The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

Addressing Class Imbalance in Federated Learning

Lixu Wang, Shichao Xu, Xiao Wang, Qi Zhu
Northwestern University, Evanston, IL, USA
{lixuwang2025, shichaoxu2023}@u.northwestern.edu, {wangxiao, qzhu}@northwestern.edu

Abstract

Federated learning (FL) is a promising approach for training decentralized data located on local client devices while improving efficiency and privacy. However, the distribution and quantity of the training data on the clients' side may lead to significant challenges such as class imbalance and non-IID (non-independent and identically distributed) data, which could greatly impact the performance of the common model. While much effort has been devoted to helping FL models converge when encountering non-IID data, the imbalance issue has not been sufficiently addressed. In particular, as FL training is executed by exchanging gradients in an encrypted form, the training data is not completely observable to either clients or server, and previous methods for class imbalance do not perform well for FL. Therefore, it is crucial to design new methods for detecting class imbalance in FL and mitigating its impact. In this work, we propose a monitoring scheme that can infer the composition of training data for each FL round, and design a new loss function — **Ratio Loss** to mitigate the impact of the imbalance. Our experiments demonstrate the importance of acknowledging class imbalance and taking measures as early as possible in FL training, and the effectiveness of our method in mitigating the impact. Our method is shown to significantly outperform previous methods, while maintaining client privacy.

Introduction

The emergence of federated learning (FL) enables multiple devices to collaboratively learn a common model without the need to collect data directly from local devices. It reduces the resource consumption on the cloud and also enhances the client privacy. FL has seen promising applications in multiple domains, including mobile phones (Hard et al. 2018; Ramaswamy et al. 2019), wearable devices (Nguyen et al. 2018), autonomous vehicles (Samarakoon et al. 2018), etc.

In standard FL, a random subset of clients will be selected in each iteration, who will upload their gradient updates to the central server. The server will then aggregate those updates and return the updated common model to all participants. During FL, one major challenge is that data owned by different clients comes from various sources and may contain their own preferences, and the resulting diversity may make the convergence of the global model challenging and slow. Moreover, the phenomenon of class imbalance happens frequently in practical scenarios, e.g., the number of patients diagnosed with different diseases varies greatly (Rao, Krishnan, and Niculescu 2006; Dong et al. 2019), and people have different preferences when typing with G-board (Ramaswamy et al. 2019) (a practical FL application proposed by Google). When a model encounters class imbalance, samples of *majority classes* account for a very large proportion of the overall training data, while those of *minority classes* account for much less. The direct impact of class imbalance is the reduction of classification accuracy on minority classes. In many practical cases, those minority classes play a much more important role beyond their proportion in data, e.g., wearable devices need to be more sensitive to abnormal heart rates than normal scenarios, and it is more important for G-board to predict SOS precisely than restaurant names.

In the literature, a number of approaches have been proposed to address class imbalance, e.g., applying various data sampling techniques (Jo and Japkowicz 2004), using generative augmentation to make up for the lack of minority samples (Lee, Park, and Kim 2016; Pouyanfar et al. 2018), and integrating cost-sensitive thoughts into model training (Sun et al. 2007). However, during FL, the communication between clients and the server is restricted to the gradients and the common model. For privacy concern, it is preferable that the server does not require clients to upload additional information about their local data (Geyer, Klein, and Nabi 2017; Hamm, Cao, and Belkin 2016). Thus, it is infeasible to gather the information of all local data and conduct an aggregated analysis globally. This makes the vast majority of imbalance solutions not applicable to FL. There are several approaches that may be applied locally, without uploading data distribution to the server (Huang et al. 2016; Wang, Ramanan, and Hebert 2017; Mikolov et al. 2013). However, due to the mismatch between local data distributions and the global one, these approaches are not effective and may even impose negative side-effect on the global model. The work in (Duan et al. 2019) directly addresses class imbalance in FL, but it requires clients to upload their local data distribution, which may expose latent backdoor to attacks and lead to privacy leakage. Moreover, it requires placing a number

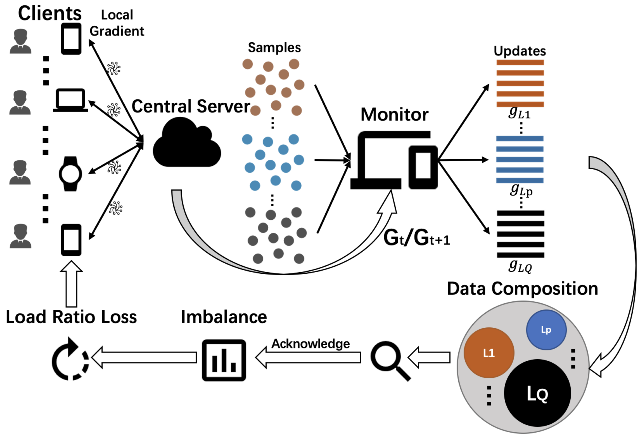

Figure 1: The monitor estimates the composition of training data round by round. When detecting a similar imbalanced composition continuously, the system will acknowledge the class imbalance and load the Ratio Loss.

of proxy servers, which increases the FL complexity and incurs more computation overhead.

In this work, we tackle the above challenges. We consider FL as a scheme that is always in training (McMahan et al. 2016). During FL, new data is constantly generated by the clients, and class imbalance could happen at any time. If such imbalance cannot be detected in time, it will induce the common model to the wrong direction in the early training phase, and thus poison the common model and deteriorate the performance. To detect the imbalance in FL timely and accurately, we propose to design a monitor that estimates the composition across classes during FL, and if a particular imbalanced composition appears continuously, the monitor will alert the administrator ($\mathcal{AD}$) to apply measures that can mitigate the negative impact. Moreover, we develop a new loss function **Ratio Loss**, and compare our approach to existing loss-based imbalance state-of-the-art solutions: **CrossEntropy Loss**, **Focal Loss** (Lin et al. 2017) and **GHMC Loss** (Li, Liu, and Wang 2019). Note that these loss functions are for general class imbalance problems, and their basis is just the output results of forward feeding. We choose them for comparison as they can also address the imbalance issue in FL without posing threats to privacy.

The basic workflow of our method is shown in Fig. 1. At round $t+1$, the monitor downloads the global model G_t of round t and feeds samples of the auxiliary data to it. For each class, the monitor obtains corresponding gradient updates g_L. And by applying our method to compare these updates with G_{t+1}, our monitor can acquire the composition of training data at round $t+1$. If a similar imbalanced composition is detected continuously, the system will acknowledge that the global model has learned imbalanced data, and then try to mitigate its impact by applying the Ratio Loss in FL.

Our contributions. More specifically, we made the following contributions in this work:

- Our approach monitors the composition of training data at each training round in a passive way. The monitor can be deployed at either the central server or a client device, and it will not incur significant computation overhead or impose threats to client privacy.
- Our works show the importance of acknowledging class imbalances as early as possible during FL and taking timely measures.
- Our approach defines two types of class imbalance in FL: **local imbalance** and **global imbalance**, and addresses class imbalance based on a new loss function (Ratio Loss). Our method is shown to significantly outperform previous state-of-the-art methods while maintaining privacy for the clients.

Related Work

Class Imbalance. In supervised learning, models require labeled training data for updating their parameters. The imbalance of the training data (i.e., the variation of the number of samples for different classes/labels) occurs in many scenarios, e.g., image recognition for disease diagnosis (Xia et al. 2020), object detection (Lin et al. 2017; Wang and Zhang 2020). Such class imbalance will worsen the performance of the learning models, especially decreasing the classification accuracy for minority classes (He and Garcia 2009). Several works have designed new metrics (Wang et al. 2016; Davis and Goadrich 2006) to quantify the model performance with class imbalance, rather than just considering the overall accuracy. Prior approaches to address class imbalance can be classified into three categories: data-level, algorithm-level, and hybrid methods. Data-level approaches leverage data resampling (Van Hulse, Khoshgoftaar, and Napolitano 2007; Mani and Zhang 2003) and data augmentation (Lee, Park, and Kim 2016; Pouyanfar et al. 2018). Algorithm-level approaches modify the training algorithm or the network structure, e.g., meta learning (Malave and Nimkar 2020; Wang et al. 2020), model tuning (Pouyanfar et al. 2018), cost-sensitive learning (Cui et al. 2019; Wang, Ramanan, and Hebert 2017), and changing the loss function (Lin et al. 2017; Li, Liu, and Wang 2019; Luo et al. 2020). Then, from the perspectives of both data and algorithm levels, hybrid methods emerge as a form of ensemble learning (Liu, Wu, and Zhou 2008; Chawla et al. 2003).

As stated in the introduction, data-level methods cannot be applied in FL due to their violation of the privacy requirements. The cost-sensitive approaches need to analyze the distribution of training data, e.g., re-weighting the loss via inverse class frequency, and are not effective for FL due to the mismatch between local data distribution and the global one. Other cost-sensitive methods need specific information of minority classes, e.g., **MFE Loss** (Wang et al. 2016) regards minority classes as positive classes, and calculates false positive and false negative to generate a new loss form. Such prior knowledge is also difficult to acquire in FL. To address class imbalance in FL, we believe that it is important to measure the imbalance according to the current common model rather than depending on the training data. We thus regard CrossEntropy Loss, Focal Loss (Lin et al. 2017) and GHMC Loss (Li, Liu, and Wang 2019) as possibly methods to solve the class imbalance problem in FL, and compare our approach with them.

Federated Learning. Due to the heavy computation burden for training deep learning models, researchers have been exploring using multiple devices to learn a common model. There are many studies on organizing multiple devices for distributed learning, with both centralized (Kim et al. 2016) and decentralized (Sergeev and Del Balso 2018) approaches. Recently, more and more local client devices (e.g, mobile phones) can participate in model learning. Under such circumstances, the training data on local devices is more personal and privacy-sensitive. In order to avoid privacy leakage, federated learning (McMahan et al. 2016; Li et al. 2020) has emerged as a promising solution, which enables a global model to be learned while keeping all training data on local devices. The privacy protection in FL training is guaranteed by secure aggregation protocols (Bonawitz et al. 2017) and differential privacy techniques (Geyer, Klein, and Nabi 2017; Niu et al. 2020). In general, with these technologies, neither local participants nor the central server of FL can observe the individual gradient in the plain form during training. Despite of various types of inference attacks (Melis et al. 2018; Zhu, Liu, and Han 2019; Wang et al. 2019; Sun et al. 2020), inferring information of particular clients is still extremely difficult. Therefore, how to extract useful information from the aggregated global gradient is interesting – and in this work, we focus on extracting such information for addressing class imbalance.

Methodology

Definition and Background

Our problem is formulated on a multi-layer feed-forward neural network. Here we consider a classifier with output size equal to the number of classes Q. It is defined over a feature space $\mathcal{X}$ and a label space $\mathcal{Y} = \{1, \cdots, Q\}$. Without losing generality for our problem, we combine all the middle layers as a hidden layer HL. If we feed the j-th sample of the class p, denoted as $X_j^{(p)}$, to the classifier, its corresponding output of HL is denoted as $Y_j^{(p)} = [y_{j,(1)}^{(p)}, ..., y_{j,(s)}^{(p)}]$. The output of the last layer is $Z_j^{(p)} = [z_{j,(1)}^{(p)}, ..., z_{j,(Q)}^{(p)}]$, followed by a softmax operation to obtain the probability vector $\mathcal{S} = [f_{j,(1)}^{(p)}, ..., f_{j,(Q)}^{(p)}]$. The HL contains s neurons. A function $f : \{\mathcal{X} \Rightarrow \mathcal{S}\}$ maps $\mathcal{X}$ to the output probability simplex $\mathcal{S}$, with f parameterizing over the hypothesis class $\mathbb{W}$, i.e., the overall parameters of the classifier. Further, the connection weight from the HL to the output layer is denoted as $\mathcal{W} = [\mathcal{W}_{(1)}, \mathcal{W}_{(2)}, ..., \mathcal{W}_{(Q)}]$, and $\mathcal{W} \in \mathbb{W}$. At each training iteration, we apply back-propagation to compute the gradient of loss $L(\mathbb{W})$ subject to $\mathbb{W}$. We use $\mathbb{W}(t)$ to denote the weights in the t-th training iteration, and λ to denote the learning rate. We then have $\mathbb{W}(t+1) = \mathbb{W}(t) - \lambda \nabla L(\mathbb{W}(t))$.

Monitoring Scheme

We define two types of class imbalance in FL: **local imbalance** and **global imbalance**. On every local client device j, the number of samples for each class p, denoted by N_p^j, may vary. The local imbalance measures the extent of such variation on each client device. Specifically, we define the local imbalance γ_j for device j as the ratio between the sample number of the majority class on j and the sample number of the minority class on j, i.e., $\gamma_j = \max_p\{N_p^j\}/\min_p\{N_p^j\}$, similar to the prevailing imbalance ratio measurement as in (Buda, Maki, and Mazurowski 2018). It is possible that $\min_p\{N_p^j\} = 0$. We regard such situation as extreme imbalance, and consider them in our experiments.

From the global perspective, we can measure the extent of global class imbalance Γ by defining it as the ratio between the total sample number of the majority class across all devices and that of the minority class, i.e., $\Gamma = \max_p\{\sum_j N_p^j\}/\min_p\{\sum_j N_p^j\}$.

In general, the local imbalance on each device may be different from the global imbalance, and in practice such difference could be quite significant. We may even encounter the cases where a particular class is the majority class on certain local devices but the minority class globally. To better quantify such mismatch between local and global imbalance, we use a vector $v_j = [N_1^j, ..., N_Q^j]$ to denote the composition of local data on device j, where Q is the overall number of classes; and we use another vector $V = [\sum_j N_1^j, ..., \sum_j N_Q^j]$ to denote the composition of global data. We then use cosine similarity (CS) score to compare their similarity, i.e., $CS_j = (v_j \cdot V)/(\|v_j\|\|V\|)$.

In regular training scenarios, there is no distinction between local and global imbalance levels since the training data is centralized and accessible, and mitigating the negative impact of imbalance is much easier than in FL. Note that in FL, the local training can be regarded as regular centralized learning. Intuitively, we could utilize existing methods to address the local imbalance issue at every round locally. However, local models exist temporarily on the selected devices. They will not be applied for users' tasks and will be replaced with the latest global model at the next round. As the result, addressing local imbalance may not have significant impact in FL. Moreover, because of the mismatch between local and global imbalance, simply adopting existing approaches at local devices is typically not effective and may even impose negative impact on the global model. Thus, we focus on addressing global imbalance in our work. To detect and mitigate the performance degradation caused by global imbalance, we develop a monitoring scheme to estimate the composition of training data during FL, as explained below.

Proportional Relation. We will first analyze the relation between the gradient magnitude and the sample quantity.

Theorem 1: *For any real-valued neural network f whose last layer is a linear layer with a softmax operation (without any parameter sharing), and for any input sample $X_i^{(p)}$ and $X_j^{(p)}$ of the same class p, if the inputs of the last layer Y_i and Y_j are identical, the gradients of link weights $\mathcal{W}$ between the last layer and its former layer induced by $X_i^{(p)}$ and $X_j^{(p)}$ are identical.*

The proof of Theorem 1 is presented in the Supplementary Materials ($\mathcal{SM}$). In the mini-batch training, gradients of samples within the batch are accumulated to update the model parameters, i.e.,

$$\Delta_{\text{batch}}\mathcal{W} = -\frac{\lambda}{n^{\text{batch}}}\sum_{p=1}^{Q}\sum_{j=1}^{n^{(p)}}\nabla_{\mathcal{W}_j^{(p)}}L(\mathbb{W}) \tag{1}$$

From our empirical study (please see $\mathcal{SM}$), we observe that the data samples of a same class p induce similar $Y^{(p)}$s, and thus their corresponding gradients are very similar. If the average value of the gradients is $\overline{\nabla_{\mathcal{W}^{(p)}}L(\mathbb{W})}$, Eq. (1) can be written as:

$$\Delta_{\text{batch}}\mathcal{W} = -\frac{\lambda}{n^{\text{batch}}}\sum_{p=1}^{Q}\left(\overline{\nabla_{\mathcal{W}^{(p)}}L(\mathbb{W})}\cdot n^{(p)}\right) \tag{2}$$

where $n^{(p)}$ is the number of samples for class p in this batch, and n^{batch} is the batch size. For one round of local training in FL, the total iteration number of local gradient update is $\left[\left(\sum_{p=1}^{Q}N_p/n^{\text{batch}}\right)\cdot \text{N}_{ep}\right]$, where N_{ep} denotes the number of local epochs. To illustrate the proportional relation between gradient magnitude and sample quantity, we assume that the parameter change is relatively small and can be neglected within a training epoch. In this case, $\overline{\nabla_{\mathcal{W}^{(p)}}L(\mathbb{W})}$ of different batches within an epoch remains the same, and we can aggregate them and obtain the weight update of one epoch as:

$$\Delta_{\text{epoch}}\mathcal{W} = -\frac{\lambda}{n^{\text{batch}}}\sum_{p=1}^{Q}\left(\overline{\nabla_{\mathcal{W}^{(p)}}L(\mathbb{W})}\cdot N_p\right) \tag{3}$$

where N_p is the overall sample number of class p.

In the setting of standard FL, the selected local gradients are aggregated by the FedAvg (McMahan et al. 2016):

$$\nabla L(\mathbb{W})_{t+1}^{Avg} = \frac{1}{K}\sum_{j=1}^{K}\nabla L(\mathbb{W})_{t+1}^{j} \tag{4}$$

where K represents the number of selected clients. In this work, we consider the case where feature spaces of data sets owned by different clients are similar (Yang et al. 2019). In the case where they have significant differences, transfer learning techniques such as domain adaptation (Ganin and Lempitsky 2015; Dong et al. 2020; Xu et al. 2021) may be needed to reduce the distribution discrepancy of different clients. This can be viewed as a problem of Federated Transfer Learning (FTL) with feature and model heterogeneity, and we plan to investigate the imbalance problem of FTL in our future work.

Based on above analysis, for any local training starting from the same current global model, data samples of the same class p output very similar Y (see $\mathcal{SM}$) and similar $\overline{\nabla_{\mathcal{W}^{(p)}}L(\mathbb{W})}$. In this case, the gradient induced by class p in one global epoch is:

$$\begin{aligned}\Delta_{\text{global}}\mathcal{W}^{(p)} &= -\frac{\lambda}{n^{\text{batch}}\cdot K}\sum_{j=1}^{K}\left(\overline{\nabla_{\mathcal{W}^{(p)}}^{j}L(\mathbb{W})}\cdot N_p^j\right)\\ &= -\frac{\lambda}{n^{\text{batch}}\cdot K}\cdot\overline{\nabla_{\mathcal{W}^{(p)}}L(\mathbb{W})}\left(\sum_{j=1}^{K}N_p^j\right)\end{aligned} \tag{5}$$

Based on this relation, we develop our monitoring scheme as follows. In round $t+1$, the monitor will feed samples of every class in the auxiliary data singly to the same global model of round t, i.e., G_t. It then obtains corresponding weight updates $\{g_{L_1}, ..., g_{L_p}, ..., g_{L_Q}\}$, where each g_{L_p} corresponds to the class p. In practice, we observe that not all weights get updated significantly – some of them increase little and thus easily get offset by the negative updates of other classes. Accordingly, we design a filter to select the weights whose updating magnitudes are relatively large. Specifically, for class p, we get the weight updates of the p-th output node (denoted as $\Delta\mathcal{W}_{(p)}^{(1\sim Q)}$) from $\{g_{L_1}, ..., g_{L_p}, ..., g_{L_Q}\}$, and compute the ratio $Ra_{p,i}$ for each weight component of $\Delta\mathcal{W}_{(p)}$ as follows:

$$Ra_{p,i} = \frac{(Q-1)\Delta\mathcal{W}_{(p,i)}^{(p)}}{\sum_{j=1}^{Q}(\Delta\mathcal{W}_{(p,i)}^{(j)}) - \Delta\mathcal{W}_{(p,i)}^{(p)}} \tag{6}$$

where $i=1, ..., s$. We set a threshold T_{Ra} ($T_{Ra}=1.25$; refer to $\mathcal{SM}$ for the experiments of setting T_{Ra}), and we select components of $\Delta\mathcal{W}_{(p)}$ whose ratios $Ra_{p,i}$ are larger than T_{Ra}. Based on the proportional relation, we formulate the accumulation of weight changes under FedAvg:

$$\begin{aligned}\Delta\mathcal{W}_{(p,i)}^{(p)}\cdot\widehat{N}_{p,i} + \left(\sum_{j=1}^{K}\sum_{p=1}^{Q}N_p^j - \widehat{N}_{p,i}\right)\frac{\Delta\mathcal{W}_{(p,i)}^{(p)}}{Ra_{p,i}}\\ = n_a^p\cdot K\left(\mathcal{W}_{(p,i)}^{G_{t+1}} - \mathcal{W}_{(p,i)}^{G_t}\right)\end{aligned} \tag{7}$$

where n_a^p is the sample number of class p in the auxiliary data, $\widehat{N}_p$ is the predicted sample quantity of class p, $\mathcal{W}_{(p,i)}^{G_t}$ and $\mathcal{W}_{(p,i)}^{G_{t+1}}$ are link weights $\mathcal{W}_{(p,i)}$ of G_t and G_{t+1}, respectively. $\sum_{p=1}^{Q}N_p^j$ is the overall number of all samples owned by client j, and we need clients to upload $\sum_{p=1}^{Q}N_p^j$ to the server. This is the only information needed from clients in our monitoring scheme. We have discussed why sharing the total sample quantity is a reasonable assumption for the monitoring scheme in our $\mathcal{SM}$, and you can get more details about the security concern from it.

Now, except for $\widehat{N}_{p,i}$, all values in Eq. (7) can be acquired by the monitor. We can then use Eq. (7) to compute $\widehat{N}_{p,i}$ for each component of the filtered $\Delta\mathcal{W}_{(p)}$. And we can obtain the final result as the average value of all calculated $\widehat{N}_{p,i}$ (denoted as $\widehat{N}_p$). After the computation for all classes, we can obtain the proportion vector of the current training round $v_{pt} = [\widehat{N}_1, ..., \widehat{N}_p, ..., \widehat{N}_Q]$, an estimation of the data composition of the current round.

Ratio Loss based Mitigation

Once our monitor detects a similar imbalanced composition continuously by checking v_{pt}, it will acknowledge that the global model has learned imbalanced data and apply a mitigation strategy that is based on our **Ratio Loss** function.

As aforementioned, applying existing approaches locally will not be effective in mitigating the impact of global imbalance. Our method instead measures the global imbalance based on the current global model. According to previous analysis, weight updates are proportional to the quantity of samples for different classes, and the current network is built by accumulating such updates round by round. Due to the difference of feature space among classes, it is likely more reasonable to use the contribution to gradient updates rather than just the number of data samples for demonstrating the imbalance problem. In other words, after feeding some data to train the network, if weights of different nodes are updated similarly in terms of magnitude, we can regard the training as balanced, and vice versa. Because the layers before output nodes are shared by all classes, we restrict our interest on link weights between the *HL* and output nodes. Specifically, we consider the imbalance problem in FL as *the weight changes of different output nodes present noticeable magnitude gap when feeding corresponding classes*.

Theorem 2: *For any real-valued neural network f whose last layer is a linear layer with a softmax operation (without any parameter sharing), and the activation function between the last layer and its former layer is non-negative (e.g., Sigmoid and ReLU), if f has learned imbalanced data set, for any majority class $\mathcal{A}$, any minority class $\mathcal{B}$, and another class $\mathcal{C}$ ($\mathcal{C} \neq \mathcal{A} and \mathcal{C} \neq \mathcal{B}$, but $\mathcal{C}$ can be any class other than chosen $\mathcal{A}$ & $\mathcal{B}$) fed to f, we have:*

$$|\nabla_{\mathcal{W}^{(\mathcal{C})}_{(\mathcal{A})}} L(\mathbb{W})| > |\nabla_{\mathcal{W}^{(\mathcal{C})}_{(\mathcal{B})}} L(\mathbb{W})| \tag{8}$$

Assumption: 1) The input similarity between class $\mathcal{C}$ and $\mathcal{A}$ is the same as between class $\mathcal{C}$ and $\mathcal{B}$. 2) The reason why there is classification accuracy degradation on the minority class $\mathcal{B}$ is that its probability result $f^{(\mathcal{B})}_{(\mathcal{B})}$ is not distinguishable with the output of other $f^{(\mathcal{B})}_{(i)}$ ($i = 1, ..., Q$ and $i \neq \mathcal{B}$). Thus minority classes can be regarded as hard samples generally, while majority classes are easy samples, i.e., $\left(f^{(\mathcal{A})}_{(\mathcal{A})} / f^{(\mathcal{A})}_{(\mathcal{B})}\right) \gg \left(f^{(\mathcal{B})}_{(\mathcal{B})} / f^{(\mathcal{B})}_{(\mathcal{A})}\right) > 1$.

The detailed proof is shown in the $\mathcal{SM}$. Based on Theorem 2, we propose to mitigate global imbalance by designing our **Ratio Loss** function, denoted as L_{RL}. Specifically, we first consider the widely-used **CrossEntropy Loss** function (denoted as L_{CE}) for the multi-class classifier:

$$L_{CE} = -p \cdot [log(\mathcal{S})] \tag{9}$$

where p is the ground-truth label and always the one-hot form in multi-class classifiers, while $\mathcal{S}$ denotes the vector of probability results. In order to address imbalance, a common method is to introduce a weight vector $\Pi = [\pi_1, ..., \pi_Q]$ and there is $\Pi \cdot L_{CE}$. Typically, π is determined by the proportions or frequencies of different classes for the overall training data. Intuitively, a larger proportion corresponds to a lower π, and vice versa.

As stated above, we use the noticeable differences of weight changes to evaluate the global imbalance. Taking L_{CE} as the basic term, we define the Ratio Loss L_{RL} as:

$$L_{RL} = (\alpha + \beta\mathbb{R}) \cdot p \cdot L_{CE} \tag{10}$$

where α and β are two hyper-parameters. In our experiments, when $\alpha = 1.0$ and $\beta = 0.1$, the mitigation results are the best (the experiments for setting α and β can be found in the $\mathcal{SM}$). After computing all $Ra_{p,i}$ for class p as Eq. (6), we can get their average value and its corresponding absolute value, denoted as Ra_p, and compose $\mathbb{R} = [Ra_1, ..., Ra_p, ..., Ra_Q]$. Finally, in the local training, when a sample $X^{(p)}$ of class p is fed to the neural network, its corresponding loss is:

$$L_{RL}(X^{(p)}) = -(\alpha + \beta Ra_p) \cdot log(f^{(p)}_{(p)}) \tag{11}$$

We mitigate the impact of class imbalance by modifying the coefficient π before L_{CE}. When the input is a minority class, according to Theorem 2, its corresponding Ra is relatively large, and then its contribution to the overall loss will increase, and vice versa. Compared with **GHMC Loss**, **Ratio Loss** pays attention to the gradient on the output node corresponding to the ground-truth label of data samples, and also considers the impact over gradients on the same node from samples of other classes. In addition, the utilization of L_{RL} does not require clients to upload their overall sample quantities – $\sum_{p=1}^{Q} N^j_p$, which maintains the client privacy.

Experimental Results

Experiment Setup

We implement the algorithms mainly in PyTorch. Our experiments follow the standard structure of FL (Konečný et al. 2016; McMahan et al. 2016). We choose four different datasets: MNIST, CIFAR10, Fer2013 (Goodfellow et al. 2013), and FEMNIST of LEAF benchmark (Caldas et al. 2018). Fer2013 relates to face recognition and has imbalance issue, and FEMNIST is a popular FL data set with great feature heterogeneity. For each data set, we utilize the following convolution neural networks: LeNet5 for MNIST, a 7-layer CNN for CIFAR10, ResNet50 (He et al. 2016) for Fer2013, and ResNet18 for FEMNIST. The local training batch size is 32, the learning rate $\lambda = 0.001$, and the optimizer is SGD. The auxiliary data is a set of samples of different classes that is fed into the current global model by the monitor. It can be acquired from the public data or be synthesized by the FL administrator who has legal access to prior knowledge about what the training data may look like. Such auxiliary data can be utilized for a long time, unless the training data of the overall FL system changes significantly. Moreover, the required size of the auxiliary data is small. In our experiments, we use only 32 samples for each class, while the sample quantity of a client is more than $10,000$. Due to the small size of the auxiliary data, the deployment of the monitor does not incur significant computation overhead, and we indeed did not observe noticeable additional processing time during our experiments. Please refer to the $\mathcal{SM}$ for more details about the auxiliary data, and the setting for hardware.

Effectiveness of Monitoring Scheme

To evaluate the effectiveness of our monitoring scheme, we design the experiments as the central server randomly selects 20 clients from 100 participants during each round of the FL training. Each client trains its model for 10 local epochs for MNIST and FEMNIST, and 5 for CIFAR10 and Fer2013 (in this case, one global round for each data set costs nearly the same amount of time). Training 30 global rounds can make the model on MNIST converge, 50 for CIFAR10, Fer2013, and FEMNIST. For each client, first the number of classes they have locally is randomly determined as an integer between 1 and Q. Then, the specific classes are randomly chosen for each client, with equal sample quantity for each class. For FEMNIST, as each writer has a relatively small number of data samples (several dozens), we group 20 writers into a new client and thus turn approximately 2,000 writers into 100 clients (we believe that the heterogeneity holds true under this allocation strategy). For all data sets, we allocate them to clients without replacement, and the detailed data splitting is visualized in the $\mathcal{SM}$. During FL, different client selections at each round lead to varying data compositions, and thus different global imbalance patterns and various non-IID situations. As introduced, our monitor computes a data composition vector v_{pt} for each training round. We can compare it against the ground truth, defined as V. Fig. 2 shows the comparison between our estimated v_{pt} and V, measured by a cosine similarity score. The closer it is to 1, the more accurate our estimation is. From the figure, we can observe that our estimation of the data composition is very close to the ground truth. Among four data sets, the average similarity score is above 0.98 and higher than 0.99 for most of the time. Such results demonstrate the effectiveness of our monitoring scheme. We also carry out experiments with different numbers of clients, and we find that the similarity score gets even closer to 1 with the increase of client number. We also find that the local batch size and epochs have little impact on the performance of the monitoring scheme. The detailed results are in the $\mathcal{SM}$.

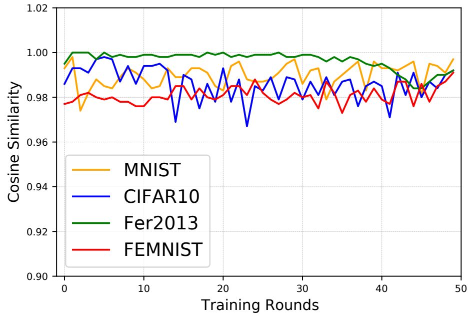

Figure 2: Similarity between our estimation of the global data composition and the ground truth.

Overall Comparison with Previous Methods

We then conduct experiments to evaluate the effectiveness of our Ratio Loss (L_{RL}), and compare it with CrossEntropy Loss (L_{CE}), Focal Loss (L_{FL}) and GHMC Loss (L_{GHMC}). We use the similar experiment setting as previous, except that we now explicitly explore different levels of global imbalance Γ, i.e., setting the ratio as $10:1$, $20:1$, $50:1$, and $100:1$, respectively, and we also include experiments when the data is balanced (B.). The evaluation metrics are the AUC score and the classification accuracy of minority classes (Ac.M). The results for majority classes can be found in the $\mathcal{SM}$. To keep Γ unchanged during training, we fix the selected clients as those chosen in the first round. (We also conduct experiments when the Γ is dynamically changing during FL training, and our L_{RL} also performs the best. Please refer to our $\mathcal{SM}$ for results.)

Data		FEMNIST				
Γ		B.	10:1	20:1	50:1	100:1
Ac.M %	L_{CE}	90.46	74.32	62.64	33.25	18.48
	L_{FL}	91.25	75.96	64.14	38.16	25.11
	L_{GHMC}	92.64	79.75	69.29	42.55	29.16
	L_{RL}(ours)	**93.46**	**88.48**	**72.29**	**51.66**	**41.45**
AUC	L_{CE}	.9652	.9441	.9187	.8979	.8667
	L_{FL}	.9650	.9540	.9291	.9011	.8774
	L_{GHMC}	.9691	.9542	.9393	.9023	.8842
	L_{RL}(ours)	**.9699**	**.9607**	**.9477**	**.9138**	**.9001**

Table 1: Comparison between our method (L_{RL}) and previous methods based on CrossEntropy Loss (L_{CE}), Focal Loss (L_{FL}) and GHMC Loss (L_{GHMC}) in federate learning, over FEMNIST and different levels of global imbalance.

After demonstrating the importance of early acknowledgment for global imbalance (please refer to our $\mathcal{SM}$), we quantitatively compare our method with previous ones in Ac.M and AUC score. The results are shown in Table 1 for FEMNIST and Table 2 for MNIST, CIFAR10 and Fer 2013 (all data points are the average of 5 runs). We can see that our method can effectively mitigate the impact of class imbalance and outperform the previous methods in almost all cases. Our improvement is particularly significant for MNIST and FEMNIST.

Moreover, we also compare our method with previous ones for the regular training of neural networks *without* FL. Table 3 demonstrates that in these cases, our method still outperforms the other three in most scenarios. This shows the broader potential of our Ratio Loss function.

Mismatch between Local and Global Imbalance

We also conduct a set of experiments to explore the impact of the mismatch between local and global imbalance. We adjust the mismatch level by setting different number of classes each client may have, i.e., from 2 to 5 out of a total number of 10 classes globally. Intuitively, the smaller the number, the less representative each client is with respect to the global training set, and hence the larger the mismatch.

To start with, we implemented **MFE Loss** (Wang et al. 2016) in FL as the representative method aiming to address the local imbalance by analyzing the local data distribution. As MFE Loss (L_{MFE}) is based on **Mean-Square-Error Loss** (L_{MSE}), we regard L_{MSE} as the baseline. As stated in the introduction, applying L_{MFE} requires knowing what minority classes specifically are. In FL, such information is difficult to acquire globally, and the standard method based on L_{MFE} can only analyze the local data of each client.

Table 4 shows the comparison between L_{MSE} and L_{MFE}. The global imbalance ratio is $\Gamma = 10:1$ (Ac.M degrades to zero when the ratio is larger than $10:1$). From the results, we can clearly see that using L_{MFE} locally has similar performance as the baseline. Moreover, we can observe

Data		**MNIST**					**CIFAR10**					**Fer2013**				
Γ		B.	10:1	20:1	50:1	100:1	B.	10:1	20:1	50:1	100:1	B.	10:1	20:1	50:1	100:1
Ac.M %	L_{CE}	**98.22**	90.19	80.04	63.66	46.84	57.57	23.43	15.17	04.93	00.97	**97.93**	23.59	12.65	05.43	02.41
	L_{FL}	96.42	84.84	75.65	63.43	41.76	50.10	26.40	17.77	06.47	01.57	85.28	21.87	12.86	05.56	**03.01**
	L_{GHMC}	93.05	81.24	64.98	61.38	20.23	50.10	27.73	19.13	**06.77**	02.53	46.54	19.76	08.72	02.44	01.47
	L_{RL}(ours)	98.04	**92.05**	**81.70**	**74.51**	**56.50**	**63.23**	**29.77**	**19.17**	**06.77**	**03.03**	97.87	**25.55**	**13.34**	**06.46**	02.95
AUC	L_{CE}	.9907	.9780	.9526	.9338	.9056	.7425	.6944	.6777	.6628	.6578	**.9893**	.7932	**.7574**	.7320	**.7275**
	L_{FL}	.9830	.9642	.9485	.9282	.8927	.7028	.6790	.6691	.6498	.6584	.9473	.7599	.7337	.7241	.7184
	L_{GHMC}	.9620	.9461	.9216	.9184	.8419	.7197	.6945	.6916	.6735	.6629	.8271	.7696	.7429	.7081	.7074
	L_{RL}(ours)	**.9908**	**.9815**	**.9644**	**.9531**	**.9213**	**.7678**	**.7179**	**.7084**	**.6844**	**.6820**	.9891	**.7962**	.7482	**.7372**	.7268

Table 2: Comparison between our method (L_{RL}) and previous methods based on CrossEntropy Loss (L_{CE}), Focal Loss (L_{FL}) and GHMC Loss (L_{GHMC}) in federate learning, over three data sets and different levels of global imbalance.

Data		**MNIST**					**CIFAR10**					**Fer2013**				
γ		B.	10:1	20:1	50:1	100:1	B.	10:1	20:1	50:1	100:1	B.	10:1	20:1	50:1	100:1
Ac.M %	L_{CE}	98.68	90.14	85.86	75.64	51.20	73.07	17.93	11.84	01.53	00.47	83.23	10.95	04.70	01.85	00.91
	L_{FL}	**99.22**	93.88	87.03	78.70	58.55	65.40	27.90	16.00	05.70	02.37	83.04	14.07	**08.18**	02.31	00.83
	L_{GHMC}	97.55	92,91	88.03	78.84	59.01	74.07	28.10	17.04	05.87	02.35	83.31	14.32	06.01	01.85	00.86
	L_{RL}(ours)	98.59	**93.99**	**89.85**	**79.41**	**60.34**	**76.33**	**29.87**	**17.70**	**06.43**	**02.40**	**83.36**	**14.93**	06.99	**02.46**	**00.93**
AUC	L_{CE}	.9929	.9793	.9729	.9543	.9108	.8429	.7354	.7183	.7078	.7068	.8831	.6975	.6853	.6752	.6745
	L_{FL}	**.9934**	.9862	.9739	.9612	.9151	.8001	.7689	.7530	.7442	.7318	.8713	.7135	.7029	.6894	.6893
	L_{GHMC}	.9895	.9862	.9799	.9615	.9237	.8470	.7773	.7658	.7444	.7309	.8881	.7233	.7030	**.7041**	.6902
	L_{RL}(ours)	.9932	**.9864**	**.9801**	**.9625**	**.9306**	**.8624**	**.7868**	**.7712**	**.7447**	**.7416**	**.8883**	**.7236**	**.7049**	.6927	**.6905**

Table 3: Comparison between our method (L_{RL}) and previous methods based on CrossEntropy Loss (L_{CE}), Focal Loss (L_{FL}) and GHMC Loss (L_{GHMC}), when the models are not trained with federate learning.

Data		**MNIST**				**CIFAR10**				**Fer2013**			
CS		0.6960	0.6158	0.5111	0.3984	0.6960	0.6158	0.5111	0.3984	0.9343	0.8489	0.7411	0.6732
Ac.M %	L_{MSE}	70.36	60.88	45.68	00.60	01.87	01.40	01.70	07.60	48.77	37.25	18.25	03.55
	L_{MFE}	**71.33**	59.71	40.42	00.00	**01.97**	01.37	01.60	07.03	46.41	35.01	16.47	03.14
AUC	L_{MSE}	.9397	.9214	.8848	.7775	.6638	.6450	.6267	.5842	.8564	.8259	.7772	.7400
	L_{MFE}	**.9417**	.9167	.8787	.7754	.6623	.6448	.6256	.5820	.8502	.8209	.7738	.7382

Table 4: Comparison between the Mean-Square-Error Loss (L_{MSE}), and the MFE Loss with local knowledge in FL setting (L_{MFE}) over three data sets and under different levels of mismatch between local and global imbalance (measured by CS).

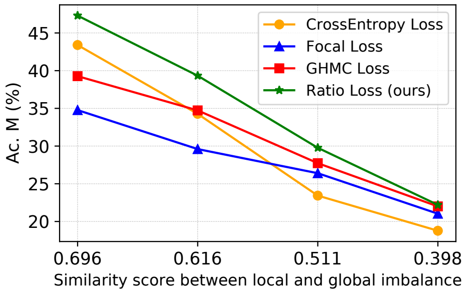

Figure 3: Comparison between Ratio Loss, CrossEntropy Loss, Focal Loss and GHMC Loss under different levels of mismatch between local and global imbalance.

that the performance of global model with L_{MFE} is worse than the baseline when the cosine similarity (CS) score is very low. This indicates the negative impact of solving the imbalance locally to the global model when there is significant mismatch between the local and global imbalance. Fig. 3 further shows the Ac.M of four methods under different levels of mismatch for CIFAR10 with $\Gamma = 10:1$ (more results for other data sets and global imbalance are in the $\mathcal{SM}$). The x-axis shows the average mismatch between local and global imbalance, measured by the average of CS scores over clients (the larger the number, the less the mismatch). From the figure, we can observe that 1) larger mismatch worsens the performance for all methods, and 2) our method outperforms the other methods under all levels of mismatch.

Conclusion

We present a novel method to address the class imbalance issue in federate learning (FL). Our approach includes a monitoring scheme that can infer the composition of training data at every FL round and detect the existence of possible global imbalance, and a new loss function (Ratio Loss) for mitigating the impact of global class imbalance. Extensive experiments demonstrate that our method can significantly outperform previous imbalance solutions. Even in regular neural network training, our method can also achieve the state-of-art performance. And our method works effectively without the sacrifice of user privacy.

Acknowledgments

We appreciate all anonymous reviewers for their valuable and detailed comments. We gratefully acknowledge the support from NSF grants 1834701, 1839511 and 1724341.

References

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191.

Buda, M.; Maki, A.; and Mazurowski, M. A. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* 106: 249–259.

Caldas, S.; Wu, P.; Li, T.; Konečnỳ, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097* .

Chawla, N. V.; Lazarevic, A.; Hall, L. O.; and Bowyer, K. W. 2003. SMOTEBoost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, 107–119. Springer.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9268–9277.

Davis, J.; and Goadrich, M. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, 233–240.

Dong, J.; Cong, Y.; Sun, G.; and Hou, D. 2019. Semantic-transferable weakly-supervised endoscopic lesions segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 10712–10721.

Dong, J.; Cong, Y.; Sun, G.; Zhong, B.; and Xu, X. 2020. What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4023–4032.

Duan, M.; Liu, D.; Chen, X.; Tan, Y.; Ren, J.; Qiao, L.; and Liang, L. 2019. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*, 246–254. IEEE.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, 1180–1189. PMLR.

Geyer, R. C.; Klein, T.; and Nabi, M. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* .

Goodfellow, I. J.; Erhan, D.; Carrier, P. L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.-H.; et al. 2013. Challenges in representation learning: A report on three machine learning contests. In *International Conference on Neural Information Processing*, 117–124. Springer.

Hamm, J.; Cao, Y.; and Belkin, M. 2016. Learning privately from multiparty data. In *International Conference on Machine Learning*, 555–563.

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

He, H.; and Garcia, E. A. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* 21(9): 1263–1284.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, C.; Li, Y.; Change Loy, C.; and Tang, X. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5375–5384.

Jo, T.; and Japkowicz, N. 2004. Class imbalances versus small disjuncts. *ACM Sigkdd Explorations Newsletter* 6(1): 40–49.

Kim, H.; Park, J.; Jang, J.; and Yoon, S. 2016. Deepspark: Spark-based deep learning supporting asynchronous updates and caffe compatibility. *arXiv preprint arXiv:1602.08191* 3.

Konečnỳ, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* .

Lee, H.; Park, M.; and Kim, J. 2016. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, 3713–3717. IEEE.

Li, B.; Liu, Y.; and Wang, X. 2019. Gradient harmonized single-stage detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8577–8584.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37(3): 50–60.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, X.-Y.; Wu, J.; and Zhou, Z.-H. 2008. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39(2): 539–550.

Luo, H.; Ji, L.; Li, T.; Duan, N.; and Jiang, D. 2020. GRACE: Gradient Harmonized and Cascaded Labeling for Aspect-based Sentiment Analysis. *arXiv preprint arXiv:2009.10557* .

Malave, N.; and Nimkar, A. V. 2020. A survey on effects of class imbalance in data pre-processing stage of classification problem. *International Journal of Computational Systems Engineering* 6(2): 63–75.

Mani, I.; and Zhang, I. 2003. kNN approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629* .

Melis, L.; Song, C.; De Cristofaro, E.; and Shmatikov, V. 2018. Exploiting unintended feature leakage in collaborative learning. *arXiv preprint arXiv:1805.04049* .

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Nguyen, T. D.; Marchal, S.; Miettinen, M.; Fereidooni, H.; Asokan, N.; and Sadeghi, A.-R. 2018. Dï IoT: A Federated Self-learning Anomaly Detection System for IoT. *arXiv preprint arXiv:1804.07474* .

Niu, C.; Wu, F.; Tang, S.; Hua, L.; Jia, R.; Lv, C.; Wu, Z.; and Chen, G. 2020. Billion-scale federated learning on mobile clients: a submodel design with tunable privacy. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 1–14.

Pouyanfar, S.; Tao, Y.; Mohan, A.; Tian, H.; Kaseb, A. S.; Gauen, K.; Dailey, R.; Aghajanzadeh, S.; Lu, Y.-H.; Chen, S.-C.; et al. 2018. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, 112–117. IEEE.

Ramaswamy, S.; Mathews, R.; Rao, K.; and Beaufays, F. 2019. Federated Learning for Emoji Prediction in a Mobile Keyboard. *arXiv preprint arXiv:1906.04329* .

Rao, R. B.; Krishnan, S.; and Niculescu, R. S. 2006. Data mining for improved cardiac care. *ACM SIGKDD Explorations Newsletter* 8(1): 3–10.

Samarakoon, S.; Bennis, M.; Saady, W.; and Debbah, M. 2018. Distributed federated learning for ultra-reliable low-latency vehicular communications. *arXiv preprint arXiv:1807.08127* .

Sergeev, A.; and Del Balso, M. 2018. Horovod: fast and easy distributed deep learning in TensorFlow. *arXiv preprint arXiv:1802.05799* .

Sun, G.; Cong, Y.; Dong, J.; Wang, Q.; and Liu, J. 2020. Data Poisoning Attacks on Federated Machine Learning. *arXiv preprint arXiv:2004.10020* .

Sun, Y.; Kamel, M. S.; Wong, A. K.; and Wang, Y. 2007. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* 40(12): 3358–3378.

Van Hulse, J.; Khoshgoftaar, T. M.; and Napolitano, A. 2007. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, 935–942.

Wang, K.; and Zhang, L. 2020. Single-Shot Two-Pronged Detector with Rectified IoU Loss. *arXiv preprint arXiv:2008.03511* .

Wang, L.; Xu, S.; Wang, X.; and Zhu, Q. 2019. Eavesdrop the Composition Proportion of Training Labels in Federated Learning. *arXiv preprint arXiv:1910.06044* .

Wang, R.; Hu, K.; Zhu, Y.; Shu, J.; Zhao, Q.; and Meng, D. 2020. Meta Feature Modulator for Long-tailed Recognition. *arXiv preprint arXiv:2008.03428* .

Wang, S.; Liu, W.; Wu, J.; Cao, L.; Meng, Q.; and Kennedy, P. J. 2016. Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, 4368–4374. IEEE.

Wang, Y.-X.; Ramanan, D.; and Hebert, M. 2017. Learning to model the tail. In *Advances in Neural Information Processing Systems*, 7029–7039.

Xia, M.; Zhang, G.; Mu, C.; Guan, B.; and Wang, M. 2020. Cervical Cancer Cell Detection Based on Deep Convolutional Neural Network. In *2020 39th Chinese Control Conference (CCC)*, 6527–6532. IEEE.

Xu, S.; Wang, L.; Wang, Y.; and Zhu, Q. 2021. Weak Adaptation Learning–Addressing Cross-domain Data Insufficiency with Weak Annotator. *arXiv preprint arXiv:2102.07358* .

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10(2): 1–19.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, 14747–14756.