ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230628

XFed: Improving Explainability in Federated Learning by Intersection Over Union Ratio Extended Client Selection

Juan Zhao [†], Yuankai Zhang [†], Ruixuan Li [*], Yuhua Li , Haozhao Wang , Xiaoquan Yi and Zhiying Deng

School of Computer Science and Technology, Huazhong University of Science and Technology, China

Abstract. Federated Learning (FL) allows massive clients to collaboratively train a global model without revealing their private data. Because of the participants' not independently and identically distributed (non-IID) statistical characteristics, it will cause divergence among the client's Deep Neural Network model weights and require more communication rounds before training can be converged. Moreover, models trained from non-IID data may also extract biased features and the rationale behind the model is still not fully analyzed and exploited. In this paper, we propose eXplainable-Fed (XFed) which is a novel client selection mechanism that takes both accuracy and explainability into account. Specifically, XFed selects participants in each round based on a small test set's accuracy via cross-entropy loss and interpretability via XAI-accuracy. XAI-accuracy is calculated by Intersection over Union Ratio between the heat map and the truth mask to evaluate the overall rationale of accuracy. The results of our experiments show that our method has comparable accuracy to state-of-the-art methods specially designed for accuracy while increasing explainability by 14%-35% in terms of rationality.

1 Introduction

Federated Learning aggregates training models from distributed clients to provide a global model without sharing clients' raw data [8]. It is crucial to adopt this collaborative paradigm in order to achieve privacy-preserving machine learning. The central server sends the global model to each client and receives the trained model from selected clients to build a new global model in each round of communication. The model is refined over a number of communication rounds until it converges.

Firstly, non-IID data among FL participants can significantly impact the global model's precision [19, 34, 25, 20], but increasing the communication rounds between the server and clients can help moderate this effect [32, 26]. As FL participants exhibit widely heterogeneous statistical characteristics, selecting clients can be an effective strategy to optimize training efficiency [15, 17, 27, 6, 23]. *Secondly*, the black-box nature of Deep Neural Network (DNN) can limit its application, making it difficult to understand the reasons behind the AI's decisions [28, 31, 7, 10, 11]. Explainable Artificial Intelligence (XAI) aims to provide interpretable and human-friendly explanations of AI decisions. By using post-hoc XAI methods [5], we can generate a salience map or a heat map that provides an understanding of the inference process.

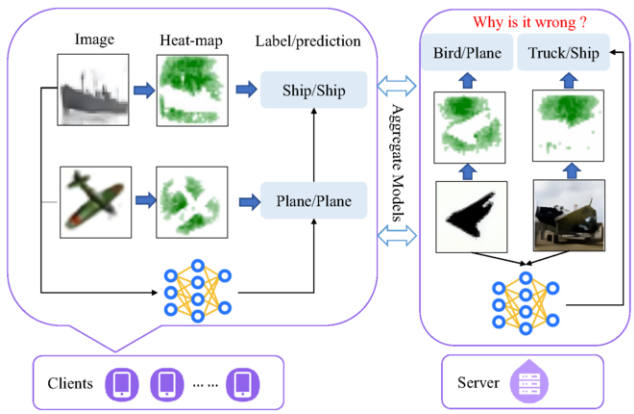

Figure 1. Why is it wrong? The client on the left has correctly identified the images of the aircraft and the ship, but the features it extracted were from the background rather than the main objects themselves. Consequently, when the server aggregates the features for the bird and the truck, incorrect features are extracted, leading to incorrect judgments being made.

However, existing client selection methods primarily focus on task performance and time to precision. Developing a more effective method that utilizes the existing solution is imperative to address the need for a model with better explainability. This is particularly challenging in Federated Learning since the samples are not independently and identically distributed, which may lead to biased feature extraction.

Figure 1 illustrates an instance of a classification task where the client's prediction is correct but based on the wrong set of features. As a consequence, the server extracted wrong features for other objects such as the bird and the truck, leading to incorrect predictions. The ultimate goal is to ensure that the high accuracy of the aggregated model is based on the right extracted features that align with human understanding. Biased feature extraction by some clients' models can negatively impact the convergence of the final model, potentially leading to incorrect or misleading features. Addressing bias in feature extraction is critical for improving the interpretability of the final model. Although post-hoc XAI methods provide valuable insights into a model's decision-making process, there is currently no universally accepted method for assessing model explainability. In the context of client selection architectures in Federated Learning,

[*] Corresponding Author. Email: rxli@hust.edu.cn
[†] Juan Zhao and Yuankai Zhang contribute equally to this work.

Please check ArXiv or contact the authors for any appendices or supplementary material mentioned in the paper.

adapting XAI evaluations can be a challenging task.

Theoretical analysis of clients' weight difference [30, 33] and Kullback-Leibler (KL) divergence [13] suggests that increasing the number of participating clients in the overall training process minimizes the cross-entropy between the model's predictions and the target distribution. However, this can be challenging in federated learning due to high communication cost and variations in the data distribution across clients.

To address these challenges, we propose XFed, a method that encourages more clients to participate in the training process while filtering out underperforming clients. To determine which clients are selected, we utilize a relatively small number of test methods with an "ideal" distribution of data. This means that the test data is selected and distributed in a way that represents the target distribution we want to achieve. We also provide a *truth_mask* [1] with our test dataset to identify the object's location for evaluating the heat map generated by the model. By employing the *truth_mask* as a sanity check, we ensure that the extracted features are valid and mitigate the risk of feature-attribution-based explanations being misleading [1]. As the training progresses, the heat map should converge towards the *truth_mask*, indicating increasing rationality of the model's predictions.

Our experimental results show that our approach increases interpretability by 14%-35% in terms of CIFAR-10 and MNIST plausibility for classification task datasets. Our major contributions are summarized in the following.

- We examine the theory of clients' weight difference and KL divergence, identifying two key aspects that can accelerate training with distributed non-IID data.
- We introduce the XFed client selection mechanism, which enhances FL accuracy while minimizing communication requirements.
- We enhance heat map measurements for XAI accuracy by extending Intersection over Union (IoU) to IoU-Ratio.

2 Related Work

Federated Learning with enhanced privacy [18] aggregates training results from multiple participants to produce a *better* DNN model with as little communication overhead as possible. In light of the FL's non-IID statistical feature [9] and the need to understand the black-box DNN model, saving communication rounds and understanding the reasoning behind DNN decisions are essential.

2.1 Client Selection in FL

Existing work typically focuses on classification tasks. To trade off between training speed and performance in federated learning (FL), Oort [15] introduces a two-phase client selection approach. In the first stage, the server assigns weights to each client based on their time cost and task loss. In the next stage, the clients with the higher weight are selected for the next round. Based on this, PyramidFL [17] provides a more fine-grained evaluation of time cost and task performance, allowing for further refinement of the client selection process. Although existing FL paradigms [14, 4] propose several optimization schemes for client selection, since they only focus on accuracy and efficiency and ignore the study of model explainability, they offer great potential for improvement in various FL applications.

[1] https://github.com/XFed2023/XFed

2.2 Explanation and Interpretability in FL

XAI aims at generating high-quality interpretable, intuitive, human-understandable explanations of AI decisions [5].

Recent research has begun to incorporate interpretable and rational analysis into FL. Liu et.al. [24] proposed Contribution-Aware FL (CAreFL) framework to provide fair and explainable FL participant contribution evaluation in an efficient and privacy-preserving manner. But the contribution-aware framework only impacts the aggregation stage of FL. It doesn't impact the training stage in the clients to provide a model with a better rationale. [22] demonstrated a novel FL system for interpretable time series classification (TSC) by extending the concept of FL to consider both stronger security and model explainability. However, these methods limit the models' type and architecture.

The explainable vertical FL (EVFL) framework [3] includes the credibility assessment strategy, the federated counterfactual explanation, and the importance rate (IR) metric. It studies the problem of how to select features under Vertical FL (VFL). However, the problem we are studying is how to improve rationale under Horizontal FL (HFL) to reduce the communication overhead and improve the explainability.

In conclusion, in spite of some cutting-edge explainability research in FL, model interpretability is not taken into consideration in current client selection procedures. Therefore, it is vital to explore this area in order to identify and optimize clients according to rational and accurate criteria. We only select the client in the early stage of training in order to use fewer rounds to achieve the specified accuracy.

3 Motivation

3.1 The Inspiration From Distribution Difference

A crucial component of FL is the iterative aggregation of model updates across multiple client devices, many of which may have slow or unstable network connections. The first step is for eligible clients to check in. In the next step, FL is performed synchronously. Clients are selected under certain criteria for each training round. Selected devices download global models from the server and train them on local datasets. The server then aggregates the updates from clients. Achieving high accuracy while minimizing communication costs is the goal.

Taking C-class classification as an example, let (x, y) denote a particular labeled sample. f_i predicts the probability that the sample belongs to the i-th class, where $\forall i \in \{1, ..., C\}$. Let w denote model weights. In FL, suppose there are K client devices checked in. The k-th device follows the data distribution p^k, which is a joint distribution of the samples x, y on this device. In round t, where $\forall t \in \{1, ..., T\}$, client k downloads the latest global weight w_{t-1} from the server and performs training to train w_t^k using learning rate η locally.The detailed math explanation for FL has described in Appendix A. By using a similar bounding technique adopted in [30, 33], we can derive the Equation (1) for weight divergence in the last round. In other words, though local models are trained with the same global weights, there is an implicit connection between the distribution of data $\sum_{i=1}^{C} \left| p^{k'}(y = i) - p^k(y = i) \right|$ on different devices and the discrepancy of local models' weight $\left\| \boldsymbol{w}_t^{k'} - \boldsymbol{w}_t^k \right\|$. The weight divergence equation is listed as below:

$$\left\|\boldsymbol{w}_T^{k'}-\boldsymbol{w}_T^{k}\right\| \leq \eta g_{\max}\left(\boldsymbol{w}_{\text{T-1}}\right) \sum_{i=1}^{C}\left|p^{k'}(y=i)-p^{k}(y=i)\right|, \quad (1)$$

where

$$g_{\max}(\boldsymbol{w}) := \max_{i=1}^{C}\left\|\nabla_{\boldsymbol{w}} \mathbb{E}_{\boldsymbol{x}|y=i}\left[\log f_i\left(\boldsymbol{x}, \boldsymbol{w}\right)\right]\right\|.$$

Convergence is a necessary condition to achieve good performance after T rounds for each client:

$$\text{minimize}\left\|\boldsymbol{w}_T^{k'}-\boldsymbol{w}_T^{k}\right\| \quad . \quad (2)$$

When we achieve the performance goal, the trained global model should show good performance to each client. It means the weight across different clients is very similar. According to Equation (1) and (2), if we want to minimize $\left\|w^{k'}-w^{k}\right\|$, we want to minimize $\left|p^{k'}-p^{k}\right|$.

We can define a virtual client, represented by the weight w^{virt} and the distribution p^{virt}, which serves as a benchmark for evaluating the performance of a model. The virtual client's distribution may be determined based on prior knowledge, assumptions about the data, or empirical observation of feature distributions.

Therefore, the goal is:

$$\text{minimize}\left\|\boldsymbol{w}_T^{k}-\mathsf{w}_{\mathsf{T}}^{virt}\right\| \quad . \quad (3)$$

Then, we can get the below equation:

$$\left\|\boldsymbol{w}_t^{k}-\mathsf{w}_{\mathsf{t}}^{virt}\right\| \leqslant \eta g_{\max}\left(\boldsymbol{w}_{\text{t-1}}\right) \sum_{i=1}^{C}\left|p^{k}(y=i)-\mathsf{p}^{virt}(\mathsf{y}=\mathsf{i})\right|. \quad (4)$$

After exploring the relationship between w_t^k and $\mathsf{w}_{\mathsf{t}}^{virt}$ in Equation (3) and (4), let's continue to discuss the distribution part: p^k and p^{virt}. KL divergence is used to calculate the difference between the actual and the observed probability distribution. Equation (5) shows KL divergence between p^k and p^{virt}:

$$\mathrm{KL}\left(\mathsf{p}^{virt} \| p^{k}\right)=\mathbb{E}_{\mathsf{x} \sim \mathsf{p}^{virt}}\left[-\log p^{k}(\mathsf{x})\right]-H\left(\mathsf{p}^{virt}\right). \quad (5)$$

$H\left(\mathsf{p}^{virt}\right)$ is the entropy function of the empirical distribution. $\mathbb{E}$ is the cross entropy of the empirical distribution p^{virt} and the predicted distribution p^k.

To minimize the KL divergence between the predicted and empirical distribution, we need to maximize the entropy function of the empirical distribution p^{virt}. This Entropy is the average amount of information we get from the empirical distribution. It is trying to be close to the real world. Its entropy will be more accurate if more data are available. We should encourage more clients to participate FL to add as much data as possible.

The virtual p^{virt} is assumed to be the empirical distribution, so the H can be treated as kind of fixed. We will now focus on the first part of this Equation (5). The Cross-Entropy of the empirical distribution and predicted distribution must be minimized in order to minimize KL divergence. As a result, KL divergence is reduced and accuracy is increased.

Therefore, leveraging more data and minimizing cross-entropy can be used to optimize the training process for non-IID data. There appears to be a contradiction between these two conditions at first glance. But it can be achieved by the following method in the joined non-IID clients: a). encouraging as many as possible clients to join the FL; b). filtering out the models with the lowest accuracy using cross-entropy loss in each round.

To evaluate the performance of the clients' models, we provide each client with a small test dataset that contains an ideally distributed statistic feature.

For example, in our experiments, we randomly sampled 3 samples from each category of CIFAR-10 to create a small test set of 30 samples, and manually annotated these samples with masks. Each category of CIFAR-10 has 5000 training samples, so CIFAR-10 can be treated as an independent and identically distributed dataset, and the small dataset obtained also satisfies the characteristics of being independent and identically distributed. This feature is representative of the true underlying distribution in the population. The client calculates p^{virt} using this feature and reports the cross-entropy to the server. The server can use this information to decide whether to include the client in this round of training.

In contrast to other methods of selecting FL clients, our method filters clients' models after the local model is trained to maximize data leverage while saving communication effort. In different rounds, the system eliminates the different lowest performers.

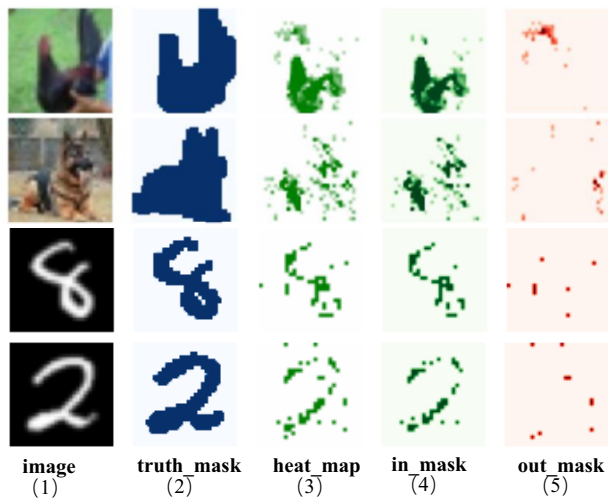

Figure 2. $Heat\_map$ and $truth\_mask$ example: the columns specifically show: the original image; the $truth\_mask$; the generated $heat\_map$; the pixels of $heat\_map$ inside the $truth\_mask$ and the pixels of $heat\_map$ outside the $truth\_mask$.

3.2 *Prediction Rationale Analysis*

By addressing non-IID clients and improving model accuracy and interpretability, XAI methods can provide insight into the importance of features, identifying biases or errors and improving model performance. Recent post-hoc analysis of explainability has brought useful methods for analyzing the willingness of model predictions. With $heat\_map$ generated by post-hoc XAI analysis, users can see why the target classification is predicted based on the model and target classification. In a training process, the local model $\mathcal{LM}$ will classify x and predict category $\mathcal{LM}(x)$ and the label is y. Mathematically, the black box local model $\mathcal{LM}$ can be analyzed using the post-hoc method to get the $heat\_map$ of $(x, y, \mathcal{LM}(x), \mathcal{LM})$ by generating an XAI $\mathcal{M}$ model from $\mathcal{LM}$. The second and third column of Figure

Algorithm 1 XAI-accuracy Calculation

Input: $\mathcal{GM}, (x, y), (s, y), A$
Output: *XAI-accuracy*
Step 3.1: Initialize and train the local model
$\mathcal{LM} \leftarrow \mathcal{GM}$
Train and update $\mathcal{LM}$ by local data (x, y)
Step 3.2: Generate XAI analysis model
$m \leftarrow \mathcal{I}(\mathcal{LM})$
$\mathcal{M} \leftarrow \mathcal{N}(m)$
Step 3.3: Calculate $heat\_map$ **and XAI-accuracy**
while each s in $\mathbb{S}$ **do**
  $B \leftarrow \mathcal{M}(s, y, \mathcal{LM}(s))$
  $u \leftarrow sum(B \cap A)/sum(A)$
  $v \leftarrow sum(B \cap \overline{A})/sum(\overline{A})$
  $r \leftarrow u/v$
  if $r \geq 1$ **then**
    $n \leftarrow n + 1$
  end if
end while
XAI-accuracy $\leftarrow n/sum(\mathbb{S})$

2 show the example of $truth\_mask$ and $heat\_map$. $\mathcal{M}$ calculation is introduced in Algorithm 1. $\mathcal{GM}$ is received global model and $\mathcal{LM}$ is the local model trained locally for explainable analysis. In order to generate reasonable and accurate $heat\_map$, we introduce Integrated Gradients SmoothGrad (IG-SmoothGrad). This method consists of two parts, Integrated Gradients (IG) and SmoothGrad, and is a widely used explainable method often employed for identifying predictive features. SmoothGrad reduces misinterpretation by averaging multiple heatmaps. Through our preliminary experiments, we have found that IG-SmoothGrad is an efficient and reasonable explanatory approach. In this process, $\mathcal{I}$ refers to the function of IG used to analyze explainable pixels, followed by $\mathcal{N}$, which is a noise reduction method called NoiseTunnel (NT) based on the SmoothGrad technique [29].

s is the image in the samples $\mathbb{S}$. y is the image label. $\mathcal{LM}(s)$ is the model's inference result. B is the heat map generated by the XAI algorithm. A is the prepared truth mask, indicating pixel collection inside the reasonable area. $\overline{A}$ is the area outside the reasonable area. $sum(area)$ is the function to sum the non-zero pixels of the area. r is the IoU-Ratio and n is the number of samples with good inference rationale. Using the example of a dog $image$ in the second row, a picture containing a dog would be classified as the *dog* category. Humans can only understand this classification by looking at the area of the dog itself in the picture. We call the graph labeled with dog regions the $truth\_mask$, the second column of Figure 2.

To evaluate whether the feature is correctly extracted, the obvious way is to check whether $\mathcal{LM}(image\&truth\_mask)$ can be predicted as the label. However, feature areas can be correctly extracted sometimes despite incorrect predictions. So, it doesn't work well if we only evaluate the prediction of $\mathcal{LM}(image\&truth\_mask)$.

Maximizing the similarity between the $Truth_mask$ and $heat_map$ is desirable. Let's use A represents $truth\_mask$ and B represents $heat\_map$. Intersection over Union (IoU) is the most popular method for comparing two regions:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Occasionally, the object in the picture is very small (A is very small), but the entire picture is quite large. Pixels in B may appear anywhere in the picture because of the noise in generated $heat\_map$. Then, any noise from B ($heat\_map$) will have a significant impact on the results. So, a better way to evaluate the model's rationale is needed here.

Let's say the entire picture consists of A and $\overline{A}$. Unlike IoU, we use percentage for comparison, so we use Intersection over Union Ratio (IoU-Ratio) to represent, as shown in Equation (6). r stands for IoU-Ratio, which calculates the ratio of u and v, where, u means $in\_mask\_percentage$ and v means $out\_mask\_percentage$. In_mask area and out_mask area are shown in Figure 2's 4^{th} and 5^{th} column. If IoU-Ratio is greater than 1, $in\_mask\_percentage$ is bigger than $out\_mask\_percentage$. Then we treat the rationale indicated by $heat\_map$ makes better sense.

$$u = \frac{|A \cap B|}{A}, v = \frac{|B \cap \overline{A}|}{\overline{A}}, \quad r = \frac{u}{v}. \tag{6}$$

In this way, we can compare the ratio of u and v, that is, IoU-Ratio to see whether most of the feature points are included inside $truth\_mask$.

Motivated by theoretical analysis of clients' weight difference, our approach encourages more clients to participate in the training process while filtering out underperforming clients using a small test dataset with an ideal distribution. Idea distribution means selecting and distributing the test data to represent the distribution we want to achieve. Additionally, we introduce a "truth mask" with the test dataset and a method to calculate Intersection over Union (IoU) ratio, ensuring the validity of the extracted features and improving the explainability of the model's decisions. By adding explainability into the client selection mechanism, XFed aims to strike a balance between accuracy and explainability, enhancing the practicality and effectiveness of FL models.

4 Methodology

4.1 Overall process

Our proposed method, XFed, aims to solve the challenge of balancing high accuracy and high explainability in Federated Learning (FL) models. Current client selection methods prioritize accuracy and time to precision, but they may not provide optimal explainability of the model's decisions, which limits their practical use in real-world scenarios. To improve the accuracy of the target distribution dataset, we propose to encourage more clients to participate in the training process while selecting out underperforming clients. In order to select the best-performing clients, we use a relatively small number of test methods with an ideal distribution to evaluate the model trained by each round of clients. By incorporating the "truth mask" technique and the Intersection over Union (IoU) ratio, our method indicates that the extracted features are valid and improves explainability. The entire XFed process is illustrated in Figure 3 and described in the list below, where we highlight the client-side selection and explainability metrics.

- Step-1. Joining Step: small group of samples are provided by the server to each newly joined client.
- Step-2. Starting of a formal process: a global model is sent to the client by the server.
- Step-3. Client training process:
  - Step-3.1. a local model ($\mathcal{LM}$) is initiated by a global model and trained with local data;

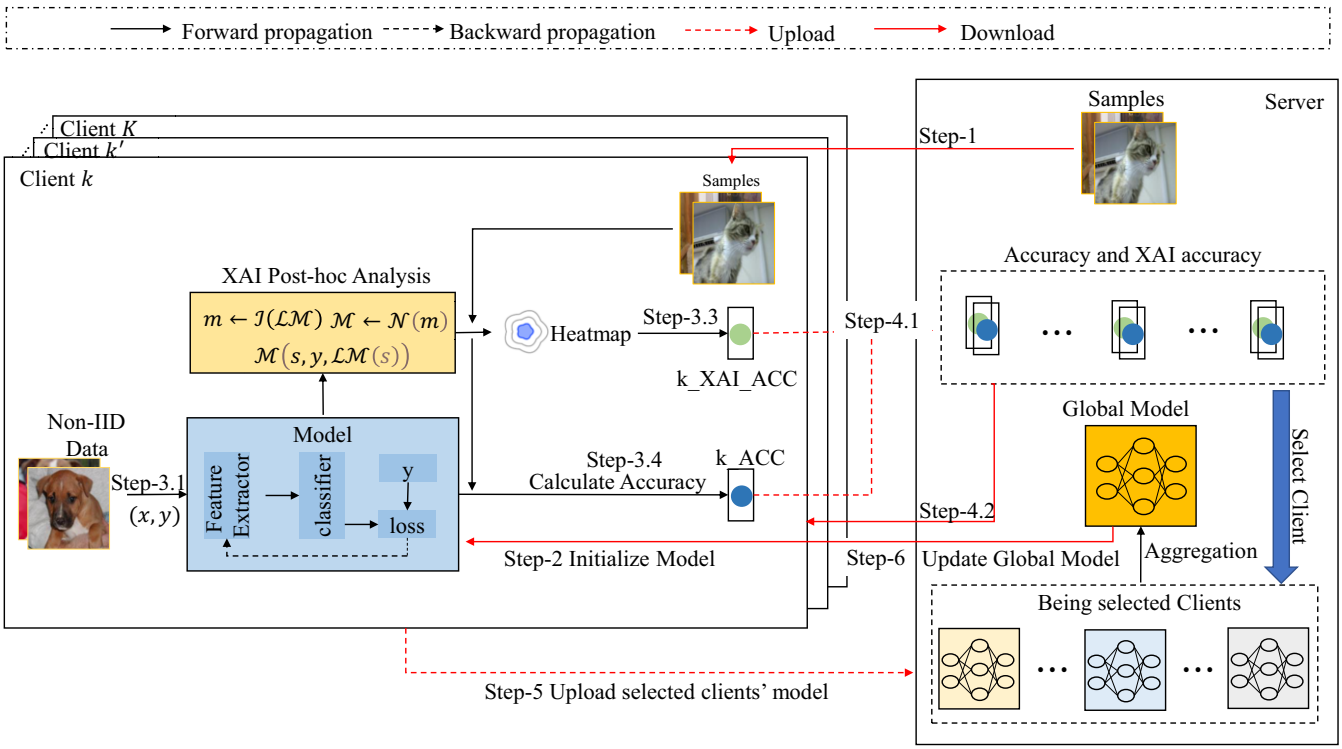

Figure 3. XFed process: this shows the full process and details will be introduced in Section 4.1.

- Step-3.2. a new XAI model ($\mathcal{M}$ in Algorithm 1) trained by XAI post-hoc analysis of the previous local model;
- Step-3.3. $heat\_map$ of test images are generated, and the XAI-accuracy are calculated;
- Step-3.4. the accuracy is calculated;

- Step-4. Client selection process:
  - Step-4.1. clients upload their own accuracy and XAI-accuracy to the server side;
  - Step-4.2. the selection list is generated by the server after the server compares the performance of each client, and eliminates the worst-performing clients.
- Step-5. The local models of the selected clients are uploaded to the server.
- Step-6. An updated global model is aggregated by the server and sent to all clients. We go back to the second step to iterate until the training is over.

4.2 *XAI evaluation and client selection*

The trained model can generate a heat map for the input sample, which we can examine to understand the inference result. We can determine the heat map's accuracy by comparing it with the truth mask and measuring the IoU-Ratio. Algorithm-1 details the method to calculate XAI accuracy.

After each client submits their accuracy and XAI accuracy to the server, the server can select clients by sorting their performance. Performance is calculated by fusing accuracy and XAI accuracy. Based on the ranking, we can select the top clients' models to upload in this round.

5 Experiments and discussions

During the joining stage, each client is provided with a small set of test data that includes a truth mask. In each communication round, the client sends its accuracy and XAI accuracy to the server. The server then sorts the clients based on their performance and selects the top 90%-95%. To reduce communication overhead, 5%-10% of each round's communications are saved. The rationale for selecting clients is primarily based on their XAI accuracy, followed by the global model's precision performance.

5.1 *Experimental Setup*

We evaluate our approach using CIFAR-10 [12] and MNIST [16] datasets, with artificially created non-IID data partitions. To simulate non-IID data partition, we use the heterogeneous partition method [2], and randomly divide the data among participant clients. We also use a small test set based on CIFAR and MNIST, which includes manually marked truth masks for 30 and 102 pictures, respectively. The truth mask indicates the position of the object to be classified and is marked through drawing software, as shown in Column 2 of Figure 2. Heat map is calculated based on IG-SmoothGard. The truth mask and generated heat map are used to calculate XAI accuracy through Algorithm 1. For evaluation, we use Acc (accuracy), XAI_Acc (XAI accuracy), and u ($in\_mask\_percentage$) of the final global model as metrics. Accuracy shows the prediction performance, while XAI accuracy indicates the rationale performance. u indicates the percentage of pixels inside the truth mask. We train the local model using cross-entropy loss function and adopt the $resnet18$ architecture with basic federated optimization methods in the model and scheduler. We use stochastic gradient descent (SGD) with global and local learning

rate adjustment for different iterations [32]. The default settings for the experiments are as follows: start learning rate of 0.01, local batch size of 128, and local epoch number of 20.

Oort [15] and **PyramidFL** [17] are considered as **state-of-the-art** client selection methods that work in two stages. In *Stage 1*, the IID constraint and utility-based method are applied to the data in the clients of the client selection orchestration center. In *Stage 2*, clients are selected based on the training loss. Adopting *Stage 1* does not impact the comparison of our methods as it is also applicable to XFed. On the other hand, non-client selection methods such as FedProx [21] and FedAVG [8] use all the training data, and as a result, the final accuracy can be higher than those of client selection methods. Therefore, we did not apply *Stage 1* of the IID constraint when using Oort and PyramidFL. In our experiments, we use **Oort'** to represent the simplified Oort and PyramidFL, and the number of selected clients is the same as XFed. **FedAVG** is a widely used baseline method in Federated Learning. **FedProx** is a **state-of-the-art** method for FL with non-IID data, which adds global training regulation to the local training process.

We use **Oort'**, **FedAVG**, and **FedProx** as baselines for comparison. The number of clients used by Oort is the same as XFed. FedAVG and FedProx use all the client's data, which can make the comparison more challenging for our methods.

5.2 *Overall Comparison*

The experiment shows that with 5%-10% communication saving and similar accuracy, our XFed method improves the XAI-accuracy 14%-35%.

XFedAVG means the method applying XFed on FedAVG, while XFedProx means the methods applying XFed on FedProx. In Figure 4 and Figure 5, we compare XFedAVG, XFedProx to Oort', FedAVG, and FedProx on CIFAR dataset selecting top 90% clients' models in each communication round. The sorting method of client rank is using both Client's accuracy and XAI-accuracy. The horizontal axis represents different communication rounds.

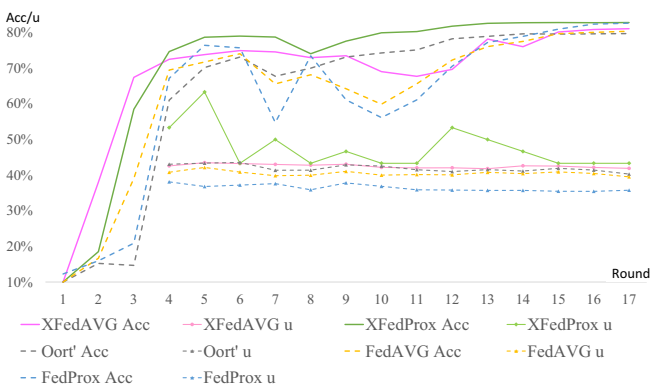

Figure 4. The comparison of different methods

The clustered columns in Figure 5 represent XAI accuracy. Our XFedAVG and XFedProx methods are significantly better than that in the early stage in terms of the accuracy and $in\_mask\_percentage$. Meanwhile, our method's XAI accuracy performs better than Oort'. The accuracy of XFedAVG is close to that of FedProx (the gap is less than 1%), while the interpretation rationality has been improved by 30%. The convergence of XFedAVG's accuracy rate is also more stable during the entire calculation process, and the rationality of the explanation is also relatively stable.

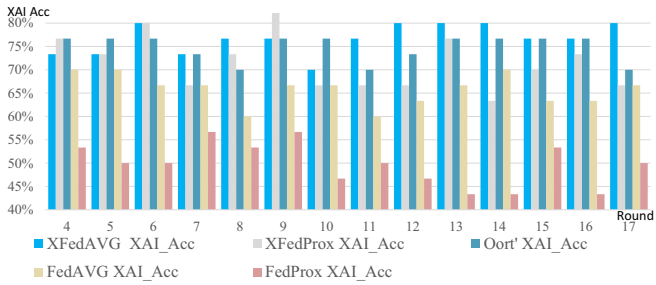

Figure 5. Rationale result comparison

Table 1 compares the performance of using XFed versus not using it. The best performance is indicated in boldface. The sorting method of client rank is using Client's accuracy. A CIFAR10 test data set is divided into 16 clients' data and 1 client is filtered out in each communication round. In the above group of precision comparison experiments, we can observe that both XFedAVG and XFedProx can achieve higher precision. XFedAVG performs best in a comparison of the second factor Rationale which is the XAI-accuracy. The interesting thing is that there is no outstanding XAI-accuracy performance of XFedProx. Upon analysis, FedProx's calculation process does not entirely rely on local data characteristics to calculate the loss, which affects the rationality of the explanation. This can also be observed in the direct comparison of FedProx and FedAVG in this second group of Table 1.

As shown in Figure 6, we compare the results of experiments conducted on FedAVG and XFedAVG on the MNIST dataset using 10 clients and select 9 clients' models in each round. According to XFedAVG's methodology, 10% of clients are eliminated during run-time. FedAVG utilizes all common clients. As we can see, the accuracy of the two methods is very close throughout the experiment, but the XAI accuracy results of XFedAVG are more reliable.

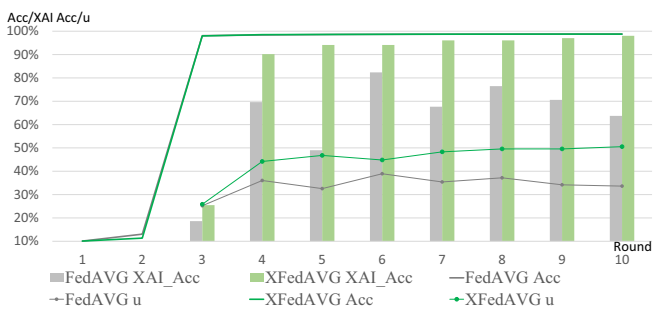

Figure 6. Evaluating the performance on the MNIST dataset

5.3 *Ablation Study*

XFed can use accuracy or XAI accuracy to select out clients to join the aggregation. So we designed the experiment to compare the difference between these two different client selection method, shown in Figure 7. Based on this experiment, we can see that client selection includes XAI accuracy shows a little improvement when comparing XAI Accuracy.

Table 1. XFed applied on FedAVG and FedProx

Round	Round 4	Round 6	Round 8	Round 10	Round 12	Round 14	Round 15	Round 16	Round 17
FedAVG Acc	69.51%	74.09%	68.19%	59.91%	72.31%	77.55%	79.73%	80.08%	80.39%
FedProx Acc	67.18%	75.74%	73.48%	56.15%	70.58%	78.96%	80.99%	82.41%	82.68%
XFedAVG Acc	72.53%	74.96%	72.98%	69.06%	69.69%	76.05%	80.19%	80.88%	81.11%
XFedProx Acc	**77.58%**	**78.04%**	**75.35%**	**70.39%**	**75.18%**	**81.69%**	**82.86%**	**83.12%**	**83.33%**
FedAVG XA_Acc	70.00%	66.67%	60.00%	66.67%	63.33%	70.00%	63.33%	63.33%	66.67%
FedProx XAI_Acc	53.33%	50.00%	53.33%	46.67%	46.67%	43.33%	53.33%	43.33%	50.00%
XFedAVG XAI_Acc	**73.33%**	**80.00%**	**76.67%**	**70.00%**	**80.00%**	**80.00%**	**76.67%**	**76.67%**	**80.00%**
XFedProx XAI_Acc	53.33%	43.33%	43.33%	43.33%	53.33%	46.67%	50.00%	46.67%	53.33%

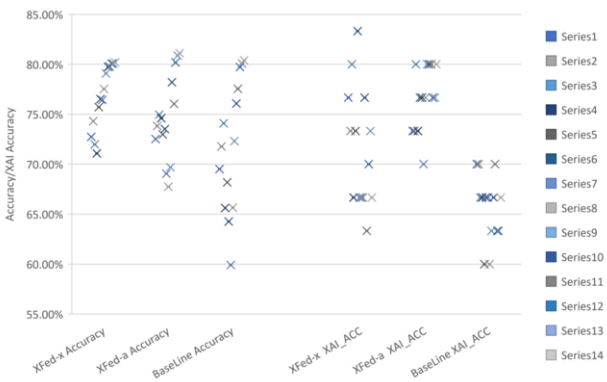

Figure 7. Ablation Study: FedAVG's Accuracy and XAI Accuracy are the base line. XFed-x shows the result when adopting XFed by using 20% XAI Accuracy and 80% Accuracy to select clients. XFed-a shows the result when adopting XFed by using Accuracy to select clients. The different series mean different rounds during the experiment.

Table 2. The model's input size is (3,32,32) where nt_samples = 5, n_steps = 5. IOU-Ratio time is measured by python's elapsed_time from IG-SmoothGrad generation to IOU-Ratio calculated.

	Million Parameters	Million Mult-Adds	IG-S Million Mult-Adds	IOU-Ratio Calc Time Per Sample
mobilenetv2	3.5	7.79	194.75	24.78ms
mobilenetv3	5.48	8.6	215	25.94ms
resnet18	11.69	37.69	942.25	15.38ms
shufflenetv2	2.28	4.08	102	25.10ms

5.4 Required Resource

XAI post-hoc computational complexity is a concern for resource-constrained clients. The required time for the XAI post-hoc model (IG-SmoothGrad) is shown in Table-2 for a CIFAR image on an NVidia P8 GPU. The IOU-Ratio time includes (IG +NT) generation but excludes image loading time. When using a test set of 30 samples, the required time is less than 0.78s. For a 720p picture with input size (3,1280,720), using the mobilenetv2 model requires 5.89G Multi-adds and takes 562 seconds to calculate the IOU-Ratio using the (IG-SmoothGrad) method described in this paper. This indicates that calculating XAI-ACC based on IOU-Ratio is promising for clients with limited resources. The memory requirements for the (IG-SmoothGrad) parameter generated by the mobilenetv2 model increase with the number of nt_samples, and at nt_sample = 5, it reaches 82.6MB, which is five times larger than the model size of mobilenetv2 for a CIFAR image.

5.5 Further Discussion

To calculate the global model, we used average aggregation after selecting the clients. The weight of each client was determined by their real rankings, resulting in potentially better results. To account for the non-IID data distribution among clients, we proposed sending a portion of the expected test data set to each client to obtain their ranking. Additionally, the server can provide an API for ranking clients who refuse to participate or are deemed untrustworthy.

Explainability accuracy on the shared dataset can be achieved without violating privacy. The explainability methods simply reveal how the model makes predictions based on certain features and do not disclose personal information. Furthermore, techniques such as differential privacy can be employed to further protect the privacy of the data. While adversarial modifications to the dataset are a concern, there are potential defenses that can be implemented, such as data integrity checks or encryption, to detect or prevent such attacks. Using a diverse set of datasets can also help reduce the impact of any malicious modifications.

6 Conclusion and Future Work

Interpretability is increasingly recognized as a crucial aspect of AI decision-making. However, in FL, the exploration of interpretable learning remains relatively novel. In this study, we investigated the use of model parameter difference and KL divergence to analyze the differences among FL clients. It revealed that FL can improve model performance by (a) incorporating more data from diverse sources in the training process; (b) minimizing the cross-entropy between the client model's prediction and the empirical distribution. This suggests that leveraging more clients and filtering out some clients with lower performance metrics can help to enhance the performance of the aggregated model over the test data. We proposed XFed as a method for selecting clients in FL. XFed sends a small group of test sets to clients and calculates XAI-accuracy using the intersection over union (IoU) ratio between the predicted $heat\_map$ and the ground truth $truth\_mask$ to measure model rationality. Our experiments show that XFed achieves accuracy close to state-of-the-art methods and improves rationale by 14%-35%, while also reducing the number of communications by 5%-10%.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. This work is supported by National Natural Science Foundation of China under grants 62206102, U1836204, U1936108, and Science and Technology Support Program of Hubei Province under grant 2022BAA046.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim, 'Sanity checks for saliency maps', *Advances in neural information processing systems*, **31**, (2018).

[2] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar, 'Leaf: A benchmark for federated settings', *arXiv preprint arXiv:1812.01097*, (2018).

[3] Peng Chen, Xin Du, Zhihui Lu, Jie Wu, and Patrick CK Hung, 'Evfl: An explainable vertical federated learning for data-oriented artificial intelligence systems', *Journal of Systems Architecture*, **126**, 102474, (2022).

[4] Yae Jee Cho, Jianyu Wang, and Gauri Joshi, 'Towards understanding biased client selection in federated learning', in *International Conference on Artificial Intelligence and Statistics*, pp. 10351–10375. PMLR, (2022).

[5] Arun Das and Paul Rad, 'Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey', *arXiv e-prints*, arXiv:2006.11371, (June 2020).

[6] Yongheng Deng, Feng Lyu, Ju Ren, Huaqing Wu, Yuezhi Zhou, Yaoxue Zhang, and Xuemin Shen, 'Auction: Automated and quality-aware client selection framework for efficient federated learning', *IEEE Transactions on Parallel and Distributed Systems*, **33**(8), 1996–2009, (2021).

[7] Krishna Gade, Sahin Cem Geyik, Krishnaram Kenthapadi, Varun Mithal, and Ankur Taly, 'Explainable ai in industry', in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3203–3204, (2019).

[8] McMahan H. B., Moore E., Ramage D., Hampson S., and Arcas B. A. y, 'Communication-efficient learning of deep networks from decentralized data', *Internet Conference on Artificial Intelligence and Statistics*, (2017).

[9] Chaoyang He, Murali Annavaram, and Salman Avestimehr, 'Group knowledge transfer: Federated learning of large cnns at the edge', *Advances in Neural Information Processing Systems*, **33**, 14068–14080, (2020).

[10] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek, 'Explainable ai methods-a brief overview', in *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pp. 13–38. Springer, (2020).

[11] Eoin M Kenny, Courtney Ford, Molly Quinn, and Mark T Keane, 'Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies', *Artificial Intelligence*, **294**, 103459, (2021).

[12] A. Krizhevsky and G. Hinton, 'Learning multiple layers of features from tiny images', *Master's thesis, Department of Computer Science, University of Toronto*, (2009).

[13] Solomon Kullback, *Information theory and statistics*, Courier Corporation, 1997.

[14] Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury, 'Fedscale: Benchmarking model and system performance of federated learning at scale', in *International Conference on Machine Learning*, pp. 11814–11827. PMLR, (2022).

[15] Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury, 'Oort: Efficient federated learning via guided participant selection', in *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, pp. 19–35, (2021).

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, **86**(11), 2278–2324, (1998).

[17] Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao, 'Pyramidfl: A fine-grained client selection framework for efficient federated learning', in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pp. 158–171, (2022).

[18] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin, 'A review of applications in federated learning', *Computers & Industrial Engineering*, **149**, 106854, (2020).

[19] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He, 'Federated learning on non-iid data silos: An experimental study', in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 965–978. IEEE, (2022).

[20] Qinbin Li, Bingsheng He, and Dawn Song, 'Model-contrastive federated learning', in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, (2021).

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith, 'Federated optimization in heterogeneous networks', *Proceedings of Machine Learning and Systems*, **2**, 429–450, (2020).

[22] Zhiyu Liang and Hongzhi Wang, 'Fedtsc: a secure federated learning system for interpretable time series classification', *Proceedings of the VLDB Endowment*, **15**(12), 3686–3689, (2022).

[23] Weiwei Lin, Yinhai Xu, Bo Liu, Dongdong Li, Tiansheng Huang, and Fang Shi, 'Contribution-based federated learning client selection', *International Journal of Intelligent Systems*, **37**(10), 7235–7260, (2022).

[24] Zelei Liu, Yuanyuan Chen, Yansong Zhao, Han Yu, Yang Liu, Renyi Bao, Jinpeng Jiang, Zaiqing Nie, Qian Xu, and Qiang Yang, 'Contribution-aware federated learning for smart healthcare', in *Proceedings of the 34th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-22)*, (2022).

[25] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng, 'No fear of heterogeneity: Classifier calibration for federated learning with non-iid data', *Advances in Neural Information Processing Systems*, **34**, 5972–5984, (2021).

[26] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand, 'A performance evaluation of federated learning algorithms', in *Proceedings of the second workshop on distributed infrastructures for deep learning*, pp. 1–8, (2018).

[27] Takayuki Nishio and Ryo Yonetani, 'Client selection for federated learning with heterogeneous resources in mobile edge', in *ICC 2019-2019 IEEE international conference on communications (ICC)*, pp. 1–7. IEEE, (2019).

[28] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, '" why should i trust you?" explaining the predictions of any classifier', in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, (2016).

[29] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg, 'Smoothgrad: removing noise by adding noise', *arXiv preprint arXiv:1706.03825*, (2017).

[30] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun 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).

[31] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu, 'Explainable ai: A brief survey on history, research areas, approaches and challenges', in *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*, pp. 563–574. Springer, (2019).

[32] Juan Zhao, Ruixuan Li, Haozhao Wang, and Zijun Xu, 'Hotfed: Hot start through self-supervised learning in federated learning', in *2021 IEEE 23rd Int Conf on High Performance Computing*, pp. 149–156. IEEE, (2021).

[33] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra, 'Federated learning with non-iid data', *arXiv preprint arXiv:1806.00582*, (2018).

[34] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin, 'Federated learning on non-iid data: A survey', *Neurocomputing*, **465**, 371–390, (2021).