IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 January 2024, accepted 23 January 2024, date of publication 10 April 2024, date of current version 17 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359251

RESEARCH ARTICLE

FedBoost: Bayesian Estimation Based Client Selection for Federated Learning

YUHANG SHENG[1], LINGGUO ZENG[2], SHUQIN CAO[3], QING DAI[4], SHASHA YANG[4], AND JIANFENG LU[3,4]

[1]School of Mathematical Science, Zhejiang Normal University, Jinhua 321004, China
[2]School of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua 321004, China
[3]School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China
[4]School of Computer Science and Technology, Zhejiang Normal University, Jinhua 321004, China

Corresponding author: Lingguo Zeng (jk63@zjnu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62372343 and Grant 62072411, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LR21F020001, and in part by the Scientific Research Fund of Zhejiang Provincial Education Department under Grant Y202147366.

ABSTRACT Although federated learning (FL) represents a distributed machine learning paradigm that ensures privacy protection, the failure of stragglers to upload local models in a timely manner results in an overall degradation of the global model's performance, and the difficulty of accurately predicting whether clients will succeed in uploading a local model makes client selection still a challenge. To address this issue, existing works mainly focus on increasing the number of clients who participate in training within a fixed time, however, the fact is that the performance of a global model depends on the data used for training. Therefore, increasing the clients' data contribution to the global model can effectively enhance the global model's performance. To this end, we propose a Bayesian estimation based FL framework, named FedBoost, to enhance the performance of the model when straggler problem exists. Specifically, we formulate a long-term problem aimed at maximizing clients' cumulative effective data contributions, while satisfying a long-term fairness constraints, which ensure a minimum selection frequency for clients. By analyzing the stability of virtual queues, we transform the long-term problem into a stepwise one via Lyapunov optimization, reducing its computational complexity. Due to the inability of the server to predict whether clients successfully upload the local model before receiving the actual upload, we use Bayesian estimation based on the observed frequency of successful uploads to estimate this probability. Last, extensive experimental results indicate that the average test accuracy of our FedBoost is up to 5.59% higher than both FedAvg and FedCS on three real-world datasets, and achieves test loss that are at most 0.1646 below the two baselines. Furthermore, the value of Lapunov function remains lower than 1.4, and at least 85% of the estimation of probabilities are in a reasonable range.

INDEX TERMS Bayesian estimation, client selection, federated learning, Lyapunov optimization.

I. INTRODUCTION

With the proliferation of smart devices, massive amounts of data are stored on distributed front-end devices, offering potential for machine learning tasks. However, due to the data sensitivity, people are reluctant to disclose their raw data to other entities. Federated learning (FL), a distributed machine learning paradigm that enables clients to co-train a global model without leaking their raw data, is considered as one of the most promising solutions to the isolated "data island" problem [1]. A typical FL system is consisted of a cloud server and distributed clients connected by the Internet [2]. Specifically, the selected clients perform several local epochs of stochastic gradient descent (SGD) on the received global model using their local datasets. Then, they upload the trained local model to the server for global model aggregation instead

The associate editor coordinating the review of this manuscript and approving it for publication was Ikramullah Lali.

of sharing their raw data. The iterative process continues until the desired testing accuracy is attained. Due to its commitment to protecting clients' data privacy, FL has attracted considerable attention across various fields, such as healthcare [3], [4], [5], *etc.*, and Internet of Things [6], [7], [8], *etc.*.

In practical applications, acquiring a high-quality machine learning model is imperative. Empirical evidence has demonstrated that boosting clients' participation accelerates the convergence of FL models [9]. However, due to practical constraints such as limited bandwidth and client availability, only a subset of clients can be selected for participation in each training round. During each training iteration, a fraction of clients' data information remains inaccessible to the global model. Thus, the client selection play a pivotal role in the convergence and performance of FL [9]. For example, clients may encounter various obstacles that prevent them from uploading their trained local models to the cloud server, including limited bandwidth, accidental closure of the training program, and active termination of the FL process by clients [10]. The straggler problem is a significant challenge for FL [11], [12]. The failure of selected clients to upload the trained local model to the server will inevitably lead to a degradation in the performance of the global model. Intuitively, we should prioritize clients with high probability of successfully uploading their local models. However, due to the heterogeneous data in FL (*i.e.*, data quantity, data quality) each client contributes differently to the global model. It is imperative to develop an effective client selection method to optimize the utilization of clients' data and enhance the performance of the global model.

The design of client selection in FL has become an increasingly prominent focus in current research. Existing works are based on the empirical observation that increased client participation in training is positively correlated with enhanced model performance [13]. However, the heterogeneous data in FL implies that selecting more clients to participate in training within a fixed time does not necessarily result in an optimal global model. Selecting a client with a large amount of high-quality data to participate in the training may contribute much more to the global model than selecting multiple clients with only a small amount of low-quality data. The straggler problem makes client selection more complicated. Even if the selected clients use substantial amounts of data to train local models, the absence of uploaded local models to the server means that they contribute nothing. Actually, the performance of the trained model improves with an increase in both the quantity and quality of data used for training [14]. Therefore, in order to enhance the performance of the global model, it is imperative to optimize the effective contribution of clients' data to the global model through meticulous client selection.

Despite recent advancements in FL, optimizing the performance of global models in the presence of stragglers remains a formidable challenge for three-fold reasons: (i) How to balance performance and long-term fairness is challenging. Long-term fairness involves ensuring the average selection rate of clients. Whereas the distinctive data from each client can enhance model performance, an excessive focus on long-term fairness may lead to a compromise in model performance. (ii) FL clients are heterogeneous. On the one hand, heterogeneous data contributes uniquely to the global model. On the other hand, device heterogeneity introduces the challenge of stragglers. (iii) Information asymmetry exists between clients and servers. The server is unaware of the probability of successful model uploads by the clients. Although we can approximate the probability of clients successfully uploading the model based on the frequency of successful uploads, there is a large deviation in this approximation when the training rounds are not large enough.

In this paper, given that the global model's performance depends on the quantity and quality of data used for training, we propose to maximize the cumulative effective data contribution while ensuring long-term fairness, thus guaranteeing the exceptional performance of the global model. Then, we take advantage of Lyapunov optimization to transform the long-term problem into a queue stability problem, systematically resolving the optimal client selection scheme step by step while ensuring the establishment of the long-term fairness constraint. This significantly diminishes the computational complexity. To address the issue of unknown clients' dropout probabilities, we employ Bayesian estimation as a solution. In contrast to the frequency approximation method, Bayesian estimation yields an approximation value that remains close to the true value even with limited training rounds. Our main contributions are summarized as follows:

- We propose a federated learning framework via Bayesian estimation based online clients selection, named FedBoost. This framework has the potential to enhance the performance of the global model by estimating the probability of clients successfully uploading their local models, thereby maximizing the cumulative effective data contribution.
- We optimize the global model by maximizing the cumulative effective data contribution from clients, even in the presence of the straggler problem. We also establish a long-term fairness constraint to ensure equitable clients selection rates. By employing virtual queues to quantify the long-term fairness constraint, and use Lyapunov optimization to transform the long-term problem into a stepwise one, greatly reducing the computational complexity.
- We conduct extensive experiments on three real-world datasets (*i.e.*, MNIST, FMNIST, and CIFAR-10) to illustrate the superiority of our algorithm. The average test accuracy of our FedBoost outperforms FedAvg and FedCS by up to 5.59% on MNIST dataset, up to 2% on FMNIST dataset, and up to 3.73% on CIFAR-10. The test loss of our FedBoost is lower than both FedAvg and FedCS by up to 0.0824 on MNIST dataset,

up to 0.0355 on FMNIST dataset, and up to 0.1646 on CIFAR-10.

In the rest of this paper, we review the related works in Section II, and construct the client selection problem in Section III. We solve it in Section IV and provide the theoretical analysis. We conduct the performance evaluation in Section V, and draw the conclusion in Section VI.

II. RELATED WORKS

In this section, we review the related works in FL from three perspectives, *i.e.*, performance promoting algorithms, client selection, and the application of Lyapunov optimization.

A. PERFORMANCE PROMOTING ALGORITHMS

FedAvg is one of the most popular FL algorithms, and serves as the foundation of FL [2]. However, the performance of FedAvg exhibits a significant decline when confronted with highly non-i.i.d. datasets. To address this problem, the authors in [15] proposed FedProx, which is a generalization of FedAvg. To accelerate the convergence speed of the global model, Yuan and Ma [16] proposed FedAc. Although FedAc speeds up the convergence of the global model, it ignores the communication cost in federated networks.

The above works considered an idealized scenario, *i.e.*, a scenario without straggler problem. In real-world scenarios, the straggler problem may arise as a result of the heterogeneity of clients' devices. Stragglers encounter difficulties in successfully uploading their local models, resulting in a performance penalty for the global model.

B. CLIENT SELECTION

To accelerate the convergence speed of the global model, Nishio and Yonetani [17] proposed the FedCS algorithm, aiming to select as many clients as possible to participate in the training while satisfying the constraints. To solve straggler problem in synchronous FL, Huang et al. [9] ensured the convergence speed of the global model by maximizing the expected cumulative effective participation. Another effective approach to expedite the convergence of the global model is to minimize the time-span of each training round, including the time required for downloading the global model, conducting local training, and uploading the local model. Huang et al. [1] constructed a long-term client selection problem to minimize the training time-span while satisfying the long-term fairness constraint. To avoid the straggler problem in FL, Zhu et al. [10] proposed an asynchronous FL framework and constructed a dynamic client selection problem to reduce the training time-span.

The aforementioned works are based on the premise that enhancing clients' participation can expedite the global model's convergence speed. The performance of the model is contingent upon the quantity and quality of the training data. By selecting appropriate clients, maximizing the effective data contribution of clients to the global model, the model performance can be effectively improved.

TABLE 1. List of notations.

Notation	Description
T, t	The number and index of global iteration rounds
$\mathcal{N}, N$	The set of clients and total number of clients
$f_n(w), F_n(w)$	Global loss function and client n's local loss function
$\mathcal{D}_n, D_n$	Client n's local dataset and the number samples in it
w_g^t, w_n^t	Global model and client n's local model in t-th round
$p_{t,n}$	Client n's aggregation weight in t-th round
$x_{t,n}$	Whether client n is selected in t-th round
$I_{t,n}$	Whether client n successfully uploads local model in t-th round
q_n, θ_n	Client n's data proportion and data quality
$\chi_{t,n}$	Client n's effective data contribution in t-th round
$Z_t, Z_{t,i}$	The set of backlogs, and the backlog of client i's virtual queue
$L(Z_t), \Delta(Z_t)$	Lyapunov function and Lyapunov drift
$\mathbb{E}$	The expectation of random variables

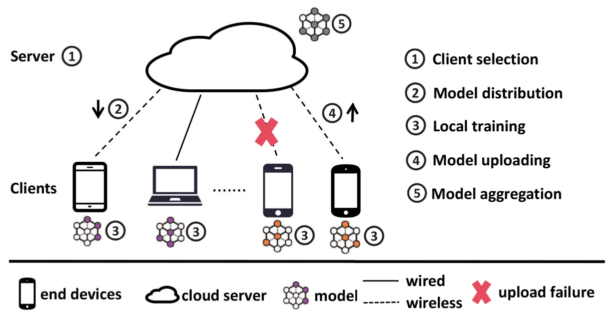

FIGURE 1. A federated learning system model for straggler problem.

C. LYAPUNOV OPTIMIZATION

Lyapunov optimization can be applied to communication and queueing systems, *e.g.*, an Internet with peer-to-peer communication [18]. The FL system can be considered an Internet with peer-to-peer communication due to its inherent structural characteristics. Hence, Lyapunov optimization can be employed for the analysis of problems in FL. Lyapunov optimization can be directly applied to scenarios with actual queues. The research conducted in [19] demonstrates that the implementation of Lyapunov optimization can effectively mitigate communication delays between vehicle users and roadside units within federated vehicle networks. When faced with long-term time-average constraints in the optimization problem, the introduction of virtual queues enables us to satisfy the constraints effectively [18]. In this case, Lyapunov optimization can be applied to the scenarios without actual queues. Battiloro et al. [20] introduced virtual queues to transform the time-average constraints into queue stability, and used Lyapunov optimization to construct a dynamic resource allocation strategy.

Different from the client selection algorithms mentioned earlier, which focus on increasing the number of clients participating in FL, this paper aims to optimize the cumulative effective data contribution from clients by devising a FL framework. Moreover, we view the entire FL process as an integrated entity. To enhance the convergence speed of the global model, we consider the impact of straggler problem on the global model from a holistic perspective.

III. PROBLEM FORMULATION

In this section, we construct a FL framework to accelerate the convergence of the global model. And we propose a client's effective data contribution and a long-term fairness constraint on clients. Then, we formulate a dynamic client selection to maximize the cumulatively effective data contribution.

A. SYSTEM MODEL

In a typical FL system, there are N clients, indexed by $\mathcal{N} = \{1, \cdots, N\}$, and one server. Each client n owns a private local dataset $\mathcal{D}_n$ with $D_n = |\mathcal{D}_n|$ data samples. Frequently used symbols are listed in Table 1.

These clients train local models on their raw data and communicate with the server periodically to co-train a global model. Typically, the clients datasets are non-i.i.d., *i.e.*, the volume and distribution of the clients datasets are different. Specifically, the local loss function of client n is given as follows [21]:

$$F_n(w) = \frac{1}{D_n} \sum_{j \in \mathcal{D}_n} l_j(w), \tag{1}$$

where j is a data sample in client n's dataset, and $l_j()$ is the experience loss function. With the formula of the local loss function, the typical goal is to minimize the global loss function, which is described as bellow:

$$\min_w f(w) = \sum_{n=1}^{N} \frac{D_n}{D} F_n(w), \tag{2}$$

where $D = \sum_{n=1}^{N} D_n$, *i.e.*, total number of data samples of all clients.

FL aims to achieve a optimal global model $w^* = \arg\min_w f(w)$ by minimizing the global loss function. To achieve this objective, the server needs to select a fraction of clients to co-train a global model based on their raw data. The straggler problem arises as a consequence of the inherent characteristics associated with FL. In reality, the occurrence of this problem can be attributed to a multitude of factors. For instance, the inadvertent shutdown of the training process by a client can lead to the straggler prblem. In FL, when stragglers fail to upload their local models properly during the training process, their data does not contribute to the global model, resulting in a deceleration of the convergence rate for the global model.

To promote the global model's performance in the presence of stragglers in the FL system, the server needs to select appropriate clients to participate in training. As shown in Figure 1, we establish a FL system model for straggler problem with five steps: (i) The server evaluates clients' effective data contributions, and selects clients according to the evaluation. (ii) The server distributes the global model to the selected clients. (iii) Each selected client trains the model using their local data. (iv) Each selected client uploads the trained local model. (v) The server aggregates the local models via our proposed aggregation rule.

Algorithm 1 Performance Boosting FL

Input: Clients Number N, Global Epochs T, Local Epochs K, Learning Rate η
Output: Global Model w_g^T

Initialize the global model w_g^0;
for $t = 1 : T$ **do**
  Server selects clients S_t according to Algorithm 2;
  if $n \in S_t$ **then**
    $w_n^{t,0} = w_g^{t-1}$;
    for $k = 1 : K$ **do**
      $\delta_n^{t,k} = \nabla F_n\left(w_n^{t,k-1}\right)$;
      $w_n^{t,k} = w_n^{t,k-1} - \eta \delta_n^{t,k}$;
    end for
    $w_n^t = w_n^{t,K}$;
    Uploads w_n^t to server;
  end if
  $p_{t,n} = \frac{\bar{p}_{t,n} I_{t,n}}{\sum_{i \in S_t} \bar{p}_{t,i} I_{t,i}}, \forall n \in S_t$;
  $w_g^t = \sum_{i \in S_t} p_{t,i} w_i^t$;
end for

B. FL FRAMEWORK FOR STRAGGLER PROBLEM

To address the straggler problem, we propose a performance boosting FL framework (named FedBoost) that can speed up global model convergence. This framework constructs a client selection method to maximize the effective data contribution to the global model. To further minimize straggler's influence, we build a rewighting aggregation rule. FedBoost consists of the following parts:

- **Client selection:** According to the problem formulated in Section III, the server selects a fraction of available clients to participate in each training round, so that the expected effective data contribution to the global model is maximized. That is,

$$S_t^* = \arg\min_{S_{t}{}_{t=1}^{\infty}} \mathcal{P}1, \tag{3}$$

  where S_t^* is the optimal clients set in t-*th* round.
- **Local training:** After receiving the global model, the selected clients run several local epochs on their local datasets to train a local model, that is, achieving a optimal solution of the local loss function with SGD optimizer, *i.e.*,

$$w_n^{t+1} = w_n^t - \eta \nabla F\left(w_n^t\right), \quad \forall n \in \mathcal{N}, \ \forall t > 0. \tag{4}$$

- **Model aggregation** When the allocated time for training and model uploading provided by the server to clients expires, the server calculates new aggregation weight for clients, who upload their local model, by the following formula:

$$p_{t,n} = \frac{\bar{p}_{t,n} I_{t,n}}{\sum_{i \in S_t} \bar{p}_{t,i} I_{t,i}}, \quad \forall n \in S_t, \tag{5}$$

  where S_t is the set of selected clients, $\bar{p}_{t,i} = D_n/D$, and $I_{t,i} \in \{0, 1\}$ indicates whether client i successfully

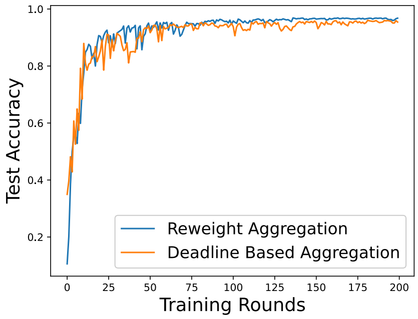

FIGURE 2. Comparison of reweight aggregation rule and deadline based aggregation rule.

uploaded the local model in t-th round. Then, the global model aggregates the local models in the following way:

$$w_g^{t+1} = \sum_{i \in S_t} p_{t,i} w_i^t, \tag{6}$$

where w_g^{t+1} is the aggregated global model, and w_i^t is the local model uploaded by client i in t-th round.

The FedBoost framework is summarized in Algorithm 1. To demonstrate the effectiveness of our reweight aggregation rule, we compare it with the deadline based aggregation rule [9], denoted as:

$$w_g^{t+1} = \sum_{i \in S_t, I_i = 1} p_{t,i} w_i^t + \sum_{i \in S_t, I_i = 0} p_{t,i} w_g^t. \tag{7}$$

Intuitively, when the selected clients do not upload the updated local model, the deadline based aggregation rule uses w_g^t to take place of w_n^t. This aggregation rule somewhat mitigated the local model loss problem caused by stragglers. However, substituting w_g^t for w_n^t for the stragglers still slows down the convergence of the model. For example, if a large percentage of clients are stragglers in t-th round, the global model makes little progress in this round. As shown in Figure 2, we compare our reweight aggregation rule with the deadline based aggregation rule. We experiment on MNIST and set the probability of clients not uploading the model to random numbers. The outcome aligns with our expectations, indicating that the reweight aggregation rule effectively mitigate the straggler problem.

C. EFFECTIVE DATA CONTRIBUTIONS

It is noteworthy that the more effective contributions the data makes to the model, the better the overall performance of the model. If the model is trained on a dataset with higher data quality and a larger quantity of data, the resultant model performance is better, leading to higher test accuracy [22]. Hence, to enhance the global model's performance, it is crucial to select the appropriate clients in a way that maximizes their effective data contributions. We denote the effective data contribution of client n in t-th round to be:

$$\chi_{t,n} = q_n \theta_n x_{t,n} I_{t,n}, \tag{8}$$

where $q_n = D_n/D$, θ_n is the data quality of client n, and $x_{t,n} \in \{0, 1\}$ denotes whether client n is selected in t-th round. In other words, if $n \in S_t$ then $x_{t,n} = 1$; otherwise $x_{t,n} = 0$. The global model in FL system is obtained through a training process. Therefore, to attain a well-performing global model after the training process, we need to maximize the cumulative effective data contribution throughout the entire training process, *i.e.*,

$$\sum_{t=1}^{T} \sum_{n=1}^{N} \chi_{t,n}. \tag{9}$$

Maximizing the cumulative effective data contribution is equivalent to maximizing the time-average cumulative effective data contribution. When the number of training rounds is sufficiently large, our objective is equivalent to maximizing the limit value of the time-average effective data contribution, *i.e.*,

$$\min \lim_{T \to \infty} -\frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \chi_{t,n}. \tag{10}$$

D. LONG-TERM FAIRNESS

Another criterion to consider is fairness, as it holds a significant influence on the performance of the global model [23]. In an ideal scenario where the server possesses accurate predictive capabilities to identify potential straggler clients, when the first m clients with the largest effective data contribution are always selected, the cumulative effective data contribution of the FL process is maximized. Yet, due to the heterogeneity of clients' data in FL, each client's dataset makes a unique contribution to enhancing the model's accuracy [10]. On the one hand, if some clients are always selected, their associated costs, such as energy consumption, may significantly exceed those incurred by other clients. On the other hand, if some clients are always ignored, due to the loss of client-specific data information, the performance of the trained global model may experience a significant decline. In this regard, greedy client selection can not optimize the global model. Thus, fairness is a crucial issue that should be taken into account. Moreover, fairness is a relative criterion, not an absolute one [24]. To model the long-term fairness, we denote $X_t = \left[x_{t,1}, \cdots, x_{t,N}\right]$, where X_t is a vector representing the client selection in t-th round. In order to ensure that each client can make effective data contributions to the global model, we construct a long-term fairness constraint, *i.e.*,

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} x_{t,n} \geq \gamma_n, \gamma_n \in (0, 1], \quad \forall n \in \mathcal{N}, \tag{11}$$

where γ_n is the expected guaranteed selection frequency for client n.

E. PROBLEM FORMULATION

In FL, having all clients participate in training creates a significant communication burden. Therefore, it is common practice to select a fraction of clients to participate in each training round. In this paper, we assume that the number of clients selected by the server to participate in each training round is at most m. Hence, we have

$$z_t = EX_t^{\mathrm{T}} \leq m, \quad \forall t > 0, \tag{12}$$

where $E = [1, \cdots, 1]$, *i.e.*, is a N-dimensional unit vector. Based on the discussions on effective data contribution and client fairness, we define the problem model as follows:

$$\begin{cases} \mathcal{P}1: \min_{\{S_t\}_{t=1}^{\infty}} \lim_{T\to\infty} -\frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N}\mathbb{E}\left[\chi_{t,n}\right], \\ \text{s.t.} \begin{cases} \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} x_{tn} \geq \gamma_n, \forall n \in \mathcal{N}, \\ \sum_{n=1}^{N} x_{tn} \leq m, \forall t > 0. \end{cases} \end{cases} \tag{13}$$

Solving $\mathcal{P}1$ presents some challenges. There are 2-fold concerns: (i) It can be noted that the object function and long-term fairness constraint in $\mathcal{P}1$ are time-coupled. Offline solving is complicated by the interplay of uncertainty and temporal coupling [25]. (ii) The server cannot determine know exactly which client will be straggler. Hence, the estimation of clients' expected effective data contribution remains unattainable, which poses a challenge in selecting the proper clients to promote the performance of the model.

IV. CLIENT SELECTION FOR MAXIMIZING EFFECTIVE DATA CONTRIBUTION

In this section, we exploit the Lyapunov optimization to transform the temporal coupling problem $\mathcal{P}1$ into a queue stability problem, which is much easier to be solved. Then, we leverage Bayesian estimation to assess the probabilities of clients becoming stragglers [26], constructing an online client select algorithm. The algorithm's time complexity, fairness guarantee and convergence are subsequently analyzed from a theoretical perspective.

A. PROBLEM TRANSFORMATION UNDER LYAPUNOV OPTIMIZATION

The management of the long-term fairness constraint poses challenges for conventional optimization methods. Lyapunov optimization employs virtual queues to transform the long-term fairness constraint to the stability of the virtual queues. This transformation enables stepwise resolution of the objective function. For the long-term fairness constraint in $\mathcal{P}1$, we introduce the following virtual queues:

$$Z_{t+1,n} = \max\left\{Z_{t,n} + \gamma_n - x_{t,n}, 0\right\}, \quad \forall n \in \mathcal{N}, \ \forall t > 0, \tag{14}$$

where $Z_{0,n} = 0, \forall n \in \mathcal{N}$. Proposition 1 is now presented to establish the fundamental principle of this transformation.

Proposition 1: The long-term fairness constraint in Eq. (11) holds, if all virtual queues remain mean rate stable during the FL process.

Proof: From the definition of the virtual queues, we can know that $Z_{t+1,n} \geq Z_{t,n} + \gamma_n - x_{t,n}, \forall t > 0, \forall n \in \mathcal{N}$. Thus, the following inequality hold:

$$\sum_{t=0}^{T}\left(Z_{t+1,n} - Z_{t,n}\right) \geq \sum_{t=1}^{T}\left(\gamma_n - x_{t,n}\right), \tag{15}$$

where the left hand side is equal to $Z_{T+1,n}$. As assumed, $Z_{t,n}$ is mean rate stable for $\forall n \in \mathcal{N}$, which means that

$$\lim_{t\to\infty}\frac{\mathbb{E}\left[Z_{t,n}\right]}{t} = 0, \tag{16}$$

where $\mathbb{E}\left[Z_{t,n}\right]$ is the expectation of $Z_{t,n}$. This implies that

$$\lim_{T\to\infty}\sum_{t=1}^{T}\mathbb{E}\left[\gamma_n - x_{t,n}\right] = 0. \tag{17}$$

The above basic mathematical operations show that the long term constraint holds. ■

The long-term fairness constraint in $\mathcal{P}1$ can be replaced by queue stability constraint, *i.e.*,

$$\lim_{T\to\infty}\frac{1}{T}Z_{T,n} = 0, \quad \forall n \in \mathcal{N}, \tag{18}$$

where $Z_{t,n}$ is the virtual queues defined in Eq. (14). Thus, the problem is transformed to maximizing the effective data contribution while ensuring all virtual queues mean rate stable. To measure the queues' length, we construct a Lyapunov function, which is a nonnegative metric of the virtual queues. The vector of virtual queues' backlogs is defined as $Z_t = \left[Z_{t,1}, \cdots, Z_{t,N}\right]$. Then, the Lyapunov function is defined as:

$$L\left(Z_t\right) = \frac{1}{2}Z_{t,n}^2. \tag{19}$$

The drift in virtual queues' length (*i.e.*, Lyapunov drift) is quantified by computing the discrepancy between the Lyapunov function of $t+1$-th round and that of t-th round, *i.e.*,

$$\Delta\left(Z_t\right) = \mathbb{E}\left[L\left(Z_{t+1}\right) - L\left(Z_t\right)\right]. \tag{20}$$

[18] states that given $\epsilon > 0$, $\alpha > 0$, and $B > 0$, such that

$$\Delta\left(Z_t\right) + \alpha\mathbb{E}\left[\sum_{n=1}^{N}\chi_{t,n} \mid Z_t\right] \leq B + \alpha\sum_{n=1}^{N}\chi_{t,n}^{*} - \epsilon\sum_{n=1}^{N}Z_{t,n}, \tag{21}$$

then all the virtual queues are mean rate stable, where $\chi_{t,n}^{*}$ is the optimal effective data contribution, α and ϵ are weight factors, and B is a constant. However, calculating $\Delta\left(Z_t\right)$ needs the information from $t+1$-th round, which is not accessible in t-th round. Thus, to tackle this problem, an upper bound of the Lyapunov drift is given as follows:

$$\begin{aligned} \Delta\left(Z_t\right) &= \mathbb{E}\left[L\left(Z_{t+1}\right) - L\left(Z_t\right)\right] \\ &\leq B + \mathbb{E}\left[\sum_{n=1}^{N}Z_{t,n}\left(\gamma_n - x_{t,n}\right) \mid Z_t\right]. \end{aligned} \tag{22}$$

Consequently, $\mathcal{P}1$ can be transformed into the following problem:

$$\begin{cases} \mathcal{P}2: \min\limits_{S_t} \ \mathbb{E}\left[-\alpha \sum\limits_{n=1}^{N} \chi_{t,n} + \sum\limits_{n=1}^{N} Z_{t,n}\left(\gamma_n - x_{t,n}\right)\right], \\ \text{s.t.} \ \sum\limits_{n=1}^{N} x_{tn} \leq m, \forall t > 0, \end{cases} \tag{23}$$

where α is a weight factor that regulates fairness and effective data contribution.

B. BAYESIAN ESTIMATION BASED ONLINE CLIENTS SELECTION

As defined in Eq. (8), the effective data contribution is a linear function of the client selection. We can substitute Eq. (8) into the object in $\mathcal{P}2$, that is,

$$-\alpha \sum_{n=1}^{N} \mathbb{E}\left[x_{tn}\left(q_n\theta_n I_{tn} + Z_{tn}\right)\right] + \gamma_n \sum_{n=1}^{N} Z_{tn}. \tag{24}$$

In this paper, we assume that the probability $\bar{d}_n$ of client n becoming a straggler is a fixed value for $\forall n \in \mathcal{N}$ (but not known to the server). The expectation of the effective data contribution is $\mathbb{E}\left[\chi_{t,n}\right] = q_n\theta_n x_{t,n} d_n$, where $d_n = 1-\bar{d}_n, \forall n \in \mathcal{N}$. With the above assumption, we can transform $\mathcal{P}2$ into the following problem:

$$\begin{cases} \mathcal{P}3: \min\limits_{S_t} \ -\alpha \sum_{n=1}^{N} x_{tn}\left(q_n\theta_n d_n + Z_{tn}\right) + \gamma_n \sum_{n=1}^{N} Z_{tn}, \\ \text{s.t.} \ \sum_{n=1}^{N} x_{tn} \leq m. \end{cases} \tag{25}$$

In each training round, q_n, θ_n, and $Z_{t,n}$ are constants. Therefore, if the probabilities $d_n, \forall n \in \mathcal{N}$ are known in advance, $\mathcal{P}3$ is quite easy to be solved. We just need to calculate the value of the objective function in $\mathcal{P}3$ for each client, and then select the first m largest clients. The law of the large number ensures when events occur enough times, we can approximate the probability of an event by the frequency of its occurrence [26]. However, when the training rounds are relatively small, the law of large numbers does not apply. For instance, if the server selects the client n three times in a row at the beginning of training (*i.e.*, $x_{1,n} = 1$, $x_{2,n} = 1$, and $x_{3,n}$=1), and client n fails to upload local model during all three rounds of training, the frequency at which client n becomes a straggler is 1. However, due to the limited number of observations, it would be inappropriate to assert that $d_n = 0$. We take advantage of Bayesian estimation to estimate the probability vector $\Theta = [d_1, \cdots, d_N]$. Bayesian estimation consists of the following five steps.

- We determine the parameters that need to be estimated $\Theta = [d_1, \cdots, d_N]$, and set a prior distribution $\mathcal{P}(\Theta)$.
- We calculate the frequency vector (*i.e.*, the frequency of successful local model upload) $R_t = \left\{r_{t,i}\right\}_{i=1}^{N}$ from the prior distribution about Θ.

Algorithm 2 Bayesian Estimation Based Online Client Selection (BEBOCS)

Input: Queues backlog vector: Z_t, Frequency vector: R_t
Output: Selected clients: S_t
Set a prior distribution $\mathcal{P}(\Theta)$;
Calculate $\mathcal{L}(R_t \mid \Theta)$ according to Eq. (26);
Calculate $\mathcal{M}(R_t \mid \Theta)$ according to Eq. (27);
Calculate $\psi(\Theta \mid R_t)$ according to Eq. (28);
$\left[\hat{d}_1, \cdots, \hat{d}_N\right] = \mathbb{E}\left[\psi(\Theta \mid R_t)\right]$;
for $n = 1 : N$ **do**
    $-\alpha \sum_{n=1}^{N} x_{tn}\left(q_n\theta_n\hat{d}_n + Z_{tn}\right) + \gamma_n \sum_{n=1}^{N} Z_{tn}$;
end for
Select the first m largest clients S_t;

- We calculate the joint probability function using the observed samples:

$$\begin{aligned} \mathcal{L}(R_t \mid \Theta) &= \mathcal{P}\left(r_{t,1}, \cdots, r_{t,N} \mid \Theta\right) \\ &= \prod_{i\in\mathcal{N}} \mathcal{P}\left(r_{t,i} \mid \Theta\right). \end{aligned} \tag{26}$$

- We calculate the probability of the marginal distribution

$$\begin{aligned} \mathcal{M}(R_t \mid \Theta) &= \int_{\Theta} \mathcal{H}(R_t \mid \Theta)\, d\Theta \\ &= \int_{\Theta} \mathcal{P}(\Theta)\, \mathcal{L}(R_t \mid \Theta)\, d\Theta, \end{aligned} \tag{27}$$

  where $\mathcal{H}(R_t \mid \Theta) = \mathcal{P}(\Theta)\, \mathcal{L}(R_t \mid \Theta)$.
- We use Bayesian formula to get the posteriori distribution for the parameters

$$\psi(\Theta \mid R_t) = \frac{\mathcal{P}(\Theta)\, \mathcal{L}(R_t \mid \Theta)}{\int_{\Theta} \mathcal{P}(\Theta)\, \mathcal{L}(R_t \mid \Theta)\, d\Theta}, \tag{28}$$

  where $\psi(\Theta \mid R_t)$ is the posteriori distribution for the parameters. We calculate $\mathbb{E}\left[\psi(\Theta \mid R_t)\right]$ as the estimation of the parameters.

Based on the Bayesian estimation, we propose a online clients selection algorithm, as summarized in Algorithm 2. When each training round begins, the server can calculate queues backlog vector Z_t and the frequency vector R_t. Then, the server can estimate the probability of successful local model upload of each clients $\hat{d}_n, \forall n \in \mathcal{N}$. Hence, the server can select a subset of clients S_t as an optimal solution of $\mathcal{P}3$.

C. THEORETICAL ANALYSIS FOR PROPOSED ALGORITHM

The time complexity of Algorithm 2, fairness guarantee, and convergence of the Algorithm 2 are analyzed theoretically.

1) TIME COMPLEXITY

Because there are N clients participating in FL, the time complexity for calculating Eq. (26) - Eq. (28) is $O(N)$. Furthermore, the time complexity for computing the effective data contribution of N clients is also $O(N)$. The backlog of each client's virtual queue is known. Our objective is to select the first m clients based on a weighted value of their effective data contributions and the backlogs of the virtual queues. Therefore, we just need to sort all clients by the weighted value. On average, the time complexity of quicksort is $O(NlogN)$. The time complexity of our BEBOCS is $O(NlogN)$.

2) FAIRNESS GUARANTEE

If the long-term fairness constraint in Eq. (11) is not satisfied, the backlogs of the virtual queues will grow to infinity. Thus, to guarantee the long-term fairness constraint, we need to make sure that all virtual queues are mean rate stable. The following theorem shows that our BEBOCS algorithm can guarantee the stability of the virtual queues.

Theorem 1: The proposed BEBOCS algorithm guarantees the long-term fairness constraint.

Proof: In our algorithm, $\mathcal{A}(t)$ is stationary. Thus, according to the Theorem 4.5 in [18], there exists an $\mathcal{A}$-only policy ϑ, for $\forall\epsilon > 0$ the following two inequality holds:

$$\mathbb{E}\left[\sum_{n=1}^{N} x_{t,n}^{\vartheta} q_n \theta_n I_{t,n}\right] \leq \sum_{n=1}^{N} x_{t,n}^{*} q_n \theta_n I_{t,n} + \epsilon,$$
$$\gamma_n \leq \mathbb{E}\left[x_{t,n}^{\vartheta}\right] + \epsilon. \tag{29}$$

Based on the definition of Lyapunov drift, we have

$$\begin{aligned}
&\mathbb{E}\left[L\left(Z_{t+1}\right) - L\left(Z_t\right)\right] \\
&\quad\leq \frac{N}{2} + \sum_{n=1}^{N} \gamma_n Z_{t,n} - \mathbb{E}\left[\sum_{n=1}^{N} x_{t,n} Z_{t,n} \mid Z_t\right] \\
&\quad\leq C + \sum_{n=1}^{N} \gamma_n Z_{t,n} - \mathbb{E}\left[\sum_{n=1}^{N} x_{t,n}\left(Z_{t,n} + \gamma_n q_n \theta_n I_{t,n}\right)\right],
\end{aligned} \tag{30}$$

where $C = \frac{N}{2} + mq\theta\gamma_n$, $q = \max_{n\in\mathcal{N}}\{q_n\}$, and $\theta = \max_{n\in\mathcal{N}}\{\theta_n\}$. And a lower bound of the last term in the right hand side of Eq. (30) is

$$\begin{aligned}
&\mathbb{E}\left[\sum_{n=1}^{N} x_{t,n}\left(Z_{t,n} + \alpha\theta_n q_n It,n\right) \mid Z_t\right] \\
&\qquad\geq \mathbb{E}\left[\sum_{n=1}^{N} x_{t,n}^{\vartheta}\left(Z_{t,n} + \alpha\theta_n q_n It,n\right) \mid Z_t\right] \\
&\qquad\geq \mathbb{E}\left[\sum_{n=1}^{N} x_{t,n}^{\vartheta} Z_{t,n} \mid Z_t\right] \\
&\qquad\geq \sum_{n=1}^{N}\left(\gamma_n - \epsilon\right) Z_t.
\end{aligned} \tag{31}$$

Let $\epsilon \to 0$, we can deduce the following inequality

$$\mathbb{E}\left[\sum_{n=1}^{N} x_{t,n}\left(Z_{t,n} + \alpha\theta_n q_n It,n\right) \mid Z_t\right] \geq \sum_{n=1}^{N} \gamma_n Z_{t,n}. \tag{32}$$

Then we substitute (32) into (30), and we add both sides of inequality from $t = 0$ to $t = T$ simultaneously. Based on $L(Z_0) = 0$ and $\mathbb{E}\left[Z_{t,n}\right]^2 \leq \mathbb{E}\left[Z_{t,n}^2\right]$, we can deduce that

$$\sum_{n=1}^{N} \mathbb{E}\left[Z_{t,n}\right]^2 \leq \sum_{n=1}^{N} \mathbb{E}\left[Z_{t,n}^2\right] \leq 2TC. \tag{33}$$

Based on Jensen's inequality, we can deduce that

$$\lim_{T\to\infty} \sum_{n=1}^{N} \leq \lim_{T\to\infty} \sqrt{\frac{2NC}{T}} = 0, \tag{34}$$

which implies $\lim_{T\to\infty} \frac{\mathbb{E}[Z_T]}{T} = 0$, *i.e.*, all virtual queues are mean rate stable. This ensures that the long-term fairness constraint holds. ■

When the long-term fairness constraint in (11) is violated, there will be some virtual queues with infinite length orientation, which means some clients are always selected. The above theorem shows that the virtual queues are mean rate stable. This implies our BEBOCS can give clients a fair selection rate.

3) CONVERGENCE ANALYSIS

We prove the convergence of the proposed performance promoting FL from the perspective of functional analysis. Banach fixed point theorem states that if f is a contraction mapping on a complete metric space, then there is a unique x such that $f(x) = x$ [27]. We make a common assumption and take advantage of Banach fixed point theorem to prove the convergence of our algorithm.

Assumption 1: The loss function of all clients $F_n(w)$, $\forall n \in \mathcal{N}$ are m-strong convex and L-smooth, *i.e.*,

$$\langle \nabla F_n(w) - \nabla F_n(\bar{w}), w - \bar{w}\rangle \geq m||w - \bar{w}||, \tag{35}$$
$$||\nabla F_n(w) - \nabla F_n(\bar{w})|| \leq L||w - \bar{w}||. \tag{36}$$

Theorem 2: If all loss function $F_n(w), n \in \mathcal{N}$ satisfy Assumption 1, given $m > L > 0$, the reweight aggregation can ensure that an appropriate learning rate η can be selected to make Algorithm 1 convergent.

Proof: First, we prove that the global loss function is L-smooth and m-strong convex. Due to $f(w) = \sum_{n\in S_t} F_n(w)$, where S_t is the selected clients in t-th round.

$$||\nabla f(w) - f(\bar{w})|| \leq \sum_{n\in S_t} p_{t,n} L||w - \bar{w}||. \tag{37}$$

And the reweight aggregation rule ensures that $\sum_{n\in S_t} p_{t,n} = 1$. Thus we have

$$||\nabla f(w) - f(\bar{w})|| \leq L||w - \bar{w}||, \tag{38}$$

which means the global loss function is L-smooth. Similarly, we have

$$\begin{aligned}
\langle \nabla f(w) - \nabla f(\bar{w}), w - \bar{w}\rangle &\geq \sum_{n\in S_t} p_{t,n} m\|w - \bar{w}\| \\
&= m\|w - \bar{w}\|,
\end{aligned} \tag{39}$$

which means the global loss function is m-strong convex. Then we can infer that the following inequality is true for $\forall w$ and $\forall \bar{w}$:

$$\|\nabla f(w) - \nabla f(\bar{w})\| \leq L\|w - \bar{w}\|,$$
$$\langle \nabla f(w) - \nabla f(\bar{w}), w - \bar{w}\rangle \geq m\|w - \bar{w}\|. \quad (40)$$

The update of the global model can be denoted as $w_g^{t+1} = w_g^t - \eta \nabla f\left(w_g^t\right)$. Let $H(w) \triangleq w - \eta \nabla f(w)$, we have

$$\|H(w) - H(\bar{w})\| = \|w - \bar{w} - \eta(\nabla f(w) - \nabla f(\bar{w}))\|$$
$$\leq \left(1 - 2m\eta + 2\eta^2 L^2\right)\|w - \bar{w}\|. \quad (41)$$

In this case, we can choose an appropriate learning rate η to make sure $0 < 1 - 2m\eta + 2\eta^2 L^2 < 1$, such that $H(w)$ is a contraction mapping. This implies our algorithm is convergent. ■

The update of the global model can be regarded as the gradient descent of the global loss function, which is proved to be a contraction mapping. This means that if we update the global model by $w_g^{t+1} = w_g^t - \eta \nabla f\left(w_g^t\right)$, then there exists a unique w_g^* such that $w_g^* = w_g^* - \eta \nabla f\left(w_g^*\right)$

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed FedBoost compared with FedAvg [2] and FedCS [17] on four real world datasets.

A. EXPERIMENTAL SETTINGS

1) DATASETS

We use three real world datasets, *i.e.*, MNIST [28], FMNIST [29], and CIFAR-10 [30]. The MNIST dataset consists of 70, 000 images, including 60, 000 for training and 10, 000 for testing. The FMNIST dataset contains 10 categories of images, including 60, 000 training samples and 10, 000 test samples. The CIFAR-10 dataset consists of 10 different categories of images, including 50, 000 training samples and 10, 000 test samples. We adopt a heterogeneous partitioning. In heterogeneous partitioning, each client has a different amount of local data. The dataset is divided by category so that each client has samples in different categories. In each category, the data samples are divided to clients via Dirichlet distribution, creating an imbalance in the amount of data.

2) MODELS

(1) Model for MNIST: On MNIST, we adopt a CNN model with the following structure: First, a 3×3 convolutional layer with 1 in-channel and 32 out-channels, followed by a 2×2 max pooling layer whose stride is 2. Then, a 3×3 convolutional layer with 32 in-channels and 64 out-channels, followed by a dropout function with 0.25 dropout rate. We use the Flatten function to flatten the input tensor to a one-dimensional tensor. Next, a liner layer with 9216 in-features and 128 out-features, followed by a dropout function with 0.5 dropout rate. Finally, a liner layer with 128 in-features and 10 out-features, followed by a ReLu function. (2) Model for FMNIST: On FMNIST, we adopt a CNN model with the following structure: Two 3×3 convolutional layers with 1 in-channel, 32 out-channels and 32 in-channels, 64 out-channels, respectively. Then, there's a 2×2 pooling layer. Next, two linear layers with 3136 in-features, 120 out-features and 120 in-features, 10 out-features, respectively. (3) Model for CIFAR-10: On CIFAR-10, we adopt a CNN model with the following structure: First, a 5×5 convolutional layer with 3 in-channels and 6 out-channels, followed by a 2×2 max pooling layer. Then, a 5×5 convolutional layer with 6 in-channels and 16 out-channels, followed by three linear layers whose in-features and out-features are 400 and 120, 120 and 84, and 84 and 10, respectively.

3) HYPERPARAMETERS

The global epochs T is set to 200 for MNIST and 150 for the other datasets, and the number of local training rounds is set to 1. The clients number in total N is set to 60, and the selected clients number per round is set to 30. We set local training batch size to 10, and use SGD optimizer to train local models. The learning rate η is set to 0.03 with a weight decay 0.001, *i.e.*, $\eta_{t+1} = (1 - 0.001)\eta_t$. We set d_n to 0.8 for all clients, and we set the estimate values to 1 for all clients at the begining of the evaluation. We set the non-IID degree to 0.5 for Dirichlet distribution. And we set $\gamma_n = \frac{1}{N}, \forall n \in \mathcal{N}$.

4) BASELINES

We compare our proposed FedBoost with the following baselines: (1) FedAvg [2]: FedAvg is one of the most popular FL algorithms, which randomly selects the clients with an equal probability. (2) FedCS [17]: FedCS selects m clients that satisfy the constraints in a greedy way.

5) METRIC

We evaluate the performance of the global model with the following metrics: (1) Average test accuracy: We compare the average test accuracy of the global model, *i.e.*, $\frac{1}{N}\sum_{n=1}^{N} acc_n$, where acc_n is the test accuracy of the global model on client n's test dataset. (2) Test loss: We compare the average test loss of the global model, *i.e.*, $\frac{1}{N}\sum_{n=1}^{N} loss_n$, where $loss_n$ is the test loss of the global model on client n's test dataset. (3) Value of Lyapunov function: We monitor the changes in the Lyapunov function (as defined in Eq. (19)) of our FedBoost across training rounds. If the value of the Lyapunov function consistently remains above a certain threshold, it indicates that the long-term fairness constraint is upheld. (4) Probability: We determine the probability of each client successfully uploading the model to the server through our FedBoost. And compare it to the real value.

B. COMPARISON OF AVERAGE TEST ACCURACY

Figure 3 depicts the comparison of average test accuracy between our proposed FedBoost and the baselines. Because

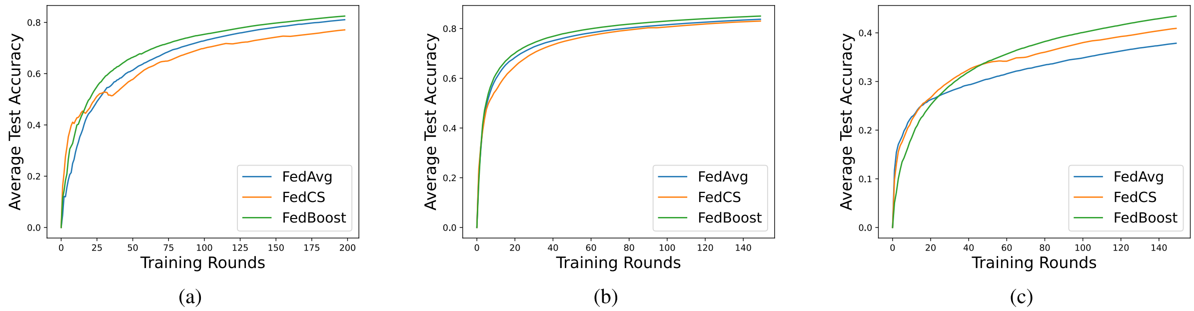

FIGURE 3. Comparison of average test accuracy on three datasets: (a) MNIST, (b) FMNIST, (c) CIFAR-10.

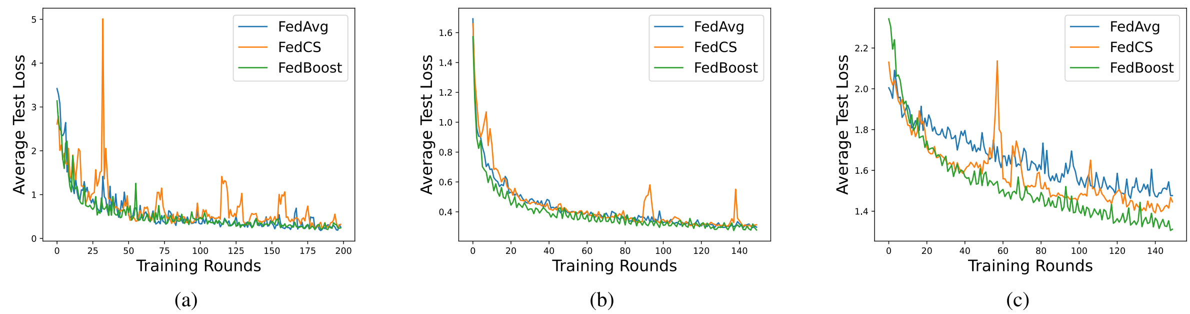

FIGURE 4. Comparison of test loss on three datasets: (a) MNIST, (b) FMNIST, (c) CIFAR-10.

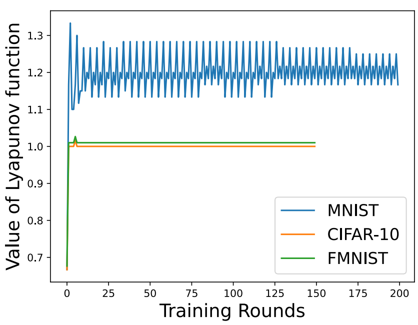

FIGURE 5. Value of Lyapunov function of our FedBoost on MNIST, FMNIST, and CIFAR-10.

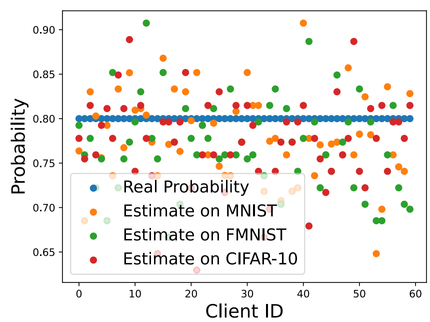

FIGURE 6. Estimation of probability of clients successfully upload local model by our FedBoost on MNIST, FMNIST, and CIFAR-10.

the classification task on MNIST and FMNIST datasets is relatively simple, the effect of FedCS algorithm on these two data sets is not ideal. On the CIFAR-10 dataset, the effect of FedCS is significantly better than that of FedAvg. The goal of FedCS is to select m clients to participate in training under constrained conditions, so as to promote model performance improvement. When there is a straggler problem, our FedBoost maximizes the clients' effective data contribution to the global model, thereby improving the test accuracy of the global model. Consequently, our FedBoost outperforms FedCS.

Specifically, with 200 training rounds on MNIST, our FedBoost achieves a average test accuracy of 82.46%, higher than the value of 81.05% and 77.08% achieved by FedAvg and FedCS, respectively. With 150 training rounds on FMNIST, our FedBoost achieves a average test accuracy of 85.02%, 1.28% higher than FedAvg and 2% higher than FedCS. Our FedBoost exhibits a distinct advantage when applied to the CIFAR-10 dataset. With 150 training rounds on CIFAR-10, our FedBoost achieves a average test accuracy of 43.44%, 5.59% higher than FedAvg, and 2.53% higher than FedCS.

C. COMPARISON OF AVERAGE TEST LOSS

Figure 4 depicts the comparison of test loss between the proposed FedBoost and the baselines. The average test loss of FedCS has experienced a sudden and significant increase, as evidenced by the data. The observed phenomenon can likely be attributed to the straggler problem affecting FedCS. Given that FedAvg employs a uniform clients selection probability and we maintain equal probabilities for clients dropout, the straggler problem may has a small impact on FedAvg. The random selection of clients with equal probability in FedAvg is insufficient for maximizing the effective data contribution of clients, thereby limiting its ability to effectively enhance model performance. Thus, when there's a straggler problem, our FedBoost ensures a selection rate of clients while prioritizing the inclusion of high-quality clients, thereby maximizing their contribution to the effective data of the global model and subsequently reducing test loss.

Specifically, with 200 training rounds on MNIST, our FedBoost achieves a test loss of 0.2402, which is lower than the value of 0.2616 and 0.3226 achieved by FedAvg and FedCS, respectively. With 150 training rounds on FMNIST, the proposed FedBoost achieves a test loss of 0.2780, 0.0191 lower than FedAvg and, 0.0355 lower than FedCS. Our FedBoost also outperforms over FedAvg and FedCS in terms of testing loss on the CIFAR-10, corresponding to the average testing accuracy. With 150 training rounds on CIFAR-10, our FedBoost achieves a test loss of 1.3117, 0.1646 lower than FedAvg, and 0.1341 lower than FedCS.

D. MEARUSRES OF FAIRNESS

Figure 5 depicts the length of virtual queues on MNIST, FMNIST, and CIFAR-10. The Lyapunov function exhibits fluctuations on the MNIST dataset. However, it is evident that this value possesses an upper bound U. This implies $\lim_{t\to\infty} \frac{1}{t} Z_{t,n} \leq \lim_{t\to\infty} \frac{U}{t} = 0, \forall n \in \mathcal{N}$, *i.e.*, all virtual queues are mean rate stable. The value of Lyapunov function on FMNIST and CIFAR-10 is almost a constant. Obviously, virtual queues are mean rate stable on both datasets. In conclusion, our FedBoost ensures the long-term fairness constraints across all datasets.

E. BAYESIAN ESTIMATION OF PROBABILITY

Figure 6 depicts the estimation of the probability of clients successfully uploading the model after the training process. In the experiments conducted on the three datasets, a significant proportion of clients were selected between 50 and 65 times. If we employ frequency of clients successfully uploading the model to estimate the probability of clients successfully uploading the model, such a number of selecting times is obviously not enough. By employing Bayesian estimation, we can obtain highly precise probability estimations. While certain estimates exhibit significant deviations from the true value (0.8), at least 85% of values fall within the range of 0.70 to 0.85.

VI. CONCLUSION

To promote the performance of the global model when straggler problem exists, we have proposed to maximize clients' cumulative effective data contribution to the global model as the optimization object. Therefore, we have proposed a Bayesian estimation based FL framework, named FedBoost, and we have analyzed its fairness and convergence theoretically. We have employed Bayesian estimation to address the challenge of unknown probabilities associated with clients successfully uploading the model. Extensive experiments on three real-world datasets have shown that our proposed FedBoost had the highest average test accuracy and the lowest test loss compared with FedAvg and FedCS.

REFERENCES

[1] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[3] S. Aich, N. K. Sinai, S. Kumar, M. Ali, Y. R. Choi, M.-I. Joo, and H.-C. Kim, "Protecting personal healthcare record using blockchain & federated learning technologies," in *Proc. 23rd Int. Conf. Adv. Commun. Technol. (ICACT)*, Pyeongchang, South Korea, Feb. 2021, pp. 109–112.

[4] S. Park, G. Kim, J. Kim, B. Kim, and J. C. Ye, "Federated split task-agnostic vision transformer for COVID-19 CXR diagnosis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24617–24630.

[5] J. Passerat-Palmbach, T. Farnan, R. Miller, M. S. Gross, H. Leigh Flannery, and B. Gleim, "A blockchain-orchestrated federated learning architecture for healthcare consortia," 2019, *arXiv:1910.12603*.

[6] A. M. Seid, J. Lu, H. N. Abishu, and T. A. Ayall, "Blockchain-enabled task offloading with energy harvesting in multi-UAV-assisted IoT networks: A multi-agent DRL approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3517–3532, Dec. 2022.

[7] J. Lu, H. Liu, R. Jia, J. Wang, L. Sun, and S. Wan, "Towards personalized federated learning via group collaboration in IIoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 8, pp. 8923–8932, Aug. 2023.

[8] S. AbdulRahman, S. Otoum, O. Bouachir, and A. Mourad, "Management of digital twin-driven IoT using federated learning," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3636–3649, Nov. 2023.

[9] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya, "Stochastic client selection for federated learning with volatile clients," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20055–20070, Oct. 2022.

[10] H. Zhu, Y. Zhou, H. Qian, Y. Shi, X. Chen, and Y. Yang, "Online client selection for asynchronous federated learning with fairness consideration," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2493–2506, Apr. 2023.

[11] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-IID data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Atlanta, GA, USA, Dec. 2020, pp. 15–24.

[12] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," 2019, *arXiv:1903.03934*.

[13] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[14] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma, "Improving data quality: Consistency and accuracy," in *Proc. VLDB*. Vienna, Austria: Univ. Vienna, Sep. 2007, pp. 315–326.

[15] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. 3rd Mach. Learn. Syst. Conf.*, Austin, TX, USA, Mar. 2020, pp. 429–450.

[16] H. Yuan and T. Ma, "Federated accelerated stochastic gradient descent," in *Proc. NeurIPS*, Dec. 2020, pp. 5332–5344.

[17] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. ICC*, Shanghai, China, May 2019, pp. 1–7.

[18] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Syn. Lecture Commun. Netw.*, vol. 3, no. 1, pp. 1–211, Jan. 2010.

[19] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, and C. Miao, "Federated learning at mobile edge networks: A tutorial," in *Federated Learning Over Wireless Edge Networks*. Cham, Switzerland: Springer, 2022, pp. 1–50.

[20] C. Battiloro, P. D. Lorenzo, M. Merluzzi, and S. Barbarossa, "Lyapunov-based optimization of edge resources for energy-efficient adaptive federated learning," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 265–280, Mar. 2023.

[21] Y. Niu and W. Deng, "Federated learning for face recognition with gradient correction," in *Proc. AAAI*, Feb./Mar. 2022, pp. 1999–2007.

[22] W. Y. B. Lim, Z. Xiong, C. Miao, D. Niyato, Q. Yang, C. Leung, and H. V. Poor, "Hierarchical incentive mechanism design for federated machine learning in mobile networks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9575–9588, Oct. 2020.
[23] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, Jun. 2019, pp. 4615–4625.
[24] J. Lu, H. Liu, R. Jia, Z. Zhang, X. Wang, and J. Wang, "Incentivizing proportional fairness for multi-task allocation in crowdsensing," *IEEE Trans. Services Comput.*, early access, Oct. 18, 2023, doi: 10.1109/TSC.2023.3325636.
[25] Z. Tong, J. Cai, J. Mei, K. Li, and K. Li, "Dynamic energy-saving offloading strategy guided by Lyapunov optimization for IoT devices," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19903–19915, Oct. 2022.
[26] L. J. Bain and M. Engelhardt, "Conditional probability and independence," in *A Modern Introduction to Probability and Statistics*. London, U.K.: Springer, 2005.
[27] L. V. Kantorovich and G. P. Akilov, "Linear operators and functionals," in *Functional Analysis*, 2nd ed. Oxford, U.K.: Pergamon, 1982, pp. 127–166.
[28] Y. Deng, F. Lyu, J. Ren, Y. C. Chen, P. Yang, Y. Zhou, and Y. Zhang, "Improving federated learning with quality-aware user incentive and auto-weighted model aggregation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4515–4529, Dec. 2022.
[29] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[30] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 18268744, 2009.

YUHANG SHENG received the B.S. degree in mathematics from the Nanjing University of Information Science and Technology, Nanjing, China, in 2021. He is currently pursuing the M.S. degree in mathematics with Zhejiang Normal University. His research interests include federated learning, Lyapunov optimization, and game theory.

LINGGUO ZENG received the B.S. degree from the Department of Information Engineering, Northeast Electric Power University, Jilin, China, in 2001, and the M.S. degree from the Computer College, East China Normal University, Shanghai, China, in 2006. He is currently a Lecturer with the College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China. His research interests include credibility calculation and federated learning.

SHUQIN CAO received the Ph.D. degree in computer science from Wuhan University, in 2023. She is currently an Assistant Professor with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. Her current research interests include the Internet of Vehicles, artificial intelligence, and traffic data mining.

QING DAI received the M.S. degree in computer science and technology from Zhejiang Normal University, in 2018. She is currently an Assistant Experimentalist with Zhejiang Normal University. Her research interests include mobile crowdsensing, incentive mechanisms, and network planning.

SHASHA YANG received the M.S. degree from the Department of Computer Science and Engineering, Zhejiang Normal University, in 2020, where she is currently pursuing the Ph.D. degree in intelligent education. Her research interests include federated learning, incentive mechanisms, and student engagement.

JIANFENG LU received the Ph.D. degree in computer application technology from the Huazhong University of Science and Technology, in 2010. He was with Zhejiang Normal University, Jinhua, China, from 2010 to 2021. He was a Visiting Researcher with the University of Pittsburgh, Pittsburgh, USA, in 2013. He is currently a Professor with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include federated learning, crowd computing, and game theory.