Dynamic Adaptive Federated Learning on Local Long-Tailed Data

Juncheng Pu, *Student Member, IEEE*, Xiaodong Fu, *Member, IEEE*, Hai Dong, *Senior Member, IEEE*, Pengcheng Zhang, *Member, IEEE* and Li Liu

Abstract—Federated learning provides privacy protection to the collaborative training of global model based on distributed private data. The local private data is often in the presence of long-tailed distribution in reality, which downgrades the performance and causes biased results. In this paper, we propose a dynamic adaptive federated learning optimization algorithm with the Grey Wolf Optimizer and Markov Chain, named FedWolf, to solve the problems of performance degradation and result bias caused by the local long-tailed data. FedWolf is launched with a set of randomly initialized parameters instead of a shared parameter employed by existing methods. Then multi-level participants are elected based on the F1 scores calculated from the uploaded parameters. A dynamic weighting strategy based on the participant level is used to adaptively update parameters without artificial control. The above parameter updating is modelled as a Markov Process. After all communication rounds are completed, the future performance (including the probability of each participant is elected as different participant level) of participants is predicted through the historical Markov states. Finally, the probability of each participant is elected as the level 1 is used as the contribution weight and the global model is obtained through dynamic contribution weight aggregating. We introduce the Gini index to evaluate the bias of classification results. Extensive experiments are conducted to validate the effectiveness of FedWolf in solving the problems of performance cracks and categorization result bias as well as the robustness of adaptive parameter updating in resisting outliers and malicious users.

Index Terms—Federated Learning, Long-Tailed Data, Grey Wolf Optimizer, Markov Chain, Dynamic Adaptive Weighting

1 Introduction

Federated learning, as a nascent distributed computing technology, ensures privacy in collaborative multi-party computation. Nevertheless, substantial heterogeneity arises among participants due to variations in factors such as environment, equipment, and statistical characteristics, which had been demonstrated to lead to instability, sluggish convergence, and even degradation in the global model's quality [1]. Enhancing performance hinges on effectively managing non-independent identically distributed (Non-IID) data, responsible for statistical heterogeneity that can result in parameter misalignment [2], [3]. Recent research has honed in label imbalance, a subset of Non-IID. Label imbalance, arising from skewed label distributions, imparts diverse gradients to participant gradient descents [4]. Particularly, the research focus has shifted towards long-tailed distribution data, characterized by a substantial number of labels and a mandatory imbalance factor [5]. This study concentrates on the local long-tailed data, signifying datasets with long-tailed distributions peculiar to each participant. Code can be found in this link[1].

Deep learning's performance is hindered by local optimization issues within local long-tailed datasets, leading to performance gaps and categorization biases in outcomes [6]. These challenges extend to federated learning. We explore how local optimization negatively impacts federated learning's ability to handle local long-tailed data through a comparative experiment. Initially, we assess the performance of a non-federated approach, optimizing globally on CIFAR-100 [7], establishing the upper limit of global model performance (Ideal model training and Ideal model testing in Fig. 1 (a)). Subsequently, CIFAR-100 is divided into ten copies with long-tailed distributions, and each participant independently trains on one copy. We randomly select Participant 6 as an example to illustrate the limitations in performance and classification bias arising from local long-tailed data without aggregation (Non-aggre training and Non-aggre testing in Fig. 1 (a), with dataset distribution and Participant 6's classification results shown in Fig. 1 (b)). Finally, we evaluate the performance of the Federated Averaging algorithm (FedAVG) [1] (FedAVG local training, FedAVG local testing, and FedAVG global in Fig. 1 (a)) using a local long-tailed data copy to demonstrate how local optimization's impact extends to federated learning.

As shown in Fig. 1, the training accuracy of all methods converge to a high upper bound, which proves the training performance is not affected by the local data with long-tailed distribution. An important experiential observation is the local long-tailed data leads to the performance crack and the categorization bias of testing results. The non-federated model (Ideal model in Fig. 1 (a)) achieves the best performance of testing accuracy and the smallest gap between training and testing accuracy. Because the centralized data

- *J. Pu, X. Fu, and L. Liu are with the Yunnan Provincial Key Laboratory of Computer Technology Application, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. E-mail: pujuncheng163@163.com, xiaodong_fu@hotmail.com, kmust_mary@163.com*
- *H. Dong is with the School of Computing Technologies, RMIT University, Melbourne, Australia. E-mail: hai.dong@rmit.edu.au*
- *P. Zhang is with the College of Computer and Information, Hohai University, Nanjing, China. E-mail: pchzhang@hhu.edu.cn*

Manuscript received xxx; revised xxx.
(Corresponding author: Xiaodong Fu. xiaodong_fu@hotmail.com)

1. https://github.com/TaisePu/FedWolf

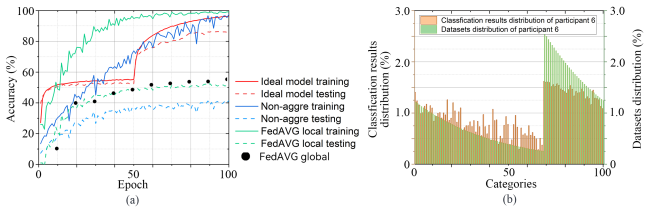

Fig. 1. (a): The training and testing accuracy of Ideal model, Non-aggre model and FedAVG [1]. The performance leaps after 50 epochs due to a smaller learning rate. (b): The bias of classification and datasets of Participant 6.

contains implicit association, which is helpful for learning data representation in deep learning. The independent performance (Non-aggre in Fig. 1 (a)) of each participant with the long-tailed data is weaker than the non-federated model. The classification results (Participant 6 in Fig. 1 (b)) of independent training by participants significantly lean towards dense categories. Researchers expect to achieve a balanced result where the categorization bias of the results is small. FedAVG [1] provides a high-performance global model by aggregating the local models of various participants. However, the performance of FedAVG [1] is still lower than the non-federated model. The above experimental results demonstrate the impact of local long-tailed data on federated learning, including the performance cracks and the bias of categorization results.

Recent work [1], [8], [9] is launched with a shared random initialization parameter to avoid the influence of outliers [1]. This setting leads to all participants' gradient decreases within a region, falling into the local optimal solution [10]. They consider the amount of data as contribution weights and repair the weak bias between local and global parameters to improve the performance of federated learning on Non-IID data. However, they do not further focus on the implications of long-tailed data, such as the performance cracks and the bias of categorization results. These methods cannot repair the huge deviation caused by long-tailed data. On the other hand, the single shared initialization parameter and fixed weights adjusted by humans may promote performance to shift towards a single direction and deviate from the global optimization. FedWolf is launched with a set of random initialization parameters to avoid local optimal solutions. We propose a dynamic adaptive weighting strategy to update parameters, which is robust in resisting outliers.

Some methods [11], [12] set a new optimization goal based on the difference between local and global parameters to reduce the bias among all participants. These methods [11], [12] rely on excellent initial parameters to ensure unbiased performance, which cannot be guaranteed in federated learning because of the individual behavior differences of participants. Some promising work [13], [14] uses the classical idea of non-federation to optimize the performance of federated learning on long-tailed distribution data, such as re-training classifiers [14] and logic adjustment [13]. These methods [13], [14] are effective, but the act of leaking intermediate products increases the risk of being attacked in federated learning. The dynamic adaptive weighting strategy in FedWolf does not require additional data transmission and strictly guarantees privacy and security.

In view of the privacy controversy of existing methods and the lack of effective methods to solve the performance crack and the bias of categorization results caused by the local long-tailed data in federated learning, we propose a novel federated learning algorithm with the Grey Wolf Optimizer [15] and Markov Chain [16], named FedWolf. Federated learning is a distributed computing technology, which provides collaborative training with multiparty participants. In this regard, we expect to use the biomimetic swarm algorithms to optimize performance and stay away from the local optima. FedWolf proposes a dynamic weighted parameter update strategy based on the Grey Wolf Optimizer to solve the impact of local long-tailed data on federated learning. Dynamic weights guide the gradient update towards a better direction, resulting in better performance compared to fixed weights. The weights are adaptively updated in FedWolf, because they are determined by the performance of participants in each communication round, rather than being manually controlled. The dynamic adaptive weights avoids performance bias towards dense categories or artificially selected weights of participants. After all communication rounds are completed, the parameter updating is modelled as a Markov Process [16]. The Markov state is used to infer the future performance of participants as contribution weights. FedWolf obtains the global model through aggregation with contribution weights. The aggregated contribution weight is also dynamic, which is determined by the performance of each participant in all communication rounds rather than fixed dataset weights. The model aggregation in FedWolf is robust in resisting outliers and malicious users, because the dynamic weights of outliers and malicious users are small and 0, respectively. The main contributions of this work are as follows:

(1) Instead of following FedAVG-related methods that start from a single shared parameter, we advocate launching federated learning with a set of random initialization parameters to keep away from local optimization. The statistical convergence of the algorithm is theoretically proved.
(2) We design a dynamic adaptive parameter update strategy based on the Grey Wolf Optimizer algorithm to solve the problem of falling into the dense head categories in local training.
(3) The parameter updating is modelled as the Markov Process. We propose a model aggregation strategy via the Markov Chain, which fully considers historical and future contributions. Thus, the aggregated global model is superior to the local model.
(4) We introduce the Gini index which is a classical theory in economics to evaluate the bias of classification results. The experimental evaluation validates the effectiveness of FedWolf in solving the performance cracks and the bias of categorization results. We also conduct empirical research to demonstrate the robustness of FedWolf in resisting outliers and malicious users.

2 Related Work

In centralized learning, the advanced methods increase the weight of sparse samples by re-sampling [17], re-weighting [18] or re-training classifier [19] to restore balance. These methods balance the distribution of head and tail data to improve the performance of deep learning, because the distribution of centralized data is easy to statistic. However, these methods are not suitable for the compulsory privacy protection strategy in federated learning because they collect data or intermediate feature.

2.1 Federated Learning on Non-IID Data

At present, a large number of promising methods have been proposed to optimize federated learning on Non-IID data. These methods are discussed as the available solutions of the local long-tailed data. Wang et al. [20] provided a generic framework based on the first principled understanding of the solution bias and the convergence slowdown due to objective inconsistency. The federated proximal algorithm (FedProx) [8] proposes a proximal term to describe the statistical heterogeneity of data and the computational heterogeneity of equipments. Karimireddy et al. [9] proposed a new stochastic controlled averaging algorithm, named SCAFFOLD. SCAFFOLD controls the reduction of variance to correct for the bias of parameters in local training. Durmus et al. [12] proposed FedDyn that updates the dynamic regularization for each participant at each round, so that the global and local model solutions are aligned. Gao et al. [11] proposed a federated algorithm with local drift decoupling and correction (FedDC). FedDC introduces a lightweight modification in training, in which each participant utilizes an auxiliary local drift variable to track the gap between the local model parameter and the global model parameter. Uddin et al. [21] proposed a disentangled information bottleneck principle-based loss function for local parameter update and suggested a model selection strategy based on the mutual information for global model aggregation.

The above methods assume that the performance drift is generated by the offset in local training and aggregation, and the proximal term is used to bring performance back on track. However, the performance of local training falls into the dense head categories, and the multi-directional pulling force makes the global model lost in the solution space instead of weak drift in the case of long-tailed data. Besides, the proximal term proposed by the above methods cannot fix the performance cracks caused by local long-tailed data.

2.2 Federated Learning on Long-Tailed Data

Some recent work [13], [14] focus on the impact of global long-tailed data on federated learning performance, which assumes the total amount of datasets for all participants follows the long-tailed distribution. Shang et al. [14] proposed a federated learning via classifier re-training with federated feature (CReFF). CReFF recommends the classifier should be re-trained with federated features to eliminate the negative influence of global long-tailed data and achieve performance comparable to real data training. Shange et al. [13] proposed a federated ensemble distillation with imbalance calibration (FEDIC), which uses the logit adjustment and calibration gating network techniques to effectively make the output of the ensemble model unbiased on global long-tailed data. The above two methods [13], [14] propose effective solutions to improve the performance on global long-tailed data, but they do not specify the global bias caused by local long-tailed data. Furthermore, additional data exchange leads to a decrease in communication efficiency and an increase in the risk of privacy leakage. Shi et al. [35] use a contrastive language-image pre-training model to optimize the federated learning between server and client models under its vision-language supervision. Xiao et al. [36] proposed a method termed Fed-GraB, comprised of a self-adjusting gradient balancer module that re-weights clients' gradients in a closed-loop manner based on the feedback of global long-tailed prior derived from a direct prior analyzer module.

Zhang et al. [5] calibrated the logit before softmax cross-entropy according to the probability of occurrence of each class to improve the federated performance of highly skewed data. Lu et al. [22] proposed a federated learning method with adversarial feature augmentation (FedAFA), which optimizes the local model for each participant by producing a balanced feature set and enhances the local minority classes with adversarial feature augmentation. The above methods alleviate the performance degradation caused by the long-tailed data. However, they ignore the impact of local long-tailed distribution data on the unbiased results. The augmentation of minority categories in local data is limited in improving the global performance and decreasing the bias of performance on local long-tailed data.

3 Dynamic Adaptive Federated Learning

The main notations involved in FedWolf are predefined, as shown in Table 1.

3.1 Overview

Since FedAVG [1] was proposed by Google as a benchmark in federated learning, FedAVG-related methods [8], [9], [11], [12] follow a unified architecture in which federated learning is described as an iterative cycle of training in Fig. 2. This architecture of FedAVG-related methods can be described as follows:

(1) The server distributes initialization parameters θ_k^0 to participant k.
(2) In communication round r, participants independently train with the Stochastic Gradient Descent (SGD) and upload parameters $\theta_k^{r,e}$ to the server.
(3) The server updates new parameters $\theta_k^{r,*}$ and distributes it to participant k. Then, loop step (1)-(3) until convergence occurs.
(4) After all communication rounds are completed, the final global model θ_{global} is obtained by weighted aggregation.

The updating and aggregating of parameters are the part of unified architecture. The key difference between recent work is the weight that considers different factors, for example, the size of datasets [11], and participants' reputation [37], [38]. As shown in Fig. 2, the modifications of FedWolf focus on Step (1), (3) and (4). A set of random parameters is

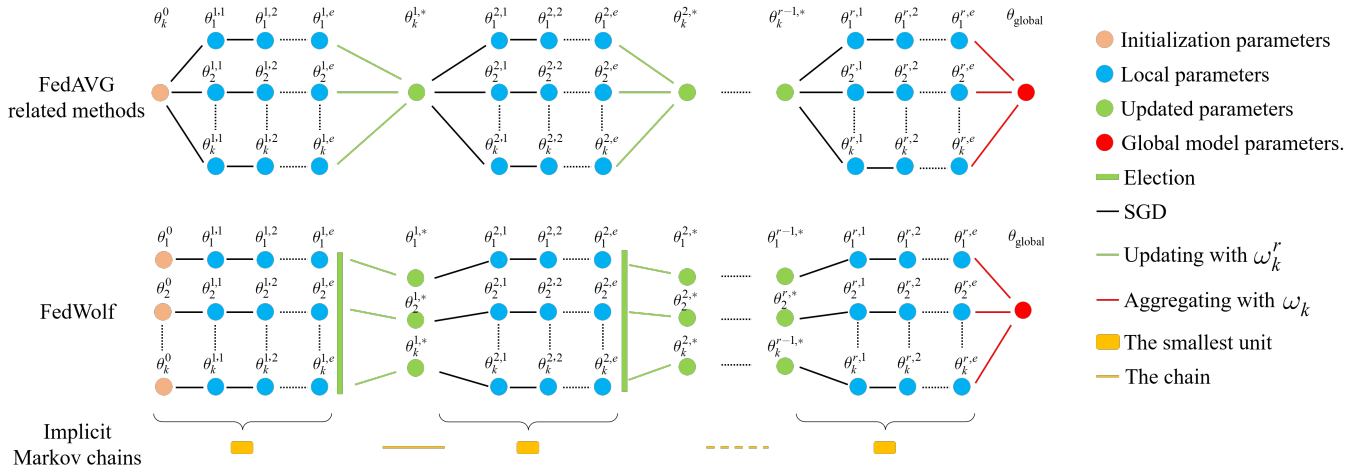

Fig. 2. Comparison between FedWolf and FedAVG-related methods [8], [9], [11], [12] in parameters updating and aggregation. Our main ideas are scattered around the initialization parameters (orange), parameters updating (green) and parameters aggregation (red).

TABLE 1
The main notations involved in FedWolf

Notations	Description
$R = \{r \mid r \in \mathbb{N}^+\}$	The set of communication rounds. $\|R\|$ is the total number of rounds and $r \in R$ is the rth communication round.
$E = \{e \mid e \in \mathbb{N}^+\}$	The set of epochs. $\|E\|$ is the total number of epochs in each communication round and $e \in E$ is the eth epoch.
$K = \{k \mid k \in \mathbb{N}^+\}$	The set of participants. $\|K\|$ is the total number of participants and $k \in K$ is the kth participant.
$L = \{1, 2, 3\}$	Markov state space of participant level.
θ_k^0	Initialization parameters of participant k.
$\theta_k^{r,e}$	Parameters of participant k in epoch e of communication round r.
$\theta_k^{r,*}$	Updated parameters of participant k after communication round r.
θ_{global}	Global model parameters.
$l_k^r \in L$	Level of participant k in round r.
$\mathbf{M}_{\|K\|\times\|R\|}$	Markov state matrix of all participants after all communication round.
$\mathbf{M}_k^{\text{tran}}$	Markov state transition matrix of participant k.
$\vec{\mathbf{m}}^r \in \mathbf{M}_{\|K\|\times\|R\|}$	State column vector of all participants in communication round r.
$\vec{\mathbf{m}}_k \in \mathbf{M}_{\|K\|\times\|R\|}$	State row vector of participant k after all communication rounds.
ω_k^r	Updated weight of participant k in round r.
ω_k	Contribution weight of participant k.
r^*	Communication round for inferring future performance in computing the contribution weight ω_k.

used as the initialization parameters θ_k^0, to keep away from the local optimization. A dynamic adaptive parameter update strategy via the Grey Wolf Optimizer [15] is proposed to solve the problem of falling into the dense head categories on the local long-tailed data. FedWolf proposes a model-weighted aggregation strategy via the Markov Chain [16], which fully considers historical and future contributions to improving the performance. This aggregation strategy with dynamic weights is believed to be more fair and effective than existing methods with fixed weights.

Specifically, we launch FedWolf with a set of initialization parameters instead of all participants downloading the same one. Then the trained parameters $\theta_k^{r,e}$ are uploaded to the server. Assume the participant k is elected according to its highest macro F1-score [23] and divided into participant level l_k^r, where Level 1 participant has better performance than Levels 2 and 3 participants. Participants hold different participant levels in different communication rounds ($l_k^1 \neq l_k^2 \neq \cdots \neq l_k^r$).

Inspired by the Grey Wolf Optimizer algorithm [15], we design a parameter update rule based on the participant level l_k^r. The updated parameters $\theta_k^{r,*}$ are distributed to all participants and the next local training is launched. Steps (1) and (2) are considered as the smallest unit that generates a set of participant level $\vec{\mathbf{m}}^r$ in the communication round r. Step (3) is a chain that links two units, as shown in Fig. 2. The participant level l_k^r is considered as the Markov state ans steps (1)-(3) can be modelled as a Markov Process [16] in FedWolf. After all communication rounds, the Markov transition matrix $\mathbf{M}_k^{\text{tran}}$ of participant k is calculated from the participant level $\vec{\mathbf{m}}_k$ to infer the participant level in future training rounds. Finally, we select the probability of Level 1 in the inferred participant level status as its contribution weight ω_k to aggregate the parameters of the global model θ_{global}.

We present the algorithm of FedWolf in Algorithm1. Four aspects distinguishing FedWolf and the existing methods are highlighted below.

(1) FedWolf is launched with a set of random initialization parameters ($\theta_1^0 \neq \theta_2^0 \neq \cdots \neq \theta_k^0$) instead of a shared parameter ($\theta_1^0 = \theta_2^0 = \cdots = \theta_k^0$ deployed by FedAVG-related methods) to avoid the local optimization.

(2) In each communication round r, the participant k is selected according to its highest macro F1-score [23] and divided into participant level $l_k^r \in L = \{1, 2, 3\}$ ($l_k^1 \neq l_k^2 \neq \cdots \neq l_k^r$).

(3) After each communication round r, the parameter

Algorithm 1 Pipeline of FedWolf.

Input: R, E, K, θ_k^0.
Output: θ_{global}.
for communication round r **do**
  for participant k **do**
    for local training epoch e **do**
      if $r == 1$ and $e == 1$ **then**
        $\theta_k^{r,e}$ = Local training with SGD and θ_k^0;
      else
        $\theta_k^{r,e}$ = Local training with SGD and $\theta_k^{r,e-1}$;
      end if
    end for
  end for
  Parameter updating (Algorithm 2);
end for
Computing the state transition matrix $\mathbf{M}_k^{\text{tran}}$ of participant k (Algorithm 3);
Computing the contribution weight ω_k with $\mathbf{M}_k^{\text{tran}}$ of participant k (Algorithm 4);
θ_{global} = Aggregating global model with contribution weight ω_k;
return θ_{global}

update strategy generates dynamic update weight ω_k^r for each participant ($\omega_k^1 \neq \omega_k^2 \neq \cdots \neq \omega_k^r$ and $\omega_1^r \neq \omega_2^r \neq \cdots \neq \omega_k^r$) based on the participant level l_k^r to reduce the bias caused by fixed update weight ($\omega_k^1 = \omega_k^2 = \cdots = \omega_k^r$ in FedAVG-related methods). The parameters are adaptively and independently updated ($\theta_1^{r,*} \neq \theta_2^{r,*} \neq \cdots \neq \theta_k^{r,*}$) rather than sharing the same parameters ($\theta_1^{r,*} = \theta_2^{r,*} = \cdots = \theta_k^{r,*}$ in FedAVG-related methods) to solve the problem of falling into the dense head categories in local training.

(4) The probability of each participant is elected as the Level 1 in the inferred future performance is used as contribution weight ω_k rather than the historical performance or the number of datasets, which is also a dynamic weight ($\omega_k \neq \omega_k^r$ in FedWolf and $\omega_k = \omega_k^r$ in FedAVG-related methods).

3.2 Parameter Update via Grey Wolf Optimizer

The core of Grey Wolf Optimizer is the dynamic selection of multi-level wolf and the adaptive search of solution space based on the wolf level, which helps the algorithm to quickly search for solution space and keep away from the local optimization [15]. Our motivation is that participants escape the local optimization caused by the local long-tailed data through a set of random initialization parameters and the Grey Wolf Optimizer [15]. We design an election strategy to rank participant level L (multi-level wolf) and an adaptive parameter update algorithm based on the participant level L (wolf level). There are 3 participant levels. Level 1 participant is a participant with the best accuracy to represent the global optimal solution. Level 2 participants are some participants with better accuracy in representing the local optimal solution. Level 3 participants are some participants with poor accuracy. Different update weights ($\omega_1^r \neq \omega_2^r \neq \cdots \neq \omega_k^r$) are used for the participants with different levels during parameter updates.

Specifically, the parameters $\theta_k^{r,e}$ of the participant k are uploaded to the server after communication round r. The server calculates the macro F1-score [23] of the participant k as the basis for electing and dividing level $l_k^r \in L$. l_k^r is recorded in $\vec{\mathbf{m}}^r = [l_1^r, l_2^r, \cdots, l_k^r]^\top$. After the election is completed, 3 sets $P_1 = \{\theta_k^{r,e} | l_k^r = 1\}$, $P_2 = \{\theta_k^{r,e} | l_k^r = 2\}$, $P_3 = \{\theta_k^{r,e} | l_k^r = 3\}$ are obtained, which contains the parameters $\theta_k^{r,e}$ of the participant k different level l_k^r.

The parameters are updated based on the participant level l_k^r. The leader ($l_k^r = 1$) holds his own parameter. Low-level participants ($l_k^r = 2, 3$) share parameters among peers and benefit from the parameters of leader. This is a dynamic weight update strategy. As the election results of each communication round change, the weight ω_k^r held by the participant k needs to be updated. FedWolf effectively avoids the global performance and the bias of classification results towards the heavy-weight participants. The new parameters are sent to the participants to start the next round of local training. We present the parameter update algorithm in Algorithm 2.

Algorithm 2 Parameter update algorithm.

Input: $\theta_k^{r,e}$, $\vec{\mathbf{m}}^r$.
Output: $\theta_k^{r,*}$.
Level1temp = $\theta_k^{r,e} \in P_1$;
Level2temp = MEAN ($\theta_k^{r,e} \in P_2$);
Level3temp = MEAN ($\theta_k^{r,e} \in P_3$);
for $l_k^r \in \vec{\mathbf{m}}^r$ and $\theta_k^{r,e}$ of participant k **do**
  if l_k^r == Level 1 **then**
    $\theta_k^{r,*} = \theta_k^{r,e}$;
  else if l_k^r == Level 2 **then**
    $\theta_k^{r,*}$ = ($\theta_k^{r,e}$ + Level1temp)/2;
  else if l_k^r == Level 3 **then**
    $\theta_k^{r,*}$ = ($\theta_k^{r,e}$ + Level2temp + Level3temp)/3;
  end if
end for
return $\theta_k^{r,*}$

3.3 Model Aggregation via Markov Chain

The process of participant level state change in Section 3.2 is modelled as a Markov Process [16] to infer the contribution weight ω_k of each participant, because of the independence of participant level states and the chain structure of FedWolf as described in Fig. 2. The advantages of the contribution weight ω_k in FedWolf are highlighted below.

(1) The state probability is consistent with the normalized weight, namely, $\omega_k \in [0, 1]$, and $\sum_{k=1}^{|K|} \omega_k = 1$.
(2) The future state probabilities are predicted from historical states, which fully consider the long-term performance of each participant.

Specifically, the participant level L is the Markov state space. After each training unit, a state sequence $\vec{\mathbf{m}}^r = [l_1^r, l_2^r, \cdots, l_k^r]^\top$ is generated and appended to the Markov state matrix.

After all communication rounds, the status sequence $\vec{\mathbf{m}}_k = [l_k^1, l_k^2, \cdots, l_k^r]$ of participant k can be extracted from

Algorithm 3 Computing the state transition matrix $\mathbf{M}_k^{\text{tran}}$.

Input: $\vec{\mathbf{m}}_k$.
Output: $\mathbf{M}_k^{\text{tran}}$.
  $\mathbf{M}_k^{\text{tran}} = \mathcal{O}_{3\times 3}$;
  for (l_k^r, l_k^{r+1}) in $\vec{\mathbf{m}}_k$ **do**
    $\mathbf{M}_k^{\text{tran}}[l_k^r][l_k^{r+1}]$ += 1;
  end for
  for row in $\mathbf{M}_k^{\text{tran}}$ **do**
    S = SUM(row);
    if $S > 0$ **then**
      row[:] = [F/S for F in row];
    else if then
      row[:] = fulfil with 0;
    end if
  end for
  return $\mathbf{M}_k^{\text{tran}}$

Algorithm 4 Computing the contribution weight ω_k.

Input: $\mathbf{M}_k^{\text{tran}}$, r^*.
Output: ω_k.
  Random initial state s;
  while $r^* > 0$ **do**
    $s = s * \mathbf{M}_k^{\text{tran}}$;
    $r^* --$;
  end while
  $\omega_k = s[0]$;
  return ω_k

the Markov state matrix $\mathbf{M}_{|K|\times|R|}$. The state transition matrix $\mathbf{M}_k^{\text{tran}}$ of participant k is calculated with the state matrix $\vec{\mathbf{m}}_k$ (Algorithm 3). Given a random initial state s, the state of participant k in future communication rounds r^* is inferred with the state transition matrix $\mathbf{M}_k^{\text{tran}}$ (Algorithm 4). We select the probability that participants become Level 1 as their contribution weights ω_k and obtain the final global model θ_{global} through weighted aggregation. Participant holds different contribution weight ω_k ($\omega_1 \neq \omega_2 \neq \cdots \neq \omega_k$), which is dynamically determined based on the performance of participant k during the whole training process, to avoid the performance bias towards the dictator.

4 Algorithm Convergence Analysis

FedWolf is launched with a set of random initialization parameters and the updated parameters are independent of the training process. To prove that the convergence of FedWolf is not compromised, the statistical convergence is given.

Problem formulation. FedWolf aims to optimize the following distributed objective function.

$$F = \text{minimize} \sum_{k=1}^{|K|} f_k(g(X, \theta_k^{r,e}), Y), \tag{1}$$

where X, Y are datasets and labels, separately. $g(X, \theta_k^{r,e})$ predicts the maximum category probability of X with $\theta_k^{r,e}$. $f_k(\cdot)$ is the loss function of the participant k. We provide a statistical convergence proof of each participant $f_k(\cdot)$, thus obtaining the proof of global F convergence.

Algorithm description. Here, we define the notation $T = \{t | t \in \mathbb{N}^+\}$, $t = r \times e + r$, $|T| = |R| \times |E| + |R|$ to describe the time of parameters change. θ_k^t is the parameter of participant k in parameters change t, including $\theta_k^{r,e}$ and θ_{global}. ω_k^t is the weight of the participant k in the parameters change t, including $\omega_k^{r,e}$ and ω_k. θ_k^* is the optimal solution. Each participant starts local training with random initialization parameters and enters the next round with the independent updated parameters in FedWolf. The process of parameter updating is described as follows.

$$\theta_k^{t+1} = \begin{cases} \theta_k^t - \eta_k^t g_k^t, & \text{local training with SGD.} \\ \sum_{k=1}^{|K|} \omega_k^t \theta_k^t, & \text{updating with weight } \omega_k^t. \end{cases} \tag{2}$$

where η_k^t is the learning rate and g_k^t=$\nabla f_k(\cdot)$ is the gradient of the participant k in time t.

Assumption. We make the following standard assumption [8], [9] on the loss function $f_k(\cdot)$. These assumptions are set in the optimization problem. They ensure that the loss decays after SGD and the amount of attenuation decreases.

(1) $f_k(\cdot)$ is convex: for all θ_k^t, $f_k(\theta_k^t) \geq f_k(\theta_k^{t-1}) + \langle \nabla f_k(\theta_k^{t-1}), \theta_k^t - \theta_k^{t-1} \rangle$.
(2) Variables bounded: $\|\theta_k^t - \theta_k^*\|_2 \leq V, \forall \theta_k^t, \theta_k^*$.
(3) Gradient bounded: $\|g_k^t\|_2 \leq G, \forall t$.

Theorem 1 (Convergence of each participant). For convex loss function $f_k(\cdot)$.

$$\begin{aligned} R(T) &= \sum_{t=1}^{|T|} \omega_k^t f_k(\theta_k^t) - \min \sum_{t=1}^{|T|} f_k(\theta_k^t) \\ &\leq \frac{V^2}{2\eta_k^t} + \frac{G^2}{2} \sum_{t=1}^{|T|} \eta_k^t \end{aligned} \tag{3}$$

Based on Assumption (2) and (3), V and G decay with T increases, so

$$\lim_{T\to\infty} \frac{R(T)}{T} = 0.$$

And the loss $\sum_{t=1}^{|T|} \omega_k^t f_k(\theta_k^t)$ is equal to the minimum loss $\min \sum_{t=1}^{|T|} f_k(\theta_k^t)$, which means the loss function $f_k(\cdot)$ converges.

Proof.
Expand function3.

$$\begin{aligned} R(T) &= \sum_{t=1}^{|T|} \omega_k^t f_k(\theta_k^t) - \min \sum_{t=1}^{|T|} f_k(\theta_k^t) \\ &= \sum_{t=1}^{|T|} \underbrace{[\omega_k^t f_k(\theta_k^t) - f_k(\theta_k^*)]}_{A}. \end{aligned} \tag{4}$$

We need to determine the upper bound of the term A in function 4. According to Assumption (1), we have the following results.

$$f_k(\theta_k^*) - f_k(\theta_k^t) \leq \langle g_k^t, \theta_k^* - \theta_k^t \rangle. \tag{5}$$

$\omega_k^t \in [0, 1]$ because the update weight $\omega_k^{r,e}$ satisfies $\omega_k^{r,e} \in (0, 1]$ and the contribution weight ω_k satisfies

$\omega_k \in [0,1]$, respectively, as described in Section 3.2 and Section 3.3.

$$\therefore \omega_k^t f_k(\theta_k^t) - f_k(\theta_k^*) \leq \langle g_k^t, \theta_k^t - \theta_k^* \rangle. \quad (6)$$

According to function 6, it can be concluded that the term A in function 4 has an upper bound.

$$\begin{aligned} R(T) &= \sum_{t=1}^{|T|} [\omega_k^t f_k(\theta_k^t) - f_k(\theta_k^*)] \\ &\leq \sum_{t=1}^{|T|} \underbrace{\langle g_k^t, \theta_k^t - \theta_k^* \rangle}_{B}. \end{aligned} \quad (7)$$

Furthermore, we verify that the term B in function 7 has an upper bound.

$$\begin{aligned} \because \theta_k^{t+1} &= \theta_k^t - \eta_k^t g_k^t \\ \therefore \|\theta_k^{t+1} - \theta_k^*\|_2^2 &= \|\theta_k^t - \theta_k^* - \eta_k^t g_k^t\|_2^2 \\ \therefore \|\theta_k^{t+1} - \theta_k^*\|_2^2 &= \|\theta_k^t - \theta_k^*\|_2^2 - 2\eta_k^t \langle g_k^t, \theta_k^t - \theta_k^* \rangle \\ &\quad + (\eta_k^t) \|g_k^t\|_2^2 \\ \therefore \langle g_k^t, \theta_k^t - \theta_k^* \rangle &= \frac{1}{2\eta_k^t} [\|\theta_k^t - \theta_k^*\|_2^2 - \|\theta_k^{t+1} - \theta_k^*\|_2^2] \\ &\quad + \frac{\eta_k^t}{2} \|g_k^t\|_2^2. \end{aligned} \quad (8)$$

Combining function 7 and 8, the upper bound of function 4 is determined.

$$\begin{aligned} R(T) \leq & \underbrace{\sum_{t=1}^{|T|} (\frac{1}{2\eta_k^t} [\|\theta_k^t - \theta_k^*\|_2^2 - \|\theta_k^{t+1} - \theta_k^*\|_2^2])}_{C} \\ & + \underbrace{\sum_{t=1}^{|T|} (\frac{\eta_k^t}{2} \|g_k^t\|_2^2)}_{D}. \end{aligned} \quad (9)$$

Then, we simplify the terms C and D in function 9.

$$\begin{aligned} C &= \sum_{t=1}^{|T|} (\frac{1}{2\eta_k^t} [\|\theta_k^t - \theta_k^*\|_2^2 - \|\theta_k^{t+1} - \theta_k^*\|_2^2]) \\ &= \frac{1}{2\eta_k^1} \|\theta_k^1 - \theta_k^*\|_2^2 + \sum_{t=2}^{|T|} (\frac{1}{2\eta_k^t} - \frac{1}{2\eta_k^{t-1}}) \|\theta_k^t - \theta_k^*\|_2^2 \\ &\quad - \frac{1}{2\eta_k^t} \|\theta_k^{t+1} - \theta_k^*\|_2^2. \end{aligned} \quad (10)$$

The last item in function 10 has an upper bound of 0.

$$-\frac{1}{2\eta_k^t} \|\theta_k^{t+1} - \theta_k^*\|_2^2 \leq 0. \quad (11)$$

The term C is simplified based on Assumption (2) and function 11.

$$\begin{aligned} C &= \frac{1}{2\eta_k^1} \|\theta_k^1 - \theta_k^*\|_2^2 + \sum_{t=2}^{|T|} (\frac{1}{2\eta_k^t} - \frac{1}{2\eta_k^{t-1}}) \|\theta_k^t - \theta_k^*\|_2^2 \\ &\quad - \frac{1}{2\eta_k^t} \|\theta_k^{t+1} - \theta_k^*\|_2^2 \\ &= \frac{V^2}{2\eta_k^t}. \end{aligned} \quad (12)$$

The upper bound of the term D is determined based on Assumption (3).

$$D = \sum_{t=1}^{|T|} \frac{\eta_k^t}{2} \|g_k^t\|_2^2 \leq \frac{G^2}{2} \sum_{t=1}^{|T|} \eta_k^t. \quad (13)$$

By substituting the items C and D into function 9, we can obtain the following results.

$$R(T) \leq \frac{V^2}{2\eta_k^t} + \frac{G^2}{2} \sum_{t=1}^{|T|} \eta_k^t. \quad (14)$$

In summary, we complete the proof of Theorem 1. The loss function of the participant k converges to the bound, and then F in function 1 is convergent.

5 Experiments and Analysis

5.1 Experimental Setting

To investigate the impact of extreme imbalance and a large number of labels with the local long-tailed distribution, we conduct experiments on the CIFAR100 [7] and ImageNet [24] datasets with the extreme imbalance factors (IF). Let c_i be the number of datasets with the i th category, $C = \max(c_1, c_2, \cdots, c_i)$ and $c = \min(c_1, c_2, \cdots, c_i)$ be the maximum and minimum of datasets $\{c_1, c_2, \cdots, c_i\}$ with all categories, respectively. IF calculates the ratio of head and tail category as follows:

$$IF = \frac{C}{c}. \quad (15)$$

Firstly, we calculate the number of data in each category with the local long-tailed distribution and IF=100. The maximum and minimum number of categories is 500 and 5 on CIFAR100. Secondly, the Dirichlet distribution [25] with the parameter $\alpha = 0.1$ is used to extract various categories images. This is a sampling with replacement, so there are some repetitions among the data of each participant. Then we randomly select the peak of head category for each participant and reassemble the data sequence. Finally, the local long-tailed data with inconsistent peaks are allocated to each participant. Besides, we partition ImageNet [24] into subsets with 300, 500, and 800 categories as the stage evaluation datasets.

ResNet-18 [26] and ResNet-34 [26] are used as the backbone network. The standards of ImageNet [24] classification competition are followed to set network hyperparameters [24], [26]. We launch 100 communication rounds and 5 local epochs in each round. There are 10 participants fully selected in each communication round. The batch size is set at 256 with a learning rate of 0.01 and SGD as the optimizer in local training. The learning rate is reset to 0.001 after half communication rounds. r^* is set at 100 in computing contribution weight.

5.2 Comparison and Analysis

We compare FedWolf with several advanced methods, including FedAVG [1], FedDC [11], FedDyn [12], FedProx [8], SCAFFOLD [9], FEDIC [13], CReFF [14], FedAFA [22], FedCLIP [35] and Fed-grab [36]. All the comparative methods are implemented with the experimental settings described

TABLE 2
Testing accuracy (%) achieved by comparison methods and FedWolf with different initialization parameters.

Methods	Single Parameter		Discrete Parameters		
	Normal	Uniform	Normal	Uniform	Normal +Uniform
FedAVG	52.13	51.57	66.29	66.02	66.66
FedDC	62.29	55.94	46.41	45.70	45.72
FedDyn	62.69	56.86	48.01	48.34	46.47
FedProx	52.31	51.12	64.12	64.50	64.52
SCAFFOLD	51.57	50.23	64.83	64.71	63.21
FEDIC	61.78	61.60	61.90	61.95	62.15
CReFF	63.94	64.02	64.12	63.97	64.25
FedAFA	60.23	60.33	60.56	61.50	60.49
FedCLIP	65.24	65.28	66.21	65.31	65.12
Fed-grab	66.12	65.48	67.33	66.34	65.98
FedWolf	**68.14**	**68.63**	**70.10**	**69.76**	**69.85**

in Section 5.1. FedAVG is a classic benchmark proposed by Google. FedDC, FedDyn, FedProx and SCAFFOLD are excellent representations of Non-IID data, which focus on minimizing bias in model parameters. FEDIC and CReFF are the promising developments in the global long-tailed distribution data. FedAFA is an outstanding personalized federated learning algorithm in dealing with the long-tailed data. FedCLIP and Fed-grab improve the performance of federated learning by text encoder and self-adjusting gradient balancer, respectively.

5.2.1 Comparison on Different Initialization Parameters

We advocate starting federated learning with a set of random initialization parameters to keep away from the local optimization. We implement the following experiments to discuss the practicality of the above idea. ResNet18 [26] is used as a backbone network.

All the methods are launched from a single parameter or some discrete random parameters with normal [28] or uniform [27] distribution respectively, to compare the differences in performance caused by different initialization parameters. To evaluate the impact of mixed parameters, we randomly selected 5 participants with normal parameters and 5 participants with uniform parameters to form the group with mixed parameters. The testing accuracy achieved by the compared methods and FedWolf is recorded in Table 2. Some important observations are presented from the experimental results.

(1) FedWolf achieves better performance than the comparison methods with different initialization parameters.
(2) The performance achieved by most methods [1], [8], [9], [13], [14], [22] with discrete parameters (columns 4, 5 in Table 2) is superior to the case with a single shared parameter (columns 2, 3 in Table 2). The performance of FedDC [11] and FedDyn [12] with discrete initialization parameters decreases, because they are designed to improve the setting of a single initialization parameter and minimize the bias between updated parameters and initial parameters, which is disrupted by discrete initial parameters.
(3) The testing accuracy achieved by FedWolf with normal discrete initialization parameters is 70.10% which is 1.96% higher than the case with normal single initialization parameters.
(4) The mixed parameters lead to a decrease in the performance of some methods [9], [22], and the performance of FedWolf is reduced by 0.25% compared to the case with normal initialization parameters.

In federated learning, an important prerequisite for achieving the best performance of the global model is that all participants are rational and harmless. The rational participants tend to configure reasonable initialization parameters to help federated learning achieve excellent global performance, for example, normal distribution [28] and pre-trained parameters. Reasonable discrete parameters are beneficial for improving the performance of federated learning. But unrestricted initialization parameters increase the risk of outliers and malicious users. Some inappropriate initialization parameters reduce learning potential and even damage the performance of federated learning. We report these negative performances caused by inappropriate initialization parameters and demonstrate the robustness of FedWolf in defensing outliers and malicious users in sections 5.4 and 5.5.

5.2.2 Comparison on Local Long-tailed Data

Table 3 reports the global model performance of the comparative methods and FedWolf on the local long-tailed data. FedWolf achieves better performance on the multi-stage datasets and different backbone networks than the comparative methods. Specifically, FedWolf achieves 17.97% performance improvement than FedAVG [1] and 6.16% performance improvement than CReFF [14] on CIFAR100 and ResNet18. Compared to the promising methods, such as FedDC [11], CReFF [14] and FedAFA [22], FedWolf improves the accuracy by 3.85%, 5.69% and 4.07% on ImageNet1000 and ResNet34, respectively. Compared to the state-of-the-art methods, such as FedCLIP [35] and Fed-grab [36], FedWolf improve the accuracy by 4.86% 3.98% on CIFAR100 and ResNet18. The above experimental results are favorable evidence that FedWolf is effective in solving the performance cracks caused by the local long-tailed data. In addition, we provide some important experimental observations as follows:

(1) ImageNet provides larger size of images than CIFAR100, which helps to improve performance. FedWolf achieves 70.10% and 75.45% testing accuracy on CIFAR100 and ImageNet300.
(2) ImageNet is also more suitable for training with the deeper network (ResNet34) compared to CIFAR100, because the larger image provides more feature information. FedWolf achieves 70.10% and 60.44% testing accuracy by ResNet18 and ResNet34 on CIFAR100, respectively.
(3) As the number of categories and images increases (from ImageNet300 to ImageNet1000), the performance of all the methods drops to a valley, which is an important observation of the performance cracks caused by the local long-tailed distribution data.
(4) The deeper network (ResNet34) provides more robust performance on imbalanced data than ResNet18. The rate of performance degradation

TABLE 3
Testing accuracy (%) achieved by comparative methods and FedWolf on different datasets with the local long-tailed distribution.

Network	Methods	CIFAR100	ImageNet300	ImageNet500	ImageNet800	ImageNet1000
ResNet18	FedAVG	52.13	53.01	50.91	46.01	38.33
	FedDC	62.29	70.30	68.28	56.97	49.64
	FedDyn	62.69	70.45	67.76	54.39	48.23
	FedProx	52.31	57.81	55.10	51.09	45.21
	SCAFFOLD	51.57	56.92	54.39	50.12	44.39
	FEDIC	61.78	64.33	59.98	53.01	48.83
	CReFF	63.94	70.88	66.15	53.98	49.90
	FedAFA	60.23	64.39	65.10	52.19	47.98
	FedCLIP	65.24	68.21	67.98	61.28	54.36
	Fed-grab	66.12	68.73	68.21	60.81	55.41
	FedWolf	**70.10**	**75.45**	**73.88**	**68.80**	**56.20**
ResNet34	FedAVG	53.21	56.98	54.31	52.12	50.90
	FedDC	52.14	73.02	69.12	68.83	68.13
	FedDyn	50.72	72.86	70.09	68.34	67.99
	FedProx	50.98	60.81	60.13	57.89	55.29
	SCAFFOLD	54.89	59.93	56.98	54.14	52.98
	FEDIC	51.29	66.38	63.94	59.87	58.84
	CReFF	52.33	71.29	69.21	67.93	66.29
	FedAFA	51.29	70.98	69.33	69.12	67.91
	FedCLIP	53.21	74.33	72.89	71.14	69.88
	Fed-grab	53.78	75.80	73.19	71.23	68.51
	FedWolf	**60.44**	**77.01**	**74.12**	**72.11**	**71.98**

TABLE 4
The Gini index and IF of classification results of comparative methods and FedWolf on different datasets with the local long-tailed distribution.

Methods	Gini		IF	
	CIFAR 100	ImageNet 1000	CIFAR 100	ImageNet 1000
FedAVG	0.20	0.32	12.28	23.10
FedDC	0.12	0.21	2.81	12.97
FedDyn	0.14	0.22	4.83	15.03
FedProx	0.19	0.32	10.75	24.89
SCAFFOLD	0.20	0.33	17.80	27.50
FEDIC	0.20	0.31	17.30	27.33
CReFF	0.12	0.28	15.03	24.00
FedAFA	0.13	0.28	8.18	20.01
FedCLIP	0.12	0.21	14.31	20.88
Fed-grab	0.13	0.20	8.72	11.03
FedWolf	**0.10**	**0.18**	**2.31**	**10.10**

on ResNet34 is slower than ResNet18 (from ImageNet300 to ImageNet1000).

The above experimental results and observations prove that FedWolf can solve the performance cracks caused by the local long-tailed data. Another aspect of the local optimization caused by the local long-tailed distribution data is the imbalanced classification results. IF is not competent in evaluating the balance of classification results, because it calculates the ratio of head and tail category to evaluate the imbalance of distribution, without considering the fluctuations in the middle category (as shown in Equation 15). There are obvious peaks at the head and tail but tend to flatten in the middle in the distribution of classification results, which causes unfair evaluation results. So, we introduce the Gini index [29] which is a classical theory in economics to objectively evaluate the bias of classification results. The Gini index is used as the basis for selecting and splitting features in decision trees [34]. Let r_i be the correct quantity of the i th category in classification results, the Gini index calculates the cumulative shift of all categories $\{r_1, r_2, \cdots, r_i\}$ to evaluate the bias of classification results as follows:

$$Gini = \frac{\frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}|r_i - r_j|}{\frac{1}{n}\sum_{i=1}^{n} r_i}, \tag{16}$$

where $\sum_{i=1}^{n}\sum_{j=1}^{n}|r_i - r_j|$ calculates the sum of offsets for all category pairs r_i and r_j in classification results. From function 15 and 16, it can be seen that the smaller the values of IF and Gini, the more balanced the results are. The Gini index is a more effective and fairer evaluation than IF to avoid erroneous results because it considers the balance of middle categories. We count the classification results of all the compared methods and use an approximate solution method [30] to calculate their Gini index. The Gini index and IF of classification results of compared methods and FedWolf are shown in Table 4 to discuss the effectiveness and fairness of the Gini index.

The imbalance of classification results is amplified with IF, such as the difference of 7.05 in IF between FedProx [8] and SCAFFOLD [9]. However, an important experimental observation is that the mid-range category distribution in classification results of FedProx [8] and SCAFFOLD [9] is flat, and IF cannot fairly evaluate this distribution of results. The difference in Gini index between FedProx [8] and SCAFFOLD [9] is only 0.01, which is consistent with our experimental observations and shows the Gini index is fairer

than IF. As the number of categories and images increases, the imbalance of classification results of all the methods increases, which also proves the damage caused by local long-tailed data to federated learning. FedWolf achieves 0.10 and 0.18 at the Gini index, respectively, which is a more balanced performance compared to the comparative methods on CIFAR100 and ImageNet1000.

5.2.3 Comparison on Uniform Data

We also evaluate the performance of comparative methods and FedWolf on the dataset with uniform distribution to explore the potential of FedWolf as a benchmark for updating and aggregating parameters in federated learning. All the participants use ResNet18 [26] as the backbone network to train local models on an independent subset of datasets. The comparison results are shown in Table 5.

The rise of performance achieved by all comparative methods on uniform data is firm evidence that proves the impact of local long-tailed data on federated learning. FedWolf achieves 75.33% testing accuracy on CIFAR100 [7] with uniform distribution, which is 4.46% higher than the advanced methods such as FedDC [11]. FedWolf achieves superior performance on all datasets, which shows the potential of FedWolf as a benchmark for updating and aggregating parameters in federated learning.

TABLE 5
Testing accuracy (%) achieved by compared methods and FedWolf on different datasets with the uniform distribution.

Methods	CIFAR 100	ImageNet 300	ImageNet 500	ImageNet 800	ImageNet 1000
FedAVG	56.98	58.21	55.94	53.43	50.79
FedDC	70.87	72.49	70.22	67.91	64.30
FedDyn	67.99	71.83	70.01	68.89	66.12
FedProx	60.21	63.01	61.09	59.82	55.98
SCAFFOLD	60.34	62.48	57.98	56.63	55.03
FEDIC	64.30	66.08	63.07	60.66	58.97
CReFF	65.68	71.98	68.90	66.31	64.09
FedAFA	65.79	68.92	67.59	65.87	62.93
FedCLIP	68.23	72.30	68.82	66.49	65.09
Fed-grab	69.97	72.81	68.77	67.18	66.38
FedWolf	**75.33**	**76.89**	**75.51**	**74.12**	**72.10**

5.2.4 Comparison on Communication Cost and Privacy

Communication cost and privacy protection play important roles in federated learning. We compare the communication rounds and additional cost required to achieve 50% testing accuracy between FedWolf and the existing methods on CIFAR100 and ResNet18 to demonstrate the positive effects of FedWolf in communication cost and privacy protection. The results are shown in Table 6.

As shown in the second column of Table 6, FedWolf only needs 10 communication rounds to achieve 50% testing accuracy, which is 9 and 50 communication rounds faster than the advanced methods (FedDC, FedDyn) and the slowest method (SCAFFOLD), respectively.

Then we measure the additional communication cost without model parameters. The FedAVG-related methods upload the number of datasets (a positive integer) to the server as a contribution. FedWolf does not need to upload the additional data to the server, so the total additional cost is 0. FEDIC uploads the number of datasets and auxiliary data to the server. Table 6 only reports the cost of uploading the number of datasets in FEDIC because the cost of uploading auxiliary data depends on the total amount of auxiliary data. CReFF and FedCLIP upload the gradient and text feature to the server, so the total cost (2,352.76 KB and 1,492.281 KB) is higher than the other methods. The above experimental results verify that FedWolf reduces the communication cost.

TABLE 6
Comparison results of additional communication cost and privacy protection required to reach 50% testing accuracy between related work and FedWolf.

Methods	Rounds	Total Additional Cost (KB)	Addtional Transmission Data
FedAVG	50	0.027	Number of datasets
FedDC	19	0.027	Number of datasets
FedDyn	19	0.027	Number of datasets
FedProx	50	0.027	Number of datasets
SCAFFOLD	60	0.027	Number of datasets
FEDIC	42	0.027	Number of datasets, Auxiliary data
CReFF	51	2,352.760	Gradient
FedAFA	35	0.027	Number of datasets
FedCLIP	30	1,492.281	Number of datasets, Text feature
Fed-grab	40	0.027	Number of datasets
FedWolf	**10**	**0**	None

The privacy protection of the comparative methods and FedWolf is investigated. FedCLIP, CReFF and FEDIC additionally upload the text feature, gradient and auxiliary data to the server, respectively, which increases the risks of privacy exposure. The additional transmission data of FedWolf is none, so FedWolf is reliable in privacy protection.

5.3 Method Validation

5.3.1 Influence of Each Module

We conduct an ablation study to evaluate the necessity of the parameter update based on the Grey Wolf Optimizer and the model aggregation based on the Markov Chain in FedWolf. The results are shown in Table 7.

Because each participant holds the independent parameters during the parameter updating phase, we report the testing accuracy achieved by the leader participant in Table 7, Line 2, to state the effectiveness of the parameter update algorithm in FedWolf. Compared to FedAVG [1] (52.13% and 38.33% as shown in Table 3), the performance of the leader participant is improved by 16.83% and 16.82% on CIFAR100 and ImageNet1000, respectively.

Then, we report the testing accuracy after model aggregation in Table 7, Line 3 to evaluate the necessity of the model aggregation algorithm in FedWolf. Compared to the performance of the leader participant, the performance of the global model is improved by 1.14% and 1.05% after the model aggregation on CIFAR100 and ImageNet1000, respectively. The global model contains the representations of all participants' data. Instead, the representations of leader participant's datasets are long-tailed and inadequate. Therefore, the global model achieves better performance than the leader participant. Federated learning improves the

performance of global model by adjusting the weights of local models in model aggregation. Some potential factors include the quality, quantity, and distribution of participants' data. FedWolf achieved competitive global performance by predicting participants' future performance as weights. The above results are empirical evidence that demonstrates the effectiveness and necessity of the two modules adopted in FedWolf.

TABLE 7
Testing accuracy (%) achieved by each module of FedWolf on CIFAR100 and ImageNet1000.

Module	CIFAR100	ImageNet1000
Wolf	68.96	55.15
Wolf + Markov	**70.10**	**56.20**

5.3.2 *Influence of Participant Pruning Strategy*

We empirically assess the impact of pruning different numbers of participants in each participant level $L = \{1, 2, 3\}$. Four strategies are provided in FedWolf, including [1, 1, 8], [1, 2, 7], [1, 3, 6], [1, 4, 5]. For example, [1, 1, 8] means the number of participants is 1, 1, 8 in level 1, 2, 3, respectively. Table 8 reports the results for these participant pruning strategies. Then, the process of participant level change with different strategies is recorded in Fig. 3 to discuss the rationality of the participant pruning strategy.

The experimental results show that FedWolf achieves the optimal performance (70.10% and 56.20%) with strategy [1, 2, 7] on CIFAR100 and ImageNet1000, respectively. The analysis on the experimental results of the four strategies is as follows:

(1) As shown in Fig. 3 (a), Participant 1 is in the dominant position, and Participants 7 and 8 are disadvantaged when the strategy is [1, 1, 8]. The contribution weights ω_{k_i} of Participants 1, 7 and 8 are 87.31%, 0% and 0% respectively in model aggregation, because Participants 7 and 8 never achieved leadership positions. The performance of the global model is similar to the performance of Participant 1.

(2) When the strategy is [1, 2, 7], the outstanding leaders (Participant 5) are elected, and the other participants also have the potential to become leaders, as shown in Fig. 3 (b), which can fairly aggregate the contributions of all participants.

(3) Increasing the number of participants at level 2 results in the superior (level 1, 2 participants) occupying a favourable position in the election for a long time but there is no outstanding leader. So, the contribution weight is divided equally when the number of participants in each level is set to [1, 3, 6] or [1, 4, 5], as shown in Fig. 3 (c) and (d).

5.4 Robustness in Resisting Outliers

Outliers are honest participants who achieve the lowest testing accuracy and damage the performance of the global model, because of the unreasonable initial parameters. FedWolf is launched with a set of inconsistent parameters,

TABLE 8
Testing accuracy (%) achieved by FedWolf with different participant pruning strategies.

Strategy	CIFAR100	ImageNet1000
1, 1, 8	65.71	54.69
1, 2, 7	**70.10**	**56.20**
1, 3, 6	69.59	55.02
1, 4, 5	69.64	55.87

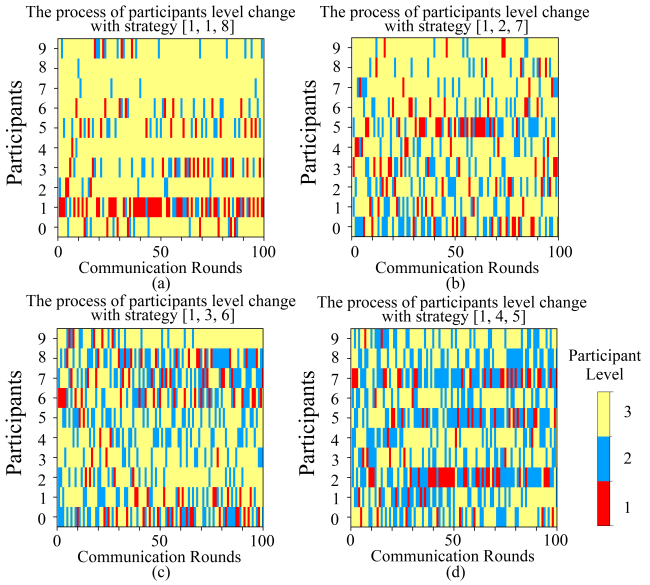

Fig. 3. The process of participant level change with four different participant pruning strategies. Different colours are used to represent the participant level $L = \{1, 2, 3\}$, as shown in the caption. The colour change in each row represents the change in participant level.

which is not recommended in FedAVG [1] due to interference from outliers. We conduct an experiment to evaluate the robustness of FedWolf in resisting outliers. We randomly select 1 and 5 participants, and fulfil the initialization parameters of them with 1 to simulate outliers, which disturbs back-propagation because of the symmetry breaking [28]. FedWolf and all the comparison methods are launched with a set of initialization parameters containing outliers on CIFAR100, and the ResNet18 is used as a backbone network.

The experimental results in Table 9 indicate that the robustness of FedWolf in resisting outliers is superior to the comparative methods. The testing accuracy achieved by all the comparative methods with outliers is less than 10.00%, which indicates they are affected by outliers. The testing accuracy achieved by FedWolf with 1 and 5 outliers is 69.90% and 69.02%, decreased by 0.2% and 1.08% than the normal performance without outliers, respectively. The performance degradation of FedWolf is lower than all the comparison methods, which proves the robustness of FedWolf in resisting outliers.

The process of participant level change and testing accuracy change in parameter updating are tracked to analyse the robustness of FedWolf in resisting outliers. The results are shown in Fig. 4. At the early stages of FedWolf (1-10 communication rounds), the outlier achieves the worst

TABLE 9
Testing accuracy (%) achieved by comparative methods and FedWolf with different numbers of outliers.

Methods	Without Outlier	1 Outlier	5 Outliers
FedAVG	52.13	1.81	1.50
FedDC	62.29	5.77	2.11
FedDyn	62.69	6.47	2.09
FedProx	52.31	1.50	1.12
SCAFFOLD	51.57	1.60	1.21
FEDIC	61.78	1.21	1.02
CReFF	63.94	1.10	0.83
FedAFA	60.23	0.89	0.51
FedCLIP	65.24	3.24	1.41
Fed-grab	66.12	2.87	1.17
FedWolf	**70.10**	**69.90**	**69.02**

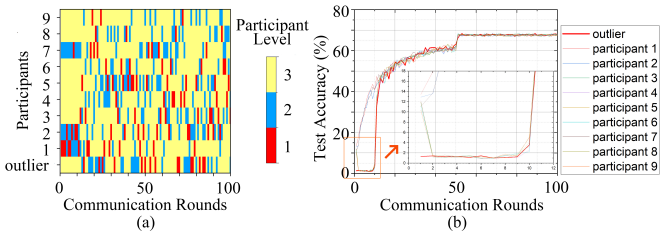

Fig. 4. (a): The process of participant level change of all participants with 1 outlier. (b): The testing accuracy achieved by all participants with 1 outlier. The partial enlargement shows that the performance of participants at level 3 decreases in communication round 2, because of outlier and increases after communication round 10. The performance leaps after 50 communication rounds due to the smaller learning rate.

performance because of the unreasonable initial parameters (as shown in Fig. 4 (b)). FedWolf effectively isolates outliers to level 3 and allows errors to propagate only among participants at level 3, which results in the performance of participants at level 3 remaining low and the performance of participants at levels 1, and 2 being unharmed (as the partial enlargement shown in Fig. 4 (b)). On the other hand, FedWolf allows the high-performance participants (levels 1, and 2) to share parameters with the weak participants (level 3), which helps the outlier to escape the trap of incorrect initialization parameters. The performance of the outlier returns to normal, after 10 communication rounds (as the partial enlargement shown in Fig. 4 (b)). The parameter update algorithm adopted by FedWolf effectively weakens the impact of outliers.

The early disadvantage makes it difficult for participants at level 3 to gain a favourable position in the election despite achieving normal accuracy performance, so they hold lower occurrences for becoming level 1 (as shown in Fig. 4 (a)) and their contribution weights ω_{k_i} are small in model aggregation. The model aggregating in FedWolf also weakens the influence of outliers.

The parameter update algorithms of the existing methods mix the parameters of all participants, which makes it easy for errors to spread to all participants and pollute the global model performance. FedWolf effectively isolates outliers to level 3 by the election mechanism, to avoid pollution spreading upwards. The proposed parameter update algorithm allows high-quality participants to propagate

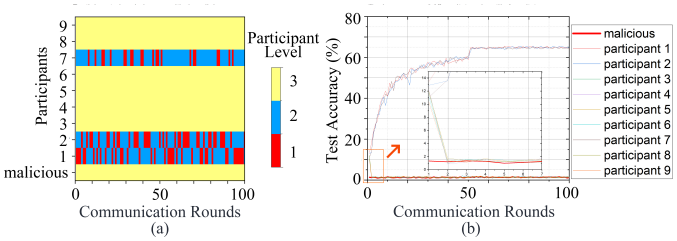

Fig. 5. (a): The process of participant level change of all participants with 1 malicious user. (b): The testing accuracy achieved by all participants with 1 malicious user. The partial enlargement shows that the performance of participants at level 3 decreases in communication round 2, because of malicious users. The performance leaps after 50 communication rounds due to the smaller learning rate.

their parameters downwards, to weaken the performance bias caused by outliers. The contribution weight ω_{k_i} of the outlier is small in model aggregation because of the early disadvantage, which also weakens the impact of outliers.

5.5 Robustness in Resisting Malicious Users

Malicious users aim to disrupt the performance by continuously poisoning during local training. We provide respective examples for data poisoning [32] and model poisoning [33] to verify the robustness of FedWolf in resisting malicious users. We randomly select 1 and 5 participants as malicious users, respectively. The datasets of the malicious users are mixed with Gaussian noise in data poisoning and the updated parameters of the malicious users are biased in model poisoning. All the comparison methods are launched with a set of initial parameters containing malicious users on CIFAR100, and the ResNet18 is used as a backbone network.

Table 10 reports the testing accuracy achieved by the comparative methods and FedWolf. The performance of all the methods is decreased in data poisoning. The testing accuracy achieved by FedWolf is 66.33% and 66.29% with 1 and 5 malicious users in data poisoning, which is decreased by 3.77% and 3.81% compared to its performance without malicious users (70.10%). All the comparison methods fail with the model poisoning and their performance is less than 1.00%. The testing accuracy achieved by FedWolf is 66.28% and 65.21% with 1 and 5 malicious users in model poisoning, which is decreased by 3.82% and 4.89% compared to its performance without malicious users (70.10%). The above experiential results show that FedWolf is more robust than existing methods in resisting malicious users.

Fig. 5 shows the participant level change and the testing accuracy change of all the participants with 1 malicious user (model poisoning) to explain the robustness of FedWolf in resisting malicious users. The malicious user continuously uploads contaminated model parameters, resulting in the lowest performance (as shown in Fig. 5 (b)). FedWolf isolates the malicious users to level 3, which causes the performance of all the participants at level 3 to be decreased (as the partial enlargement shown in Fig. 5 (b)) and prevents contamination from spreading to better-performing participants (at levels 1 and 2). In model aggregation, the contribution weight ω_{k_i} of the malicious user and the infected participants (malicious user and Participants 3,4,5,6,8,9 in Fig. 5 (a)) is 0, because they never become level 1. The aggre-

TABLE 10
Testing accuracy (%) achieved by comparing existing methods and FedWolf with different numbers of malicious users.

Methods	Without Malicious User	Data Poisoning 1 Malicious User	Data Poisoning 5 Malicious Users	Model Poisoning 1 Malicious User	Model Poisoning 5 Malicious Users
FedAVG	52.13	43.19	38.19	0.55	0.31
FedDC	62.29	47.68	40.01	0.48	0.44
FedDyn	62.69	47.43	41.03	0.19	0.25
FedProx	52.31	45.90	39.23	0.32	0.31
SCAFFOLD	51.57	41.28	38.19	0.18	0.10
FEDIC	61.78	43.09	37.98	0.19	0.11
CReFF	63.94	48.25	37.66	0.10	0.08
FedAFA	60.23	46.12	36.29	0.29	0.21
FedCLIP	65.24	44.36	35.19	0.41	0.14
Fed-grab	66.12	47.18	38.22	0.27	0.12
FedWolf	**70.10**	**66.33**	**66.29**	**66.28**	**65.21**

gation with dynamic adaptive contribution weights achieve exclusion, which means the malicious users completed all training, but their parameters are not aggregated into the global model.

The difference in mechanism for resisting outliers and malicious users is the weight of aggregation. Outliers hold small weight than other participants and the weight of malicious users is 0. FedWolf does not perform any additional detection to outliers and malicious users during the training process. This method relies solely on the dynamic adaptive weights to eliminate the adverse effects caused by outliers and malicious users on the performance of federated learning.

6 CONCLUSION

FedWolf is proposed to solve the performance cracks and the bias of categorization results caused by the local long-tailed data, which is a dynamic adaptive federated learning algorithm with the Grey Wolf Optimizer and Markov Chain. FedWolf is launched with a set of initialization parameters and the parameters are updated based on the Grey Wolf Optimizer. After all communication rounds are completed, the parameter update is modelled as a Markov Process and the future performance of each participant is inferred from the Markov state as a contribution weight to aggregate the global model. We provide the convergence analysis and validity analysis of FedWolf. Besides, the Gini index is introduced to evaluate the bias of classification results in this work. Extensive experimental evaluation validates the effectiveness of FedWolf in solving the performance cracks and the bias of categorization results caused by the local long-tailed data. We also conduct empirical research to demonstrate the robustness of FedWolf in resisting outliers and malicious users.

FedWolf effectively improves the global performance and alleviates the bias of categorization results on the local long-tailed data. There are some limitations and challenges during the experiment. For example, the impact of data repetition and missing on the performance of federated learning is significant. However, due to the limitation of dataset' size, these challenges have not been explored on the long-tailed distribution data.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 62362043, 62262036, 61962030), the Xingdian Talent Support Project (No. KKXY202203008), the Science and Technology Plan Projects of Yunnan Province (No. 202005AC160036, 202205AF150003, 202204BQ040010), and the Australian Research Council Discovery Projects Scheme (DP220101823).

REFERENCES

[1] McMahan B, Moore E, Ramage D, et al., "Communication-efficient learning of deep networks from decentralized data," Proceedings of the Artificial Intelligence and Statistics. 2017: 1273-1282.
[2] Kairouz P, McMahan H B, Avent B, et al., "Advances and open problems in federated learning," Foundations and Trends® in Machine Learning," 2021, 14(1–2): 1-210.
[3] Li T, Sahu A K, Talwalkar A, et al., "Federated learning: Challenges, methods, and future directions," IEEE Signal Processing Magazine, 2020, 37(3): 50-60.
[4] Gong B, Xing T, Liu Z, et al., "Adaptive client clustering for efficient federated learning over non-iid and imbalanced data," IEEE Transactions on Big Data, 2022: Early Access.
[5] Zhang J, Li Z, Li B, et al., "Federated learning with label distribution skew via logits calibration," Proceedings of the International Conference on Machine Learning. 2022: 26311-26329.
[6] Du F, Yang P, Jia Q, et al., "Global and local mixture consistency cumulative learning for long-tailed visual recognitions," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 15814-15823.
[7] Hinton Geoffrey Krizhevsky Alex, "Learning multiple layers of features from tiny images," in Tech. Rep. 2009: 32–33.
[8] Li T, Sahu A K, Zaheer M, et al., "Federated optimization in heterogeneous networks," Proceedings of the Machine Learning and Systems, 2020: 429-450.
[9] Karimireddy S P, Kale S, Mohri M, et al., "Scaffold: Stochastic controlled averaging for federated learning," Proceedings of the International Conference on Machine Learning. 2020: 5132-5143.
[10] Kawaguchi K, Huang J, Kaelbling L P. "Every local minimum value is the global minimum value of induced model in non-convex machine learning," Neural Computation, 2019, 31(12): 2293-2323.
[11] Gao L, Fu H, Li L, et al., "Feddc: Federated learning with non-iid data via local drift decoupling and correction," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 10112-10121.
[12] Durmus A E, Yue Z, Ramon M, et al., "Federated learning based on dynamic regularization," Proceedings of the International Conference on Learning Representations. 2021: 1-36.
[13] Shang X, Lu Y, Cheung Y, et al., "Fedic: Federated learning on non-iid and long-tailed data via calibrated distillation," Proceedings of the International Conference on Multimedia and Expo. 2022: 1-6.

[14] Shang X, Lu Y, Huang G, et al., "Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features," Proceedings of the International Joint Conference on Artificial Intelligence. 2022: 1-7.
[15] Mirjalili S, Mirjalili S M, Lewis A., "Grey wolf optimizer," Advances in engineering software, 2014, 69: 46-61.
[16] Chung K L., "Markov chains," Springer-Verlag, New York, 1967.
[17] Cui Y, Jia M, Lin T Y, et al., "Class-balanced loss based on effective number of samples," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 9268-9277.
[18] Yang Y, Xu Z., "Rethinking the value of labels for improving class-imbalanced learning," Advances in Neural Information Processing Systems, 2020: 19290-19301.
[19] Alshammari S, Wang Y X, Ramanan D, et al., "Long-tailed recognition via weight balancing," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2022: 6897-6907.
[20] Wang J, Liu Q, Liang H, et al., "Tackling the objective inconsistency problem in heterogeneous federated optimization," Advances in neural information processing systems, 2020: 7611-7623.
[21] Uddin M P, Xiang Y, Lu X, et al., "Federated Learning via Disentangled Information Bottleneck," IEEE Transactions on Services Computing, 2022: Early Access.
[22] Lu Y, Qian P, Huang G, et al., "Personalized Federated Learning on Long-Tailed Data via Adversarial Feature Augmentation," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2023: 1-5.
[23] Perozzi B, Al-Rfou R, Skiena S., "Deepwalk: Online learning of social representations," Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. 2014: 701-710.
[24] Deng J, Dong W, Socher R, et al., "Imagenet: A large-scale hierarchical image database," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2009: 248-255.
[25] Yurochkin M, Agarwal M, Ghosh S, et al., "Bayesian nonparametric federated learning of neural networks," Proceedings of the International Conference on Machine Learning. 2019: 7252-7261.
[26] He K, Zhang X, Ren S, et al., "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.
[27] Glorot X, Bengio Y., "Understanding the difficulty of training deep feedforward neural networks," Proceedings of the International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, 2010: 249-256.
[28] He K, Zhang X, Ren S, et al., "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," Proceedings of the IEEE International Conference on Computer Vision. 2015: 1026-1034.
[29] Corrado Gini, "Measurement of Inequality of Incomes," The Economic Journal, 1921: 31(121) 124–125.
[30] Dagum C. A new approach to the decomposition of the Gini income inequality ratio[M]. Physica-Verlag HD, 1998.
[31] Gharibi M, Rao P., "Refinedfed: A refining algorithm for federated learning," Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop. 2020: 1-5.
[32] Biggio B, Nelson B, Laskov P., "Poisoning attacks against support vector machines," Proceedings of the International Coference on International Conference on Machine Learning. 2012: 1467-1474.
[33] Bhagoji A N, Chakraborty S, Mittal P, et al., "Analyzing federated learning through an adversarial lens," Proceedings of the International Conference on Machine Learning. 2019: 634-643.
[34] Breiman L. Classification and Regression Trees[J]. The Wadsworth Brooks/Cole, 1984.
[35] Shi J, Zheng S, Yin X, et al. CLIP-Guided Federated Learning on Heterogeneity and Long-Tailed Data[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38(13): 14955-14963.
[36] Xiao Z, Chen Z, Liu S, et al. Fed-grab: Federated long-tailed learning with self-adjusting gradient balancer[J]. Advances in Neural Information Processing Systems, 2024, 36.
[37] Zhang Y, Wang K, He Q, et al. Covering-based web service quality prediction via neighborhood-aware matrix factorization[J]. IEEE Transactions on Services Computing, 2019, 14(5): 1333-1344.
[38] Murhekar A, Yuan Z, Ray Chaudhury B, et al. Incentives in federated learning: Equilibria, dynamics, and mechanisms for welfare maximization[J]. Advances in Neural Information Processing Systems, 2024, 36.

Juncheng Pu is currently working toward the PhD degree at Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. His research interests include distributed machine learning and federated learning.

Xiaodong Fu is a professor at Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. His research interests include services computing, federated learning, and intelligent decision making. He received his PhD degree in management science and engineering from Kunming University of Science and Technology. He is an IEEE member, a senior member of China Computer Federation and a member of China Computer Federation Technical Committee on Service Computing.

Hai Dong is a senior lecturer at School of Computing Technologies in RMIT University, Melbourne, Australia. He received a PhD from Curtin University, Perth, Australia. His research interests include Service Computing, Edge Computing, Blockchain, Cyber Security, Machine Learning and Data Science. He is a senior member of the IEEE.

Pengcheng Zhang is a professor at College of Computer and Information, Hohai University, Nanjing, China. He received the Ph.D. degree in computer science from Southeast University in 2010. His research interests include software engineering, service computing and data mining. He has published in premiere or famous computer science journals. He was the co-chair of IEEE AI Testing 2019. He served as technical program committee member on various international conferences. He is an IEEE member.

Li Liu is a professor at Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. She received her PhD degree from Sun Yat-sen University, China. Her research interests include computer vision and graphics and machine learning.