MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM) | 979-8-3503-2181-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/MILCOM58377.2023.10356215

MINDFL: Mitigating the Impact of Imbalanced and Noisy-labeled Data in Federated Learning with Quality and Fairness-Aware Client Selection

Chaoyu Zhang*, Ning Wang*, Shanghao Shi*, Changlai Du*, Wenjing Lou*, and Y. Thomas Hou*
* Virginia Tech, VA, USA

***Abstract*—Federated Learning (FL) has been acclaimed for enhancing machine learning privacy, but it also faces criticism for its high communication overhead due to the need for a large number of interactions between participants and the server. The conventional approach of randomly selecting clients for FL participation can reduce communication costs, but it often slows down the convergence of the global model and lowering its overall quality. Although various advanced methods have been proposed for client selection, focusing on the selection of the most informative local models, these solutions frequently assume clean training data. They often overlook the impact of noisy-labeled and imbalanced local data among clients, which can significantly hinder training efficiency.**

To address these challenges, we present MINDFL, a client selection mechanism that prioritizes quality and fairness considerations. MINDFL tackles the problem of utility divergence among local models caused by noisy-labeled and imbalanced data. It also ensures fairness in the selection process. In particular, we introduce a novel metric called Quality-of-Model (QoM), which assesses the contribution of local models to the aggregated global model. MINDFL selects clients with high QoM, representing the most informative model updates, in each FL round to maximize learning efficiency. Further, to enhance participant diversity while maintaining clients' model quality, we utilize a sortition-inspired selection method to choose clients with high QoM randomly. Our comprehensive experimental evaluations demonstrate the effectiveness of MINDFL in improving learning speed and reducing communication overhead.

***Index Terms*—Federated learning, Quality-of-Model, Client Selection, Imbalanced and Noisy data**

I. Introduction

Federated learning is an emerging ML framework that coordinates distributed intelligent clients, such as mobile and IoT devices, to collaboratively train an ML model without collecting their local data. Such privacy preservation property renders FL increasingly popular for a broad spectrum of ML-based applications, such as next-word prediction [1], human activity monitoring [2], and credit risk controlled by WeBank [3]. The FL framework consists of a central parameter server (PS) and a set of FL clients. Within each FL training iteration, PS first broadcasts a global model to FL clients. Clients will then train the global model with their local data for several iterations before providing updates. Then the PS aggregates FL clients' feedback with an aggregation algorithm, e.g., FedAvg [4], and updates the global model. This training process, performed jointly by the PS and FL clients, is repeated for multiple rounds until the global model converges.

However, the repeated communication processes between the PS and clients introduce a significant overhead for the FL system, especially when the client number is large, e.g., in an IoT scenario with hundreds/thousands of clients. Considering the large client number and model dimensionality, communication overhead becomes a bottleneck in FL. Recent FL systems proposed two different directions to deal with the limitation: 1) model compression, including parameter quantization [5], knowledge distillation [6], and model pruning [7]. 2) per-iteration participation reduction that selects a small fraction of clients to participate in each training iteration. This paper focuses on the second direction–client selection–as it is less researched. Note that the two types of methodology are orthogonal and can be applied simultaneously to boost communication efficiency.

Client selection mechanisms, while effectively reducing communication overhead, can negatively impact learning efficiency due to reduced client participation. The most widely-used client selection is random client selection which has achieved great success on high-quality and independently and identically distributed (IID) datasets [4]. However, in scenarios with imperfect data, characterized by noise and non-IID patterns, there is still room for improvement in client selection to enhance learning efficiency. Zhang et al. [8] utilized weight divergence to recognize the non-IID degrees of clients and select the clients with a lower degree of non-IID data more frequently. Wang et al. [9] designed a reinforcement-learning-based client selection scheme to counter the bias in data. Several works [8], [10], [11] proposed various metrics to estimate data quality and select models with higher data quality. However, these approaches commonly assume that clients' local data are clean.

The devices utilized in modern IoT networks, such as signal sensors, are deployed in diverse environments with varying capabilities for data collection [12], [13]. Consequently, the local data available on different devices exhibit imbalances in terms of data distribution and quantity [14], [15]. Additionally, the

data collected by these local devices are susceptible to noise stemming from factors like software bugs, hardware faults, and environmental influences [16], [17], as well as potential errors introduced during manual data labeling. To provide clarity, it is important to define the terms used: **imbalanced data** refers to the disparity in the quantity of training data among clients, while **noisy-labeled data** describes a scenario where the assigned labels contain errors, deviating from the true classes.

The primary objective of this research paper is to enhance the efficiency of federated learning when faced with communication constraints in the presence of noisy-labeled and imbalanced data. To address the aforementioned data challenges, we propose a novel metric called Quality-of-Model (QoM) that quantifies the contribution of local models to the aggregated global model. By maintaining a low client selection rate to accommodate the communication bottleneck, our approach, referred to as MINDFL, selects the most informative model updates (i.e., models with high QoM values) in each iteration. In our client selection method, PS first collects the local losses of the clients. Subsequently, the PS evaluates the QoM for each client based on the received model losses. The PS selects a subset of clients with higher QoM values, considering the communication constraint. Moreover, to ensure client representation, we introduce randomness into the selection scheme. Specifically, the received models are clustered into groups based on quality, distinguishing between low-quality and high-quality groups. Random selection is then performed within the high-quality group. This selection method draws inspiration from the sortition method employed in governance [18], which leverages random representative sampling for the selection of public officials or jurors. This approach effectively mitigates the risk of exclusively selecting top-ranked clients, preventing overfitting to a small subset of clients within the federated learning system.

We implemented the selection mechanism in a general FL system. Our contributions can be summarized as follows:

- We proposed MINDFL, a client selection scheme in FL systems, to deal with noisy and imbalanced data with a communication capability constraint. We proposed a novel QoM metric to measure the most informative clients based on their loss value.
- We proposed a scheme to broaden the representation of clients and avoid selection biases over clients. The scheme employs a clustering model to divide all clients into two groups — the high QoM group and the low QoM group. A random selection is conducted on the high QoM group, which maintains learning efficiency and broadens client representation simultaneously.
- Comprehensive results demonstrate the effectiveness of MINDFL in improving learning speed and decreasing communication overhead. The high learning speed also provides by-product benefits of energy efficiency.

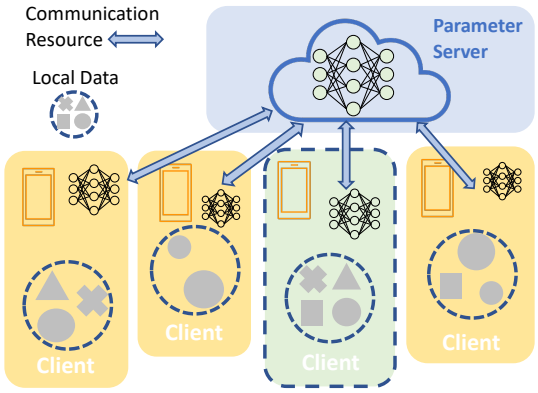


Fig. 1: Overview of client selection problem in federated learning with various client local models. To earn the best quality global model, selecting the most informative clients needs to be considered in depth.

II. Background and Motivation

A. Client Selection in FL systems

In the FL framework, the computation is mainly conducted on the client side. Clients train the model locally and parallelly and submit the model updates to PS periodically [19]. Client selection mechanisms are promising solutions to address the issues. In client selection mechanisms, PS chooses a fraction of the FL clients in each FL training round to reduce the communication overhead. The model update submissions introduce significant communication overhead to the participants. When the data at different clients are identically distributed and with the same quality, a random selection is demonstrated to achieve optimal performance. However, clients' local datasets are usually imbalanced since clients are notorious deployed in heterogeneous environments. And manual data labeling also introduces noise to data labels. As a result, the accuracy of the global FL model can be affected by clients with biased/noisy data. Therefore, a more advanced client selection that can thwart the effects of biased/noisy data is needed.

B. Challenges

Data Imbalance. In real-world FL systems, the local clients (e.g., mobile and IoT devices) usually exhibit heterogeneous data because of different functionality and usage environment. The heterogeneity brings some unprecedented challenges to the whole FL system. One of the typical challenges is imbalanced data, known as local data size variety. Some clients generate a large number of data while others may only collect a small amount of data. Another challenge is non-iid data [8], [20], where the data distributions (represented by data class) are different among clients. In this work, we focus on data imbalance and set non-iid data as a potential future work.

Data Noise. Local data is usually generated in independent contexts and tightly coupled with particular surrounding environments so that its quality cannot be guaranteed. On the one hand, the collected features may suffer from noises because of

software bugs, hardware faults, or environmental impacts. On the other hand, data labeling can also introduce noise. A noisy label usually results in a larger bad impact on a learning system since it can direct the model in a direction that significantly deviates from the optimal direction. A noisy label for a data point is a label that is different from the true class. In this paper, we focus on noisy-labeled data.

We assume PS has limited client state information (e.g., data quality, data amount) because of the privacy settings in FL systems. A Random client selection method may inadvertently involve the local updates trained by low-quality raw data (e.g., noisy data, which may jeopardize the global model quality. The goal of this paper is to design a client selection mechanism without knowing the explicit data information on clients.

III. System Model and Problem Formulation

A. System Model

We consider a federated learning system with N local clients and a central parameter server. We use $\{N\}$ to denote $\{1, 2, \ldots, N\}$ for simplicity. We use $C_{i|i\in\{N\}}$ to represent a client and $\mathcal{D}_i$ to represent the corresponding local data. The size of the local data is denoted by $|\mathcal{D}_i|$. Every data sample consists of a feature vector x_j and corresponding label y_j.

In each iteration, k from the total N clients will be selected for learning. We denote the selected set of clients as $\mathcal{S}$. The local model weights of C_i are w_i in the parameter space $\mathbf{W}$, $w_i \in \mathbf{W} \subseteq \mathbb{R}^d$, d is the model dimensionality. The entropy-based loss value for the local model is $\mathcal{L}(w_i, \mathcal{D}_i)$. The total loss function $\mathcal{F}(\theta)$ for the global model $\theta \in \mathbf{W}$ is calculated using the loss of the k selected clients and can be expressed as:

$$\mathcal{F}(\theta) = \frac{\sum_{i=1}^{N} \mathbb{1}(C_i \in \mathcal{S}) \sum_{\xi\in\mathcal{D}_i} \mathcal{L}(w_i, \xi)}{\sum_{i=1}^{N} \mathbb{1}(C_i \in \mathcal{S})|\mathcal{D}_i|} \tag{1}$$

where $\mathbb{1}(C_i \in \mathcal{S})$ is an indicator function, and it equals 1 when C_i is a member of set $\mathcal{S}$ (i.e., the condition is true), $|\mathcal{D}_i|$ denotes cardinality of $\mathcal{D}_i$(i.e., the number of elements in the set), and $\mathcal{L}(w_i, \xi)$ is C_i's loss value for sample ξ. The goal of the FL system is to jointly minimize the loss $\mathcal{F}(\theta)$.

We assume all clients follow the FL protocol honestly and are dedicated to providing their accurate information to PS with the aim of training an accurate global model. The local data bias is from environments and unintentional mistakes in labeling. Manipulations on feature vectors, data labels, model parameters, and any other information are out of the research scope of this paper.

B. Problem Formulation

This paper aims to improve the model accuracy of an FL system under imbalanced and noisy data with a constrained communication capability. We use k to denote the number of clients the FL system can support in each iteration and T to denote the number of total FL iterations. The objective of this work is to minimize the loss value (i.e., maximize model accuracy) by selecting the most informative clients in each iteration. The objective function is depicted below:

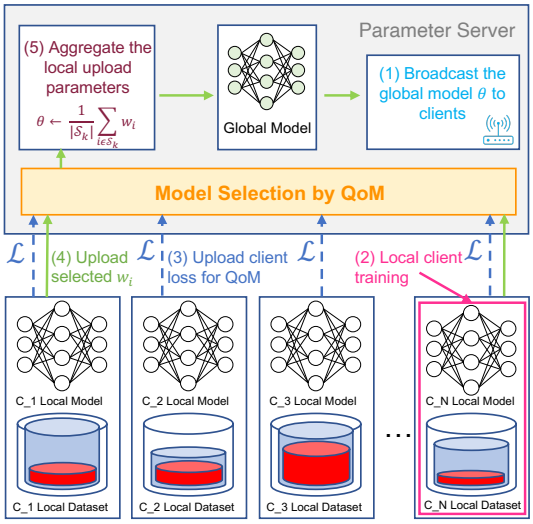


Fig. 2: MINDFL Design: Utilizing QoM metrics to measure local model contributions and mitigate selection biases in federated learning

$$\begin{aligned} \min_{\mathcal{S}_k, k\in\{T^*\}} \quad & \mathcal{F}(\theta_{T^*}) \\ s.t. \quad & |\mathcal{S}_i| \le k, \forall i = 1, 2, ..., T, \\ & T^* \le T \end{aligned} \tag{2}$$

where $\mathcal{S}_i$ denotes the set of selected clients in the i-th iteration, $|S_i|$ denotes cardinality of S_i(i.e., the number of elements in the set) and θ_T is the global model at iteration T. We use T^* to denote the number of iterations when the FL system achieves minimal loss. The goal aligns with the goal of general FL systems, which is to minimize the loss function. There are two conditions–the first condition implies that the number of clients selected in one iteration can not exceeds k. The second condition indicates that the total number of learning iterations can not exceed the maximal T.

IV. Detailed Design of MINDFL

We proposed MINDFL to improve the model accuracy of an FL system under imbalanced and noisy data with a constrained communication capability. The core of our FL algorithm is a client ranking and selection algorithm. We propose a metric to support the ranking algorithm, Quality of Model (QoM). Though multiple methods have been proposed to evaluate model quality using the model update/gradient [21], [22], we need to measure model quality without seeing the model update/gradient because of communication overhead concern. In this section, we will first introduce the building block, QoM, and then describe the FL process in detail with a focus on client selection.

A. Quality of Model

We design a metric, QoM, to capture client data utility and the local model's contribution towards the global model. Motivate by [23] that approximates the statistical utility by local loss, we also incorporate the local loss value into the design of QoM. Generally, a loss value measures the estimation error between model predictions and the ground truth. Since a gradient is calculated by taking the derivative of the loss concerning the current local parameters, a larger loss results in a larger gradient and, thus, a larger model update step [24], [25]. Therefore, [23] regards a client with a more significant loss as a more informative one.

In contrast to their design, we propose that a larger loss may indicate noisy and imbalanced data. Theoretical analysis [24], [25] has demonstrated that noisy data can increase loss value significantly. The loss value of a client can be impacted by both the model utility itself and corresponding validation data. To prevent the influence of the diverse validation data on clients, PS distributes small validation data $\mathcal{D}_{val}$ to all clients at the beginning of the federated learning training process. And the loss of all clients is evaluated on the same validation dataset.

Based on this, we further demonstrate that the noisy and imbalanced data usually result in a significantly large loss which is larger than the loss of a normal informative local model. Based on this observation, we use the loss value to indicate the model quality. And the QoM of client C_i is designed as:

$$\mathcal{Q}_i = \frac{1}{\sum_{\xi \in \mathcal{D}_{val}} \mathcal{L}(w_i, \xi)} \tag{3}$$

where $|\mathcal{D}_{val}|$ denotes cardinality of $\mathcal{D}_{val}$(i.e., the number of elements in the set). The QoM is a float number and its range is related to the learning task difficulty (see Fig. 3).

B. Federated Learning with Client Selection

Although we can use QoM to differentiate models with high utility, persistent use of high-QoM models may result in biased client participation. To address this problem, inspired by the philosophy of sortition [18], [26], we proposed to incorporate randomness into our QoM-based client selection to balance model quality and client diversity/fairness. This section provides an overview of the entire learning process, emphasizing the proposed client selection mechanism

PS collects the QoM values from all clients and proceeds to classify them into two clusters based on their QoM values. The preferred cluster comprises clients with the top 50 percent (configurable) of QoM values. The subsequent section outlines the detailed process of federated learning.

As shown in Fig. 2, when the system starts up, PS initializes the global model and works by iterations as follows:

1) PS broadcasts the global model parameter θ together with a small validation dataset $\mathcal{D}_{val}$ to all the N clients.
2) Client $C_i|_{i\in\{N\}}$ initializes the local model with θ. Client $C_i|_{i\in\{N\}}$ continue to train the local model using local dataset $\mathcal{D}_i$.
3) After local training is completed, all clients upload their entropy-based loss value $\mathcal{L}(w_i, \mathcal{D}_{val})$ to PS for model quality ranking.
4) PS utilizes $\mathcal{L}(w_i, \mathcal{D}_{val})$ to evaluate the QoM Q_i of each client C_i, and cluster the QoM scores into two clusters: $\mathcal{S}_h$ denoting the cluster with higher QoM scores and $\mathcal{S}_l$ denoting the cluster with lower QoM scores. A random selection is then performed over cluster $\mathcal{S}_h$, and the set of selected clients $\mathcal{S}$ is represented by:

$$\mathcal{S} = \begin{cases} \mathcal{S}_h & \text{if } k \geq |\mathcal{S}_h| \\ \text{RandomSample}(\mathcal{S}_h, k) & \text{otherwise} \end{cases} \tag{4}$$

   where RandomSample$(\mathcal{S}_h, k)$ represents randomly selecting k members from the set $\mathcal{S}_h$. This selection method is motivated by sortition in governance. Sortition, also known as selection by lottery, is originally proposed to select public officials or jurors using a random representative sample [26]. The random selection effectively mitigates the risk of selecting only top k clients, thus avoiding FL systems overfitting to such a small portion of clients.
5) PS sends a notification to the selected clients and requests their local model updates $\delta = (w_i - \theta)_{|C_i \in \mathcal{S}}$. PS then aggregates the received models and updates the global model by $\theta \leftarrow \theta + \frac{\alpha}{|\mathcal{S}|} \sum_{C_i \in \mathcal{S}} \delta_i$ where α denotes the learning rate.

The FL system repeats the steps until the global model converges or exceeds a predefined iteration number T. PS outputs the final global model at the end of the learning process. Model consumers can customize the final global model to their task by continually training with their local data.

V. System Evaluations

We simulate an FL system with imbalanced and noisy local data to evaluate the effectiveness of MINDFL in improving model accuracy. The system is implemented using the PyTorch framework. We ran all the experiments on a server equipped with an Intel Core i9-11900K CPU 3.50GHz×16, a GeForce RTX 3080 GPU, and Ubuntu 20.04.5 LTS.

A. Experimental Settings

The experimental evaluation of MINDFL is performed on two benchmark datasets, namely, the MNIST dataset [27] and the N-Baiot dataset [28]. **MNIST** is a collection of 70,000 grayscale images of handwritten digits from 0 to 9, with dimensions of 28x28 pixels. It is divided into a training set of 60,000 samples and a testing set of 10,000 samples. **N-BaIoT** is a comprehensive collection of network traffic data of IoT devices. It is designed to evaluate intrusion detection systems (IDS) in IoT networks. It contains a diverse range

TABLE I: N-BaIoT Dataset

Index	Devices Make and Model	Device Type
1	Danmini	Doorbell
2	Ennio	Doorbell
3	Ecobee	Thermostat
4	Phillips B120N/10	Baby monitor
5	Provision PT-737E	Security camera
6	Provision PT-838	Security camera
7	SimpleHome XCS7-1002-WHT	Security camera
8	SimpleHome XCS7-1003-WHT	Security camera
9	Samsung SNH 1011 N	Webcam

of network traffic samples, including the benign class and ten malicious classes: five classes from the Mirai attack family and the other five from the BASHLITE attack family. The whole dataset contains 6,506,674 malicious records and 556,932 benign records, and each record contains 115 attributes. They are generated by nine commercial IoT devices as listed in Table. I.

The neural network used for MNIST dataset is the Resnet18 [29]. For N-BaIoT, we devise a fully connected neural network comprising one input layer, three hidden layers, and an output layer. The model accepts inputs with 115 features and makes predictions for the 11 possible output classes. The three hidden layers are with 64, 32, and 16 nodes, respectively. To introduce non-linearity into the network, the ReLU (Rectified Linear Unit) activation function is applied after each hidden layer.

In the FL system, we set the default client number as 50. Each client trains its local model using an Adam optimizer with a learning rate of $\alpha = 0.0001$ and a batch size of $b = 100$, with a local epoch of $e = 1$. The number of clients selected for uploading models in each FL iteration is $k = 5$.

B. Evaluation Metric and Baselines

To evaluate MINDFL, we employ global model test accuracy throughout the FL training iterations as the major metric. We set the maximal iteration number as T. We compared MINDFL with three baseline selection methods, including the well-known random selection method [4] and two state-of-the-art schemes in [23], namely Oort1 and Oort2. We briefly summarized the three client selection methods as follows.

- **Random** selection methods randomly select k clients out of N in each FL iteration.
- **Oort1** [23] defines a utility function for each client as: $\mathcal{U}(i) = |\mathcal{D}_i|\sqrt{\frac{1}{|\mathcal{D}_i|}\sum_{\xi\in\mathcal{D}_i}\mathcal{L}(\xi)^2}$. In Oort1, a client with a larger utility value will be selected for the upcoming iteration.
- **Oort2** is an advanced version built upon Oort1. It incorporates a fairness component into the selection to increase the likelihood of selecting clients that have not been previously selected.

C. Evaluation Results

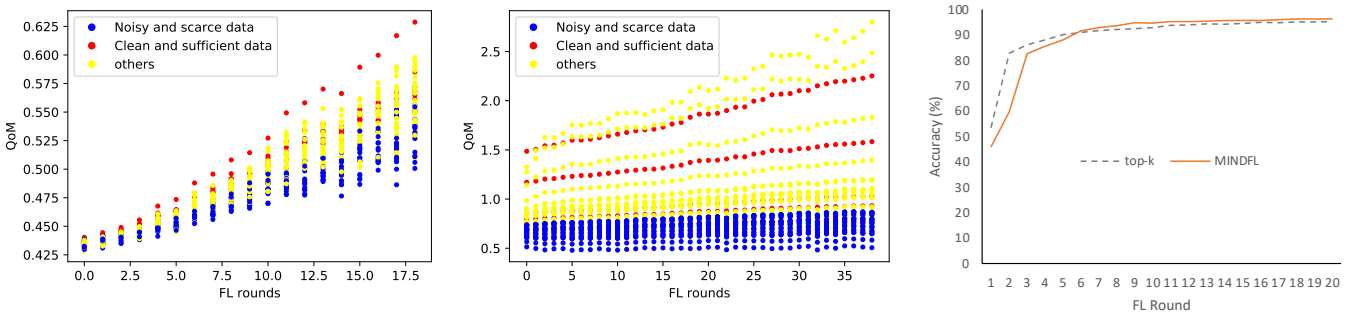
The proposed client selection mechanism aims to improve accuracy under imbalanced and noisy data and minimize the accuracy loss under clean and balanced data. With this goal, our evaluation includes two parts: results under noisy and imbalanced data and results under clean and balanced data.

1) Noisy and Imbalanced Data: In order to assess the effectiveness of MINDFL in improving model accuracy under noisy and imbalanced data, we simulate the data distributions over clients. We provide two different levels of imbalance (denoted by **IMB1** and **IMB2**). When we simulate noisy data, we assume that a small portion of a client's data can be with incorrect labels. For the small portion of data, we randomly generate a label for a data record different from the original/true class. And the ratio of incorrect labels for different clients can vary. we assume the ratio of incorrect labels among clients follows a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ where μ denotes the mean grey value and σ denotes its standard deviation. Two levels of label noise (denoted by **N1** and **N2**) are considered. The detailed settings are introduced below.

- **IMB1**: In this setting, 10% of the clients possess 90% of the data, while the remaining 90% of clients hold 10% of the data.
- **IMB2:** This setting allocates 80% of the data to 20% of the clients, and the other 80% of clients possess only 20% of the data.
- **N1**: We consider a Gaussian noise $\mathcal{N}(\mu = 0.1, \sigma = 0.1)$, indicating the average ratio of incorrect label among clients are 10%.
- **N2**: We consider a Gaussian noise $\mathcal{N}(\mu = 0.2, \sigma = 0.1)$, indicating the average ratio of incorrect label among clients are 20%.

2) The Power of QoM in Differentiating Noisy and Imbalanced Data : Before evaluating the performance of the MINDFL system, we first illustrate the power of the proposed QoM metric in differentiating models trained by noisy and scarce data from models trained by clean and sufficient data. As discussed in Sec. V-C1, there will be four types of clients: noisy and scarce data (denoted by D1), noisy and sufficient data (denoted by D2), clean and scarce data (denoted by D3), and clean and sufficient data (denoted by D4). In Fig. 3a and Fig. 3b, we use red dots to represent and clean and sufficient data D4, blue dots to represent noisy and scarce data D1, and yellow dots to denote D2 and D3. The results demonstrates the effectiveness of QoM in differentiating bad data from good data. And this observation further confirms the QoM-based client selection is suitable for FL systems with noisy and imbalanced data.

3) Advances of Sortition-based Selection: To demonstrate the effectiveness of the proposed sortition-based selection, we compare MINDFL with a selection mechanism that selects the clients with top-k QoMs (denoted as top-k). All other settings are the same in the comparison. As shown in Fig. 3c, we can see MINDFL achieves higher accuracy at the end of the training though its accuracy grows slower at the first few FL rounds. The results demonstrate that MINDFL incorporating



(a) (b) (c)

Fig. 3: (a) QoM of clients throughout the learning process on the MNIST dataset. (b) QoM of clients on the N-BaIoT dataset. (c) Model Accuracy on the MNIST dataset.

the sortition mechanism generalizes better than the top-k method because it can utilize more diverse data.

4) Effectiveness of MINDFL in Noisy and Imbalanced Data: We present the test accuracy of the proposed FL system on the MNIST dataset and the N-BaIoT dataset in Figure.5. The first four sub-figures (a)(b)(c)(d) show the results on the MNIST dataset with different combinations of imbalance level and noise level. The figures show that the proposed method achieves higher accuracy than other baselines. Additionally, our method's model convergence speed is higher than other methods. It is worth mentioning that random selection has higher robustness against noisy and imbalanced data than other priority-based methods, such as oort1 and oort2. The random selection method achieves a little lower accuracy than the proposed method but with a much lower convergence speed.

To further verify the practicality of our method, we experiment MINDFL with the N-BaIoT dataset. For the BaIoT dataset, we classify the ten types of attacks listed in Table.I. As shown in Figure.5 (e) and (f), our method achieved higher accuracy than other baselines. Similarly, when we increase the noise level to N2 in Figure.5(g) and (h), the experimental results show the same trend. And our proposed method achieves the highest accuracy. We find that for the N-BaIoT dataset, although we could not achieve the fastest convergence, the accuracy of our model gradually improved with the increase of the FL rounds through training. The random selection method could reach convergence faster, but the final accuracy remains lower than MINDFL. The possible reason is that the model has been directed to a local optimum because of the low-quality data. Similarly, both Oort1 and Oort2 achieve a lower accuracy than our proposed method, indicating that the two methods are not effective when facing noisy and imbalanced data.

In Table II, we further present the highest accuracy achieved by MINDFL and baseline methods ('rand', 'oort1', 'oort2') in various data distribution settings. From the table, we can see our method consistently outperforms other baselines by achieving the highest accuracy across the two datasets.

The convergence speed of FL is an important metric since it is related to both the computation and communication

TABLE II: Test accuracy

Dataset	**MNIST**			
Setting	**IMB1+N1**	**IMB2+N1**	**IMB1+N2**	**IMB2+N2**
rand	94.13	94.76	92.67	94.41
MINDFL	**96.52**	**96.38**	**94.65**	**96.23**
oort1	46.18	78.02	50.13	68.56
oort2	51.41	76.59	46.99	68.40
Dataset	**N-BaIoT**			
rand	73.93	78.99	74.93	74.81
MINDFL	**79.27**	**80.98**	**81.99**	**76.51**
oort1	75.21	73.71	70.35	70.68
oort2	73.57	75.47	70.68	69.58

overheads of FL systems. A higher convergence speed implies saving more computation and communication resources among clients and PS. We demonstrate the convergence performance of MINDFL across FL rounds in Figure.6. Our proposed method achieves the highest convergence speed on the MNIST dataset. In the more complex Ba-IoT dataset, while our model achieves relatively high accuracy quickly, its convergence speed is relatively slow due to the gradual improvement of accuracy in subsequent training.

Additionally, we evaluate the MINDFL under a limited training iteration budget considering the fact that wireless devices often have limited computing power and battery life [30], [31]. We vary the total learning round T from 4 to 10 and show the results in Table III. Table III highlights that MINDFL outperforms the other methods in terms of test accuracy for iteration budget setting to $T = 4, 6, 8, 10$. The orange row in the table represents our algorithm, and the results are based on the IMB1+N1 setting of the two datasets. Results under other experimental settings are similar, and we do not include these results because of space limitations. The observations demonstrate that MINDFL is suitable for FL systems with a limited learning iteration budget.

5) Performance of MINDFL Under Clean and Balanced Data: We further evaluated the performance of MINDFL using a clean and balanced dataset. In this scenario, the entire dataset is evenly divided into 50 clients and no label noise is included. As shown in Fig. 7, MINDFL has a similar learning curve with the baselines in terms of test accuracy and

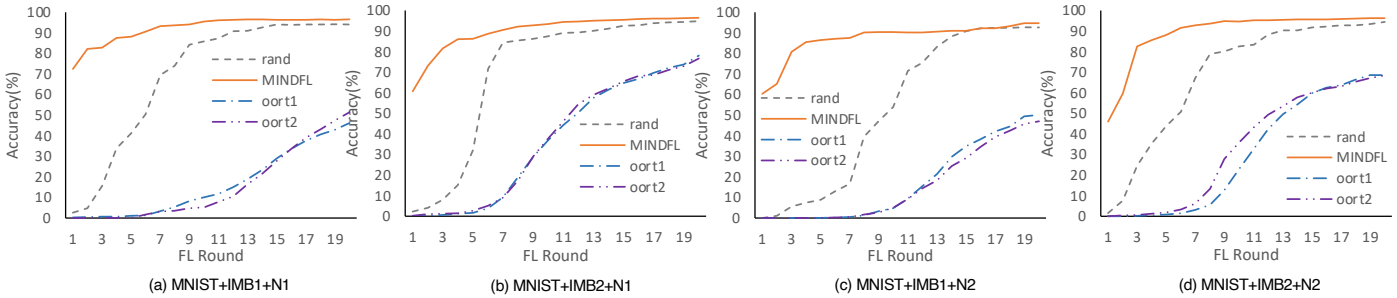


Fig. 4: Round-to-Accuracy performance of MINDFL with Random Selection, Oort1, and Oort2 in N1 and N2 noise settings and IMB1, IMB2 imbalance settings on MNIST dataset.

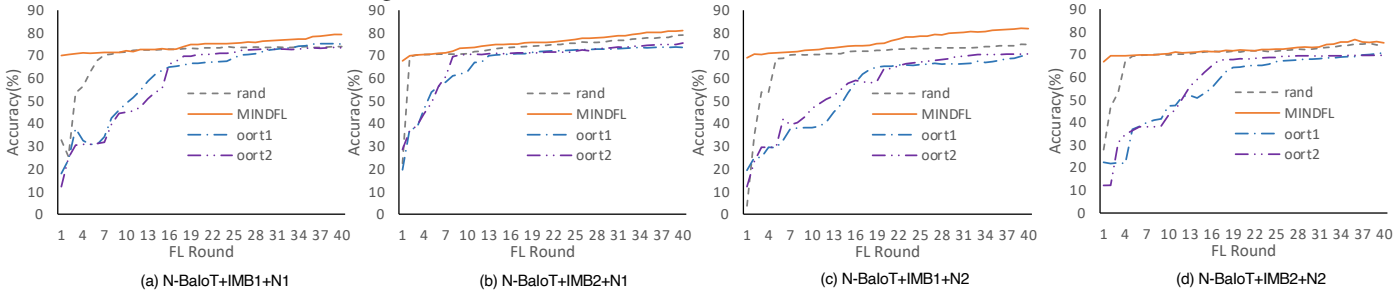


Fig. 5: Round-to-Accuracy performance of MINDFL, Random Selection, Oort1, and Oort2 in N1 and N2 noise settings and IMB1, IMB2 imbalance settings on N-BaIoT dataset.

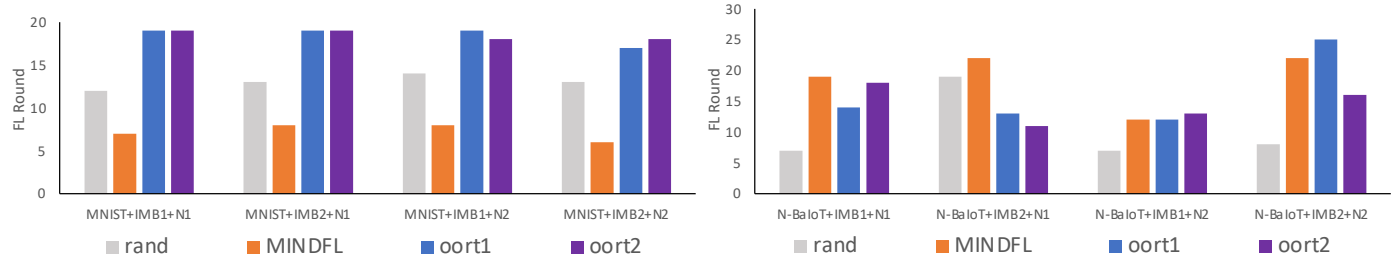


Fig. 6: Empirical convergence performance of MINDFL, Random Selection, Oort1, and Oort2, in N1 and N2 noise settings and IMB1, IMB2 imbalance settings.

TABLE III: Test Accuracy under Limited FL Round

Setting	Method	T=4	T=6	T=8	T=10
MNIST IMB1+N1	rand	33.75%	50.52%	73.90%	85.93%
	MINDFL	87.41%	90.55%	93.57%	95.41%
	oort1	0.69%	1.29%	5.38%	10.15%
	oort2	0.02%	0.08%	1.62%	5.12%
N-BaIoT IMB1+N1	rand	56.35%	67.80%	70.55%	71.51%
	MINDFL	71.14%	71.13%	71.25%	72.03%
	oort1	32.53%	30.80%	42.40%	48.82%
	oort2	30.80%	31.09%	39.71%	45.00%

convergence speed on the MNIST dataset. For the N-BaIoT dataset, MINDFL achieves a little lower accuracy compared to random selection, Oort1, and Oort2. This is because MINDFL favors model updates with smaller loss values, and this design slows down the learning process under ideal data conditions. In contrast, random selection allows for broader participation among clients, while Oort's method targets clients with a higher model utility.

6) Summary: There is always a trade-off between filtering bad models and broadening good model participation. The proposed MINDFL is effective in filtering models with noisy and imbalanced data and inevitably filters out some good models when all models are trained by clean and balanced data. In summary, the proposed client selection mechanism in MINDFL is effective in improving model accuracy under imbalanced and noisy data by only sacrificing a small accuracy under clean and balanced data.

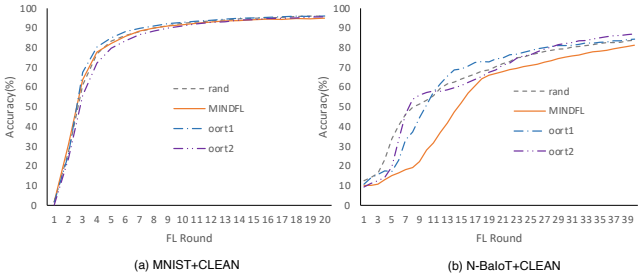


Fig. 7: Round-to-Accuracy Performance of MINDFL with Random Selection, Oort1, and Oort2 in **balanced** and **noise-free** distribution on MNIST and N-BaIoT datasets.

D. Discussions

We empirically demonstrated the convergence of the proposed approach, and the theoretical analysis on convergence could be an extension for future work.

VI. CONCLUSION

The client selection problem in federated learning has received significant attention in wireless networks considering the communication bandwidth limitation. We propose MINDFL to prioritize clients with the most valuable data for aggregation. We use a Quality-of-Model metric to evaluate the contribution of each client to the global model. Additionally, we group clients into two clusters and use random selection over the cluster with higher QoM to balance the selection diversity and model quality. Our comprehensive experimental evaluations demonstrate that MINDFL achieves the highest model test accuracy in the presence of noise and imbalanced data. Overall, MINDFL mitigates the problems posed by data imbalance and noise, making it a promising solution for improving the performance of federated learning in practice.

ACKNOWLEDGMENT

This work was supported in part by the Office of Naval Research under grant N00014-19-1-2621, and by the US National Science Foundation under grant 2312447.

REFERENCES

[1] 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.

[2] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "Clusterfl: a similarity-aware federated learning system for human activity recognition," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 54–66, 2021.

[3] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[5] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," *Advances in neural information processing systems*, vol. 30, 2017.

[6] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, p. 2032, 2022.

[7] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[8] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang, "Client selection for federated learning with non-iid data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24462–24474, 2021.

[9] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1698–1707, IEEE, 2020.

[10] Y. J. Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 10351–10375, PMLR, 2022.

[11] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv:2010.13723*, 2020.

[12] E. Tsogbaatar, M. H. Bhuyan, Y. Taenaka, D. Fall, K. Gonchigsumlaa, E. Elmroth, and Y. Kadobayashi, "Del-iot: A deep ensemble learning approach to uncover anomalies in iot," *Internet of Things*, vol. 14, p. 100391, 2021.

[13] Y. Li, Y. Chen, K. Zhu, C. Bai, and J. Zhang, "An effective federated learning verification strategy and its applications for fault diagnosis in industrial iot systems," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 16835–16849, 2022.

[14] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, and L. Liang, "Self-balancing federated learning with global imbalanced data in mobile systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 59–71, 2020.

[15] Y. Wang, G. Gui, H. Gacanin, B. Adebisi, H. Sari, and F. Adachi, "Federated learning for automatic modulation classification under class imbalance and varying noise condition," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 86–96, 2021.

[16] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa, "Noise removal in the presence of significant anomalies for industrial iot sensor data in manufacturing," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7084–7096, 2020.

[17] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, "An overview of iot sensor data processing, fusion, and analysis techniques," *Sensors*, vol. 20, no. 21, p. 6076, 2020.

[18] O. Dowlen, *The political potential of sortition: A study of the random selection of citizens for public office*, vol. 4. Andrews UK Limited, 2017.

[19] C. Zhang, R. Li, and H. Jiang, "Optimization of gpu kernels for sparse matrix computations in hypre," 2019.

[20] 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.

[21] A. Katharopoulos and F. Fleuret, "Not all samples are created equal: Deep learning with importance sampling," in *International conference on machine learning*, pp. 2525–2534, PMLR, 2018.

[22] P. Zhao and T. Zhang, "Stochastic optimization with importance sampling for regularized loss minimization," in *international conference on machine learning*, pp. 1–9, PMLR, 2015.

[23] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection.," in *OSDI*, pp. 19–35, 2021.

[24] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[25] T. Wang, Q. Sun, S. Pranata, K. Jayashree, and H. Zhang, "Equivariance and invariance inductive bias for learning from insufficient data," *arXiv preprint arXiv:2207.12258*, 2022.

[26] H. Landemore, "Deliberation, represenation, and the epistemic function of parliamentary assemblies: a burkean argument in favor of descriptive representation," in *International Conference on Democracy as Idea and Practice*, pp. 13–15, 2010.

[27] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[28] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[30] C. Zhang, H. Yu, Y. Zhou, and H. Jiang, "High-performance and energy-efficient fpga-gpu-cpu heterogeneous system implementation," in *Advances in Parallel & Distributed Processing, and Applications: Proceedings from PDPTA'20, CSC'20, MSV'20, and GCC'20*, pp. 477–492, Springer, 2021.

[31] H. Yu, C. Zhang, and H. Jiang, "A fpga-based heterogeneous implementation of ntruencrypt," in *Advances in Parallel & Distributed Processing, and Applications: Proceedings from PDPTA'20, CSC'20, MSV'20, and GCC'20*, pp. 461–475, Springer, 2021.