_\pi\left\|\frac{1}{m_t}\sum_{i=1}^{m_t}\nabla F_i(w^t) - \frac{1}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_i}p_i^k\nabla F_i\left(y_{i,q}^{t-r_i}\right)\right\|^2\\ &\le \frac{1}{m_t}\sum_{i=1}^{m_t}\mathbb{E}_\pi\left\|\sum_{q=1}^{\epsilon_i}p_i^k\left[\nabla F_i(w^t) - \nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right]\right\|^2\\ &\le \frac{1}{m_t}\sum_{i=1}^{m_t}\mathbb{E}_\pi\sum_{q=1}^{\epsilon_i}p_i^k\left\|\left[\nabla F_i(w^t) - \nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right]\right\|^2.\end{aligned} \tag{14}$$

By defining $\gamma_i(t) = \sum_{q=1}^{\epsilon_i}p_i^k$, further, T_3 can be expressed as the error caused by obsolescence and local drift.

$$\begin{aligned}\mathbb{E}[T_3(t)] &\le \frac{2}{m}\sum_{i=1}^{m}\mathbb{E}_\pi\gamma_i(t)(O_e + C_d)\\ &\le \frac{2}{m}\sum_{i=1}^{m}\left(L^2\mathbb{E}_\pi\gamma_i(t)\left\|w^t - w^{t-\tau_i}\right\|^2 + L^2\mathbb{E}_\pi\gamma_i(t)\left\|w^{t-\tau_i} - y_{i,q}^{t-\tau_i}\right\|^2\right).\end{aligned} \tag{15}$$

$$O_e = \left\|\nabla F_i\left(w^t\right) - \nabla F_i\left(w^{t-\tau_i}\right)\right\|^2 \tag{16}$$

$$C_d = \left\|\nabla F_i\left(w^{t-\tau_i}\right) - \nabla F_i\left(y_{t,q}^{t-\tau_i}\right)\right\|^2 \tag{17}$$

Where,O_e is the Obsolescence error, and C_d is the client drift.

The errors caused by obsolescence can be mitigated by accumulating them as model updates between rounds

$$\begin{aligned}\left\|w^t - w^{t-\tau_i}\right\|^2 &= \left\|\sum_{\rho=t-\tau_i}^{t-1}\left(w^{\rho+1} - w^\rho\right)\right\|^2\\ &= \left\|\sum_{\rho=t-\tau_i}^{t-1}\frac{\eta_g}{K}\sum_{j_\rho\in S_\rho}\Delta_{j_\rho}^\rho\right\|^2\\ &= \frac{\eta_g^2}{K^2}\left\|\sum_{\rho=t-\tau_i}^{t-1}\sum_{j_\rho\in S_\rho}\sum_{l=0}^{Q-1}\eta_l^{(l)}g_{j_\rho}\left(y_{j_\rho,l}^\rho\right)\right\|^2.\end{aligned} \tag{18}$$

The upper bound for computing its expectation can be obtained

$$\begin{aligned}\gamma_i(t)\mathbb{E}_\pi\left\|w^t - w^{t-\tau_i}\right\|^2 &\le \frac{\eta_g^2\tau_i}{K}\mathbb{E}_\pi(\gamma_i(t)\epsilon_i)\sum_{\rho=t-\tau_i}^{t-1}\sum_{j_\rho\in S_\rho}\sum_{l=0}^{\epsilon}\left(\eta_l^{(l)}\right)^2\mathbb{E}\left\|g_{j_\rho}\left(y_{j_\rho,l}^p\right)\right\|^2\\ &\le 3\eta_g^2\mathbb{E}_\pi(\gamma_i(t)\epsilon_i)\max_{\tau_i}\tau_i^2\left(\sum_{l=1}^{\epsilon_l}\left(\eta_l^{(l)}\right)^2\right)\left(\sigma_l^2 + \sigma_g^2 + G\right)\\ &\le 3\eta_g^2\eta_l^2\mathbb{E}_\pi\left(\gamma_i(t)\epsilon^2\right)\tau_{\max,K}^2\left(\sigma_l^2 + \sigma_g^2 + G\right),\end{aligned} \tag{19}$$

The second inequality utilizes Lemma 2 for bounding, and the last inequality utilizes $\mathbb{E}(X)^2 \le \mathbb{E}\left(x^2\right)$. The expected error caused by local drift can be similarly constrained as:

$$\begin{aligned}\mathbb{E}[T_3] &\le 6\left(L^2\eta_g^2\eta_l^2\mathbb{E}_\pi\left(\gamma_i(t)\epsilon^2\right)\tau_{\max,K}^2\left(\sigma_l^2+\sigma_g^2+G\right)+L^2q\left(\sum_{i=0}^{g-1}\left(\eta_l^{(i)}\right)^2\right)\left(\sigma_l^2+\sigma_g^2+G\right)\right)\\ &\le 6L^2\mathbb{E}_\pi\left(\gamma_i(t)\epsilon^2\right)\left(\eta_g^2\tau_{\max,K}^2+\frac{1}{2}\right)\left(\sigma_l^2+\sigma_g^2+G\right),\end{aligned} \tag{20}$$

Substituting the constraint of T_3 back into T_1 yields:

$$\mathbb{E}[T_1] \le -\frac{\eta_g\eta_l}{2}\left\|\nabla f\left(w^t\right)\right\|^2+\frac{\eta_g\eta_l}{2}\mathbb{E}[T_3]-\mathbb{E}_\pi\left\|\frac{1}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_i}\eta_l^{(q)}p_i^k\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2. \tag{21}$$

Let $\beta(Q) := \sum_{q=0}^{Q-1}\left(\eta_\ell^{(q)}\right)^2$. Therefore, we have:

$$\begin{aligned}(T_4+T_5) &= -\mathbb{E}_\pi\left\|\frac{1}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_i}\eta_l^{(q)}p_i^k\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2+\frac{L\eta_g^2\mathbb{E}_\pi\bar{\epsilon}}{2K}\sum_{k\in S_i}\left(\eta_\ell^{(q)}\right)^2\mathbb{E}_\pi\left[\frac{p_i^t}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_k}\left\|\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2\right]\\ &\le -\mathbb{E}_\pi\left\|\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_i}\eta_l^{(q)}\frac{p_i^k}{m_t}\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2+\frac{L\eta_g^2\mathbb{E}_\pi\bar{\epsilon}}{2K}\sum_{k\in S_i}\left(\eta_\ell^{(q)}\right)^2\mathbb{E}_\pi\left[\frac{p_i^t}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_k}\left\|\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2\right]\\ &= \left(-\eta_g-L\mathbb{E}_\pi\bar{\epsilon}\eta_g^2\eta_l^2\right)\mathbb{E}_\pi\left\|\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_i}\eta_l^{(q)}\frac{p_i^k}{m_t}\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2.\end{aligned} \tag{24}$$

$$\begin{aligned}\mathbb{E}[T_t] \le &-\frac{\eta_g\eta_l}{2}\left\|\nabla f\left(w^t\right)\right\|^2+3L^2\mathbb{E}_\pi\left(\gamma_i(t)\epsilon^2\right)\left(\eta_g^2\tau_{\max,K}^2+\frac{1}{2}\right)\left(\sigma_l^2+\sigma_g^2+G\right)\\ &\underbrace{-\mathbb{E}_\pi\left\|\frac{1}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_i}\eta_l^{(q)}p_i^k\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2}_{T_4}.\end{aligned} \tag{22}$$

Therefore, in order to ensure that $T_4+T_5\le 0$, it is required that for all local gradient descent steps, $\eta_g\eta_\ell\mathbb{E}_\pi\bar{\epsilon}\le\frac{1}{L}$. Finally, combining$T1$, $T2$ provides the expected improvement in performance between two adjacent global models:

For the constraint on the expected value of T_2, we have:

$$\begin{aligned}\mathbb{E}[T_2(t)] &= \mathrm{E}\left[\frac{\mathrm{L}\eta_g^2}{2\mathrm{K}^2}\left\|\sum_{\mathrm{k}\in\mathrm{S_t}}\sum_{\mathrm{q}=0}^{\epsilon_\mathrm{k}}\eta_\ell^{(\mathrm{q})}\mathrm{p_k^t}\mathrm{g_k}\left(\mathrm{y_{k,q}^{t-\tau_k}}\right)\right\|^2\right]\\ &= \mathbb{E}\left[\frac{L\eta_g^2}{2K^2}\left\|\sum_{k\in S_k}\sum_{q=1}^{\epsilon_k}\eta_k^{(q)}p_k^t\left(g_k\left(y_{k,q}^{t-\tau_k}\right)-\nabla F_k\left(y_{k,q}^{t-\tau_k}\right)\right)+\sum_{k\in S_k}\sum_{q=1}^{\epsilon_k}\eta_k^{(q)}p_k^t\nabla F_k\left(y_{k,q}^{t-\tau_k}\right)\right\|^2\right]\\ &= \frac{L\eta_q^2}{2K^2}\mathbb{E}\left\|\sum_{k\in S,\,q=1}^{\epsilon_k}\eta_\ell^{(q)}p_k^t\left(g_k\left(y_{k,q}^{t-r_k}\right)-\nabla F_k\left(y_{k,q}^{t-r_k}\right)\right)\right\|^2+\frac{L\eta_g^2}{2K^2}\mathbb{E}\left\|\sum_{k\in S}\sum_{q=1}^{\epsilon_k}\eta_\ell^{(q)}p_k^t\nabla F_k\left(y_{k,q}^{t-r_k}\right)\right\|^2\\ &= \frac{L\eta_g^2}{2}\sum_{k\in S}\sum_{q=1}^{\epsilon_k}\left(\eta_\ell^{(q)}p_k^t\right)^2\mathbb{E}\left\|\left(q_k\left(y_{k,q}^{t-\tau_k}\right)-\nabla F_k\left(y_{k,q}^{t-\tau_k}\right)\right)\right\|^2\\ &\quad+\frac{L\eta_g^2}{2K^2}\mathbb{E}_\pi\bar{\epsilon}\mathbb{E}\sum_{k\in S,q=1}^{\epsilon_k}\left\|\eta_\ell^{(q)}p_k^t\nabla F_k\left(y_{k,q}^{t-\tau_k}\right)\right\|^2\\ &\le \frac{L\eta_g^2\eta_l^2\zeta(t)\sigma_\ell^2}{2}+\frac{L\eta_g^2\mathbb{E}_\pi\bar{\epsilon}}{2K}\sum_{k\in S_k}\left(\eta_\ell^{(q)}\right)^2\mathbb{E}_\pi\mathbb{E}_{k\sim[m_t]\pi}p_k^t\|\sum_{q=1}^{\epsilon_k}\nabla F_k\left(y_{k,q}^{t-\tau_k)}\right)\|^2\\ &= \frac{L\eta_g^2\eta_l^2\zeta(t)\sigma_\ell^2}{2}+\underbrace{\frac{L\eta_g^2\mathbb{E}_\pi\bar{\epsilon}}{2K}\sum_{k\in S_i}\left(\eta_\ell^{(q)}\right)^2\mathbb{E}_\pi\left[\frac{p_i^t}{m_t}\sum_{i=1}^{m_t}\sum_{q=1}^{\epsilon_k}\left\|\nabla F_i\left(y_{i,q}^{t-\tau_i}\right)\right\|^2\right]}_{T_5},\end{aligned} \tag{23}$$

$$\mathbb{E}\left[f\left(w^{t+1}\right)\right] \leq \mathbb{E}\left[f\left(w^{t}\right)\right] - \frac{\eta_g \gamma(t}{2}\left\|\nabla f\left(w^{t}\right)\right\|^2 + 3\phi_g L^2 Q\gamma(t)\zeta(t)\left(\eta_g^2 \pi_{\max,K^2} + \frac{1}{2}\right)\left(\sigma_l^2 + \sigma_g^2 + G\right) + \frac{L}{2}\eta_g^2\zeta(t)\sigma_l^2 \tag{25}$$

After nested summation from $t = 1, \cdots, T$, the above equation can be obtained.

$$\begin{aligned}
&\sum_{t=0}^{T-1} \eta_g \gamma(t)\left\|\nabla f\left(w^{t}\right)\right\|^2 \\
&\leq \sum_{t=0}^{T-1} 2\left(\mathbb{E}\left[f\left(w^{t}\right)\right] - \mathbb{E}\left[f\left(w^{t+1}\right)\right]\right) + 3\sum_{t=0}^{T-1} \eta_g L^2 \mathbb{E}_\pi \bar{\epsilon}\gamma(t)\zeta(t)\left(\eta_g^2\tau_{\max,K}^2 + 1\right)\left(\sigma_t^2 + \sigma_g^2 + G\right) + \frac{L}{2}\eta_g^2\zeta(t)\sigma_l^2 \\
&\leq 2\left(f\left(w^0\right) - f\left(w^n\right)\right) + 3\sum_{s=0}^{T-1} \eta_g L^2 \gamma(t)\zeta(t)\left(\eta_g^2\tau_{\max,K}^2 + 1/2\right)\left(\sigma_l^2 + \sigma_g^2 + G\right) + \frac{L}{2}\eta_g^2\zeta(t)\sigma_\ell^2
\end{aligned} \tag{26}$$

Therefore, Theorem 1 can be obtained.

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla f\left(w^{t}\right)\right\|^2 \leq\ & \frac{2\left(f\left(w^0\right) - f(w^*)\right)}{\eta_g \cdot \alpha(Q) T} \\
&+ 3L^2 Q\beta(Q)\left(\eta_g^2\tau_{\max,K}^2 + 1\right)\left(\sigma_l^2 + \sigma_g^2 + G\right) + \frac{L}{2}\frac{\eta_g \beta(Q)}{\alpha(Q)}\sigma_l^2
\end{aligned} \tag{27}$$

Proof

Lemma 2 □

$$\begin{aligned}
\left\|\mathbf{w}_{t+1} - \mathbf{w}^\star\right\|^2 &= \left\|\mathbf{w}_t + \eta_g \sum_{k\in S^t} p_k \left(-\eta_\ell \sum_{q=1}^{Q} g_k\left(y_{k,q}^{t-\tau_k(t)}\right)\right)\right. \\
&= \left\|\mathbf{w}_t - \mathbf{w}^\star\right|^2 + T_1 + T_2
\end{aligned} \tag{28}$$

$$T_1 = 2\eta_t(\overline{\mathbf{w}}_t - \mathbf{w}^\star, +\eta_g \sum_{k\in S^t} p_k \left(-\eta_\ell \sum_{q=1}^{Q} g_k\left(y_{k,q}^{t-\tau_k(t)}\right)\right) \tag{29}$$

$$T_2 = \eta_g^2 \left\| \sum_{k\in S^t} p_k \left(-\eta_\ell \sum_{q=1}^{Q} g_k\left(y_{k,q}^{t-\tau_k(t)}\right)\right)\right\|^2 \tag{30}$$

Experiment and analysis

To understand the impact of various hyperparameters in the convergence process and computational resource consumption in federated learning, this study conducted an analysis from two perspectives: the influence of hyperparameters on the performance of FedEem and the computational speed. By controlling variables and conducting a series of comparative experiments, we demonstrated the efficiency and fairness of FedEem .

Experimental setup

To ensure fairness in the experimental comparison, this paper primarily focuses on comparing the number of rounds required for the global model to achieve a specific accuracy threshold (e.g., 95% accuracy on the MNIST dataset). The experiments involve a fixed total of 20 clients.

The federated learning process is simulated using FLsim, a simulator specifically designed for experimental research [36]. FLsim utilizes JSON files to manage the configuration parameters of federated learning simulations and provides a general template along with three pre-configured simulation files for the CIFAR-10, FashionMNIST, and MNIST datasets. In this study, we implemented federated learning algorithms such as FedBuff for conducting comparative experiments.

All simulation experiments were performed on a PC server running Ubuntu Linux 21.1.0. The server is equipped with an Intel i5-10600KF (4.10GHz) processor, 64GB RAM, and 4 NVIDIA TITAN-V GPUs. The experimental environment utilizes Python 3.9.5 and PyTorch 1.8.1.

Analysis of experimental results

Figures 2 and 3 show the performance comparison of FedEem with other state-of-the-art algorithms under the scenarios of uniform and randomly independent distributions. Due to the presence of the early exit mechanism, FedEem has a significant advantage in terms of convergence speed compared to other asynchronous federated learning algorithms. In addition, the aggregation mechanism optimized by FedEem allows for a more stable convergence process, as it is less affected by the obsolescence of model updates and interference caused by large gradient differences.

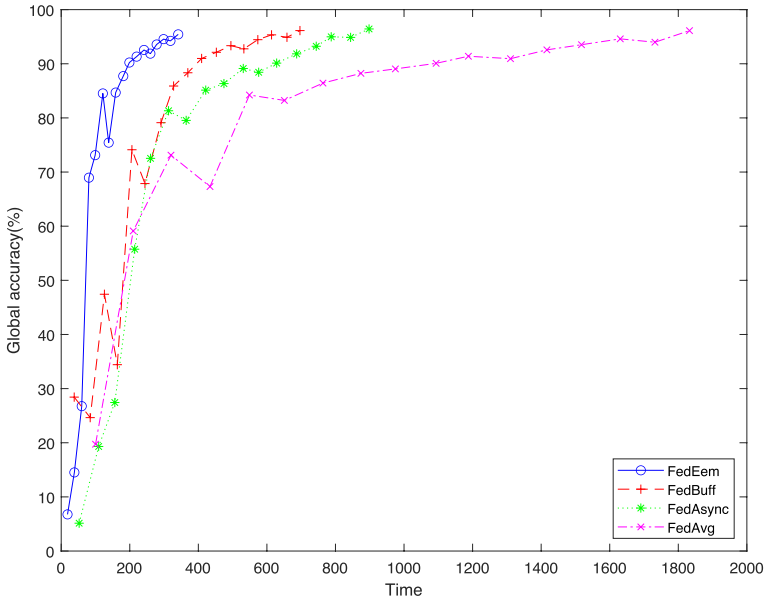

Fig. 2 Concurrency level is 10, with each client having 120 data samples. The data is uniformly distributed with a Non-IID pattern

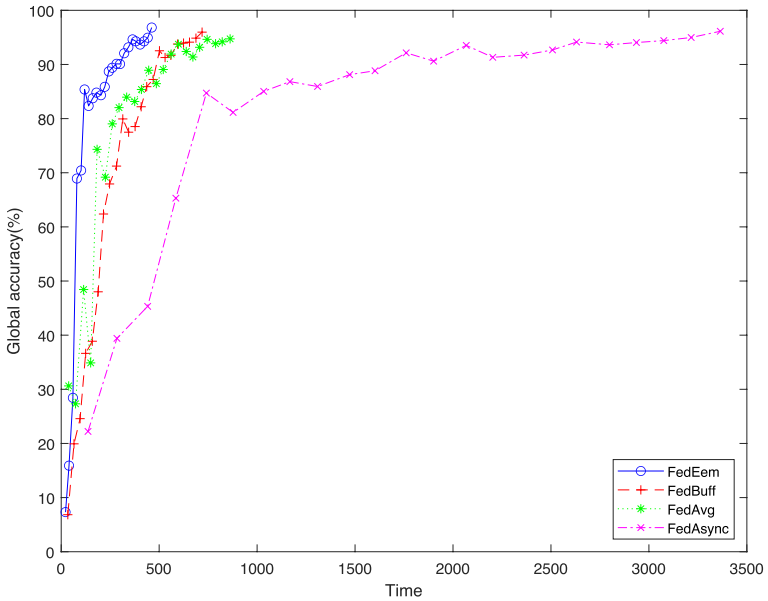

Fig. 3 The concurrent number is 10, and each client has 120 data, with data randomly distributed in a Non-IID manner

Figure 4 illustrates the convergence process of FedEem under different choices of regularization weights, with 30 clients and four repetitions of experiments. It can be observed that different hyperparameter choices have significant differences in terms of time and round consumption, but are not consistent in terms of variance. Therefore, making intelligent decisions regarding hyperparameters in the asynchronous federated learning process is necessary.

Conclusion

This paper investigates an optimized mechanism for asynchronous federated learning in the context of edge computing scenarios. Firstly, the necessity of the

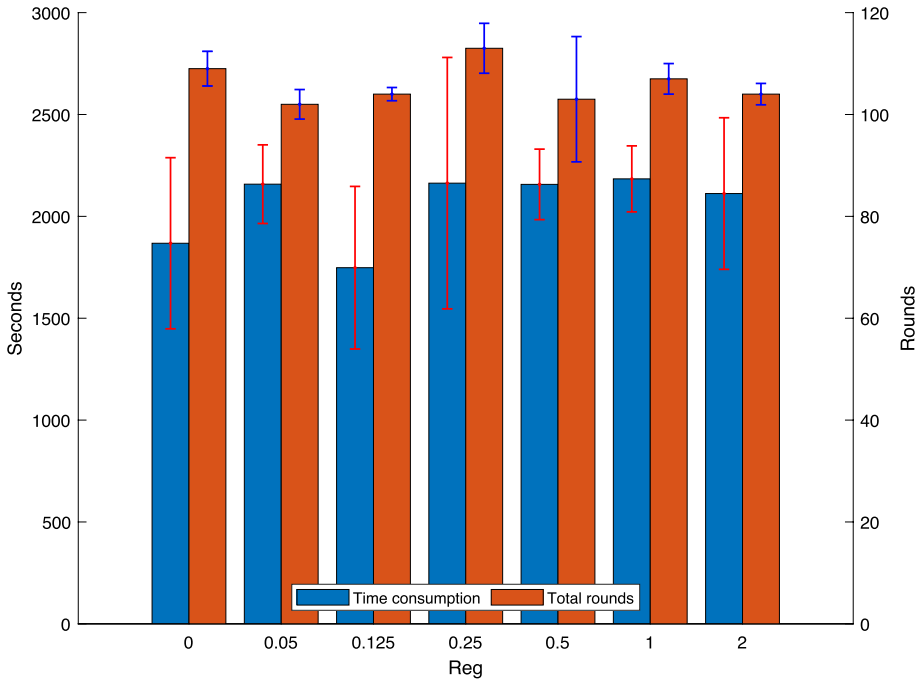

Fig. 4 Performance comparison of asynchronous federated learning under different numbers of clients

asynchronous mechanism in highly heterogeneous federated learning is analyzed. The paper also addresses the fairness issues in previous asynchronous federated learning algorithms and proposes an optimized mechanism called FedEem . This mechanism includes a weight aggregation mechanism that incorporates timeliness and fairness considerations, as well as an early exit mechanism. Experimental results demonstrate that the proposed algorithm achieves significant improvements in both convergence time and fairness under various data distributions and device heterogeneity.

Acknowledgements
We would like to express our sincere gratitude to the editors and reviewers for their invaluable feedback and comments on this paper.

Authors' contributions
Author Contributions Statement: Each author has made significant contributions to the research and preparation of this manuscript. [G.] conceived the research idea, designed the experiments, and conducted the data analysis. Additionally, [G.] contributed to the literature review, data collection, and manuscript writing. [Z.] provided technical guidance, reviewed and revised the manuscript. [Z.] also contributed to the experimental design, conducted the experiments, and analyzed the results. Furthermore, [Z.] provided critical feedback and contributed to the interpretation of the findings. All authors have read and approved the final version of the manuscript and take full responsibility for its content.

Funding
Not applicable.

Availability of data and materials
All code used to support this work is available from the authors upon request.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 21 August 2023 Accepted: 25 October 2023
Published online: 09 November 2023

References

1. Gong B, Xing T, Liu Z, Wang J, Liu X (2022) Adaptive clustered federated learning for heterogeneous data in edge computing. Mob Netw Appl 27(4):1520–1530
2. Xu X, Li H, Li Z, Zhou X (2022) Safe: Synergic data filtering for federated learning in cloud-edge computing. IEEE Trans Ind Inform 19(2):1655–1665
3. Wu S, Shen S, Xu X, Chen Y, Zhou X, Liu D, Xue X, Qi L (2022) Popularity-aware and diverse web apis recommendation based on correlation graph. IEEE Trans Comput Soc Syst 10(2):771–782
4. Wang F, Zhu H, Srivastava G, Li S, Khosravi MR, Qi L (2021) Robust collaborative filtering recommendation with user-item-trust records. IEEE Trans Comput Soc Syst 9(4):986–996
5. Liang F, Yang Q, Liu R, Wang J, Sato K, Guo J (2022) Semi-synchronous federated learning protocol with dynamic aggregation in internet of vehicles. IEEE Trans Veh Technol 71(5):4677–4691
6. You L, Liu S, Chang Y, Yuen C (2022) A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement. IEEE Internet Things J 9(23):24199–24211
7. Hu CH, Chen Z, Larsson EG (2023) Scheduling and aggregation design for asynchronous federated learning over wireless networks. IEEE J Sel Areas Commun 41(4):874–886
8. Liu Y, Zhou X, Kou H, Zhao Y, Xu X, Zhang X, Qi L (2023) Privacy-preserving point-of-interest recommendation based on simplified graph

convolutional network for geological traveling. ACM Trans Intell Syst Technol. https://doi.org/10.1145/3620677
9. Hosseini SM, Sikaroudi M, Babaie M, Tizhoosh H (2023) Proportionally fair hospital collaborations in federated learning of histopathology images. IEEE Trans Med Imaging 42(7):1982–1995
10. Chen Y, Ning Y, Slawski M, Rangwala H (2020) Asynchronous online federated learning for edge devices with non-iid data. In: 2020 IEEE International Conference on Big Data (Big Data). IEEE, Piscataway, p 15–24
11. Nguyen J, Malik K, Zhan H, Yousefpour A, Rabbat M, Malek M, Huba D (2022) Federated learning with buffered asynchronous aggregation. In: International Conference on Artificial Intelligence and Statistics. PMLR, NY, p 3581–3607
12. Su N, Li B (2022) How asynchronous can federated learning be? In: 2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS). IEEE, Piscataway, p 1–11
13. Tan AZ, Yu H, Cui L, Yang Q (2022) Towards personalized federated learning. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2022.3160699
14. Li Q, Diao Y, Chen Q, He B (2022) Federated learning on non-iid data silos: An experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE, Piscataway, p 965–978
15. Qi L, Lin W, Zhang X, Dou W, Xu X, Chen J (2022) A correlation graph based approach for personalized and compatible web apis recommendation in mobile app development. IEEE Trans Knowl Data Eng 35(6):5444–5457
16. Chai Z, Ali A, Zawad S, Truex S, Anwar A, Baracaldo N, Zhou Y, Ludwig H, Yan F, Cheng Y (2020) Tifl: A tier-based federated learning system. In: Proceedings of the 29th international symposium on high-performance parallel and distributed computing. ACM, New York, p 125–136
17. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2020) Federated optimization in heterogeneous networks. Proc Mach Learn Syst 2:429–450
18. Jiang Y, Konečnỳ J, Rush K, Kannan S (2019) Improving federated learning personalization via model agnostic meta learning. arXiv preprint arXiv:1909.12488. https://doi.org/10.48550/arXiv.1909.12488
19. Yang H, He H, Zhang W, Cao X (2020) Fedsteg: A federated transfer learning framework for secure image steganalysis. IEEE Trans Netw Sci Eng 8(2):1084–1094
20. Duan M, Liu D, Ji X, Liu R, Liang L, Chen X, Tan Y (2021) Fedgroup: Efficient federated learning via decomposed similarity-based clustering. In: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, Piscataway, p 228–237
21. Huang Y, Chu L, Zhou Z, Wang L, Liu J, Pei J, Zhang Y (2021) Personalized cross-silo federated learning on non-iid data. In: Proceedings of the AAAI conference on artificial intelligence. Menlo Park, 35:7865–7873
22. Li D, Wang J (2019) Fedmd: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581. https://doi.org/10.48550/arXiv.1910.03581
23. Zhou Z, Chu L, Liu C, Wang L, Pei J, Zhang Y (2021) Towards fair federated learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. ACM, New York, p 4100–4101
24. Lyu L, Xu X, Wang Q, Yu H (2020) Collaborative fairness in federated learning. Federated Learn Priv Incent 12500:189–204
25. Lim WYB, Xiong Z, Miao C, Niyato D, Yang Q, Leung C, Poor HV (2020) Hierarchical incentive mechanism design for federated machine learning in mobile networks. IEEE Internet Things J 7(10):9575–9588
26. Zhan Y, Li P, Wang K, Guo S, Xia Y (2020) Big data analytics by crowdlearning: Architecture and mechanism design. IEEE Netw 34(3):143–147
27. Zhan Y, Zhang J, Li P, Xia Y (2019) Crowdtraining: Architecture and incentive mechanism for deep learning training in the internet of things. IEEE Netw 33(5):89–95
28. Du W, Xu D, Wu X, Tong H (2021) Fairness-aware agnostic federated learning. In: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM). SIAM, p 181–189
29. Liang PP, Liu T, Ziyin L, Allen NB, Auerbach RP, Brent D, Salakhutdinov R, Morency LP (2020) Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523. https://doi.org/10.48550/arXiv.2001.01523
30. Xu X, Tang S, Qi L, Zhou X, Dai F, Dou W (2023) Cnn partitioning and offloading for vehicular edge networks in web3. IEEE Commun Mag 61(8):36–42
31. Zhou X, Bilal M, Dou R, Rodrigues JJ, Zhao Q, Dai J, Xu X (2023) Edge computation offloading with content caching in 6g-enabled iov. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2023.3239599
32. Xu X, Yang C, Bilal M, Li W, Wang H (2022) Computation offloading for energy and delay trade-offs with traffic flow prediction in edge computing-enabled iov. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2022.3221975
33. Wu J, Zhang J, Zhang Y, Wen Y (2023) Constraint-aware and multi-objective optimization for micro-service composition in mobile edge computing. Softw Pract Exp. https://doi.org/10.1002/spe.3217
34. Qi L, Xu X, Wu X, Ni Q, Yuan Y, Zhang X (2023) Digital-twin-enabled 6g mobile network video streaming using mobile crowdsourcing. IEEE J Sel Areas Commun. https://doi.org/10.1109/JSAC.2023.3310077
35. Zhan Y, Zhang J, Hong Z, Wu L, Li P, Guo S (2021) A survey of incentive mechanism design for federated learning. IEEE Trans Emerg Top Comput 10(2):1035–1044
36. Wang H, Kaplan Z, Niu D, Li B (2020) Optimizing federated learning on non-iid data with reinforcement learning. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, Piscataway, p 1698–1707

Publisher's Note