FedACS: Federated Skewness Analytics in Heterogeneous Decentralized Data Environments

Zibo Wang*, Yifei Zhu*, Dan Wang†, and Zhu Han‡
*UM-SJTU Joint Institute, Shanghai Jiao Tong University
†Department of Computing, The Hong Kong Polytechnic University
‡Department of Electrical and Computer Engineering, University of Houston
Email: wangzibo@sjtu.edu.cn, yifei.zhu@sjtu.edu.cn, csdwang@comp.polyu.edu.hk, zhan2@uh.edu

***Abstract*—The emerging federated optimization paradigm performs data mining or artificial intelligence techniques locally on the edge devices, enabling scientists and engineers to utilize the blooming edge data with privacy protection. In such a paradigm, since data cannot be shared or gathered, data heterogeneity naturally emerges, which significantly degrades the performance of federated optimization, ultimately leading to poor quality of federated services. In this paper, we present the first work on characterizing the data heterogeneity in the framework of *federated analytics*, *i.e.*, to collectively carry out analytics tasks without raw data sharing, and use the information to create a desirable data environment via intelligent client selection. Our proposed Analytics-driven Client Selection framework, named FedACS, tackles the data heterogeneity problem in three steps. First, clients are in charge of generating insights about local data without disclosure of sensitive information. Then, the server uses these insights to infer the situation of clients' data heterogeneity based on the Hoeffding's inequality. Finally, a dueling bandit is formulated to intelligently select clients with slighter data heterogeneity to form a desirable client pool. FedACS can be universally applied to all kinds of federated optimization tasks, and gains benefits including privacy protection, infrastructure reuse, and client load reduction. To test its efficiency, we further customize it to assist federated learning, a popular scenario of federated optimization. According to experiment results, FedACS reduces the accuracy degrading by up to 65.6%, and speeds up the convergence for up to 2.4 times.**

***Index Terms*—federated analytics, data heterogeneity, federated learning, dueling bandit**

I. Introduction

As we step into the era of data explosion, an exponential amount of data are being generated by smartphones and IoT devices. In 2020, 5.3 billion people were networked via cellular service, generating 1.2 trillion digital photographs via smartphones, and 8.7 billion networked IoT devices were deployed in the world [1], [2]. These big data play an important role in driving the data science and artificial intelligence algorithms from labs to real world applications. However, with the increasing awareness of data privacy, laws and regulations, such as EU General Data Protection Regulation (GDPR) [3], are established worldwide to protect raw data from being collected into one centralized place and conducting further intelligence extraction processes.

The increasing regulations on data privacy and the growing computing capability at the edge side thus motivate the wide study of federated optimization techniques, represented by federated learning (FL), in recent years [4]. Under the orchestration of a centralized server, researchers train a deep learning model (neural network) across multiple decentralized edge devices holding local data samples. Without having access to these raw data directly, the distilled updates, usually weight information, are uploaded from the edge devices to the server for an immediate aggregation process.

While FL has been proved to be effective in multiple areas, it still focuses on the tasks and scenarios requiring complex deep learning models, like natural language processing [5], [6] or computer vision [7], [8] applications. Meanwhile, a wide range of analytic applications that rely on data science methods, like heavy hitter discovery, outlier detection, histogram construction, *etc*. are left without discussion. As can be seen, in these applications, the studied questions are no longer simply "how to collaboratively train a model to do the prediction or classification task", but rather "what is the most frequent word used by users?", "what is the underlying distribution of the dataset?", *etc*. These tasks usually do not need complicated prediction models, rather they require data insights obtained by analyzing the decentralized datasets, which becomes harder due to the restrictions on accessing raw data.

In May 2020, Google presented the next evolution of federated optimizations: *federated analytics* (FA) [9]. In the new FA framework, individual clients collectively carry out a *non-training* analytic task, rather than training a neural network in FL, and send derived insights, not just weight updates in FL, to the servers. Though the newly introduced FA still follows the federation paradigm as its predecessor, the central aggregation part and local analytics part in FA are *application-specific*, which calls for careful design to guarantee the privacy of raw data and the accuracy of the extracted insights. For example, in a federated frequent word analytic scenario [10], a prefix tree is constructed as the FA model. Edge devices provide their insights by adding a character as a leaf of the tree. The server aggregates all trees to get an estimation of the most frequent word used among devices.

This work is supported by a SJTU Explore-X Research Grant. Dan Wang's work is supported in part by GRF 15210119, 15209220, ITF-ITSP ITS/070/19FP, CRF C5026-18G, C5018-20G, PolyU 1-ZVPZ and a Huawei Collaborative Project. Zhu Han's work is partially supported by NSF EARS-1839818, CNS1717454, CNS-1731424, and CNS-1702850. The corresponding author is Yifei Zhu.

978-0-7381-3207-5/21/$31.00 ©2021 IEEE

For any federated system, because the raw data have to be processed locally in a privacy-preserving way, diverse situations at the client[1] side (usually termed as data heterogeneity, device heterogeneity, *etc*.) greatly affect the efficiency of the federated systems. For example, data heterogeneity (non-IID[2] datasets and imbalanced datasets) degrades FL with longer convergence time and lower accuracy [11], [12]. Device heterogeneity (varying availability and computation capability of the edge devices) introduces significant uncertainty to the system and affects the operation of the federated system [13], [14]. *Obviously, characterizing these heterogeneities with privacy preserved can help understand the underlying federated system and improve its quality of services.*

In this paper, we present the first work on characterizing the class distribution heterogeneity in federated systems and use this insight to create a desirable data environment via intelligent client selection. Unlike learning an unknown probability distribution from random samples [15] as studied in previous differential privacy field or manually selecting features to heuristically determine a client's data heterogeneity, we use the term *skewness*[3] to describe the severity of the local class distribution of a client skew from the global, virtually centralized, one and aim at gaining a provable estimation on it. The derived estimations about the skewness of the clients are further used to select a group of low skewness clients, creating an environment closed to the ideal IID environment, desirable by a variety of federated tasks.

However, there are several significant challenges we have to solve in order to achieve this. Since skewness measures the divergence of class distribution in one client from the global class distribution, we have to rely on aggregation of insights from multiple clients to form the skewness measure on one specific client. Specifically, first, for the clients, the class distribution information of each client has to be both representative and aggregatable. Second, for the central server, the results based on aggregating these local insights have to describe the skewness of each client in a mathematically provable and practically effective way. Third, from the federation's perspective, the solution should not introduce any potential privacy leakage when deriving insights from clients. Last, the client selection algorithm has to be capable of reliably selecting low skewness clients even when the skewness information derived in previous steps is stochastic or uncertain.

To this end, following the FA framework, we present a Federated Analtyics-driven Client Selection (FedACS) framework to help federated optimization tasks collaboratively profile the class distribution at the client side and intelligently select low skewness clients. The cycle of FedACS is synchronous to the host federated task. Each cycle of FedACS includes three parts: the insight derivation part that provides indirect insight about local data, the skewness estimation part that aggregate all insights to infer about clients' skewness, and the client selection part that iteratively selects clients with a low skewness level to participate in the federated tasks. FedACS itself is a typical instance of FA, and can be universally applied to all kinds of federated optimization tasks. To test its efficiency, we further use FL as the representative host federated task and use FedACS to assist it. The skewness estimation part is further designed to reuse the infrastructure of FL and the FA workloads can be minimized. With the assistance of FedACS, the degrading effects of FL caused by the non-IID environment are heavily reduced.

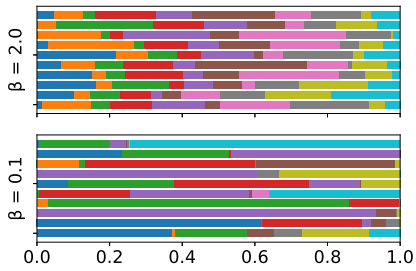


Fig. 1. Class distributions of β-Dirichlet clients with different parameters.

In summary, our contributions are

- We present the first work on federated skewness analytic following the framework of FA.
- Based on the Hoeffding's inequality, our approach quantifies the class distribution heterogeneity in the federated environment in a mathematical provable way.
- We formulate the client selection problem into a novel dueling bandit problem to cater to the unique characteristics of the client skewness estimation and solve it using a Thompson Sampling based approach.
- Implementations under various non-IID environments demonstrate that, with the assistance of FA, the host FL task can reduce $\sim 65.6\%$ of accuracy degrading caused by data heterogeneity and speed up the model convergence for $\sim 2.4\times$.

The rest of this paper is organized as follows: Section II introduces the system model and the problem formulation of FedACS. Section III provides detailed information of major components of FedACS. Section IV presents the evaluation results. Related work is surveyed in Section V, followed by the conclusion in Section VI.

II. System Model and Problem Formulation

In this section, we present the system models and formulate the problem FedACS solves. Section II-A provides a modeling of the non-IID environment. Section II-B presents an overview of FedACS. Section II-C introduces its problem formulation.

A. Non-IID Environment Modelling

In this part, we model the non-IID environment with two steps. We first model the class distribution at the client level,

[1]We use client and edge device interchangeable in this paper

[2]IID: independent and identically distributed

[3]Note that skewness has a different definition in statistics. In this paper, we follow a similar definition to describe the label distribution skew as in [4] and [11].

and then model the heterogeneity of these distributions across different clients at the population level.

The first step to model a non-IID distribution is to determine the class distribution of each client. Dirichlet distribution has been used to model non-IID environments at the client side by various FL studies [16]–[18]. It generates a list of random variables with an invariant sum, which can be naturally converted to the proportion of data belonging to each class, and is therefore an effective solution to model the client-level class distribution. Based on this, we present the following definition to describe the class distribution in one client:

Definition 1 (β-Dirichlet client)**.** *A β-Dirichlet client has class distribution following the Dirichlet distribution, with concentration parameter β.*

In practice, the skewness of clients is not only different, but also heavily diverges. For example, if an ordinary person takes photos of everything, then pictures in his/her smartphone will be close to the global distribution (or slightly skewed). Meanwhile, if a photographer attends auto-shows everyday, then his/her pictures would be heavily skewed ones.

Therefore, the next step of modeling the non-IID environment is to let the skewness levels of different clients diverge, so that the aforementioned characteristics can be properly modeled. We achieve this by having each client's concentration parameter β diverges. Since β has an uncertain but strong influence on the skewness of clients, if the clients are assigned with different values of β, the skewness of clients will diverge at the population level. Based on this, we present the final model of the non-IID environment:

Definition 2 (Dirichlet skewness environment)**.** *In a Dirichlet skewness environment, all clients are β-Dirichlet clients. Half of the client has β values following continuous uniform distribution in range $(0, x_{med}]$, and those of the rest clients are uniformly distributed in range $(x_{med}, x_{max}]$.*

In the Dirichlet skewness environment, the values of β follow a layered uniform distribution, whose median and maximum are predefined, rather than an ordinary uniform distribution. The reason is that the change of skewness is not proportional to the change of β, *e.g.* if we change β from 0.2 to 0.1, the change of client skewness is much more violent than when changing β from 1.0 to 0.9. By layering β, we guarantee that client skewness is evenly distributed to different levels.

We visualize the Dirichlet skewness environment for a better understanding. We choose two representative values of β, 0.1 and 2.0, and generate the class distributions of ten clients for each value of β. The results are shown in Fig. 1, where each row describes the class distribution in one client, and colors in each row represent raw data from different classes. As can be seen from the figure, the skewness of clients with $\beta = 0.1$ is much higher than those with $\beta = 2.0$.

B. System Overview

FedACS aims at profiling the class distribution heterogeneity in federated systems following the federated analytics framework and uses the derived insights to create a near IID environment via intelligent client selection. FedACS can work on its own as a stand-alone analytic system or form a symbiosis with other federated tasks to help improve these host tasks' quality of service. To fully demonstrate the power of FedACS, we choose the latter form and use FL as the host task in this paper. The overall structure of FedACS and the assisted FL are presented in Fig.2.

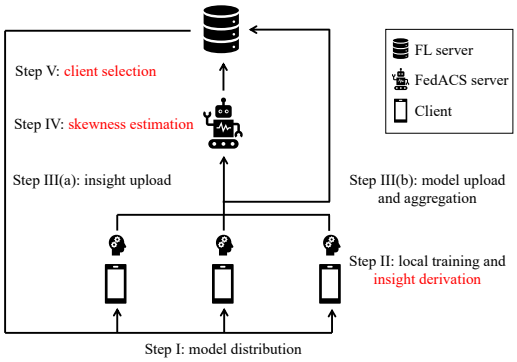


Fig. 2. An overview of FedACS assisted FL. The insight derivation part and the skewness estimation part are demonstrated in Section III-A together with Section III-B. The client selection part is demonstrated in Section III-C.

The whole system has a set of clients C represented by index $\{1, 2, 3, ...N\}$, and executes two task cycles. In the FL task cycle, each client is in charge of performing the local training phase in FL. In each round, κ clients are selected by the server to participate in FL. The focused updates are sent to the server for model aggregation. The clients have different intrinsic data skewness. Therefore, the benefit provided by each client varies. To maximize the overall benefit, FedACS introduces a new FA task cycle synchronous to the host FL cycle. This FA cycle helps the server discover the low skewness client and select those clients to join the FL training process in each round. Specifically, the FA cycle consists of three modules: insight derivation, skewness estimation, and client selection.

In the insight derivation part, each participating client generates insight, which will be utilized by the server to infer its skewness. The design of insight is of a high degree of freedom, but should not expose direct information about raw data. In this paper, the insight is in the form of the gradient to cater to the host task. Such a design reuses the infrastructure of FL and reserves the privacy protection level as the FL.

In the skewness estimation part, the server transforms the insights derived by clients into estimations of client skewness. The Hoeffding's inequality is applied to bridge the connection between the uploaded insights and the client skewness. For each participating client i, a reward value R_i is simultaneously derived as an inverse reflection of its skewness, *i.e.*, a client with a lower skewness obtains a higher R_i.

In the client selection part, FedACS receives the reward information, *i.e.*, inversed skewness estimations, from the previous part, and selects clients with low skewness to participate

in federated tasks. Since FedACS carries out an exploration-exploitation tradeoff under the challenge of stochastic and uncertain reward, it formulates this problem into a multi-dueling bandit problem. FedACS also takes into account the tradeoff between the reward maximization and the limited sample size to guarantee the performance of its host task.

C. FedACS: A Dueling Bandit Formulation

We next formulate the federated skewness analytics problem and client selection problem in FedACS into a dueling bandit problem. Dueling bandit problems consider a different scenario than the conventional stochastic bandit problems. Being firstly introduced in [19], a dueling bandit selects two arms to perform one comparison (dueling) in each round and receives the noisy comparison result of these two arms. For each pair of arms i and j, the probability for i to be stronger than j is represented by,

$$\mathbb{P}(i \succ j) = \phi(i,j) + \frac{1}{2}, \tag{1}$$

where $\phi(i,j)$ denotes the dueling preference between i and j. The goal of the conventional dueling bandit is to find the *Condorcet winner*, which beats all other arms with a probability not lower than 0.5. Multi-dueling bandit [20] extends the original dueling bandit, allowing simultaneous dueling between multiple arms and targeting at identifying multiple optimal arms.

Since we need to select multiple clients in each round to help the host task and the derived skewness estimations are uncertain over time, multi-dueling bandit naturally suits our scenario. We let participating clients in each round "duel" with each other using their R_i values, and update the bandit with the dueling results. The objective eventually is to select a set of clients that can beat others, *i.e.*, with low skewness.

Therefore, we present the formal formulation of our problem as follows. Recall that the set of clients is denoted as C. For each client $i \in C$, ψ_i indicates the quantified intrinsic skewness of client i. For each pair of clients i and j, if they are both participating clients and given rewards (R_i and R_j) in the same round, and then the quantitative comparison of R_i and R_j has a noisy negative correlation with ψ_i and ψ_j, *i.e.*,

$$\mathbb{P}(R_i > R_j) > 0.5 \iff \psi_i < \psi_j. \tag{2}$$

Similar to (1), we donate the stochastic preference between client a and b as $\phi(a,b)$,

$$\phi(i,j) = \mathbb{P}(R_i > R_j) - 0.5. \tag{3}$$

FedACS aims at

$$\underset{S'}{\operatorname{argmin}} \left\{ \sum_{t=1}^{T} \sum_{i \in S'} \phi(i^{(*)}, i) \right\}, \tag{4}$$

where S' is a fix-size set of desirable clients, T is the total number of communication rounds, and $i^{(*)}$ is the client with the lowest skewness.

We aim at finding the clients with the lowest skewness so that the regret defined in the objective function in (4) can be minimized. As can be seen in (4), our problem can be decomposed into two sub-problems.

- We need to determine the dueling results $\phi(i,j)$ in (3) based on skewness analytics.
- We need to select a set of clients to minimize the objective value in (4).

III. Skewness Analytics and Client Selection

In this section, technological details about skewness analytics and client selection algorithm to solve the previous two subproblems are present. In Section III-A, we show how the Hoeffding's inequality is employed in FedACS to estimate the client skewness. In Section III-B, we conclude a practical estimation of the client skewness, which will be used as the reward for the bandit. In Section III-C, detailed client selection algorithm in FedACS is demonstrated.

A. Connection between the Hoeffding's Inequality and the Client Skewness

In this part, we show the procedure of inferring about the client skewness based on the Hoeffding's inequality. We first apply Hoeffding's inequality to the results of gradient descent. After that, we bridge the derived value to the client skewness by converting it to the possibility of accepting a hypothesis, assuming data in the client is IID.

The Hoeffding's inequality is a statistical tool first introduced in [21]. It estimates the deviation of the average of independent random variables from its exception and provides a probabilistic bound given the deviation of $\overline{X}$ from its exception [22], [23]. As the cornerstone for federated skewness analytics, we formally present this theorem as follows.

Theorem 1 (Hoeffding's inequality)**.** *Supposed $X_1, ..., X_n$ are independent variables, $X_i \in [a_i, b_i]$, $\overline{X}$ is the average of X_i, there's*

$$\mathbb{P}(|\overline{X} - \mathbb{E}(\overline{X})|) \geq \epsilon) \leq 2exp\bigg(- \frac{2\epsilon^2 n^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \bigg). \tag{5}$$

Next, we demonstrate how the Hoeffding's inequality is applied in skewness estimation. Since the insight we used is in the form of a gradient (weight change) from the neural network, we first demonstrate how the gradient is derived, and then show how the Hoeffding's inequality is linked to it.

In the system, there are N clients in total. Donate $d_{i,m}$ as the m-th datum in the i-th client. M is the number of datum in each client. The procedure of calculating gradient in a neural network is called backward propagation. First, the client calculates a loss function $Loss(d)$ for each datum d indicating how the prediction of one datum d is closed to the truth. Then, the client averages the loss function of all data it owns to form a cost function $Cost_i$. Finally, the client calculates the weight change (gradient) of the neural network. Donate the dimension index of weight as k, the weight change are derived by the backward propagation that

$$\Delta w_i^{(k)} = \gamma \times \frac{\partial Cost_i}{\partial w^{(k)}}, \tag{6}$$

where Δw_i^k is the weight change of client i in dimension k, and γ is a preset learning rate. In FL, client i upload Δw_i, with K dimensions, to the server.

Above is the full procedure of generating gradients in a neural network. Then we link the final result Δw_i to the Hoeffding's inequality.

Donate $z_{i,m}^{(k)}$ as the k-th dimension of gradient derived from the m-th datum in the i-th client, times the learning rate γ.

$$z_{i,m}^{(k)} = \gamma \times \frac{\partial Loss(d_{i,m})}{\partial w^{(k)}} \tag{7}$$

Donate $z_i^{(k)}$ as the average of $z_{i,m}^{(k)}$, consider the calculation of the weight change in deep learning in (6),

$$\begin{aligned} z_i^{(k)} &= \frac{1}{M}\sum_{m=1}^{M} \gamma \frac{\partial Loss(d_{i,m})}{\partial w^{(k)}} \\ &= \gamma \frac{\partial \frac{1}{M}\sum_{m=1}^{M} Loss(d_{i,m})}{\partial w^{(k)}} \\ &= \gamma \frac{\partial Cost_i}{\partial w^{(k)}} \\ &= \Delta w_i^{(k)}. \end{aligned} \tag{8}$$

$z_{i,m}^{(k)}$ are derived from different independent samples, so they are also independent random variables, while the uploaded weight change $\Delta w_i^{(k)}$ is the average value of $z_{i,m}^{(k)}$. We apply the Hoeffding's Inequality in (5) to $z_{i,m}^{(k)}$, and get p_i^k ,the probability that k-dimension of weight change from client i diverges from its exception for a fixed value ϵ. Namely,

$$\begin{aligned} p_i^k &= \mathbb{P}(|\Delta w_i^{(k)} - \mathbb{E}(\Delta w_i^{(k)})|) \geq \epsilon) \\ &\leq 2exp\bigg(-\frac{2\epsilon^2 M^2}{\sum_{j=1}^{M}(b^{(k)} - a^{(k)})^2}\bigg) \\ &= 2exp\bigg(-\frac{2\epsilon^2 M}{(b^{(k)} - a^{(k)})^2}\bigg). \end{aligned} \tag{9}$$

$b^{(k)}$ and $a^{(k)}$ are the upper and lower bounds of $z_{i,m}^{(k)}$. To make our estimation comparable for different client (i) and datum (j), we use the same bounds $b^{(k)}$ and $a^{(k)}$ instead of $b_{i,m}^{(k)}$ and $a_{i,m}^{(k)}$. We are safe to do this because the Hoeffding's inequality in (5) does not require a tight bound.

Recall that $z_{i,m}^{(k)}$ are gradient derived by one datum. Although the server does not have knowledge about the datum $d_{i,m}$, we can estimate its skewness by the skewness of $z_{i,m}^{(k)}$, which is a mapping of $d_{i,m}$. Furthermore, in FL, values of $z_{i,m}^{(k)}$ is also private. Therefore, FedACS use the Hoeffding's inequality to estimate skewness of $z_{i,m}^{(k)}$ based on $\Delta w_i^{(k)}$.

In order to link p_i^k to the client skewness, we start with a hypothesis H:

H: Data in client i is IID distributed.

We utilize H via a generalized *reduction to absurdity*: we first accept H anyway, so that we can calculate p_i^k with (9). Since the value in (9) is a possibility bound, it represents how rare the situation of accepting H is. A rarer situation indicates that we are less likely to accept H in the first place, which means that data distribution is distant from the assumption made by H, indicating a higher skewness.

Following the aforementioned rationale, we first link the possibility of accepting H to the value of p_i^k derived by Hoeffding's inequality, as presented in Lemma 1.

Lemma 1. *p_i^k has a positive correlation to the likelihood of accepting H.*

Proof. See Appendix A □

The likelihood of accepting H can be naturally linked to client skewness: if we have a high confidence to claim that client i is IID, client i will be more likely to have a low skewness. From the insight above, we can build the connection between client skewness and p_i^k, that high p_i^k indicates low skewness of client i.

Also, when the assumption is made that data in client i is IID, the expectation of $z_i^{(k)}$ should be equal to the global one $z^{(k)}$. Based on that, a new expression of p_i^k is derived as a side product from Lemma 1, *i.e.*,

$$\begin{aligned} p_i^k &= \mathbb{P}(|\Delta w_i^{(k)} - \mathbb{E}(z^{(k)})|) \geq \epsilon) \\ &\leq 2exp\bigg(-\frac{2\epsilon^2 M}{(b^{(k)} - a^{(k)})^2}\bigg). \end{aligned} \tag{10}$$

B. Derivation of the Rewards

In this part, we transform the skewness estimation in (10) into a more practical representation R_i, which will be used by the bandit in the next component of FedACS. We first solve a challenge by providing an estimation for a variable to be used, whose exact value is impossible to obtain. Next, we combine multiple dimensions of the gradient to a single value R_i, so that the credibility of our estimation is increased.

According to (10), the calculation of p_i^k requires the value of the expectation of $z^{(k)}$, which is used in deriving ϵ. The exact value of $\mathbb{E}(z^{(k)})$ is the average of $z_{i,m}^{(k)}$ of all data in all clients ($\forall i, m$). However, not all clients participate in each round. Therefore, the exact value is impossible to obtain. To tackle this challenge, we used the average of all data in all participating clients instead, as a reliable estimation.

The rationale of estimating $\mathbb{E}(z^{(k)})$ is concluded into the following theorem:

Theorem 2. *The expectation of $z^{(k)}$ can be estimated by the average of uploaded weight changes of all participating clients. Namely,*

$$\mathbb{E}(z^{(k)}) \approx \overline{\Delta w}^{(k)}, \tag{11}$$

where $\overline{\Delta w}^{(k)}$ indicates the average uploaded weight changes of all participating clients at dimension k.

Proof. See Appendix B □

As the estimation of $\mathbb{E}(z^{(k)})$ has been given, we are able to propose a more practical estimation of client skewness. Rewrite (10), we have,

$$P_i^k = 2exp\bigg(-\frac{2(\epsilon^{(k)})^2 M}{(b^{(k)} - a^{(k)})^2}\bigg) \tag{12}$$

where,

$$\epsilon^{(k)} = |\Delta w_i^{(k)} - \overline{\Delta w}^{(k)}|. \tag{13}$$

Eq. (12) provides an estimation about the skewness of all clients. However, it only utilizes one dimension of weight changes. It will be more accurate and robust when considering estimations for all dimensions.

Since the combination is not mathematically purposeful, we choose to multiply R_i^k among all dimensions. The rationale of choosing multiplication and the detailed procedure of combining all dimensions can be found in appendix C. A final result of combination R_i is derived:

$$R_i = -||\Delta w_i - \overline{\Delta w}||_2 \tag{14}$$

where Δw_i indicates the uploaded gradient from client i, and $\overline{\Delta w}$ indicates the average of uploaded gradients among all participating clients. The values of R_i have a negative correlation to clients' skewness, *i.e.*, a higher R_i value indicates a lower client skewness, which is desired by the bandit.

C. Client Selection: A Thompson Sampling Approach

In this part, the detailed algorithm of the multi-dueling bandit is present. FedACS builds a multi-dueling bandit based on INDSELFSPARRING, an effective multi-dueling bandit algorithm [20]. It borrows the power of Thompson sampling to handle the dueling results in the way of stochastic bandits.

Compared to the original INDSELFSPARRING algorithm, the client selection scheme of FedACS takes an extra tradeoff into consideration in order to guarantee its effectiveness in assisting FL, that the balance of data number to be utilized. If we decrease the utilized data, FedACS can select the most perfect clients with low skewness, but the neural network will lack raw samples for training; if the utilized data increases, the neural network will be trained with sufficient samples, but the overall skewness of participating client will be higher. To tackle the challenge, we introduce a meta parameter λ, which describes our tolerance to client skewness: a higher λ indicates we tolerant more heavily skewed clients to participate in FL, in order to feed the neural network with more raw samples.

With the introduction of λ, the procedure of client selection is modified. In each round, the bandit first provides a client pool consisting of λN desirable clients. Then, we randomly select κ clients from the client pool as the participating clients. Finally, duels are performed by the participating clients, and the bandit is updated with the dueling results.

When λ takes its minimum, κ/N, it becomes a vanilla multi-dueling bandit, which always tries to use top κ clients with the lowest skewness. On the contrary, when $\lambda = 1$, it falls back to randomly selecting clients (the default policy in the existing FL protocol). By wisely selecting the parameter λ, we can both restrict participating clients to be with low skewness, and provide the neural network with sufficient raw sample by extending the client pool.

The detailed algorithm for selecting clients and updating the key parameters with rewards are presented in Algorithms 1 and 2, respectively. In Algorithm 1, the bandit requires all clients to sample from their own beta distributions, and repeatedly chooses clients with the highest sampling result. Compared to the original INDSELFSPARRING, we add a new feature that the bandit first forms a desirable client pool S' with size λN, and then randomly select κ participant from S' to form the actual selected client set S.

In Algorithm 2, the parameters of each client A_i and B_i are modified based on its dueling results, which shape its beta distribution. If a client is more likely to defeat others in duels, its beta distribution will be more likely to return high sampling results. Unlike traditional dueling bandits where the dueling results are naturally given, in FedACS, participating clients in the same round have to generate the dueling results beforehand, by comparing their R_i values with each other.

Algorithm 1 Process of client selection

Input: Parameters for beta distribution: A, B; Number of clients: N, Set of all clients: C; Number of clients to be selected in each round: κ; Skewness tolerance parameter: λ.

Output: Set of selected clients: S.

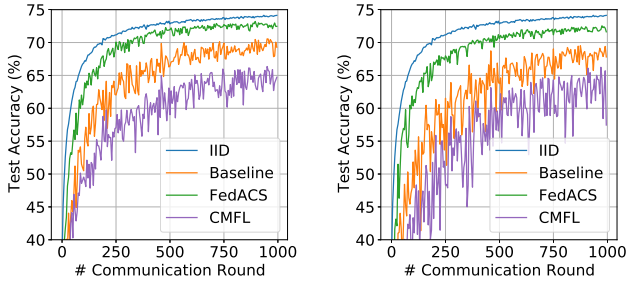
$S' \leftarrow$ empty set ▷ the desirable client pool
$P \leftarrow N \cdot \lambda$ ▷ size of the desirable client pool
for $t = 1, 2, ..., P$ **do** ▷ repeat Thompson sampling
  for $i \in C$ **do**
    sample θ_i by *Beta*$(A_i + 1, B_i + 1)$
  end for
  $g \leftarrow \mathrm{argmax_i}\theta_\mathrm{i}$ ▷ a desirable client
  append g to S'
  remove g from C
end for
$S \leftarrow$ randomly draw κ clients from S'
return S

Algorithm 2 Key parameters update

Input: Parameters for beta distribution: A, B; Set of participating clients: S; Rewards of participating clients: R; Learning rate: η.

Output: Updated parameters for beta distribution: A', B'.

$A' \leftarrow A$
$B' \leftarrow B$
for $i \leftarrow$ clients in S **do**
  for $j \leftarrow$ clients in S **do**
    if $R_i > R_j$ **then** ▷ dueling between participants
      $A'_i \leftarrow A'_i + \eta$
      $B'_j \leftarrow B'_j + \eta$
    end if
  end for
end for
return A', B'



(a) Low (b) High

Fig. 3. Test accuracy v.s. communication rounds on different heterogeneity settings.

TABLE I
SUMMARY OF TERMINAL ACCURACY AND RELATIVE IMPROVEMENT

Environment	Method	Accuracy (%)	Improvement (%)
Low	IID	74.0	100
	baseline	69.7	0
	CMFL	64.8	−112.2
	FedACS	72.5	65.6
High	IID	74.0	100
	baseline	68.4	0
	CMFL	62.9	−96.7
	FedACS	72.1	65.5

TABLE II
SUMMARY OF ROUNDS TO TARGET AND RELATIVE SPEEDUP

Environment	Method	Rounds to target	Speedup
Low	IID	85	3.2x
	baseline	270	1.0x
	CMFL	620	0.4x
	FedACS	130	2.1x
High	IID	85	4.3x
	baseline	365	1.0x
	CMFL	915	0.4x
	FedACS	155	2.4x

IV. Performance Evaluation

A. Experiment Setup

Settings: We evaluate FedACS assisted FL on a popular dataset, CIFAR-10 [24]. Raw data are distributed to 200 clients with the Dirichlet skewness environment defined according to Definition 2. In this paper, we provides two practical settings on the Dirichlet skewness environment: *low heterogeneity*, where $x_{med} = 0.2, x_{max} = 3$, and *high heterogeneity*, where $x_{med} = 0.1, x_{max} = 5$, referred as *low* and *high*, respectively. Each client holds $M = 2000$ samples. We use the same CNN model as described in [25]. Local epoch and local batch size are set to $E = 5, B = 400$. Learning rate and learning rate decay per local epoch are set to $\gamma = 0.1$, $\gamma_d = 0.9993$. The FL model is trained for five repeat trials, and the medians are recorded. In Algorithm 2, the learning rate of the dueling bandit η is set to be 1.0. The skewness tolerance parameter λ equals 0.4 if not mentioned otherwise.

Baseline and benchmark: The baseline in our experiments is the settings of vanilla FL, where participating clients are randomly selected. In addition, the performance of FL in the IID environment is measured as a reference, which represents a theoretical upper bound of FedACS. One state-of-the-art solution, named CMFL [26], designed for improving FL performance under the non-IID environment is further implemented as the benchmark. CMFL calculates the similarity of clients' gradient and global gradient based on the sign count of all dimensions, and removes "diverging" gradients in the model aggregation to accelerate convergence.

Metrics: We introduce two sets of metrics to evaluate the performance of FedACS. The first set is *terminal accuracy*, which is defined as the average test accuracy in the last 50 rounds, and *relative improvement*, which is defined as the improvement of terminal accuracy, compared to the degrading effect of the non-IID environment. Another set of metrics is *convergence speed*, which is represented by the number of rounds taken for each method to reach the target accuracy 65%; and *speedup* of methods, compared to the baseline.

B. Results and Analysis

Overall performance: To provide an overview of the performance of FedACS and other approaches, we plot their round-accuracy curves in Fig. 3. As can be seen from the figure, first, both non-IID environment settings degrade the performance of FL with slower convergence and lower test accuracy. Second, FedACS greatly reduces the degrading effect in both settings. Last, unfortunately, the performance of CMFL turns out to be even worse than the baseline. The reason is that sign count, a manually selected feature, is not an effective indicator for client skewness in our heterogeneous data environment. It may mistakenly remove some of the uploaded gradients and thus degrades the overall performance.

Specifically, we summarize the performances of FedACS and other methods in Table I and II. FedACS increases the terminal accuracy for $\sim 3.7\%$, compared to the baseline, and reduces the terminal accuracy degrading of non-IID environment for $\sim 65.6\%$. FedACS takes much fewer rounds to reach the target accuracy than the baseline and the benchmark, speeding up for $\sim 2.4\times$.

Parameter sensitivity analysis: Skewness tolerance λ is a critical parameter for the performance of FedACS. To investigate its influence, we test FedACS with different values of λ in the low heterogeneity environment. Experiment results are concluded in Fig. 4. When λ takes its minimum, 0.05, although the harm of skewness is minimized, the performance of the host task turns out to be even lower than the baseline, due to the severe lacking of raw samples. The performance of the host task increases when λ increases from 0.05 to 0.4. Furthermore, when λ is set to 0.4, the host task has the best performance regarding terminal accuracy and convergence speed. The performance degrades when λ becomes 0.6. This is because the overall skewness of participating clients increases. Therefore, a good choice of λ should be neither too high, which connives skewed clients, nor too low, which limits the number of utilized samples.

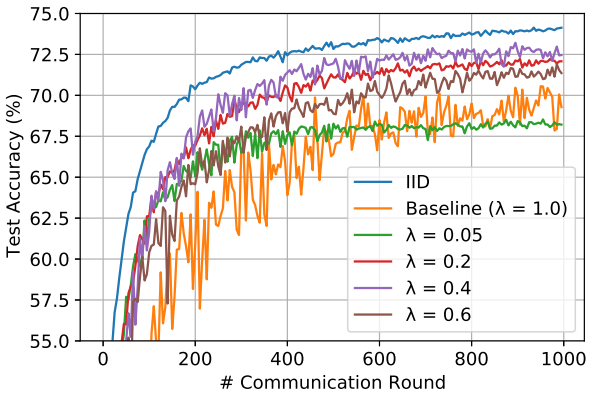


Fig. 4. Performance of FL with different values of λ.

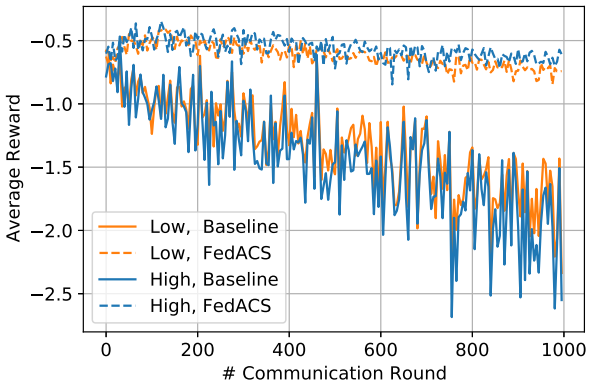


Fig. 5. Average R_i of participating clients in each round.

Performance of the bandit: The most direct objective of the bandit in FedACS is to select the clients with high R_i values (derived in (14)). To test whether the bandit takes effect, we record the average R_i values of the selected participating clients in each round, and compare them with the baseline. Fig. 5 shows the experiment results. Compared to the baseline, the bandit in FedACS fulfills its job of selecting clients with higher potential R_i, which indicates a lower level of client skewness. In addition, both R_i values of the baseline and FedACS are not stationary and decrease over communication rounds. It shows the uncertainty of the derived reward, which reinforces the necessity of our employed multi-dueling bandit formulation.

V. Related Work

A. Federated Analytics

As a newly introduced concept, we acknowledge that there are still a few works in the field of FA. Unlike traditional geo-distributed data analytics [27], [28], FA focuses on the close collaboration of the clients. Currently, FA can be categorized into two types: interactive FA, where the insight derivation procedure requires a global model, and non-interactive FA, where clients do not need any information from the server to perform insight derivation [29]. Interactive FA has been applied in heavy hitter discovery [10], model evaluation [9], and song recognition [9], while the example of non-interactive FA can be found in privacy-preserved data uploading scenario [30]. Our proposed FedACS has the flexibility of being either interactive or non-interactive depending on its relationship with the host task. In this paper, since FedACS reuses the global model of FL to derive insight in (7), it falls into the interactive FA. On the other hand, FedACS also enables users to design other forms of $z_{i,m}^{(k)}$ in (7) that does not rely on the global model, where FedACS will become non-interactive.

B. Application of the Hoeffding's inequality

The Hoeffding's inequality has been widely applied in the field of distributed systems [31]–[33]. In Oort, the Hoeffding's inequality estimates the number of clients required to test the performance of the FL model [31]. In [32], the Hoeffding's inequality for Markov chains is employed for optimizing caching systems in small-cell networks. In [33], the Hoeffding's inequality derives a lower bound of detection rate of the wormhole attack detection algorithm. These applications show the distinctive advantage of the Hoeffding's inequality in providing theoretical bound for various stochastic events.

C. FL in non-IID environment

The non-IID environment is a major challenge for FL, and has attracted worldwide interest from both industry and academia. Various methods have been proposed to reduce the negative effect of the non-IID environment in FL [11], [34]. In [11], the server shares some reserved IID raw data to clients, in order to reduce client skewness. In [34], reinforcement learning helps find clients with higher potential benefit for FL. Personalized Federated Learning, as an emerging variation of traditional FL, breaks the limit that there can be only one global model, and is therefore considered as an effective solution to the non-IID environment [35], [36].

VI. Conclusion

Data heterogeneity is a critical challenge for federated optimization tasks and greatly affects their quality of services. In this paper, we follow the framework of federated analytics to present the first work on federated skewness analytic and client selection, referred to as FedACS. FedACS first uses local-derived insights to infer about clients' data heterogeneity with privacy protected based on the Hoeffding's inequality. After that, it intelligently selects low skewness clients to form an IID environment based on a Thompson sampling based approach. Our proposed FedACS could serve as a standalone federated analytic tool for the distribution characterization purpose or symbiose with other host federated tasks to improve their quality of services. Extensive experiments demonstrate that, when assisting federated learning, FedACS reduces the accuracy degrading by $\sim 65.6\%$, and accelerates the FL' s convergence for $\sim 2.4\times$.

References

[1] Cisco Annual Internet Report (2018–2023). Cisco. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf

[2] D. Carrington, "How many photos will be taken in 2020?" Feb 2020. [Online]. Available: https://focus.mylio.com/tech-today/how-many-photos-will-be-taken-in-2020

[3] 2018 reform of EU data protection rules. European Commission. [Online]. Available: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[5] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," *arXiv preprint arXiv:1812.02903*, 2018.

[6] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[7] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, and Q. Yang, "Fedvision: An online visual object detection platform powered by federated learning," in *Proc. Conf. AAAI Artif. Intell.*, vol. 34, no. 08, New York, NY, USA, Apr. 2020, pp. 13 172–13 179.

[8] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.

[9] Federated analytics: Collaborative data science without data collection. Google AI. [Online]. Available: https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html

[10] W. Zhu, P. Kairouz, B. McMahan, H. Sun, and W. Li, "Federated heavy hitters discovery with differential privacy," in *Proc. Int. Conf. Artif. Intell. Statist.*, Palermo, Sicily, Italy, June 2020, pp. 3837–3847.

[11] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[12] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the Convergence of FedAvg on Non-IID Data," in *Proc. Int. Conf. Learn. Represent.*, Addis Ababa, Ethiopia, Apr. 2020, pp. 1–26.

[13] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[14] L. Li, J. Wang, X. Chen, and C.-Z. Xu, "Multi-layer coordination for high-performance energy-efficient federated learning," in *IEEE/ACM Int. Symp. Qual. Service*, Hangzhou, China, June 2020, pp. 1–10.

[15] I. Diakonikolas, M. Hardt, and L. Schmidt, "Differentially private learning of structured discrete distributions." in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 2566–2574.

[16] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," *arXiv preprint arXiv:2002.04758*, 2020.

[17] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[18] T. Kim, S. Bae, J.-w. Lee, and S. Yun, "Accurate and fast federated learning via combinatorial multi-armed bandits," *arXiv preprint arXiv:2012.03270*, 2020.

[19] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims, "The k-armed dueling bandits problem," *J. Comput. Syst. Sci.*, vol. 78, no. 5, pp. 1538–1556, Sept. 2012.

[20] Y. Sui, V. Zhuang, J. W. Burdick, and Y. Yue, "Multi-dueling bandits with dependent arms," *arXiv preprint arXiv:1705.00253*, 2017.

[21] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.

[22] D. Wang and Z. Han, *Sublinear algorithms for big data applications*. Springer, 2015.

[23] Z. Han, M. Hong, and D. Wang, *Signal processing and networking for big data applications*. Cambridge University Press, 2017.

[24] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Tech. Rep.*, 2009.

[25] "Training a classifier - pytorch tutorials 1.7.1 documentation." [Online]. Available: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial

[26] W. Luping, W. Wei, and L. Bo, "CMFL: Mitigating communication overhead for federated learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, Dallas, TX, USA, July 2019, pp. 954–964.

[27] T. Wang, Z. Qian, L. Jiao, X. Li, and S. Lu, "Geoclone: Online task replication and scheduling for geo-distributed analytics under uncertainties," in *IEEE/ACM Int. Symp. Qual. Service*, Hangzhou, China, June 2020, pp. 1–10.

[28] W. Li, R. Xu, H. Qi, K. Li, and X. Zhou, "Optimizing the cost-performance tradeoff for geo-distributed data analytics with uncertain demand," in *IEEE/ACM Int. Symp. Qual. Service*, Vilanova i la Geltrú, Spain, June 2017, pp. 1–6.

[29] P. Kairouz, B. McMahan, and V. Smith. Federated Learning Tutorial. [Online]. Available: https://sites.google.com/view/fl-tutorial/home

[30] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Scottsdale, AZ, USA, Nov. 2014, pp. 1054–1067.

[31] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Informed participant selection for scalable federated learning," *arXiv preprint arXiv:2010.06081*, 2020.

[32] K. Poularakis and L. Tassiulas, "Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 675–687, June 2016.

[33] S. Ji, T. Chen, and S. Zhong, "Wormhole attack detection algorithms in wireless network coding systems," *IEEE Trans. Mobile Comput.*, vol. 14, no. 3, pp. 660–674, May 2014.

[34] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, July 2020, pp. 1698–1707.

[35] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized federated learning: An attentive collaboration approach," *arXiv preprint arXiv:2007.03797*, 2020.

[36] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mobile Comput.*, Dec. 2020.

APPENDIX A
PROOF OF LEMMA 1

Donate the distribution of data in client i as $\mathcal{D}(d_i)$, and the global distribution in all clients as $\mathcal{D}(d)$. Similarly, we donate the distribution of $z_{i,m}^{(k)}$ in each clients and the global distribution as $\mathcal{D}(z_i^{(k)})$ and $\mathcal{D}(z^{(k)})$, respectively.

Suppose that we have accepted H, then the distribution of $d_{i,m}$ in client i is identical to the distribution of sum-up data in all clients,

$$\mathcal{D}(d_i) = \mathcal{D}(d). \tag{15}$$

As a mapping of $d_{i,m}$, distribution of $z_{i,m}^{(k)}$ is also identical to the overall distribution,

$$\mathcal{D}(z_{i,m}^{(k)}) = \mathcal{D}(z^{(k)}). \tag{16}$$

First, the exception of z_i^k is equal to the exception of $z_{i,m}^k$, since the former is simply arithmetic average of M samples of latter, *i.e.*,

$$\mathbb{E}(\Delta w_i^{(k)}) = \mathbb{E}(z_i^{(k)}) = \mathbb{E}(z_{i,m}^{(k)}). \tag{17}$$

Eq. (16) gives that distribution of $z_{i,m}^k$ is identical to $z^{(k)}$ for all i, m, their exception is also equal:

$$\mathbb{E}(z_{i,m}^{(k)}) = \mathbb{E}(z^{(k)}). \tag{18}$$

Given these insights, we can rewrite (9) as:

$$\begin{aligned} p_i^k &= \mathbb{P}(|\Delta w_i^{(k)} - \mathbb{E}(\Delta w_i^{(k)})| \geq \epsilon) \\ &= \mathbb{P}(|\Delta w_i^{(k)} - \mathbb{E}(z^{(k)})| \geq \epsilon) \\ &\leq 2exp\bigg(- \frac{2\epsilon^2 M}{(b^{(k)} - a^{(k)})^2} \bigg). \end{aligned} \tag{19}$$

p_i^k gives the probabilistic relationship between uploaded weight change and the exception of $z^{(k)}$. When difference ϵ is obtained, we can use p_i^k to define how rare the situation is. In other words, when p_i^k is small, the case here is rare, and we have little confidence to accept H, which is a premise at the beginning.

Supposed the likelihood of accepting client i being IID is low, we can then conclude that the skewness of client i is high.

APPENDIX B
PROOF OF THEOREM 2

We cannot claim that the expectation of z^k is equal to the average of all participating clients, because they are not guaranteed to fully characterize the global data. Instead, we bound the error between $\mathbb{E}(z^{(k)})$ and $\overline{\Delta w}^{(k)}$, showing that $\overline{\Delta w}^{(k)}$ is practically effective as an estimation of $\mathbb{E}(z^{(k)})$.

Donate the clients participated in round t as S_t. Consider (7) and (8), we have:

$$\begin{aligned}\mathbb{E}(z^{(k)}) &= \frac{1}{\sum_{i=1}^{N} M}\sum_{i=1}^{N}\sum_{j=1}^{M} z_{i,m}^{(k)} \\ &\approx \frac{1}{\sum_{i\in S_t} M}\sum_{i\in S_t}\sum_{j=1}^{M} z_{i,m}^{(k)} \\ &= \frac{1}{\sum_{i\in S_t} M}\sum_{i\in S_t} M\Delta w_i^{(k)} \\ &= \overline{\Delta w}^{(k)}. \end{aligned} \tag{20}$$

From (20), we conclude that the estimation of $\mathbb{E}(z^{(k)})$ is given by the weighted average of uploaded weight changes, weighted by their numbers of data.

Credibility of $\overline{\Delta w}^{(k)}$ as an estimation of $\mathbb{E}(z^{(k)})$ can be analyzed via the Hoeffding's inequality. Recall (20), $\overline{\Delta w}^{(k)}$ is the average of $z_{i,m}^{(k)}$ in all clients in S_t. Eq. (5) yields,

$$\begin{aligned}&\mathbb{P}(|\overline{\Delta w}^{(k)} - \mathbb{E}(\overline{\Delta w}^{(k)})| \geq \epsilon) \\ &= \mathbb{P}(|\overline{\Delta w}^{(k)} - \mathbb{E}(z^{(k)})| \geq \epsilon) \\ &\leq 2exp\bigg(-\frac{2\epsilon^2 M}{(b^{(k)}-a^{(k)})^2}\bigg),\end{aligned} \tag{21}$$

It may seem illogical that the estimation of $\mathbb{E}(z^{(k)})$, which will be used to give a probabilistic bound by the Hoeffding's inequality in (19), is also bounded by the Hoeffding's inequality in (21). However, it is numerically reasonable, because the latter bound is much tighter than the former. When (19) and (21) are given the same confidence level, the bound of ϵ in the latter estimation will be κ times tighter than the former, where κ indicates the number of participating clients in each round. Therefore, in (19), the uncertainty given by estimating $\mathbb{E}(z^{(k)})$ is comparatively negligible.

APPENDIX C
DERIVATION OF R_i

The combination of estimation from different dimensions is not mathematically purposeful, but consider that the nature of P_i^k is a probability. Multiply P_i^k among all dimensions seems plausible, as it can be understood as the logical operator "and".

As a result, a skewness estimation of client i based on all dimensions are given by

$$P_i = \prod_{k=1}^{K} 2exp\bigg(-\frac{2(\epsilon^{(k)})^2 M}{(b^{(k)}-a^{(k)})^2}\bigg) \tag{22}$$

Recall that higher P_i^k indicates lower skewness, and the range of P_i^k is $[0,1]$. Therefore, a higher P_i also indicates a lower skewness.

$b^{(k)}$ and $a^{(k)}$ are the upper and lower bounds of $z_{i,m}^k$. A normal method is to request the minimum and maximum from all participating clients and to derive the tightest bound. However, it increases the communication overhead by $2\times$, and breaks the strict privacy restriction of FL. Therefore, we used a looser bound, which is equal for all dimensions. Donate them as b_{max} and a_{min}, *i.e.*,

$$b_{max} = \max_{\forall i,m,k}\big(z_{i,m}^{(k)}\big),\ a_{min} = \min_{\forall i,m,k}\big(z_{i,m}^{(k)}\big). \tag{23}$$

Since (5) only requires a and b as bounds, without requirement of tightness. We are able to use b_{max} and a_{min} to take place of $b^{(k)}$ and $a^{(k)}$ in all dimensions, without loss of mathematical correctness. Rewrite (22),

$$P_i = \prod_{k=1}^{K} 2exp\bigg(-\frac{2(\epsilon^{(k)})^2 M}{(b_{max}-a_{min})^2}\bigg). \tag{24}$$

Take the logarithm on both sides, and simplify the form,

$$\frac{(K\ln 2 - P_i)(b_{max}-a_{min})^2}{2M} = \sum_{k=1}^{K}\big((\epsilon^{(k)})^2\big) \tag{25}$$

Recall (13), we can find that the sum of $(\epsilon^{(k)})^2$ among all dimensions is the sequre of L^2 norm between Δw_i and $\overline{\Delta w}$, and derive the resulted form as Q_i:

$$\begin{aligned}Q_i &= \sqrt{\frac{(K\ln 2 - P_i)(b_{max}-a_{min})^2}{2M}} \\ &= ||\Delta w_i - \overline{\Delta w}||_2\end{aligned} \tag{26}$$

where,

$$\overline{\Delta w} = \frac{1}{N}\sum_{i\in S_t}\Delta w_i \tag{27}$$

Eq. (26) shows that Q_i is an inverse transformation of P_i, where lower Q_i indicates lower skewness. However, a MAB is pursuing arms with higher rewards, so we should assign a higher reward to the clients with lower skewness. As a result, we used the inverse of Q_i as the reward R_i, *i.e.*,

$$R_i = -Q_i = -||\Delta w_i - \overline{\Delta w}||_2 \tag{28}$$