Crowdsourcing-based Model Testing in Federated Learning

Yunpeng Yi[1], Hongtao Lv[1], Tie Luo[2], Junfeng Yang[3], Lei Liu[1,4✉] and Lizhen Cui[1]
[1]Shandong Unversity, Jinan, China
[2]Missouri University of Science and Technology, Rolla, USA
[3]Beijing aerospace automatic control institute, Beijing, China
[4]Shandong Research Institute of Industrial Technology, Jinan, China
Email: l.liu@sdu.edu.cn

***Abstract*—Federated Learning (FL) is a distributed machine learning technique that trains models on local devices to preserve data privacy. In FL, evaluating model quality is crucial for detecting malicious clients and improving model accuracy. However, existing methods typically require a representative public testing dataset on the server, which is often unavailable in practical federated learning scenarios. To address this problem, we propose a novel four-step framework, taking a crowdsourcing approach. The basic idea is to distribute the model to be evaluated as a task to a set of testing clients selected from the original clients pool, who evaluate the model quality using their local datasets. By consolidating these individual evaluations, we obtain the overall model quality. To select a suitable number of testing clients, we propose an exploration-exploitation-based framework. Furthermore, to safeguard against attacks from potential malicious testing clients, we introduce a Correlated Agreement (CA) mechanism. This is achieved by comparing correlations of accuracy among the same set of testing clients (who were selected for the aforementioned evaluation task). Extensive experiments demonstrate the effectiveness of our approach, which yields accuracy comparable to methods that rely on a public testing dataset on the server. Moreover, our approach can identify and filter out dishonest testing clients and thereby ensure model quality even in adversarial settings.**

***Index Terms*—Federated learning, crowdsourcing, model testing, correlated agreement.**

I. Introduction

Federated Learning (FL) is a distributed machine learning approach that aims to tackle data privacy and security concerns using a decentralized framework [1–3]. Unlike conventional machine learning techniques, which use centralized datasets for training and pose risks associated with data transfer and centralized storage, FL empowers edge devices to train models using their respective local datasets. The training process occurs locally on these devices (clients), eliminating the need to transmit raw data to a central server. Each device updates the model locally and shares the parameters with a central server, aggregating them to generate a global model. Subsequently, the global model is distributed back to the local devices, enabling the repetition of the training process. By iteratively performing these steps, FL achieves the aggregation of intelligence from distributed devices while preserving data privacy, making it attractive for various applications [4–6].

In FL, the evaluation of model quality and accordingly, the evaluation of client contribution, hold significant importance in several aspects. First, they allow for the detection of malicious clients which can intentionally provide incorrect updates or tamper with the model. Identifying these malicious clients by their model quality helps ensure the trustworthiness of the model's training process and results. Second, they are essential for making informed decisions about its deployment and potential improvements, such as incentive mechanism design and determining when to terminate the training process, preventing overfitting or unnecessary iterations. Thirdly, by computing metrics such as accuracy, precision, and recall values on the validation set, one can evaluate the quality of the uploaded models of clients or the aggregated model in FL. On top of these metrics, the contribution of each client in FL can be accessed by Shapley value [7, 8]. This method involves considering all possible combinations of participants and evaluating their contributions by measuring the expected marginal gain when they join the federation. Leave-One-Out (LOO) [7] is a simpler approximate method to reduce the complexity of Shapley value, which estimates the contribution of each individual to the cooperative benefit by excluding it.

However, implementing the model evaluation in FL is challenging in practice, due to the unavailability of a public testing dataset, which is difficult to acquire due to the privacy and security concerns in FL. Constructing a representative, diverse, and accessible public testing dataset may harm clients' privacy and result in sensitive information leakage, contrary to the original intention of FL. Furthermore, some existing approaches suggest the utilization of a small, rather than representative, public testing dataset. However, when the dataset size is limited, it may exhibit variations in data distribution and features due to the heterogeneous nature of data distribution in FL, which may significantly harm the performance of model quality evaluation.

In this paper, we present a crowdsourcing-based framework to tackle the issue of testing dataset construction by leveraging the local datasets of individual users for distributing the model and evaluating its accuracy. Furthermore, we consolidate these individual accuracy evaluations to derive an overall model

This work was partially supported by National Natural Science Foundation of China (NSFC No. 62220106004, 62302267), the National Science Foundation (NSF) under Grant No. 2008878, the Natural Science Foundation of Shandong (Shandong NSF No. ZR2021LZH006), Taishan Scholars Program.

accuracy. We also address the challenge of ensuring the truthfulness of participating clients to provide genuine and high-quality evaluations by introducing a reputation mechanism based on the Correlated Agreement (CA) mechanism in the evaluation process [9].

This paper makes the following contributions.

1. We propose a crowdsourcing framework for FL to perform model testing without a public test dataset by leveraging the local datasets owned by individual FL clients. This is the first work that introduces the use of crowdsourcing for FL model testing. Not only does it overcome the lack of testing sets in real-world scenarios of FL, but it also eliminates the need for additional data transmission (in assembling a testing set), thereby mitigating data privacy concerns and improving communication efficiency.

2. We propose an exploration-exploitation-based scheme under a stochastic optimization framework to address the problem of selecting a suitable number of testing clients in a stochastic optimization framework. By iteratively evaluating the performance of different subsets of testing clients and refining the selection, our scheme achieves an optimal trade-off between exploring new clients and exploiting the information gained from existing ones. This approach enables us to efficiently utilize resources while maximizing the accuracy and reliability of the evaluation process.

3. After selecting the appropriate number of clients, we propose to utilize the CA mechanism to incentivize clients to perform the crowdsourcing tasks honestly and with high quality. This mechanism enables the detection and removal of malicious clients from the evaluation process.

II. Related Work

Our work is closely related to the research from two branches: contribution evaluation in FL and FL with crowdsourcing.

A. Contribution Evaluation in Federated Learning

The effective evaluation of participant contributions in federated cooperation is a crucial issue for the practical application and long-term advancement of FL. Prior research on contribution evaluation in FL has explored various approaches, including the Shapley value-based approach and the LOO method [10]. In 1953, the Shapley value was proposed as a solution to cooperative game problems where the Shapley value of each participant is defined as its average marginal contribution [11]. Recently, Shapley value has gained widespread usage in evaluating the contributions of participants in FL. Liu et al. [12] highlights that existing methods based on Shapley value involve substantial computational costs, which limits their practical applicability. To tackle this challenge, they propose the GTG-Shapley approach which reconstructs FL models using gradient updates instead of repetitive training with various participant combinations. Another frequently utilized method to evaluate participant contributions is the LOO method, commonly used in statistical analysis and machine learning, which has been adapted to assess individual contributions in FL. For example, Wang et al. [7] proposed to use LOO in FL by removing a participant from the entire federated group and measuring the resulting loss in data value as that participant's contribution. These approaches rely on the availability of a representative benchmark dataset within the FL setting. Our work addresses this challenge by leveraging the local datasets on each client and employing crowdsourcing techniques. This approach enables the evaluation of client contributions even when a benchmark dataset is not present.

B. Federated Learning with Crowdsourcing

FL itself can be viewed as a crowdsourcing process [13, 14], as edge devices in FL have autonomy and contribute their own data to the training process, resembling a form of crowdsourcing. Previous research has focused on incentive mechanisms, participant selection, and handling dishonest participants during the model training process, treating it as a crowdsourcing task. The classical FL framework in [1] is to involve a large number of participants in the training process by aggregating their local models or gradients to create a global model. These collective efforts contribute to enhancing the overall performance and generalization capabilities of the model. Additionally, Pandey et al. [15] proposes a crowdsourcing framework considering communication efficiency to address the challenge of user participation in building a high-quality global model in FL. Different from the above studies, we are the first to consider the model testing phase as a crowdsourcing task and investigate participant selection and detection of dishonest participants in this context.

III. System Overview

In this section, we present a framework consisting of a four-step process for performing model testing. Figure 1 shows the process of FL with crowdsourcing-based model testing.

A. Task Generation

The Leave-One-Out (LOO) method is utilized for generating the model evaluation tasks, wherein the model is trained using N clients, and upon completing the training process, we iterate through each client's model parameters and exclude them individually. Subsequently, we aggregate the model parameters contributed by the remaining clients. This iterative process yields $N+1$ distinct models for comprehensive evaluation and analysis, which includes a global aggregated model without excluding any client. We note that our approach is a technique for general distributed model testing scenarios, applicable to any contribution evaluation method that requires model testing, such as Shapley value-based approaches. We choose to adopt the LOO method only for the sake of clarity in this paper.

B. Recruitment of Test Participants

An important challenge in this recuited process is the uncertainty of the data quantity of local datasets available to each test participant. The aim is to identify a minimized number of participants to ensure that the combined local datasets sufficiently meet the criteria for assessing the model's accuracy.

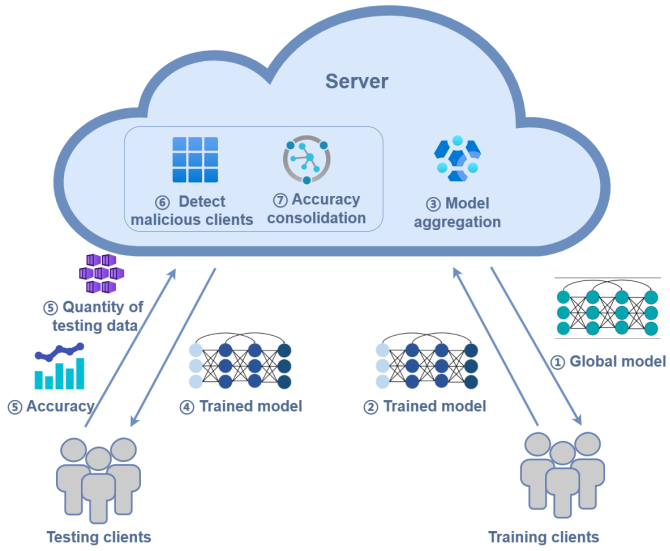


Fig. 1: The process of federated learning with crowdsourcing-based model testing.

To tackle the above challenge, we employ an exploration-exploitation approach: During the exploration phase, we initially recruit a random number of test participants. This allows us to explore the range of dataset sizes and distributions among the potential participants. Based on the insights gained from the information about their dataset sizes, we transition from the exploration phase to the exploitation phase. During this phase, we strategically recruit additional test participants which ensures that the recruited test participants collectively provide a sufficient amount of data to accurately evaluate the model.

C. Model Testing and Aggregation

After recruiting the testing population, the subsequent step involves conducting model testing and aggregation. Each testing participant is assigned a specific set of model tasks. Utilizing their local datasets, they evaluate the assigned model and provide individual assessments of its accuracy. To ensure a fair and reliable aggregation of evaluations, a weighting scheme is employed by taking into account the size of each participant's dataset to determine the weight assigned to their evaluation.

D. Testing Quality Evaluation

The objective of this step is to ensure testing clients invest genuine effort in accurately testing the model's quality and to detect clients who falsely report or conduct testing solely on a subset of the dataset. To achieve this, we employ a Correlated Agreement (CA) mechanism for evaluating and motivating the clients. The CA mechanism calculates the contribution of each client by measuring the correlation of their test results on the same task. Based on the correlation measure, the contribution of each client can be determined for model aggregation or resource allocation purposes. This method can detect the quality of model test results and is used to exclude malicious testing clients and other untrustworthy behaviors. We utilize this method to accomplish two tasks: determining the probability of being selected for future tasks and rewarding their contributions. By employing a correlation-based mechanism, we can assess the reliability of testing results of clients.

IV. Methodology

In this section, we present a detailed description of each step in the framework and provide specific methods that meet the requirements of the framework.

A. Task Generation

As mentioned above, to evaluate model quality, a LOO approach is used to generate tasks. After the model training in each round of FL, one client's model is removed at a time, and the remaining $N-1$ clients' models are aggregated with weights based on their data size, resulting in N models. Additionally, there is a globally aggregated model without removing any client, so there are $N+1$ models totally for testing. This methodology enables the generation of multiple tasks and the assessment of each client's trained model quality.

B. Recruitment of Test Participants

How to determine the number of testing clients to recruit for assessments of model quality is a crucial issue. We now define the recruiting problem mathematically.

Definition 1 (Problem Statement): The objective is to determine the minimum number of testing clients n to recruit, such that the probability that the total amount of data leveraged by all testing clients is less than m is smaller than a given threshold θ. Formally,

$$\begin{aligned} & min \quad n \\ s.t. \quad & P\left(\sum_{i\in\mathcal{K}} \tau_i < m\right) \le \theta \\ & |\mathcal{K}| = n, \end{aligned} \tag{1}$$

where $\mathcal{K}$ is the set of selected testing clients, τ_i represents the amount of data used by testing client i to perform the model testing task, and $\sum_{i\in\mathcal{K}} \tau_i$ denotes the total amount of data leveraged by all testing clients.

We assume that the random variable τ_i follows an identical Gaussian distribution, i.e., $\tau_i \sim N(\mu, \sigma^2)$, with the probability density function given by $f(\tau) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\tau-\mu}{\sigma}\right)^2}$, where σ is the standard deviation and μ is the mean. Let $T_n = \sum_{i\in\mathcal{K}} \tau_i$ denote the sum of n random variables τ_i. One can establish that the sum of random variables induced by a Gaussian distribution also follows a Gaussian distribution, that is, T_n is characterized by a Gaussian distribution, specifically $T_n \sim N(n\mu, n\sigma^2)$, where n is the number of selected testing clients, μ and σ are the mean and the standard deviation as given above, respectively.

To obtain an analytical solution to the above problem, we can rephrase the constraint as follows:

$$F_{n\mu, n\sigma^2}(m) < \theta, \tag{2}$$

where $F_{n\mu,n\sigma^2}(\cdot)$ represents the cumulative distribution function of the random variable T_n. The subscript $n\mu$ and $n\sigma^2$ denote the mean and variance of the random variable T_n, respectively. Since the mean μ and standard deviation σ remain unchanged in the subsequent derivations, we omit them and denote $F_{n\mu,n\sigma^2}(\cdot)$ as $F_n(\cdot)$ for simplicity. As n increases, it can be easily proven that $F_n(\cdot)$ decreases. Therefore, we can formulate the constraint (1) as follows:

$$n^* = \arg\min_n \quad F_n(m) \quad s.t.\ F_n(m) < \theta. \tag{3}$$

Next, we employ an exploration-exploitation approach to obtain the expected value μ and variance σ of this distribution.

- Exploration: During the exploration phase, we randomly select l testing clients and obtain information about the amount of data they have collected, denoted as $W = \{w_i | i = 1, 2, ..., l\}$. Here, w_i represents the number of samples in the local dataset used by testing clients i, which is reported by the testing clients themselves. We assume that testing clients honestly report their data amounts, which is a mild assumption because there is no benefit to overstate the amount of data. Hence, the expected value of the reported data from the l testing clients is $\mu = \frac{\sum_{i=1}^{l} w_i}{l}$, and the variance is $\sigma^2 = \frac{\sum_{i=1}^{l}(w_i-\mu)^2}{l}$.
- Exploitation: After obtaining μ and σ in the exploration phase, we can fit the cumulative distribution function $F_{\mu,\sigma}(\cdot)$. Then, we can use binary search to find the value of n. As n increases, $n\mu$ and $n\sigma^2$ also increase. By substituting these values into the constraint condition, we can identify a threshold that satisfies the constraint, allowing us to determine the required number of participants.

The exploration-exploitation phase occurs throughout the entire FL process. For example, in an FL process consisting of a total of R rounds, the first u rounds are exploration phases, while the remaining $R - u$ rounds are exploitation phases. In the exploration phase, the data distribution of the testing clients is learned, and in the subsequent exploitation phase, the corresponding number of participants is selected based on this distribution for testing purposes.

C. Model Testing and Aggregation

Each of the recruited testing clients is assigned a set of model tasks generated in the first step and uploads the corresponding model accuracies. We consolidate the accuracy provided by the testing clients using a simple weighting scheme. The weight assigned to each testing client is determined based on the data size of the datasets they contribute. Let's denote the set of weights for the N testing clients as $W = \{w_i \mid i = 1, 2, ..., N\}$. The weight assigned to the i-th testing clients is denoted as λ_i, calculated as $\lambda_i = \frac{w_i}{\sum_{j=1}^{N} w_j}$. Finally, we compute the overall accuracy, denoted as η, by summing up the individual accuracies weighted by their λ values: $\eta = \sum_{i=1}^{N} \lambda_i \eta_i$.

D. Testing Quality Evaluation

We now introduce how to assess the quality of testing clients' execution of model testing tasks, to detect malicious or low-quality testing clients. To address this issue, we employ a scoring mechanism known as the CA mechanism. It involves the following steps:

- *Step D-1*: We distribute a set of tasks $T = \{t_i | i = 1, 2, ..., x\}$, where each task represents the parameters of a model to be evaluated, to a group of testing clients $A = \{a_i | i = 1, 2, ..., k\}$. Each testing client a_i receives the complete group of tasks, and each task is completed by different testing clients. Each testing client a_i tests the corresponding model parameters using their local data and reports the model accuracy η_i.
- *Step D-2*: We define the correlation matrix Δ as in [16] and we utilize the received accuracy to calculate it. We define η_p^i as the test result of testing clients a_i on task t_p. The CA mechanism is built upon the Δ matrix, which describes the correlation between η_p^i and η_p^j:

  $$\Delta(\eta_p^i, \eta_p^j) = P(\eta_p^i, \eta_p^j) - P(\eta_p^i)P(\eta_p^j). \tag{4}$$

  In Equation (4), $P(\eta_p^i, \eta_p^j)$ represents the joint probability of testing clients a_i's result η_p^i and another testing clients a_j's result η_p^j on the same task t_p. $P(\eta_p^i)$ and $P(\eta_p^j)$ denote the marginal probabilities. When $\Delta(a, b) > 0$, it indicates a positive correlation between the two results; if $\Delta(a, b) = 0$, we have that a and b are independent; otherwise, they are negatively correlated. Without loss of generality, we assume that there is at least one element in the matrix that is non-zero. The scoring function $S(\cdot)$ is defined as follows:

  $$S(a, b) = \begin{cases} 1, \Delta(a, b) > 0 \\ 0, \Delta(a, b) \leq 0. \end{cases} \tag{5}$$

  To illustrate the idea of delta matrix Δ more clearly, one can consider a 100×100 matrix representing accuracy levels from 1% to 100%. Then we can obtain a delta matrix Δ with the size of 100×100. For example, a common $S(\cdot)$ in practice could be

  $$S(\cdot) = \begin{bmatrix} 1 & 1 & 1 & 0 \cdots & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \cdots & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \cdots & 1 & 1 & 1 \end{bmatrix},$$

  which means that two clients are more likely to report close accuracy for the same task.
- *Step D-3*: Next, with a given delta matrix Δ, we describe the classical CA method introduced by [17] for our setting. We put the pseudocode to calculate the contribution of each client in Algorithm 1. We use Q_p^i to represent the contribution of testing clients a_i to task t_p:

  $$C_p^{ij} = S(\eta_p^i, \eta_p^j) - S(\eta_q^i, \eta_z^j), \tag{6}$$

  where η_p^i denotes the accuracy of the client i for task p.
- *Step D-4*: We sum up the total contribution value of each client to the whole task set as her contribution, then we can establish a threshold value, denoted as T. If the

Algorithm 1 Correlated Agreement (CA) Mechanism

Input: The set of selected users $\mathcal{K}$, a set of task Z, the number of peers m and the delta matrix Δ.

Output: The contribution Q^i of each user $a_i \in \mathcal{K}_t$.

1: **for** each user $i \in \mathcal{K}$ **do**
2: $D^j \leftarrow$ a set of randomly selected m testing clients as peers;
3: **for** each task t_p in Z **do**
4: **for** each peer client $a_j \in D^i$ **do**
5: Calculate η_p^i ,η_p^j for a_i and a_j on task t_p;
6: Calculate η_q^i for a_i on task t_q and η_z^j for a_j on task t_z;
7: $C_p^{ij} \leftarrow S\left(\eta_p^i, \eta_p^j\right) - S\left(\eta_q^i, \eta_z^j\right)$;
8: **end for**
Repeat Lines 3 - 8 for each task;
$Q_p^i \leftarrow \frac{1}{|D^i||P|} \sum_{p \in P} \sum_{j \in D^i} C_p^{ij}$.
9: **end for**
10: **end for**

contribution value of a client falls below this threshold, we classify the client as a "bad client" and exclude the accuracy calculated by this client from the accuracy aggregation process.

V. Performance Evaluation

A. Experimental Setting

In our study, we evaluate the performance of our method using CIFAR-100 dataset [18]. CIFAR-100 is a classic computer vision dataset used for image classification tasks. It consists of 100 categories of colour images, with 600 images per category. These categories are divided into 20 superclasses, each containing 5 subclasses. The CIFAR-100 dataset was created to evaluate the performance and robustness of image classification algorithms, and it has been widely applied in machine learning research.

For the training model, we follow a similar approach as described in [19] by employing a simple Convolutional Neural Network (CNN) with two convolutional layers, one pooling layer, and three fully connected layers. Among them, the learning rate is 0.01, the local epoch is 50, and the local batch size is 5.

We establish a pool of 370 clients and allocate 60,000 data points from CIFAR-100 among these clients. Each client's data is further divided into a training set and a testing set while guaranteeing that there is no overlap between the two sets. The data distribution adheres to a Dirichlet distribution [20] with parameters $\alpha = 0.1$ and $\beta = 0.7$, where α controls the amount of data allocated to each client, and a larger α value makes the distribution more concentrated and peaked. β affects the number of sample categories assigned to each client, and a larger β value indicates a smaller difference in the number of sample categories assigned to each client.

Furthermore, we designate a global training dataset of 50,000 images and a testing dataset containing 10,000 images. From the pool of clients, we select 18 of them as training clients. Employing the leave-one-out method, we iterate through each client's trained model parameters, excluding one client's model from the aggregation at a time. This iterative process generates a total of 19 tasks (including the globally aggregated model without excluding any client).

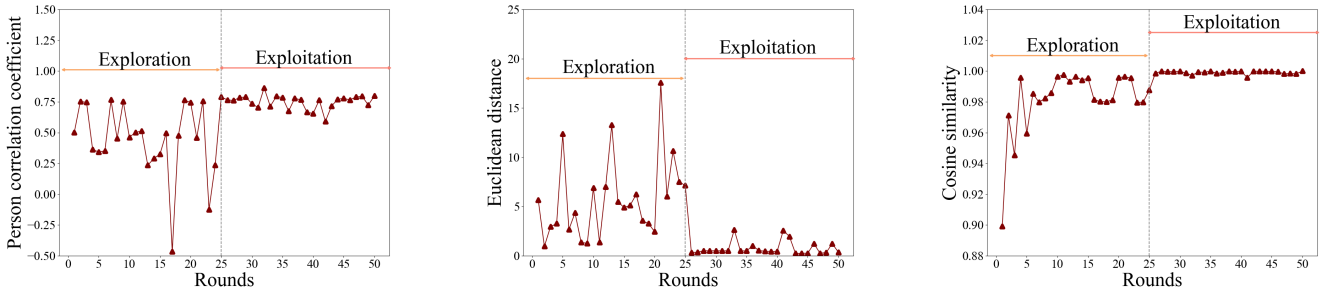
B. Experimental Results

We conduct the following two experiments to validate the effectiveness of our approach:

Firstly, we conduct FL for a total of 50 rounds, divided into an initial exploration phase of 25 rounds followed by an exploitation phase of 25 rounds. During the exploration phase, we randomly select an integer between 1 and 40 to determine the number of testing clients included in each round. The mean value μ of the clients' data amount is calculated to be 27.212, with a variance value σ of 8.522. These values are utilized to derive the truncated normal distribution function, equation (3), which guides the selection of testing clients based on their data characteristics. By utilizing the binary search method with m = 5000 and θ = 0.1, we obtain that the minimum value of n is 171. Therefore, during the remaining 25 rounds, we select 171 testing clients for evaluation purposes. In each round, the training clients utilize the global training dataset for training, while the testing clients use their local testing set for evaluation. The model accuracies are then consolidated, taking into account the weights assigned based on the respective client's data size. This process generates a list of model accuracies. To compare this list of model accuracies with the ground truth, *i.e.*, the list of accuracies generated from a public testing dataset, we employ metrics including the Pearson correlation coefficient [21], Euclidean distance [22], and cosine similarity [23]. These metrics provide insights into the similarity or dissimilarity between the two sets of accuracies.

Figure 2 depicts the trends of the Pearson correlation coefficient, Euclidean distance, and cosine similarity between the two accuracy lists generated by our approach and the one with a public testing dataset during the exploration and exploitation phases. Figure 2(a) represents the Pearson correlation coefficient, which ranges from -1 to 1. A value closer to 1 indicates a stronger positive correlation between the two lists. It can be observed that during the exploration phase, the Pearson correlation coefficient fluctuates significantly due to the random selection of testing clients. However, during the exploitation phase, the coefficient stabilizes around 0.9. Figure 2(b) illustrates the Euclidean distance, which takes non-negative values. A smaller value indicates a closer distance between the points. Figure 2(c) represents the cosine similarity, ranging from -1 to 1. A cosine similarity close to 1 indicates a small angle between the two vectors, implying a high degree of similarity in direction. These trends indicate the feasibility of using individual testing datasets instead of a public testing dataset.

Secondly, assuming the presence of malicious clients during the exploitation phase, we randomly select 5 testing clients



(a) Pearson correlation coefficient (b) Euclidean distance (c) Cosine similarity

Fig. 2: Correlation between model accuracies with and without a common testing set during the exploration-exploitation process.

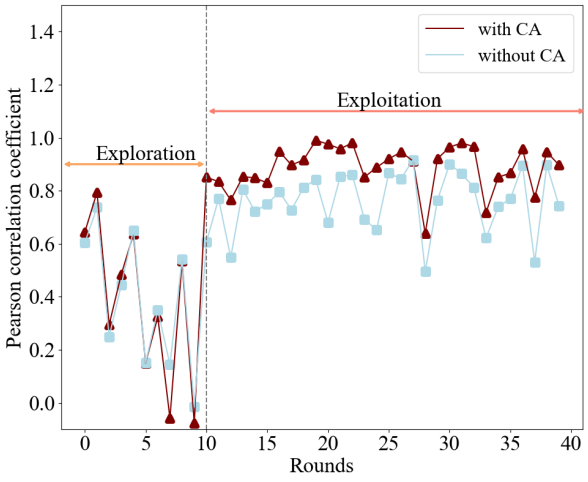


Fig. 3: Pearson correlation coefficients with and without CA method in the presence of malicious testing clients.

as malicious clients. These clients conduct dishonest behavior by adding Gaussian noise to their reported accuracies. The added noise follows a Gaussian distribution with a mean of 0 and a standard deviation of 0.2. We establish a 100×100 pentadiagonal matrix as the Δ matrix in CA. Each testing client randomly selects 5 other clients as its peer clients. By applying the lookup table, we compute the contribution values for each testing client according to Equation (6). We then set a threshold of 2.5, considering any client with a contribution value below this threshold as a malicious client. During the accuracy aggregation process, these identified malicious clients are excluded. Subsequently, we compare the Pearson correlation coefficient between the accuracy lists obtained by using the CA approach (with the malicious clients removed) and the accuracy lists obtained without removing malicious clients. This comparison allows us to assess the similarity between the two sets of accuracies and evaluate the effectiveness of using the CA approach in mitigating the impact of malicious clients.

Figure 3 illustrates the variation range of Pearson correlation coefficients whether the bad testing clients are excluded by this method. The first 10 rounds are designated as the exploration phase, while the remaining 30 rounds represent the exploitation phase. During the exploitation phase, the Pearson correlation coefficients obtained with CA exhibit clearly higher values compared to those without CA, where the testing clients with added noise are retained. This indicates the effectiveness of detecting malicious clients with the CA method.

VI. Conclusion

In this paper, we propose a novel approach for model testing in FL without the need for a public testing dataset. It leverages the power of crowdsourcing by taking advantage of the local datasets held by the FL clients themselves, while taking a cross-validation approach. Furthermore, our method incorporates a CA mechanism to detect malicious clients and thereby enhances robustness. Through extensive experiments, we demonstrate the effectiveness of our proposed method which yields accuracy comparable to methods that rely on a public testing dataset on the server.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[2] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019.

[3] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *ACM Computing Surveys*, 53(2):1–33, 2020.

[4] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale:

System design. *Proceedings of Machine Learning and Systems*, 1:374–388, 2019.

[5] Mohamed Elmahallawy and Tie Luo. One-shot federated learning for LEO constellations that reduces convergence time from days to 90 minutes. In *24th IEEE International Conference on Mobile Data Management*, pages 45–54, 2023.

[6] Mohamed Elmahallawy, Tie Luo, and Mohamed I Ibrahem. Secure and efficient federated learning in leo constellations using decentralized key generation and on-orbit model aggregation. *arXiv preprint arXiv:2309.01828*, 2023.

[7] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. Measure contribution of participants in federated learning. In *2019 IEEE International Conference on Big Data*, pages 2597–2604, 2019.

[8] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A principled approach to data valuation for federated learning. *Federated Learning: Privacy and Incentive*, pages 153–167, 2020.

[9] Dan Wang, Ju Ren, Zhibo Wang, Yichuan Wang, and Yaoxue Zhang. Privaim: A dual-privacy preserving and quality-aware incentive mechanism for federated learning. *IEEE Transactions on Computers*, 2022.

[10] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6):10700–10714, 2019.

[11] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games (AM-28), Volume II*, pages 307–318, 1953.

[12] Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. GTG-Shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Transactions on Intelligent Systems and Technology*, 13 (4):1–21, 2022.

[13] Ece Kamar and Eric Horvitz. Incentives for truthful reporting in crowdsourcing. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems*, volume 12, pages 1329–1330, 2012.

[14] Xiao Xue, Guanding Li, Deyu Zhou, Yepeng Zhang, Lu Zhang, Yang Zhao, Zhiyong Feng, Lizhen Cui, Zhangbing Zhou, Xiao Sun, et al. Research roadmap of service ecosystems: A crowd intelligence perspective. *International Journal of Crowd Science*, 6(4):195–222, 2022.

[15] Shashi Raj Pandey, Nguyen H Tran, Mehdi Bennis, Yan Kyaw Tun, Aunas Manzoor, and Choong Seon Hong. A crowdsourcing framework for on-device federated learning. *IEEE Transactions on Wireless Communications*, 19(5):3241–3256, 2020.

[16] Luc Devroye and Gábor Lugosi. *Combinatorial Methods in Density Estimation*. 2001.

[17] Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C Parkes. Informed truthfulness in multi-task peer prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 179–196, 2016.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] Saeed Vahidian, Mahdi Morafah, Weijia Wang, Vyacheslav Kungurtsev, Chen Chen, Mubarak Shah, and Bill Lin. Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces. *ArXiv Preprint ArXiv:2209.10526*, 2022.

[20] Kai Wang Ng, Guo-Liang Tian, and Man-Lai Tang. Dirichlet and related distributions: Theory, methods and applications. 2011.

[21] Karl Pearson. X. Contributions to the mathematical theory of evolution.—ii. skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London*, (186):343–414, 1895.

[22] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory—ICDT 2001: 8th International Conference*, pages 420–434, 2001.

[23] Christopher D Manning. *An introduction to information retrieval*. 2009.