Ahmad Khalil, Aidmar Wainakh, Ephraim Zimmer, Javier Parra-Arnau, Antonio Fernández Anta, Ralf Steinmetz. *IEEE International Conference on Fog and Mobile Edge Computing (2023)*

# Label-Aware Aggregation for Improved Federated Learning

Ahmad Khalil*, Aidmar Wainakh†, Ephraim Zimmer†, Javier Parra-Arnau‡, Antonio Fernandez Anta§, Tobias Meuser*, and Ralf Steinmetz*
*Multimedia Communications Lab, Technical University of Darmstadt, Germany
{ahmad.khalil, tobias.meuser, ralf.steinmetz}@kom.tu-darmstadt.de
†Telecooperation Lab, Technical University of Darmstadt, Germany
wainakh@tk.tu-darmstadt.de, zimmer@privacy-trust.tu-darmstadt.de
‡Network Engineering Dept., Universitat Politècnica de Catalunya, Spain
javier.parra@upc.edu
§IMDEA Networks Institute, Spain
antonio.fernandez@imdea.org

***Abstract*—Federated Averaging (FedAvg) is the most common aggregation method used in Federated learning, which performs a weighted averaging of the updates based on the sizes of the individual datasets of each client. A raising discussion in the research community suggests that FedAvg might not be the optimal method since, for instance, it does not fully take into account the variety of the client data distributions. In this paper, we propose a label-aware aggregation method FedLA, that addresses the biased models issue by considering the variety of labels in the weighted averaging. It combines two main properties of the client data, namely data size and label distribution. Through extensive experiments, we demonstrate that FedLA is particularly effective in several heterogeneous data distribution scenarios. Especially when only a small group of the clients is participating in the Federated Learning process. Furthermore, we argue that accurately describing the data distribution is crucial in selecting the appropriate aggregation method. In this regard, we discuss various properties that can be used to describe data distribution and illustrate how these properties can guide the choice of an aggregation method for specific data distributions.**

***Index Terms*—Federated learning, Heterogeneous data distribution, non-IID**

## I. Introduction

Federated Learning (FL) has gained significant attention in recent years due to its ability to address two critical issues in distributed learning: 1) data privacy and 2) communication efficiency. Unlike traditional machine learning approaches that require centralizing the training data on a single central server, FL allows the clients to train the model locally and share only their model updates with the central server, thereby reducing significantly the amount of data exchanged between the clients and the server. Moreover, as the user data is maintained locally, FL provides enhanced user privacy – although recent research has highlighted potential information leakages in FL [5], [13], [16].

To aggregate the updates obtained from the clients, the server uses an aggregation operation that is crucial for the convergence of the model. McMahan et al. [12], the founders of FL, proposed the Federated Averaging (FedAvg) algorithm to aggregate the updates. FedAvg has been widely adopted and serves as a fundamental building block for many FL algorithms. FedAvg assigns weights to client updates solely based on the size of their local data, making it suitable for Independent and Identically Distributed (IID) data. However, for non-Independent and Identically Distributed (non-IID) data, FedAvg may require a large number of communication rounds to converge [12].

In fact, in real FL scenarios, data is typically heterogeneous: non-IID and imbalanced [12]. Ignoring this heterogeneity can lead to biased models and unfairness [8]. Recent studies indicate that FedAvg can lead to low accuracy in some cases [14] and may not be optimal [18]. To address these issues, various approaches [11], [17] have been proposed to improve the aggregation process. For instance, FedProx [11] includes a proximal term in the objective function to encourage client models to stay close to the global model. On its hand, FedNova [17] uses a second-order optimization method to estimate the curvature of the loss surface and adjust the step size of the updates accordingly. While these approaches have shown promise, they offer limited improvements over FedAvg, and their conditions to be effective are not always clear.

Hence, the data distribution is a major factor when measuring the effectiveness of aggregation methods. Despite this, the existing literature on aggregation methods primarily focuses on scenarios where data is IID, while the complexities of non-IID data distributions remain relatively unexplored. Consequently, two major issues arise. Firstly, it is unclear which aggregation methods perform better under which non-IID distribution. Secondly, it is challenging to compare different aggregation methods against each other. In light of these limitations, further research is needed to explore the potential of aggregation methods in non-IID settings and to develop evaluation metrics that account for the variability of non-IID data distributions.

The following outlines the contributions presented in our paper:

- We propose Federated Learning with Label Awareness (FedLA), an aggregation method that addresses the non-IID data distribution by considering the distribution

of classification labels in the weighted averaging. Our method is simple, yet effective in several scenarios (we discuss them in more detail in Section IV). FedLA computes a weighted average of the models trained on each client, where the weights are proportional to the number of instances of each label in the local dataset. Moreover, FedLA outperforms FedAvg, especially in cases where clients' participation is considerably small.

- In addition, we argue that the key point for selecting a proper aggregation method is describing the data distribution (e.g., non-IID) as precisely as possible. Therefore, we elaborate on the different properties to describe data distribution. We demonstrate how these properties can be used to decide which aggregation method is more suitable for a particular data distribution.
- Finally, the complete source code, encompassing the data distribution scenarios and the full corresponding experimental outcomes considering different performance metrics, are openly available at [1]

## II. Literature Review

The proper specification of data distribution is crucial for designing and implementing machine learning models, especially in the context of FL. In FL, the IID data distribution is commonly used as a standard, where each client's data follows the same distribution as the population, and the clients' distributions are independent of each other. Implementing this scenario in empirical experiments is straightforward. However, dealing with non-IID data distribution is challenging due to various potential sources of heterogeneity. Kairouz et al. [8] identified five distinct ways in which data can deviate from the IID case:

- Feature distribution skew: the feature distribution varies across clients.
- Label distribution skew: the label distribution varies across clients.
- Feature-label mapping skew: different features across clients yield the same label.
- Label-feature mapping skew: different labels across clients are mapped with the same features.
- Quantity skew: different clients possess significantly different amounts of data.

To consider the heterogeneity of the data and non-IID distributions, numerous approaches have been proposed in the literature. Broadly speaking, these approaches can be classified into two main categories: (1) data-based and (2) parameter-based approaches.

**Data-based approaches.** These approaches elicit properties of the data and incorporate these properties into the aggregation method. One of the most common approaches in this category is FedAvg [12], where the data size of the clients is used as a property into the weighted averaging process. However, FedAvg is mainly addressing the *quantity skew* issue, while overlooking the other non-IID cases leading to underestimation of the full extent of data heterogeneity in FL. Empirical evidence demonstrates that combining data from different sources without considering the diversity of the sources can yield a misleading interpretation of results [15]. Xiao et al. [18] studied the behavior of the model parameters of different clients and demonstrated that, with the increase of training iterations, the model parameters of different clients become more correlated but not necessarily closer in value. This indicates that the commonly used averaging technique may not be the most effective approach for parameter aggregation. Anelli et al. [1] studied clients' contributions in FL and proposed criteria to measure their quality. They focused on dataset properties for image classification and identified key factors: dataset size, label diversity, model divergence, class balance, and image sharpness. The authors showed that these criteria influenced the overall FL training process and the resulting model's performance, with certain factors being more impactful than others.

**Parameter-based approaches.** These approaches optimize model parameters to enhance accuracy, without modifying the data distribution to suit the model. The core idea is that training a machine learning model on non-IID data reflects the statistical properties of the model parameters. Reyes et al. [15] proposed an averaging algorithm that incorporates estimated parameter variances. They penalized model uncertainty at the client level by using inverse variance as weights during averaging. However, their study mainly focused on a specific non-IID scenario with two classes per client, limiting the generalizability of their findings to other non-IID cases. Hsu et al. [7] proposed two novel algorithms to improve aggregation in FL. They focused on addressing two key issues: label distribution skew and quantity skew. The first algorithm incorporates importance weights into local optimization to obtain an unbiased estimator of the loss. However, it requires knowledge of the target distribution at the server. Nonetheless, this approach greatly enhances FL model accuracy. The second algorithm introduces virtual clients to tackle the quantity skew problem, ensuring equal data contribution from all clients. This helps minimize the impact of imbalanced data and results in more robust and accurate models. Zhuo et al. [19] introduced a novel approach to enhance the performance of FL models by filtering and re-weighting client model parameters. This approach is particularly useful for both IID and non-IID data distributions. The method consists of two steps: weighting the parameters of the client model's final layer based on class sample counts and removing nodes or kernels with high variance in local client models during aggregation. While the approach shows promising results, the study lacks a clear and thorough description of non-IID data distributions, which would provide better understanding of its practical applicability. Oza et al. [14] proposed a novel method to train the global model on the server side using statistical properties of local model parameters. They utilized the mean and variance of these parameters to train the global model. However, their method is limited to scenarios where clients have an equal

[1] https : / / github . com / AhmadMkhalil / Label - Aware - Aggregation - in - Federated-Learning

number of samples. Despite this limitation, their approach demonstrates promising results in improving the accuracy of image-based authentication models in FL settings.

Overall, the impact of different approaches described in the literature has not been thoroughly analyzed yet to understand their behavior under diverse data distributions. One of the primary reasons for this gap is the inconsistent means used by researchers to describe data distributions in their experiments. Many researchers rely on vague terms like non-IID without considering various factors related to data distribution mentioned earlier. Consequently, the interpretation of the results obtained from such experiments remains a matter of debate. To address this issue, our research paper takes a significant step by providing a more precise description of the data distribution. This step will ultimately help researchers to accurately evaluate the effectiveness of various proposed approaches.

## III. Method

In this Section, we introduce our approach to addressing label distribution skew in FL. We propose a straightforward aggregation method called FedLA, which utilizes weighted averaging. Additionally, we present three properties for describing label distribution skew in non-IID data. These contributions aim to improve model performance and mitigate bias in FL systems.

### A. FedLA

The weights used in our method are based on a combination of the data size and the variety of labels per client. It is crucial to note that in this context, we assume that the central server, responsible for the averaging process, possesses knowledge about the label distribution information of all clients participating in the communication round. To achieve this, during each communication round, every client shares additional information (metadata) with the server, which includes the label distribution of the data used to train the local model, along with the model update.

As depicted in **Algorithm** 1, let $k$ denote the number of clients participating in one communication round, and $n$ denote the number of labels, and $S(c_i)$ denote the number of samples held by each client $i$ participating in the communication round. For the label with index $j$ (namely, $l_j$), let $S(l_j)$ denote the number of samples with label $l_j$, and $S(c_i, l_j)$ the number of samples with label $l_j$ at client $c_i$. To calculate the weights for FedLA, we take three steps as follows:

1) Client weight per label: To compute the weights, we first calculate the client weight per label $l_j$. To do so, we sum all the samples of the label, $S(l_j)$, as follows

$$S(l_j) = \sum_{i=1}^{k} S(c_i, l_j).$$

Then, we divide the number of samples of client $c_i$ with label $l_j$ by this number to obtain its weight with respect to (w.r.t.) the label,

$$W(c_i, l_j) = \frac{S(c_i, l_j)}{S(l_j)}.$$

This step ensures that clients with a higher number of labeled samples for a particular label have a higher weight for that label, thereby improving the overall performance of the model.

2) Client weight w.r.t. all labels: Next, we calculate the client weight for each client $c_i$ with respect to all labels by summing all the weights computed in the previous step.

$$W(c_i) = \sum_{j=1}^{n} W(c_i, l_j)$$

This step ensures that clients with a higher overall number of labeled samples have a higher weight, regardless of the label.

3) Client update weight: Finally, we compute the client update weight, which is the ratio of a client's weight to the sum of all clients' weights. This weight is used in a weighted average to obtain the aggregate.

$$W_{FedLA}(c_i) = \frac{W(c_i)}{\sum_{x=1}^{k} W(c_x)}$$

By using this weighted aggregation scheme, we can improve the performance of the model on clients with a smaller number of labeled samples or a more diverse label distribution.

**Algorithm 1** FedLA Weights Calculation in Each Communication Round

Initialize: $k$ , $S(c_i, l_j)$ (for all clients participating in the communication round)
**for** each label $l_j$ **do**
  Calculate label total samples $S(l_j) = \sum_{i=1}^{k} S(c_i, l_j)$
  **for** each client $c_i$ **do**
    Compute client weight per label: $W(c_i, l_j) = \frac{S(c_i, l_j)}{S(l_j)}$
**for** each client $c_i$ **do**
  Compute client weight w.r.t. all labels: $W(c_i) = \sum_{j=1}^{n} W(c_i, l_j)$
  Compute client update weight: $W_{FedLA}(c_i) = \frac{W(c_i)}{\sum_{x=1}^{k} W(c_x)}$

This weighted aggregation ensures proportional representation of all existing labels in the aggregated model, regardless of sample count. It mitigates bias towards frequently appearing labels by considering each client's contribution based on the diversity of its local dataset. For example, a label that appears at one user with 1000 samples will yield label weight of 1 for that user. The same weight also 1 will be assigned to a label that appears in one user with 1 sample.

We demonstrate the effectiveness of our proposed approach with the following example in Table I. We assume three clients $c_i : i \in \{1, 2, 3\}$ and three labels $l_j : j \in \{a, b, c\}$. The weight of client $c_1$ w.r.t. label $l_a$ is $W(c_1, l_a) = 700/1000 = 0.7$. Then the client weight for all labels $W(c_1) = \sum_j^n W(c_1, l_j) =$

| **Labels** | **Client** $c_1$ | | **Client** $c_2$ | | **Client** $c_3$ | | $S(l_j)$ |
|---|---|---|---|---|---|---|---|
| | $S(c_1, l_j)$ | $W(c_1, l_j)$ | $S(c_2, l_j)$ | $W(c_2, l_j)$ | $S(c_3, l_j)$ | $W(c_3, l_j)$ | |
| a | 700 | 0.7 | 200 | 0.2 | 100 | 0.1 | 1000 |
| b | 0 | 0 | 100 | 1 | 0 | 0 | 100 |
| c | 0 | 0 | 25 | 0.5 | 25 | 0.5 | 50 |
| $S(c_i)$ | 700 | 0.7 | 325 | 1.7 | 125 | 0.6 | |
| **FedAvg** | | 0.61 | | 0.28 | | 0.11 | |
| **FedLA** | | **0.23** | | **0.56** | | **0.20** | |

TABLE I: Example of client weights calculated using classical FedAvg and FedLA. $S$ refers to sample size, and $W$ refers to weight. $S(c_i) = \sum_{j=1}^{n} S(c_i, l_j)$. Client $c_1$, Client $c_2$, and Client $c_3$ are participating in the FL communication round.

0.7. Finally, the weight $W_{FedLA}(c_1) = W(c_1)/\sum_x^k W(c_x) = 0.7/(0.7 + 1.7 + 0.6) = 0.23$.

When comparing FedAvg and our proposed method, we notice a significant difference in the weight assigned to Client $c_1$. While FedAvg gives it a weight of $0.61$, our method assigns a lower weight of $0.23$. This adjustment aims to balance the contributions of different clients, considering factors such as label diversity. For instance, Client $c_2$ is assigned a weight of $0.56$ due to its more varied labels.

It is pertinent to acknowledge that FedLA operates under the premise that the central server is endowed with comprehensive insights into the distribution of labels across all participating clients during each communication round. However, it's essential to consider that this assumption may prove impractical within certain FL applications. Instances where privacy considerations restrict clients from divulging this particular information are noteworthy. Thus, to render FedLA applicable in scenarios characterized by stringent privacy constraints, novel approaches are requisite. Implementing privacy-preserving strategies like secure multi-party computation [3] could be pivotal in aligning modified FedLA with privacy mandates. On the other hand, in the context of FL within some vehicular applications, which revolves around training object detection models [9], FedLA could be applied without unduly compromising privacy. In this application, when the clients (e.g., cars) share lightweight anonymized label information (such as cars, vegetation, etc.) alongside their respective frequencies, this might not inherently pose significant privacy risks.

In order to substantiate the effectiveness of our method, we thoroughly present and analyze the complete set of experiments in the subsequent Section IV. We encompasse diverse scenarios and datasets, ensuring a comprehensive evaluation of FedLA. By doing so, we aim to foster a deeper comprehension of our findings and facilitate a more precise interpretation of the obtained results.

### *B. Describing Label Distribution Skew*

Our second contribution in this work is to focus on describing non-IID data, specifically the label distribution skew. To better understand this phenomenon, we propose three properties to consider.

1) Firstly, the label distribution of client data, which reflects the diversity of labels per client. We use a histogram as shown in Figure 1 to depict this property, where the x-axis represents the labels, and the y-axis represents the number of samples per label. When the client has all

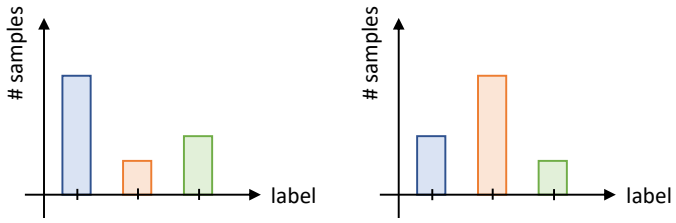



Fig. 1: Two examples of label distributions, corresponding to two different clients.

labels with a uniform distribution, this represents one example of the IID. Conversely, when the client possesses a solitary label, it deviates significantly from the principle of IID data and leans towards the realm of non-IID data. Conversely, in case where the client has only one label, this considered to be far from IID and near to non-IID

2) Secondly, we introduce the concept of *client distribution consistency*, which refers to the consistency of the label distribution between clients (see Figure 2). We measure

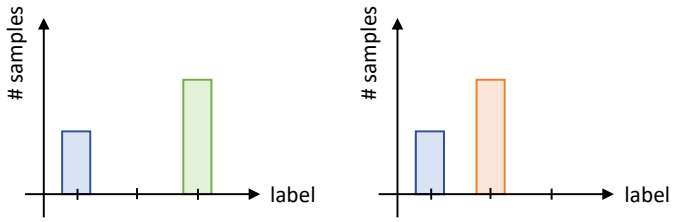



Fig. 2: Two examples of label distributions, corresponding to two differents clients, that exhibit distribution consistency.

this property by calculating the distance between client distributions, regardless of the labels' IDs. We consider two clients to have similar distributions if they have the same mean and variance but with different labels. The optimal case of this property happens when all clients have the same distribution. Several metrics, such as

Earthmover's distance and Hellinger distance, can be used to measure the distance.

3) Finally, we explore the inter-client distribution, which indicates the extent to which clients share the same labels, i.e., the overlap between the labels belonging to different clients. As we can see in Figure 3, we use a histogram to depict this property, where the x-axis represents the clients, and the y-axis represents the number of samples with shared labels. Thus, the data distribution scenarios

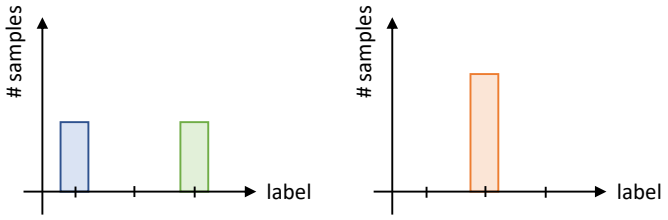



Fig. 3: Two examples of label distributions, corresponding to two differents clients, that do not overlap each other.

can range from a situation where all clients possess the same $n$ labels to the cases where there is no overlap, implying that each client has a unique set of labels that are not shared by any other client.

By considering these properties, we can better design algorithms that can handle non-IID data effectively (*label distribution skew*), leading to improved model performance and better overall results. In the following Section IV, we conduct a thorough investigation into how our model aggregation method performs concerning the variations in label distribution skew properties.

## IV. Experiments

In order to comprehensively address the label distribution skew properties mentioned earlier (Section III), we construct diverse data distribution scenarios. We investigate how these different data distribution scenarios influence the performance of the FL with classical FedAvg. On the other hand, we show different scenarios where our proposed FedLA algorithm outperforms FedAvg. To achieve this goal, we conduct a series of experiments to evaluate the performance of our proposed method.

### A. Experimental Setup

To evaluate our approach, we adopt two simple classification tasks.

**Classification of handwritten letters and digits:** For this task, we use the EMNIST-balanced dataset [4]. This dataset is balanced, thus, each class has the same number of samples. The total number of classes is 47, so that $j \in \{1, ..., 47\}$ , and for each class there are 2400 samples.
We conducted the experiment for this task using the FL implementation found on Github [2], where we employed the *CNNMnist* architecture for the purpose of classification, which consists of two convolutional, one dropout, and two linear layers. We selected the learning rate $lr = 0.01$, with a batch size $b = 256$. We used Stochastic Gradient Descent (SGD) for optimization.
**Classification of multi-class images:** For this task, we use the CIFAR-100 [10] dataset. This dataset contains 100 classes $j \in \{1, ..., 100\}$, with 600 samples each.
When developing the model, we were inspired by the design proposed by He et al. [6], which encompasses 5 convolutional blocks and 3 residual blocks. We selected the learning rate $lr = 0.001$, with a batch size $b = 256$. We used SGD for optimization.

*Federated Learning settings*: we assume that the total number of clients is 10, $i \in \{1, ..., 10\}$. For the first classification task where the EMNIST-balanced dataset is used, each client has 2400 samples $S(c_i) = 2400$. On the other hand, for the second classification task where the CIFAR-100 dataset is used, each client has 1000 samples, meaning we have no quantity skew in both tasks. We train the models for 100 communication rounds $g_e = 100$. Between communication rounds, each of the participating clients performs 10 local epochs, $l_e = 10$.
Finally, we trained the models using the hardware RTX3090 TI GPU. We repeated each experiment 10 times and reported the average model test accuracy.
*Data distribution scenarios*: As depicted in Figure 4, we divide the clients into two groups; in the first group the data is IID, while the second group has non-IID data.

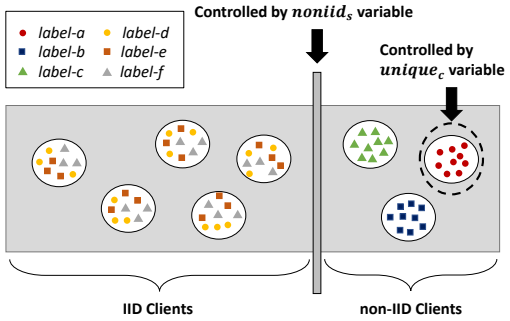



Fig. 4: Dummy example of clients data distribution in two IID and non-IID with six different labels. The size of each clients group is determined by the variable $noniid_s$, while the variable $unique_c$ determines the number of distinct classes (labels) held by each client within the non-IID group.

We provide more details on the data distribution in the following.

To control the data distribution we define three variables:

- $unique_c$ refers to the number of unique classes (labels) held by each *non-IID* client. For EMNIST-balanced dataset, we define five possible values for this variable $unique_c \in \{1, 2, 3, 4, 6\}$. On the other hand, for CIFAR-100 dataset, we define three possible values for this variable $unique_c \in \{2, 5, 10\}$. This is applied to all

*non-IID* clients, thus, having $unique_c$=1 means that all *non-IID* clients have all $S(c_i)$ samples of a unique class, and there is no other client that possesses this class in both *non-IID* and *IID* clients groups.

- $noniid_s$ refers to the *non-IID* group size ratio w.r.t. the total number of the participants. We define four possible values for this variable $noniid_s \in \{0.1, 0.3, 0.5, 0.7\}$, where $noniid_s = 1$ indicates that all the clients belong to the non-IID group.
- $k_r$ refers to the participation ratio in the current communication round. We define four different values control the number of the participants in each communication round, such that $k_r \in \{0.1, 0.3, 0.5, 0.7\}$.

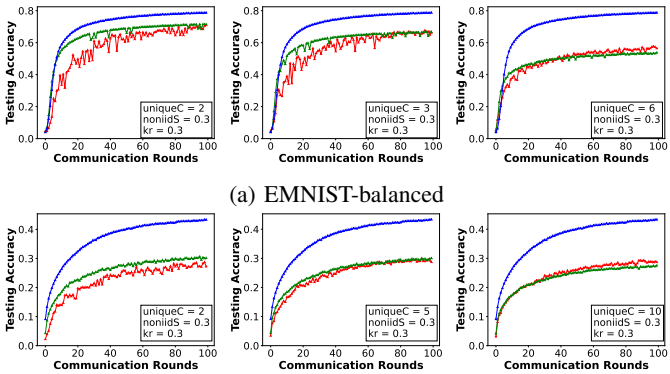
It is important to mention that the values of $noniid_s$ and $unique_c$ directly impact the number of classes within the *IID* group; larger values of $noniid_s$ and $unique_c$ correspond to a reduced number of classes in the *IID* group. Now and after defining the different data distribution control variables, in the following we provide an example of a possible data distribution setup: Let $unique_c = 4$, $noniid_s = 0.5$, $k_r = 0.3$. Thus, each client in *non-IID* has four different unique classes, 25% of the clients' samples belongs to one of the four classes. There is no other client that has these four classes in both *non-IID* and *IID*. Moreover, *non-iid* includes 50% of the total number of the clients, and the rest of the clients are in the *IID* group. $k_r = 0.3$ means that 30% of clients will be randomly selected to participate in each communication round.

In order to thoroughly explore the multitude of potential data distribution scenarios, our experimental design encompassed a wide array of 80 distinct data distributions for the first classification task and an additional 60 distributions for the second task. By systematically varying the data distributions, we aimed to capture a comprehensive representation of the diverse patterns and characteristics that may arise in real-world scenarios. To ensure robustness and account for potential variations, we repeated each training iteration a total of 10 times. We calculated and reported the average values derived from these repeated experiments. In the subsequent sections, we present our analysis of the outcomes obtained from this extensive experimentation. We critically examine and interpret the results, striving to uncover key insights and patterns that emerged across the diverse data distribution scenarios. Our focus is not only on evaluating the performance of the novel approach of FedLA but also on comparing it with the classical approach of FedAvg. We also show the performance of FedAvg on fully IID dataset as a reference in our comparisons.

### *B. Results Discussion*

In this section, we discuss the most interesting results of our experiments. The complete set of the results can be found in our `Github repository`. In the following, we delve into selected significant experimental findings.

*1) Individual Data Distribution Parameters:* $\boldsymbol{unique_c}$: We observed a negative impact of increasing the value of the parameter $unique_c$ on the accuracy of FedAvg and FedLA on both datasets (when fixing the other data distribution parameters). However, the results in Figure 5 show that the FedAvg approach exhibits similar resilience to this increase compared to FedLA. This is expected because the non-IID nature of the data not only affects data distribution in non-IID group, but also influences the data distribution of the IID group. Increasing parameter $unique_c$ yields two fundamental outcomes: (1) an increase in the number of classes within the non-IID group, coupled with a decrease in the number of classes within the IID group, (2) considering the fixed number of samples per client, a decrease in the number of training samples available per unique class specifically within the non-IID group.




(a) EMNIST-balanced

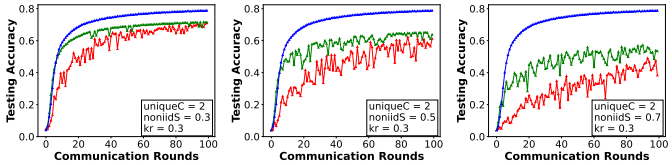



(b) CIFAR-100

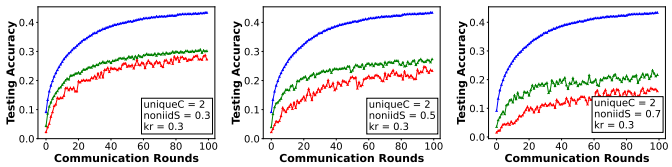
Fig. 5: Comparison between FedAvg, FedLA, and IID baseline when $unique_C$ changes.

In the case of FedAvg, an increase in the number of unique classes assigned to each client leads to a proportional decrease in the number of samples per unique class. Consequently, the model's performance deteriorates due to reduced training on these classes. However, the class distribution imbalance results in an increased number of samples per class in the IID group, enhancing overall model stability. Conversely, FedLA assigns significantly higher weights to clients in the non-IID group compared to those in the IID group, disregarding the potential positive impact from the IID group users' data.

$\boldsymbol{noniid_s}$: Increasing the value of the parameter $noniid_s$ results in two effects: (1) an increase in the number of classes in the non-IID group, coupled with a decrease in the number of classes in the IID group, and (2) a reduction in the size of the IID group, leading to a decline in the positive effect of IID group data. Typically, an increase in the value of parameter $noniid_s$ leads to a decrease in accuracy, indicating a higher degree of non-iidness. However, our proposed approach demonstrates enhanced resilience to variations in $noniid_s$ when the value of $unique_c$ is small. This is clearly depicted in Figure 6. This observation can be justified by considering the impact of $noniid_s$ in FedLA, where an increase in the value of this paramter diminishes the influence of non-IID clients relative to IID clients. In contrast, in the case of FedAvg, all the clients are assigned equal weights because no quantity

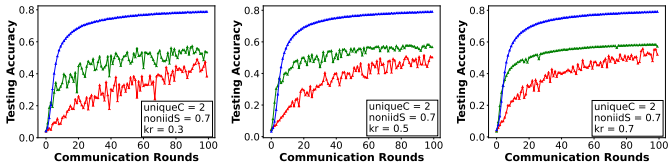



(a) EMNIST-balanced

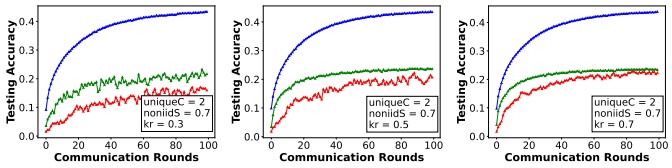



(b) CIFAR-100

Fig. 6: Comparison between FedAvg, FedLA, and IID baseline when $noniid_S$ changes.




(a) EMNIST-balanced




(b) CIFAR-100

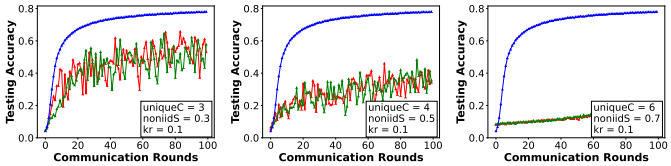
Fig. 7: Comparison between FedAvg, FedLA, and IID baseline when $k_r$ changes.

skew is considered, thereby magnifying the influence of non-IID clients.

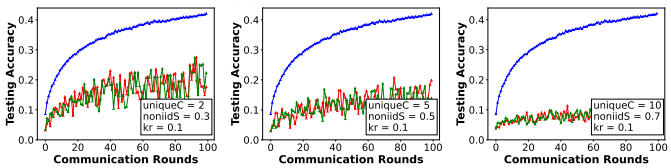
***$k_r$***: In Figure 7, we observe that an increase in the value of variable $k_r$ leads to greater stability of the accuracy chart for both methods. However, our proposed method achieves increased stability at a faster rate by effectively moderating the instability caused by high non-IIDness using label-aware crafted weights. FedLA enables rapid stabilization compared to FedAvg. By leveraging label information, our method effectively counteracts the destabilizing effects of non-IID data, resulting in enhanced stability.

*2) Investigating the Correlation between Different Data Distribution Parameters:* The increase in the values of both parameters $unique_c$ and $noniid_s$ can lead to severe outcomes when the value of parameter $k_r$ is considerably small (e.g., $k_r = 0.1$). Figure 8 reveals a clear inverse relationship between the rising values of the parameters $unique_c$ and $noniid_s$, and the attained accuracy, especially in cases where $k_r$ is considerably small.

This can be attributed to the fact that minimizing the value of $k_r$ leads to increased possibility of selecting only a small number of clients from the non-IID group during certain




(a) EMNIST-balanced

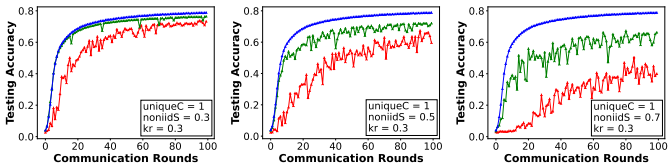



(b) CIFAR-100

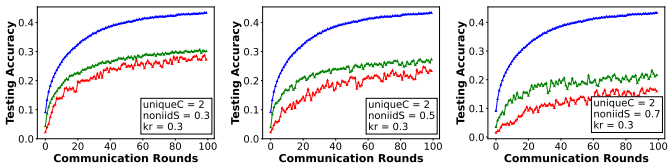
Fig. 8: Comparison between FedAvg, FedLA, and IID baseline. A clear inverse relationship between the rising values of $unique_C$ and $noniid_S$ and the attained accuracy, especially when $k_r$ is considerably small (e.g., $k_r = 0.1$ in these figures).

training iterations. This will amplify their adverse impact on the overall convergence of the model.

**Concluding observation:** Based on the aforementioned obser-




(a) EMNIST-balanced




(b) CIFAR-100

Fig. 9: Comparison between FedAvg, FedLA, and IID baseline. This shows one case in which FedLA outperforms FedAvg.

vations, it can be concluded that FedLA demonstrates superior performance compared to the conventional FedAvg method in various non-IID data distribution scenarios, particularly in cases where there is:

- A small number of unique classes ($unique_c$) per non-IID clients. This can be seen in application where clients are capturing highly personalized data as images of themselves or their pets.
- A large number of non-IID group clients ($noniid_s$). This could be the case in applications where clients are collecting data infrequently, thus, their data starts as non-IID and slowly transform to IID. Another example could be in applications where a big number of clients belong to highly diverse and distinct environments.

- A small number of participating clients ($k_r$). This can be the case, e.g., in the initial phase of newly launched applications, or when a big number of clients are offline or do not meet the FL client selection criteria.

This can be clearly observed in Figure 9. Finally, it is evident that FedLA exhibits superior performance in the first classification task, where the EMNIST-balanced dataset with a smaller number of classes and a higher number of samples per class was utilized, as compared to the second task involving the use of the CIFAR-100 dataset.

## V. Conclusion

In this paper, we introduced FedLA as a label-aware aggregation method that addresses biased models in Federated Learning. Moreover, we presented extensive experimental results highlighting the effectiveness of FedLA in various scenarios, especially in cases where clients' participation is considerably small. Additionally, we emphasized the significance of considering data distribution properties in selecting aggregation methods, and we provided insights into how these properties can guide the choice of aggregation methods for specific data distributions. By scrutinizing the diverse factors associated with data distribution, we aim to shed light on the underlying aspects that contribute to either favorable or unfavorable performance when utilizing either the traditional FedAvg or the innovative FedLA. Through this comprehensive investigation, we seek to deepen our understanding of the interplay between data distribution and the performance of FL methods, ultimately paving the way for more effective and efficient strategies in future endeavors. As a future work, further experimentation and investigation are required to fully uncover the advantages of our method. Moreover, while this paper primarily addresses label distribution skew, future studies should extend their focus to investigate other types of skew.

## Acknowledgment

This work has been funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 MAKI, the Federal Ministry of Education and Research of Germany in the project "Open6GHub" (16KISK014), and the Research Training Group (RTG) 2050 Privacy & Trust.

J.P.-A. is a "Ramón y Cajal" fellow (RYC2021-034256-I) funded by MCIN/AEI/10.13039/501100011033 and "NextGenerationEU"/PRTR.

## References

[1] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, and Antonio Ferrara. Prioritized multi-criteria federated learning. *Intelligenza Artificiale*, 14(2):183–200, 2020.

[2] AshwinRJ. Federated-learning-pytorch. https://github.com/AshwinRJ/Federated-Learning-PyTorch, 2021.

[3] David Byrd and Antigoni Polychroniadou. Differentially private secure multi-party computation for federated learning in financial applications. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–9, 2020.

[4] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters (2017). *arXiv preprint arXiv:1702.05373*, 2017.

[5] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1623–1640, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 76–92. Springer, 2020.

[8] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[9] Ahmad Khalil, Tobias Meuser, Yassin Alkhalili, Antonio Fernández Anta, Lukas Staecker, Ralf Steinmetz, et al. Situational collective perception: Adaptive and efficient collective perception in future vehicular systems. In *International Conference on Vehicle Technology and Intelligent Transport Systems*, pages 346–352, 2022.

[10] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). 2009.

[11] 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.

[12] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[13] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[14] Poojan Oza and Vishal M Patel. Federated learning-based active authentication on mobile devices. *arXiv preprint arXiv:2104.07158*, 2021.

[15] Jonatan Reyes, Lisa Di Jorio, Cecile Low-Kam, and Marta Kersten-Oertel. Precision-weighted federated learning. *arXiv preprint arXiv:2107.09627*, 2021.

[16] Nuria Rodríguez-Barroso, Daniel Jiménez-López, M Victoria Luzón, Francisco Herrera, and Eugenio Martínez-Cámara. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90:148–173, 2023.

[17] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[18] Peng Xiao, Samuel Cheng, Vladimir Stankovic, and Dejan Vukobratovic. Averaging is probably not the optimum way of aggregating parameters in federated learning. *Entropy*, 22(3):314, 2020.

[19] Yaoxin Zhuo and Baoxin Li. Fedns: Improving federated learning for collaborative image classification on mobile clients. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.