


Exploring Normalization for High Convergence on Federated Learning for Drones

Flávio Vieira   [Federal University of the State of Rio de Janeiro | flaviovieira@edu.unirio.br]

Carlos Alberto V. Campos   [Federal University of the State of Rio de Janeiro | beto@uniriotec.br]

 Exact Sciences and Technology Center, Av. Pasteur, 458 - Botafogo, Rio de Janeiro, RJ, 22290-250, Brazil.

Received: 01 March 2024 • Accepted: 20 August 2024 • Published: 23 October 2024

Abstract The usage of mobile devices like drones has been increasing in various fields, ranging from package delivery to emergency services and environmental monitoring. Intelligent services increasingly use the processing power of these devices in conjunction with techniques such as Federated Learning (FL), which allows machine learning to be carried out in a decentralized way using data accessed by clients or devices. However, in normal operations, the data accessed by clients is distributed heterogeneously among themselves, negatively impacting learning results. This article discusses the normalization in Federated Learning local training to mitigate results obtained in heterogeneous distributions. In this context, we propose Federated Learning with Weight Standardization on Convolutional Neural Networks (FedWS) and evaluate it with Batch Normalization, Layer Normalization, and Group Normalization in experiments with heterogeneous data distributions. The experiments demonstrated that FedWS achieved higher accuracy results ranging from 3% to 6% and reduced the computational and communication costs between 25% and 40%, being more suitable for use in devices with computational resource limitations.

Keywords: Federated Learning, Heterogeneous Data, Edge Devices, Data Transformation, Image Classification

1 Introduction

Recent technological advances have allowed the development of services using the capabilities of mobile devices like drones or as part of various activities such as air mobility, ranging from package deliveries to emergency services [Butler *et al.*, 2020]. In turn, the need to use processing power of drones and other mobile devices in an Internet of Things context took advantage of the concept of Mobile Edge Computing, proposing the positioning of servers close to the devices to provide and support the use of intelligent services involving recognition and sensing activities, environmental monitoring, and use of virtual reality [Alsamhi *et al.*, 2022].

Drones have a variety of sensors at their disposal, such as RGB Camera, LiDAR, Hyperspectral sensors, Lightweight cameras, and thermal infra-red sensors that are used to capture data used for visual analysis, terrain mapping, object detection, people and vehicle tracking activities that are commonly carried out in the context of agriculture, industry 4.0, environment, health and emergency, smart cities, natural disaster tracking, and construction. The use of deep learning (DL) techniques in these activities can provide intelligence to the services provided but has limitations regarding latency, energy demands, and privacy [Yazid *et al.*, 2021].

The need to transfer data from the drone to the central server to use DL techniques in image classification can be a risk to data privacy, as the data may contain sensitive information, such as location and personal information. Furthermore, sending many images to the central server will increase transmission bandwidth and energy consumption even before the learning process begins [Alsamhi *et al.*, 2022].

Considering this context, Federated Learning (FL) was presented by McMahan *et al.* [2017] as a decentralized ma-

chine learning approach, which allows clients with access to data and communication with a central server, to perform local learning tasks on their data, sending only the learned parameters to the central server, without the need of data transfer in the learning activity, reducing privacy risks and communication and energy costs, compared with a centralized machine learning approach.

An essential factor for the use of drones and mobile devices in FL are issues related to limitation of resources that these types of devices have about their processing capacity, memory, and battery [Asad *et al.*, 2021]. In this way, issues such as the cost of processing and communication in FL can be decisive for the use of drones in FL safely and satisfactorily.

However, one of the main characteristics of drones is the mobility and the possibility of having contact with a variety of large datasets that generally have a degree of heterogeneity among them. Using FL in heterogeneous data distributions between clients can lead to poor learning accuracy results and low convergence, increasing costs in FL, and the consumption of drones' computational resources [Li *et al.*, 2021].

Zhu *et al.* [2021] highlighted the adverse effects caused by the heterogeneous distribution of data among FL clients as a challenge, highlighting different techniques in a taxonomy that classify them according to the type of activity within FL. Algorithm techniques are notable for enhancing local FL convergence and expediting the process. Amongst them stand out the normalization techniques such as Group Normalization (GN) [Wu and He, 2018] and Batch Normalization (BN) [Ioffe and Szegedy, 2015]. However, there is a need to delve deeper into the study of the use of normalization techniques, considering the impact of data heterogeneity in FL on questions of costs that may impact the use of drones in FL activi-

ties.

Motivated by the need to advance in treating heterogeneous data distributions in FL using drones in image classification activities and considering the factors relevant to using mobile devices highlighted previously, we propose a solution in this work.

In our previous work [Vieira and Campos, 2023], we introduced the Federated Learning Algorithm technique with Weight Standardization on Convolutional Layers (FedWS) which aims to reduce the adverse effects caused by heterogeneous data distributions in FL in the accuracy and communication costs. FedWS uses Weight Standardization [Qiao *et al.*, 2019] applied to convolutional layers of a Convolutional Neural Network (CNN) in the local FL training stage, obtaining improvements in accuracy results and reducing FL costs in heterogeneous data scenarios.

This paper advances the work of [Vieira and Campos, 2023], deepening the understanding of the global effects and impacts on FL caused by heterogeneous data distribution, and how at a local level, other effects also contribute to a scenario of worsening FL results. Furthermore, we discuss in greater detail how the technique proposes to improve local training behavior, reducing the impact of global effects on FL results. To highlight the improvement, new experiments were conducted that expanded the understanding of how convolutional layers behave in situations of heterogeneous distribution, such as classification results reflect the improvements proposed by the technique and how the behavior of losses confirms the expected behavior. Finally, new communication cost results were expanded for comparison with processing costs aiming to show how the behavior of the technique can benefit devices with few computational resources.

As contributions during the work, we can highlight the following items:

- We advance the discussion of how improving local training behavior in the context of data heterogeneity can improve global FL results and how the FedWS proposes to address this situation.
- We show how the EMD (Earth Mover's Distance) metric can portray heterogeneity more realistically, considering not only the absence of classes in clients but also the quantitative imbalance of data, allowing experiments to be carried out on data closer to reality.
- We advanced the discussion of how data heterogeneity affects the convolutional layers of FL clients' neural networks, contributing to the reduction of test accuracy in this context.
- We expand the analysis of the impacts of data heterogeneity on FL tasks in data classification, also considering the differences between training and testing, how the techniques behave with increasing sample EMD, and other evaluation metrics. In this context, FedWS managed to stand out from other techniques, showing results that demonstrate the reduction of the impact of EMD on classification metrics.
- We discuss how the increase in local epochs does not influence the total accuracy result but increases the convergence of normalization techniques.

- Our empirical evidence shows that FedWS offers communication costs that other techniques can only achieve by increasing local epochs, which comes at the expense of increased computational cost. FedWS proves that it can deliver superior results with fewer local times, making it a more viable option for use on devices with limited computing resources.

The rest of this article is organized as follows. Section 2 presents related work. In Section 3, the theoretical foundations involved in the study are described. Section 4 presents the proposed method. Section 5 describes the experiments carried out and analyzes the results obtained. And finally, Section 6 presents the conclusion.

2 Related Work

Federated Learning was presented as a machine learning decentralized approach by McMahan *et al.* [2017] with significant advances in preserving data privacy in training and taking advantage of decentralized computing resources, also called FedAvg, showing significant results even in situations of heterogeneous data distribution.

However, subsequent studies Zhao *et al.* [2018] point out that the federated process in situations of heterogeneity suffers from degradation in test accuracy, mainly attributed to divergence of weights of local model layers caused by the heterogeneous data distribution among FL clients.

According to Zhu *et al.* [2021], one of the ways to mitigate the effects caused by heterogeneous distributions is to focus on techniques that improve local client training, such as normalization techniques such as BN and GN, reducing the impact of non-uniform distributions on DNN weights.

Originally, normalization techniques, most notably BN, were designed for centralized training to address the problem called Internal Covariance Shift (ICS), which is a variation of the inputs of the layers of a DNN that generated outputs that were very far from each other, leading to saturation of the [Ioffe and Szegedy, 2015] neurons. Subsequently, several beneficial effects were identified to stabilize the training of DNNs, such as reducing the dependence on initialization values, greater convergence due to removing outliers, and smoothing training errors. [Lubana *et al.*, 2021]. These features were later explored to improve local FL training behavior when clients use a DNN.

Several studies use this strategy to improve local training on FL using normalization techniques. FedBN [Li *et al.*, 2021] proposed using batch normalization to solve the problem caused by data distribution, but BN has some setbacks when used with FedAvg in this context. Fedavg uses the calculation of the weighted sum of the DNN parameters learned by each client to generate the global model, and BN has extra trained parameters that are calculated based on the data distribution of each client that do not behave well when being aggregated by FedAvg [Li *et al.*, 2022].

Fed2 [Yu *et al.*, 2021] proposed the use of GN in conjunction with feature store, considering heterogeneous data samples with distributions very simple with only the presence and absence of classes in clients without, however, making

a comparison with other normalization techniques. Subsequently, Zhang *et al.* [2021] proposed the replacement of BN by GN in semi-supervised training, and its results showed that BN achieved lower results than GN, which managed to improve the behavior of gradients and, consequently, training accuracy.

Subsequently, FedNorm [Du *et al.*, 2022] carried out a comparative study observing BN, GN, and *Layer Normalization* (LN), which presented the best test accuracy and convergence results. However, the GN results were very close to LN and were not observed in all experiments, not presenting a justification that differentiates the choice of one or the other as the most appropriate technique.

All of these studies were based on treating problems caused by data heterogeneity using normalization techniques to improve the local training behavior of clients in FL. However, it was not observed how these techniques behave considering the communication and processing costs in FL using mobile devices with processing capacity restrictions.

In our work, we present a new technique using a new normalization approach aiming to improve the results of federated training in heterogeneous distributions with different levels of imbalance, considering the communication and processing costs and the capacity and energy constraints of FL clients.

3 Theoretical Foundation

3.1 Federated Learning Process

Federated Learning (FL) is a decentralized machine learning approach whose architecture comprises a set of clients with access to local data and a central server that coordinates and manages the learning process. In this approach, data is not shared with the central server, and between clients; only the parameters learned during training are transmitted [Duan *et al.*, 2023].

Figure 1 shows the training process used by FL, which starts with the central server sending the global model, composed of the neural network parameters (weights and biases), initially with random values, to the clients. Each client then uses the global model to perform a local learning task on the data it has access to, and after a determined number of local epochs, generates the local model that is sent to the central server [Khan *et al.*, 2021].

After the central server receives all local models from all clients, it will aggregate the parameters and update the global model generating a new global model. As proposed by McMahan *et al.* [2017], the most used way to perform this aggregation is called FedAVG and uses a weighted sum of the local model parameters to perform the aggregation.

The stage of sending the global model, local learning by clients, aggregation performed by the central server, and updating the global model is called a round. After each completed round, the central server identifies whether the objective of the federated training reaches a certain number of rounds or whether accuracy has been achieved. Otherwise, another round begins with sending the new global model to clients.

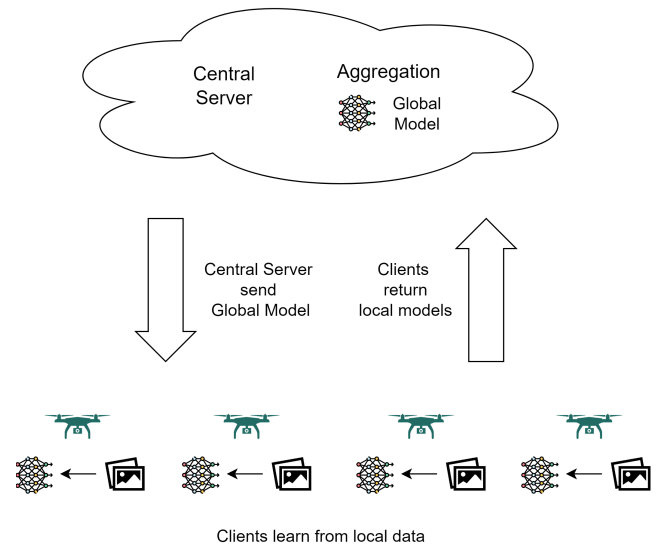


Figure 1. Federated Learning Training

Using FL by drones allows access to data captured by sensors, enriching training with image data obtained in different locations and conditions. However, these devices have peculiarities that must be considered when used as clients in FL, such as battery and processing power limitations, reducing the number of local iterations these clients can perform without compromising their functioning [Asad *et al.*, 2021]. Furthermore, more complex neural network architectures require more computational resources and time to complete the local training task, increasing energy consumption.

3.2 Communication and Computational Costs on Federated Learning

According to Alsamhi *et al.* [2022], one of the great motivators for using FL in drones is its ability to reduce the communication cost of the learning process compared to the decentralized process. Therefore, communication cost is highlighted as one of the determining factors for using machine learning in drones due to its limitation in processing capacity and battery.

The total value of the communication cost W depends on the amount of data transmitted in each transmission, the number of federated training rounds and the number of [Shome *et al.*, 2022] clients, and can be defined as [Liu *et al.*, 2021]:

$$W = 2T(K \cdot \omega^*), \quad (1)$$

where T is the total number of FL rounds, multiplied by 2 representing the send and the return of information, K is the number of clients, and ω^* is the size of weights of the local model.

Based on this definition, the value of W is strongly influenced by the number of training rounds T and the size of ω^* . The convergence capacity of the FL algorithm used to achieve target accuracy is decisive in this aspect, as low convergence can result in many training rounds that lead to an increase in the W cost. In turn, the value of ω^* derives from the complexity of the neural network used in local training. Considering the requirements for using drones previously discussed in Section 2, the neural networks used will

preferably be more straightforward to reduce the consumption of energy resources.

According to Khan *et al.* [2021], another relevant factor for the use of FL considering the energy consumption factor in mobile devices is the computational or processing cost. The local computational cost can be defined as the number of local iterations on clients to perform local training, represented by the number of epochs (E) of the local machine learning task. From this information, we can derive the total computational cost as:

$$C = T \times K \times E, \quad (2)$$

which represents the total number of local iterations, considering K clients executing a fixed number of iterations/epochs (E) and T rounds necessary to achieve a target global accuracy.

According to Xu *et al.* [2022], one of the ways to reduce the cost of communication in FL is to increase the number of local client iterations, with positive results in learning convergence and reducing the number of FL rounds to achieve a specific target accuracy and consequently the communication cost, but increasing the computational cost. However, from the perspective of using FL for mobile devices and drones, the increase in local rounds represents an increase in energy consumption for each client, which may make using FL unfeasible for some types of devices.

3.3 Heterogeneous Data Distributions and its Impacts on Federated Learning

The way the data accessed by FL clients is distributed has a significant impact on the behavior and results obtained by this machine learning strategy. As for the type of distribution, they are generally characterized as IID (Independent and Identically Distributed), when each client's data is distributed uniformly about the total distribution and non-IID (non Independent and Identically Distributed) which represents a distribution heterogeneous data [Rodríguez-Barroso *et al.*, 2023].

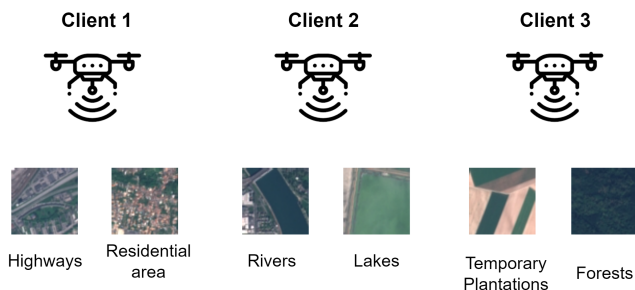


Figure 2. Heterogeneous distribution in federated learning on the EuroSAT dataset

Typically, in real-use scenarios, FL clients are subjected to different environments that increase the possibility that their data is heterogeneous (non-IID) reducing convergence performance and test accuracy, increasing the need for more training rounds to achieve the desired test accuracy and consequently raising communication and processing costs of FL [Wen *et al.*, 2023].

Drones have a great capacity to monitor activities using sensors in heterogeneous environments [Yazid *et al.*, 2021], and there is an excellent possibility that clients have data with different classes. Therefore, this work will be developed on this type of heterogeneity, where each drone has data with an imbalance between its classes.

Figure 2 exemplifies this type of distribution, taking the EuroSAT dataset [Helber *et al.*, 2019] as an example and considering that the drones are in regions with access to different types of terrain. In this case, each drone would only have access to data representing part of the classes in the total data set. The EuroSAT dataset has ten classes representing different types of terrain (permanent and temporary plantations, highways, rivers, forests, pastures, lakes, diverse vegetation, industrial and residential areas).

According to Zhao *et al.* [2018], the presence of heterogeneous data distributions in clients in FL causes a negative effect on the neural network weights called weight divergence, which is defined as the relationship between the weights generated in centralized learning (Stochastic Gradient Descent - SGD) performed on the total data set compared to the weights obtained in FL using the same data set. The greater the heterogeneity of the data, the greater the divergence of weights and the lower the final test accuracy obtained in FL.

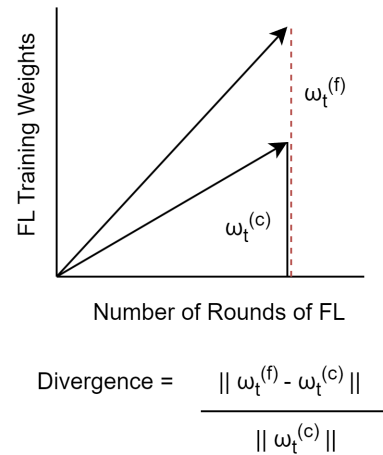


Figure 3. Weight divergence on Federated Learning

Considering the weights $\omega_t^{(c)}$ and $\omega_t^{(f)}$ as resulting respectively from SGD training and the aggregation of client weights in FL, in a heterogeneous data configuration, the divergence will initially be created by the distribution of data from the first round of training. It will subsequently increase cumulatively with each subsequent training round, as illustrated by Figure 3. Both the initial value and the subsequent accumulated value are more significant in distributions with higher data heterogeneity.

To measure the impact of different levels of heterogeneous data distributions with class imbalance, the EMD metric can be used, which is defined as the data distribution ratio per class for each client compared to the total distribution [Zhao *et al.*, 2018]. It is calculated as $\sum_{i=1}^C ||p^{(k)}(y=i) - p(y=i)||$, where $p^{(k)}$ is the data distribution according to classes C for each client k and p is the total distribution.

Distributions with greater heterogeneity present higher

EMD values and, consequently, a greater impact on test accuracy in FL. This may lead to a need to increase training rounds T , and consequently, the communication cost W in FL.

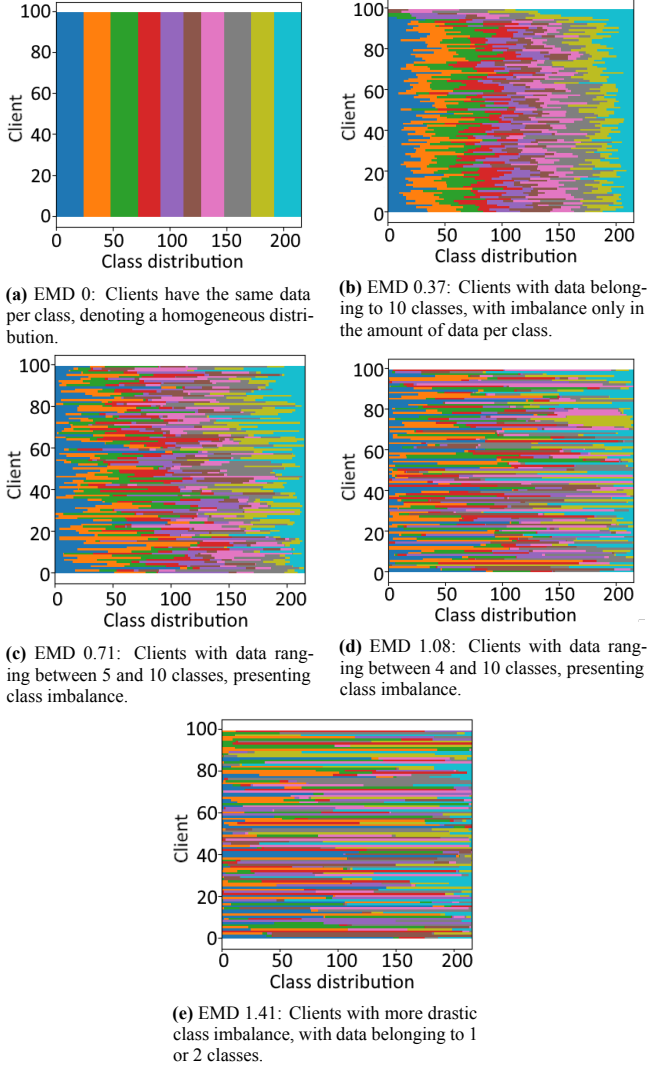


Figure 4. EuroSAT dataset: Data distribution by Classes in 100 clients considering EMD values: 0, 0.37, 0.71, 1.08 and 1.41.

Considering the EuroSAT dataset, we can investigate how the class distribution would look at different EMD levels. Figure 4 exemplifies the distribution in three EMD values: 0, 0.37, 0.71, 1.08, and 1.41, in a scenario with 100 clients with access to local data. When observing the distribution, we notice that the lowest EMD values (0.37) present an imbalance that does not necessarily indicate the absence of a class in a given client but rather the reduction of data that represents this class. Intermediate values (0.71 to 1.08) already show the absence of some classes, while level 1.41 shows a level of data concentration in 1 to 2 classes in the clients involved, denoting a more acute unbalanced scenario.

Finally, Table 1 consolidates the main impacts of heterogeneous distributions in FL and shows us how a local customer problem, their data distribution, and the result of local training generates impacts at a global level in FL and mainly increasing the costs of communication and FL processing, consequently increasing the energy consumption of drones.

Table 1. Summary of the Main Impacts of Heterogeneous Distributions in FL

Root cause	Impacts and Symptoms
Heterogeneous data distributions on FL clients	<ul style="list-style-type: none"> - Reduces convergence of FL training - Reduces FL test accuracy - Increases weight divergence - Increases communication and processing costs

4 Proposed Method

This section presents the proposed method, called *Federated Learning Algorithm with Weight Standardization on Convolutional Layers (FedWS)*.

Given what was exposed in Section 3.3, we can investigate how this problem influences clients' local training to the point of harming the final FL result, aiming to propose an approach that can reduce the impact of heterogeneous distributions in FL when drones are used.

Local training in FL is conventional DL training carried out by clients. Within the context of DL, the heterogeneous data distribution within neural networks causes the ICS effect, which is a change in the parameters of a CNN, generating weight values that are very far apart, generating saturation of neurons and resulting in low convergence of training.

As noted by Santurkar *et al.* [2018], another effect caused by ICS is the increase in losses during centralized learning, damaging both convergence and the final test accuracy. Normalization techniques such as BN can improve the distribution of weights within the layers, reducing and smoothing losses, increasing convergence, and improving the final accuracy result.

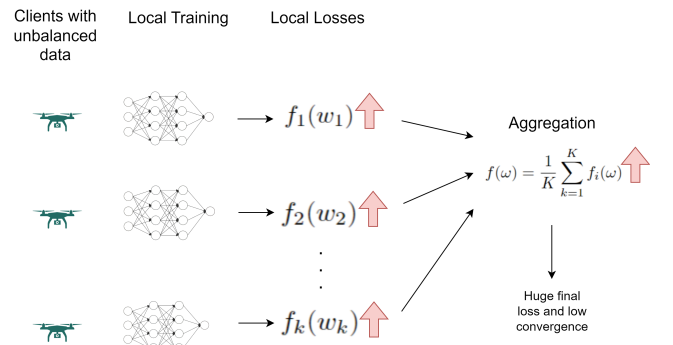


Figure 5. Impact of Global Losses Caused by Heterogeneous Data.

The behavior of the objective function during the process becomes a good indicator of how the learning convergence will be when the data is distributed in a non-uniform way. Considering the objective function for a centralized DL problem $f_i(\omega) = l(x_i, y_i; \omega)$, which calculates the loss value on an example (x_i, y_i) , using the parameters ω , McMahan *et al.* [2017] defined the objective function for FL, considering K clients that calculates global FL losses as:

$$f(\omega) = \frac{1}{K} \sum_{k=1}^K f_i(\omega). \quad (3)$$

For clients with non-uniform distributions, the loss values produced at the end of local training will be aggregated,

generating high aggregated loss values and global weights that are distant from each other, contributing to the increase in FL's accumulated loss and low convergence, as shown in Figure reffig:gbloss. In this way, the impact of local training contributes to the final result, generating the problems listed in Table 1.

The importance of loss behavior is also evidenced by Zhao *et al.* [2018], which identifies that the impact of heterogeneous distributions, measured by EMD, can be mitigated by the way gradients direct losses. In this way, the use of techniques that enable the smoothing of losses in local training tends to generate losses that contribute to reducing customer data's impact on global training results.

One way to move in this direction is to use normalization techniques. In centralized training, the Weight Standardization (WS) technique is a normalization approach whose objective was to achieve good results with reduced batch sizes with the ability to smooth gradients in a CNN, providing losses with smoother behavior than other normalization techniques [Qiao *et al.*, 2019].

In local training in FL, updating the weights during learning is carried out as defined in (4), where ω are the local weights, η is the learning rate, ∇ is the gradient, and $l(w; b)$ is the error of the local training step.

$$\omega \leftarrow \omega - \eta \nabla l(w; b) \quad (4)$$

The FedWS strategy uses the normalization proposed by WS in the local training phase of FL, reducing the local impact of losses generated by the non-uniform distribution of data. This type of normalization consists of acting on the output of the convolutional layer, using the mean (μ^ω) and standard deviation (σ^ω) of the set of weights ω to generate a new normalized set of weights ω^{ws} of the form:

$$\omega^{ws} = \frac{\omega - \mu^\omega}{\sigma^\omega}. \quad (5)$$

Another expected effect is that the impact of weight divergence is reduced due to the normalization of the convolutional layers, producing local weights with less divergence between them ω^{ws} so that the weighted sum produces a model global with a more minor divergence of weights represented by the aggregation:

$$\omega_{t+1}^{ws} = \sum_{k=1}^K \frac{n_k}{n} \omega_{t+1}^{wsk}, \quad (6)$$

, where n is the total set of samples from the dataset, n_k is the set of samples for the client, and ω_{t+1}^{wsk} are the weights that the clients sent from local training.

Figure 6 summarizes the expected effects of using FedWS in FL considering the use of non-uniformed data by clients. The smoothed loss values at the end of each FL round will allow for greater test accuracy due to the reduction of local ICS effects and greater convergence due to standardization. Furthermore, with increased convergence, fewer local and global training iterations will be required, reducing communication and processing costs. In turn, the smoother behavior of losses will make FL results less sensitive to the increase in the EMD of client data distributions, which exist in more heterogeneous distributions.

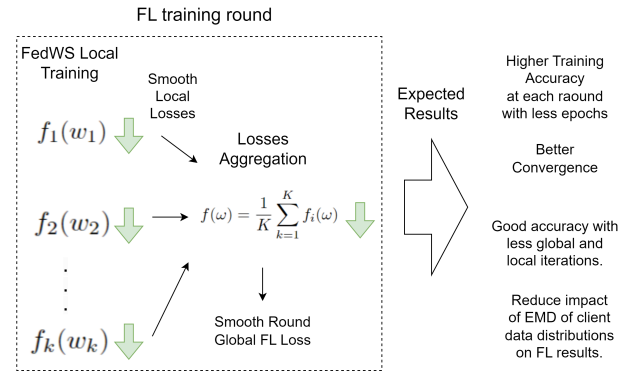


Figure 6. Expected Results of FedWS Training on Heterogeneous Data.

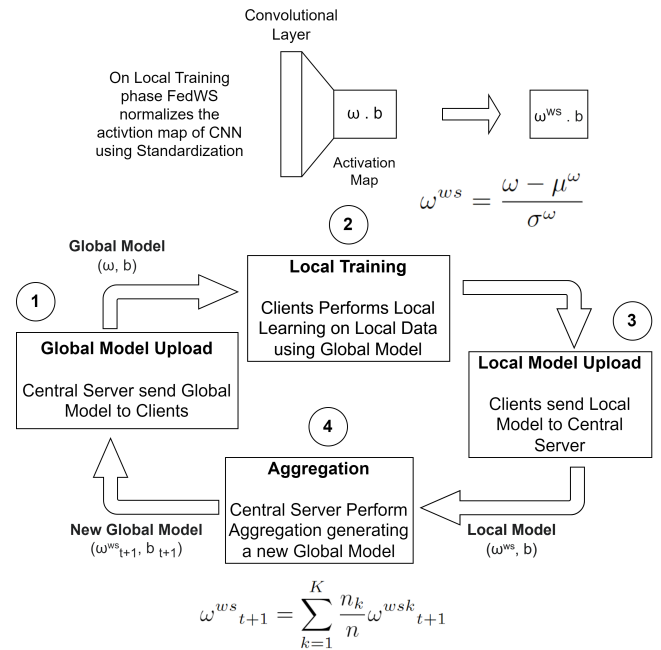


Figure 7. FedWS Schema of Federated Training.

Figure 7, presents a resume of FedWS federated training, FedWS starts the FL process at stage 1, with **Global Model Upload**, which consists of sending the global model, composed of weights and biases for customers participating in the learning. Then, in stage 2, clients perform what is called **Local Training**, where they perform a learning task using neural networks on their data from the global model. At this stage, FedWS normalizes the output of the convolutional layers, more precisely the activation map, using (5) and generating a new normalized activation map. This activation map is used in subsequent neural network layers to produce lower loss values. Upon completion of local training, clients complete stage 3 (**Local Model Upload**), which is the return of local models to the central server.

After receiving all local models, at stage 4 (**Aggregation**), the central server performs local model aggregation using (6) and generates an update of the global model and uses this update to generate a new global model, ending the current round of FL. At the end of the round, if the FL objectives are achieved, federated learning ends; otherwise, the new global model is sent to a new set of data aimed at the continuity of training.

Another critical point is that the application of the normalization shown in (5) is not a local training parameter and,

therefore, does not suffer side effects due to the aggregation performed by FedAvg.

5 Experiments

Aiming to advance the discussion of how normalization techniques can positively affect local training and reduce the impact of heterogeneous distributions in FL, experiments were conducted using the EuroSAT dataset [Causa *et al.*, 2023]. EuroSAT comprises 27,000 images captured via satellite, with dimensions of 64 by 64 pixels divided into ten classes representing different types of terrain (permanent and temporary plantations, highways, rivers, forests, pastures, lakes, diverse vegetation, industrial and residential areas). The choice of this dataset was based on the use of drones to capture and analyze images in different regions, generating heterogeneity, as discussed in Section 3.3.

5.1 Setup and Dataset

The experiments were conducted on a notebook with a Core i5 processor, 16 GB of RAM, and an Nvidia GTX 1650 graphics card with 4 GB of memory and CUDA support. The experiment utilized a modified version of a *framework*¹ to implement and conduct the experiments in a Python 3.8 and Pytorch 1.19 environment.

The EuroSAT dataset was set up with an 80% training and 20% testing split, and the heterogeneous data distribution was achieved through five data arrangements representing EMD values of 0, 0.37, 0.74, 1.08, and 1.41 [Zhao *et al.*, 2018], leading to a class imbalance in client data.

The federated training was configured with 200 rounds (T=200), a learning rate of $\mu = 0.01$, a local training batch size of 50 (B=50), five local training epochs, and ten clients (K=10). The local model was a CNN, as defined by McMahan *et al.* [2017].

The scenarios considered were: No Normalization (NN), where no normalization technique was applied to the model, three others that represent the techniques used by related works: FedBN(BN) [Li *et al.*, 2021], Fed2(GN) [Yu *et al.*, 2021] and FedNorm (LN) [Du *et al.*, 2022] and, finally, a scenario using FedWS, executed on the predicted 5 distinct EMD dataset distributions.

The CNN architecture used for the tests is the Lenet5 LeCun *et al.* [2015], and its parameters are detailed in Table 2, which shows where the normalization layers are positioned within the architecture. This architecture was used considering the limited processing capacity that mobile devices have to execute the local training stage in FL.

5.2 Impact of Heterogeneous distributions on Convolutional Layers of CNN

Considering the use of normalization techniques mentioned previously to mitigate the effects of heterogeneous distributions in FL and that these techniques are widely used in conjunction with CNN architectures, an essential factor is to evaluate the impact of data heterogeneity on the weights

Table 2. Architecture of CNN Used on Experiments (Lenet5).

No.	Layer	Kernel	Shape	Parameters	Activation
01	Input	-	(3 × 24 × 24)	-	-
02	Conv2D	(5 × 5)	(64 × 24 × 24)	4,864	ReLU
03	Normalization	-	-	-	ReLU
04	MaxPooling2D	-	(64 × 12 × 12)	-	-
05	Conv2D	(5 × 5)	(64 × 12 × 12)	102,464	ReLU
06	Normalization	-	-	-	ReLU
07	MaxPooling2D	-	(64 × 6 × 6)	-	-
08	Fully Connected	-	(384)	885,120	ReLU
09	Fully Connected	-	(192)	73,920	ReLU
10	Fully Connected	-	(10)	10	Softmax

Normalization layers are positioned in the 3 and 6 layers of the Lenet5. If normalization is not used these layers are ignored.

of convolutional layers. Figure 8 presents a histogram of the weight distributions of the last convolutional layer of a Lenet5 network [LeCun *et al.*, 2015] used as architecture in local training of FL clients and extracted after aggregation using FedAvg, represented by the number 05 in the Table 2. As we can see, the homogeneous distribution (EMD 0) presents a normalized distribution, and when we observe the distribution with a higher EMD (1.41), the histogram shows a denormalized weight distribution.

When using normalization techniques in conjunction with Lenet5 convolution layers, we observed in Figure 8 that even with values of EMD 1.41, the weights appear to be more homogeneous, close to a normal shape.

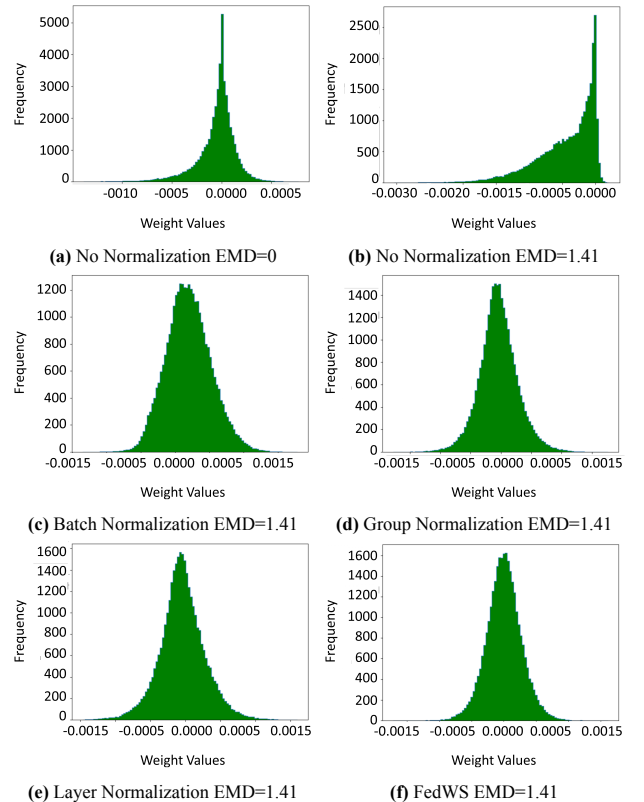


Figure 8. Histogram of Convolutional Layer on Round 1 of FL on EuroSAT Dataset

Considering the use of the proposed technique, it is essential to evaluate how the outputs of the convolution layer behave when subjected to heterogeneous distributions. Figure 8 shows the histogram of the Lenet5 architecture used as lo-

¹<https://github.com/c-gabri/Federated-Learning-PyTorch>

cal training applying WS to the outputs of the convolutional layers. We can see in (Figure 8(f)) that the result obtained by FedWS is a more uniform and normal distribution compared to other normalization techniques, demonstrating the potential of using the FedWS technique.

5.3 Using FL for Image Classification in Heterogeneous Data

Observing the results in Table 3 we can evaluate the results of using normalization techniques in FL for image classification. In this context, it was shown that the FedWS technique achieved results between 3% and 6% higher than other normalization techniques evaluated, observing the results in training accuracy.

Table 3. Federated Learning Accuracy Results using Normalization Techniques on EuroSAT Dataset with Different EMDs

EMD	0	0.37	0.71	1.08	1.41
NN	75.7 (0.2)	74.6 (0.3)	73.3 (2.1)	69.8 (0.8)	64.8 (0.8)
BN	82.4 (0.2)	79.6 (0.7)	79.5 (0.7)	75.0 (0.3)	72.5 (3.9)
GN	81.1 (0.3)	80.4 (0.1)	79.4 (0.9)	74.0 (0.1)	71.3 (1.2)
LN	80.4 (0.2)	79.9 (0.3)	78.6 (1.1)	73.3 (0.3)	70.4 (0.5)
FedWS	85.5 (0.6)	86.0 (0.6)	85.0 (0.8)	80.7 (0.3)	77.0 (0.1)

Standard deviation in parentheses

(a) Percentage Training Accuracy

EMD	0	0.37	0.71	1.08	1.41
NN	77.1 (0.7)	76.2 (0.5)	74.7 (1.7)	71.9 (0.1)	61.7 (1.1)
BN	84.7 (0.3)	82.4 (0.1)	82.0 (1.2)	77.8 (0.3)	69.7 (3.1)
GN	82.0 (0.9)	80.9 (0.3)	80.1 (1.5)	75.0 (0.6)	69.2 (0.8)
LN	81.0 (0.1)	80.9 (0.2)	79.7 (1.9)	74.0 (0.2)	68.8 (0.1)
FedWS	84.4 (2.9)	84.5 (0.9)	84.3 (2.1)	80.3 (0.9)	76.9 (0.1)

Standard deviation in parentheses

(b) Percentage Test Accuracy

Furthermore, observing the results considering an increase in the heterogeneity of the data, represented by the increase in the EMD in the samples used, it was observed that the proposed technique obtained a more linear result than the others, being less affected by the negative effects of this type of distribution in the accuracy results.

However, the results obtained in test accuracy must also be considered, as they reflect how the federated global model behaves in more real test results. In this context, the results in Table 3 present a very close result between using BN and FedWS when the data is homogeneous. However, as the EMD value increases, FedWS presents test accuracies between 2% and 7% higher than the best results of other techniques.

Considering the adverse effects that increased data heterogeneity causes on accuracy results, the increase in EMD is expected to result in lower accuracies compared to the homogeneous scenario (EMD=0) [Zhao *et al.*, 2018]. One way to measure this impact is to calculate the difference between the test accuracy results with EMD equal to 0 and the result obtained by each heterogeneous EMD sampling.

Figure 9 shows that the observed normalization techniques have a smaller impact in magnitude when compared to not using normalization (NN). However, they present similar behavior considering the behavior. In contrast, FedWS

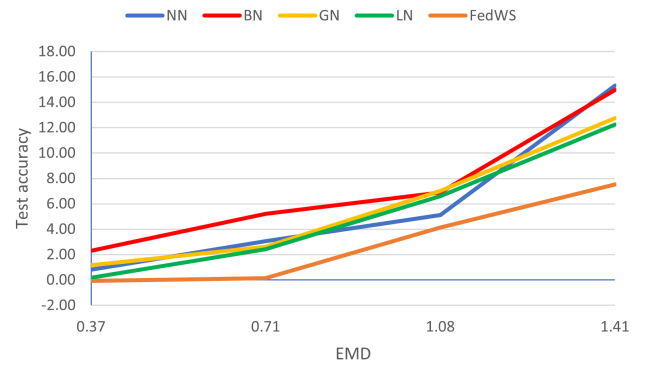


Figure 9. Impact of increased heterogeneity on test accuracy for each observed technique

presents results that show that the technique is less affected by samples with larger EMD and presents more linear results. Considering EMD 1.41, which represents a greater degree of heterogeneity, the impact on FedWS accuracy was close to 7%, while the others had an impact between 12% and 15%. As shown in Figure 4 EMD 1.41 represents a type of sample where clients have images belonging to only between 1 and 2 classes for the EuroSAT Dataset, showing that the technique presents good results even in situations with greater heterogeneity. In distributions with EMD 0.37 and 0.71, FedWS was practically not affected by the increase in EMD, considering that at these levels of heterogeneity, they do not present a very large unbalance, as shown in Figure 4.

Table 4. Federated Learning Precision, Recall and F1-Score Results on Normalization Techniques on EuroSAT Dataset with Different EMDs

Technique	Metric	0	0.37	0.71	1.08	1.41
NN	precision	0.771	0.762	0.747	0.759	0.667
	recall	0.771	0.762	0.747	0.719	0.617
	f1-score	0.761	0.762	0.737	0.699	0.597
BN	precision	0.847	0.834	0.820	0.818	0.747
	recall	0.847	0.824	0.820	0.778	0.697
	f1-score	0.847	0.824	0.820	0.758	0.677
GN	precision	0.820	0.809	0.801	0.770	0.712
	recall	0.820	0.809	0.691	0.750	0.692
	f1-score	0.820	0.809	0.671	0.740	0.682
LN	precision	0.810	0.809	0.807	0.770	0.708
	recall	0.810	0.809	0.797	0.740	0.688
	f1-score	0.810	0.809	0.797	0.730	0.678
FedWS	precision	0.844	0.855	0.853	0.813	0.779
	recall	0.844	0.845	0.843	0.803	0.769
	f1-score	0.844	0.845	0.843	0.793	0.769

To evaluate the impact of the diversity of data distributions, other relevant metrics that can be observed are precision, recall, and f1-score. Precision measures the proportion of true positive predictions among all positive predictions, informing which positive results really are positive. Recall measures the proportion of true positive predictions among all true positive instances, telling us about the data points that should be predicted as true and how many we correctly predicted as true. F1-score presents the relationship between precision and recall.

Table 4 shows the results of the weighted average score of the precision, recall, and f1 score metrics in FL on the Eu-

roSAT dataset of the observed techniques, considering different EMDs. These values were calculated as the weighted average of the values obtained by the ten classes of the EuroSAT dataset in the image classification experiment. Considering the results shown, if compared with Table 3b, the techniques presented uniform results consistent with the test accuracy at EMDs 0 and 0.37. From EMD 0.71 onwards, the NN, BN, GN, and LN techniques began to present recall values lower than precision. Looking more directly at the f1-score results, we observe that NN, BN, GN, and LN have a lower test accuracy than the f1-score results, indicating that these techniques generate models that possibly cannot be effective for inference but classes fewer representatives, considering a scenario with class imbalance.

In this context, FedWS presented f1-score results superior to other techniques by between 2% to 9% considering the best results of all techniques and did not present such a large difference between precision and recall.

Table 5. F1-Score Results Considering the Three Classes with the Worst Results on EuroSAT with EMD 0.71, 1.08 and 1.41.

EMD	NN	BN	GN	LN	FedWS
0.71	0.470	0.650	0.601	0.587	0.723
	0.597	0.720	0.681	0.677	0.733
	0.637	0.740	0.711	0.707	0.803
1.08	0.469	0.448	0.620	0.550	0.608
	0.519	0.678	0.600	0.560	0.718
	0.679	0.678	0.680	0.680	0.738
1.41	0.227	0.327	0.462	0.392	0.589
	0.447	0.347	0.482	0.482	0.589
	0.447	0.537	0.522	0.562	0.649

However, we would also like to evaluate how the techniques behaved in the worst cases, considering the results by class. Therefore, Table 5 presents the f1-score values for the three classes with the worst results for all techniques considering the most critical EMD values (0.71, 1.08, and 1.41) observed in Table 4. The results confirm that FedWS presents higher values even considering the worst f1-score results in an unbalanced data scenario.

Another point to be analyzed is the comparison between training and testing accuracy results, which can indicate overfitting tendencies when the neural network obtained a training result superior to the testing result, indicating that the generated model may not present a generalization aimed at using in real scenarios. On the contrary, underfitting occurs when the testing accuracy is higher than the training accuracy, the neural network may need to learn more from the training data.

Figure 10 shows that the observed techniques generally presented an underfitting situation with a tendency to increase as the heterogeneity of the data samples increases. The difference values varied between -3% to 1.5%, indicating that, in general, the difference was not so significant as to cause any problem. Nevertheless, it is worth highlighting the behavior of some of the techniques: BN showed a more linear behavior with a higher underfitting value compared to the other techniques, LN showed a linear underfitting behavior that grew as the EMD value increased, and FedWS showed overfitting in the samples less heterogeneous and becoming almost imperceptible underfitting in the sample with EMD

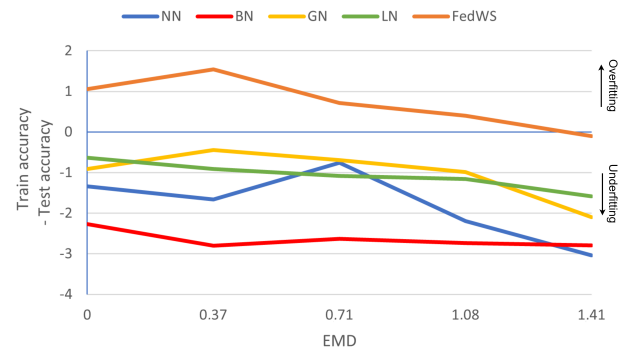


Figure 10. Percentage difference between training and testing results of observed normalization techniques

1.41.

According to Zhao *et al.* [2018], the number of local epochs influences EMD's impact on federated training in heterogeneous data distributions. Therefore, it is interesting to check the effect of increasing the number of local epochs (E) on test accuracy. Table 6 shows that all techniques increased accuracy, varying between 1.1% to 6.7%. Considering that the local epochs were doubled, the gain in accuracy was not very significant.

Table 6. Test accuracy percentage in the EuroSAT Dataset using ten local epochs, showing the comparative gain with five epochs

EMD	0	0,37	0,71	1,08	1,41
NN	81.9 (4.9)	80.9 (4.7)	80.0 (5.2)	75.2 (3.3)	69.7 (4.9)
BN	86.0 (1.3)	84.3 (1.9)	84.8 (2.7)	79.9 (2.1)	74.4 (1.9)
GN	85.0 (3.0)	83.7 (2.9)	83.3 (3.2)	77.9 (2.9)	75.4 (4.0)
LN	84.4 (3.3)	83.8 (2.9)	83.0 (3.3)	77.5 (3.1)	74.2 (3.8)
FedWS	87.7 (3.3)	87.1 (2.6)	86.7 (2.5)	83.5 (3.2)	79.6 (2.5)

Comparative with test accuracy between parentheses. Positive values indicate that the execution with ten epochs obtained greater test accuracy

The convergence of accuracy indicates how long the learning process took to reach a sure accuracy. This is another characteristic that can be observed when evaluating the behavior of normalization techniques in heterogeneous data distributions. Figure 11 shows the test accuracy obtained by FL techniques on the EuroSAT Dataset with EMD 0, 0.71, and 1.41, considering 5 and 10 epochs of local training at clients. Considering the ability of the techniques to reach a level of 75% accuracy, the results for five epochs of local training show that FedWS managed to reach this accuracy using 50, 60, and 170 rounds, while the other techniques required 80 and 100 rounds, and for EMD 1.41 were unable to reach this level. For ten local rounds, we observed that the increase in local epochs increased the convergence of the techniques and reduced the number of rounds to 75% at 30, 30, and 70 for FedWS and 50, 60, and 200 for the other techniques. These results indicate that the increase in local epochs, despite not significantly affecting the final result as observed in Table 6, has a more beneficial effect on the convergence of techniques.

According to Section 3.1, the divergence of client weights increases with each round of federated training, contributing to the reduction of training and testing accuracy. However, the impact of the EMD term on these results is influenced by the errors presented during the federated training. Analyzing the average of errors detected by the techniques, shown in Ta-

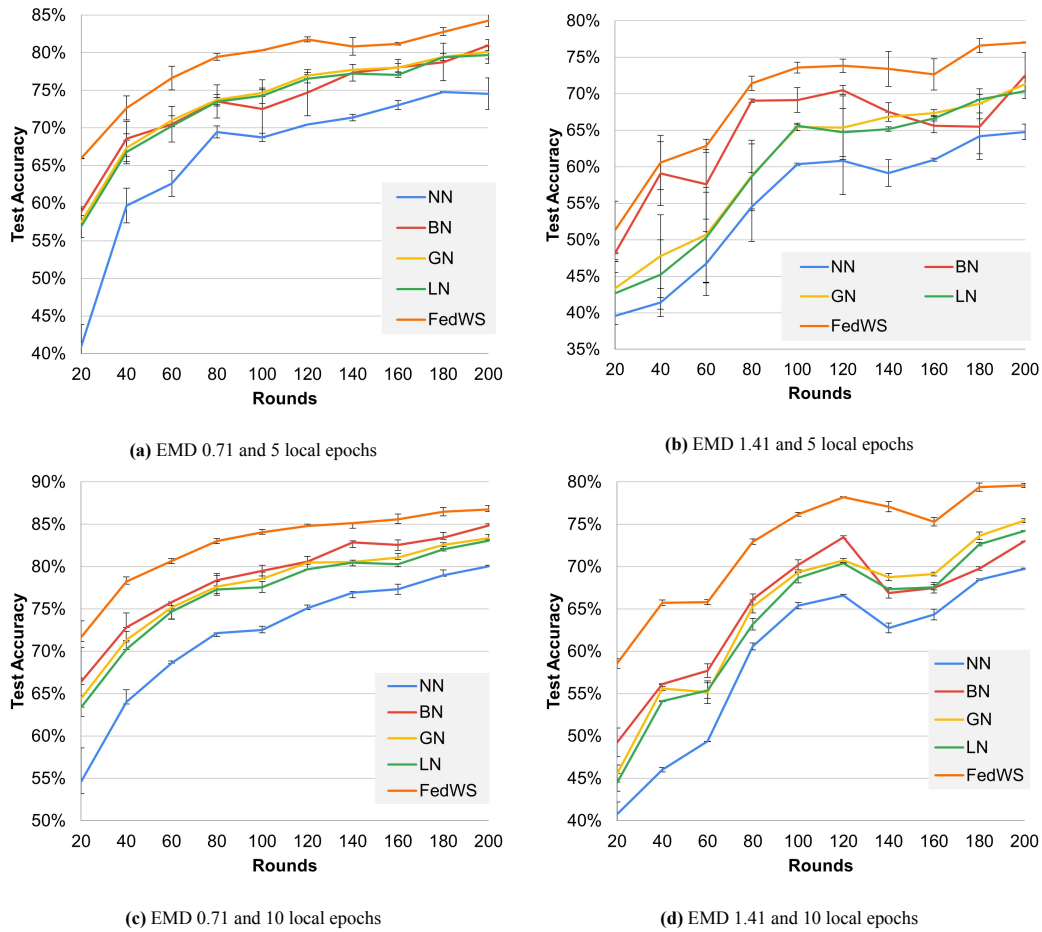


Figure 11. Convergence analysis in classification task on EuroSAT

Table 7. Average losses of techniques on different EMD distributions.

EMD	0	0,37	0,71	1,08	1,41
NN	1.05 (0.24)	1.01 (0.22)	0.88 (0.21)	0.68 (0.19)	0.44 (0.10)
BN	0.86 (0.20)	0.80 (0.18)	0.70 (0.17)	0.53 (0.15)	0.33 (0.08)
GN	0.85 (0.22)	0.81 (0.20)	0.69 (0.19)	0.55 (0.16)	0.36 (0.10)
LN	0.88 (0.22)	0.83 (0.20)	0.73 (0.19)	0.56 (0.17)	0.37 (0.10)
FedWS	0.54 (0.22)	0.47 (0.21)	0.43 (0.19)	0.36 (0.15)	0.23 (0.07)

Standard deviation between parentheses.

In table 7 we observed that FedWS has the lowest averages and standard deviation even with the increase in data heterogeneity, represented by the increase in EMD. Considering that the client weights in local training are also affected by errors as seen in (4), the results of the errors obtained by FedWS prove to be decisive so that the technique is less affected by the increase in data heterogeneity, as indicate that the technique can produce weight updates more smoothly, avoiding jumps in weight values, in addition to alleviating the impact of EMD on weight divergence.

5.4 Federated Learning Communication and Computational Costs

Communication cost is significant in the FL training process, indicating when a technique is more appropriate or consumes less training time to achieve a specific objective, avoiding saving communication and computational resources. Aiming to compare how the techniques behave in terms of com-

munication cost, we use the definition presented in (1) to calculate the communication cost in megabytes (MB) using the number of rounds (T), the number of clients (K) and the size of the local model weights (ω).

Based on the architecture used in Lenet5 training, shown in Table 2, the size of the local model, measured via experiment, is 4.27 MB for all techniques observed. This shows that normalization techniques do not impact the size of the generated model. Considering that we have ten clients per round $K = 10$, we can apply the values of K and ω to the definition of (1), resulting in $W = 2T(42.7)$. This shows that since the techniques did not modify the size of the local model, the communication cost will be defined by the number of rounds needed to achieve sure accuracy.

Table 8 shows the results of the communication cost obtained by the techniques aiming to reach 75% in federated training. Using five local epochs, FedWS presented lower costs between 25% and 40% compared to the best results of other techniques. Furthermore, considering the increase in cost concerning the homogeneous distribution scenario (EMD = 0), it presented the exact cost for EMD = 0.71 and a cost three times higher in the scenario with EMD 1.41, in which the others were not even able to achieve the desired accuracy. Considering that EMD 0.71 has a medium unbalance of classes in clients (5 to 10). EMD 1.41 has a more acute unbalance (1 to 2 classes), which shows that the technique's cost is little affected in lighter unbalances. In more acute ones, despite achieving accuracies not achieved by others, it

significantly impacts cost more significantly.

When analyzing the impact of increasing local epochs in training, from 5 to 10, we observed that previous results showed a relative gain in accuracy, as shown in Table 6, but a substantial gain in convergence, as shown in Figure 11. In Table 8, we see that the increase in local epochs resulted in a reduction in communication cost between 25% and 50% for all techniques. Looking at these results alone, increasing local epochs can be used to reduce communication costs. However, the processing cost must be taken into account.

Table 8. Communication Cost (MB) for Normalization Techniques to Achieve 75% Test Accuracy

Local Epochs	EMD	NN	BN	GN	LN	FedWS
5	0	13,665	6,832	8,540	8,540	5,124
	0.71	15,372	10,248	8,540	8,540	5,124
	1.41	-	-	-	-	15,372
10	0	8,540	5,124	5,124	5,124	3,416
	0.71	10,248	5,124	5,124	5,124	3,416
	1.41	-	-	17,080	-	8,540

As defined by (2), the processing cost represents the total number of iterations required for all clients to execute a FL task to achieve the desired accuracy. Using this definition, Table 9 compares FL techniques' performance comprehensively. It shows the influence of increasing local epochs on the result of total iterations, aiming to achieve 75% test accuracy, for each technique in data distributions with EMD 0, 0.71 and 1.41. The results demonstrate that in the scenario with 5 local epochs, FedWS required 25% fewer iterations to reach the target accuracy in a homogeneous dataset (EMD=0) and 40% fewer iterations for a dataset with an EMD of 0.71. For the most acute case of data heterogeneity observed (1.41), only FedWS managed to achieve 75% accuracy, making it the best approach in such scenarios.

Table 9. Computational Cost (total iterations) to Achieve 75% of Test Accuracy

Local Epochs	EMD	NN	BN	GN	LN	FedWS
5	0	6,000	4,000	5,000	5,000	3,000
	0.71	9,000	6,000	5,000	6,000	3,000
	1.41	-	-	-	-	9,000
10	0	10,000	6,000	6,000	6,000	4,000
	0.71	12,000	6,000	6,000	6,000	4,000
	1.41	-	-	20,000	-	10,000

Considering a scenario with 10 local epochs and an EMD of 0.71, normalization techniques increased the processing cost between 20% to 25%. Compared to the reduction in communication cost observed in Table 8 between 25% to 50% in this context, the increase can be beneficial for most techniques aiming to obtain greater accuracy and reduce the cost of communication. However, the processing cost was measured considering the training process for all clients involved. If we consider each client individually, the increase in processing cost is 100% as the times increased from 5 to 10. In a scenario where clients have battery and processing capacity restrictions, this increase may not be feasible. In this case, FedWS can be an excellent alternative, as the reduction in communication costs obtained by other techniques when increasing local epochs was 20% to 50%, while FedWS us-

ing 5 local epochs achieved results of 25% and 40% the rest. This can be seen in Table 8 where the cost of FedWS for 5 local epochs was 5,124 MB for EMD 0 and 0.71 while the best technique using 10 local epochs reached the same cost. In other words, FedWS can be used instead of other techniques for use on devices that can use small local epochs due to capacity limitations.

6 Conclusion

In this paper, we advance the discussion of how data heterogeneity can negatively affect FL, considering its use in conjunction with CNN for image classification activities. Using EMD to define samples with heterogeneity made it possible to identify that samples with slight quantitative imbalances between classes generate less impact. In contrast, cases in which a lack of data representativeness in the classes affect accuracy and convergence more acutely. The perception of the impact of heterogeneity in the convolutional layers to this degree is apparent, as shown in the histograms, which also demonstrate how normalization techniques can mitigate this effect. The test and training accuracy results confirmed that FedWS achieved test accuracy results between 3% and 6% higher than other techniques, in addition to suffering less impact compared to the homogeneous distribution result. Considering the increase in local epochs, we observed that their increase was little beneficial to all the techniques observed, considering the percentage increase in epochs and the test accuracy result. However, when observing the convergence of techniques, there was a reduction of 75% to 100% in the number of rounds needed to achieve the desired objective. Delving deeper into the issue of communication and computational costs, the results show that FedWS reduced the communication cost between 25% and 40% compared to other techniques using five local training epochs. When we increased the local training epochs to 10, there was a reduction in the communication cost between 20% to 50% for all techniques, but this increased the computational cost between 20% to 25%. When comparing the results, we observed that FedWS achieves communication costs that the others only achieved with an increase in local epochs and, consequently, in the computational cost. In this scenario, FedWS becomes more appropriate for devices with processing and battery resource constraints, like drones, which are unsuitable for longer time periods.

Declarations

Authors' Contributions

Both authors contributed to writing and editing the text. Flavio was responsible for conceiving the work, developing the methodology, implementing the algorithms, conducting the experiments, and analyzing the results. Carlos contributed to the conception, provided supervision, and assisted in analyzing the results. Both authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets generated and/or analyzed during the current study and the source codes of FedWS are available at <https://github.com/flaviovieiraunirio/fedws>.

References

- Alsamhi, S. H., Shvetsov, A. V., Kumar, S., Hassan, J., Alhartomi, M. A., Shvetsova, S. V., Sahal, R., and Hawbani, A. (2022). Computing in the sky: A survey on intelligent ubiquitous computing for uav-assisted 6g networks and industry 4.0/5.0. *Drones*, 6(7):177. DOI: 10.3390/drones6070177.
- Asad, M., Moustafa, A., Ito, T., and Aslam, M. (2021). Evaluating the communication efficiency in federated learning algorithms. In *24th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 552–557. DOI: 10.1109/CSCWD49262.2021.9437738.
- Butler, L., Yigitcanlar, T., and Paz, A. (2020). Smart urban mobility innovations: A comprehensive review and evaluation. *IEEE ACCESS*, 8:196034–196049. DOI: 10.1109/ACCESS.2020.3034596.
- Causa, F., Franzone, A., and Fasano, G. (2023). Strategic and tactical path planning for urban air mobility: Overview and application to real-world use cases. *Drones*, 7(1):11. DOI: 10.3390/drones7010011.
- Du, Z., Sun, J., Li, A., Chen, P.-Y., Zhang, J., Li, H. H., and Chen, Y. (2022). Rethinking normalization methods in federated learning. In *3rd International Workshop on Distributed Machine Learning*, pages 16–22. DOI: 10.1145/3565010.3569062.
- Duan, Q., Huang, J., Hu, S., Deng, R., Lu, Z., and Yu, S. (2023). Combining federated learning and edge computing toward ubiquitous intelligence in 6g network: Challenges, recent advances, and future directions. *IEEE Communications Surveys & Tutorials*. DOI: 10.1109/COMST.2023.3316615.
- Helber, P., Bischke, B., Dengel, A., and Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226. DOI: 10.1109/JS-TARS.2019.2918242.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456. Available at: https://asvk.cs.msu.ru/~sveta/%D1%80%D0%B5%D1%84%D0%B5%D1%80%D0%B0%D1%82/batch_normalization.pdf.
- Khan, L. U., Saad, W., Han, Z., Hossain, E., and Hong, C. S. (2021). Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 23(3):1759–1799. DOI: 10.1109/COMST.2021.3090430.
- LeCun, Y. *et al.* (2015). Lenet-5, convolutional neural networks. URL: <http://yann.lecun.com/exdb/lenet>, 20(5):14. Available at: <https://yann.lecun.com/exdb/lenet/>.
- Li, Q., Diao, Y., Chen, Q., and He, B. (2022). Federated learning on non-iid data silos: An experimental study. In *38th IEEE International Conference on Data Engineering (ICDE)*, pages 965–978. DOI: 10.1109/ICDE53745.2022.00077.
- Li, X., Jiang, M., Zhang, X., Kamp, M., and Dou, Q. (2021). Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*. DOI: 10.48550/arXiv.2102.07623.
- Liu, S., Yu, J., Deng, X., and Wan, S. (2021). Fedcpf: An efficient-communication federated learning approach for vehicular edge computing in 6g communication networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):1616–1629. DOI: 10.1109/TITS.2021.3099368.
- Lubana, E. S., Dick, R., and Tanaka, H. (2021). Beyond batchnorm: towards a unified understanding of normalization in deep learning. *Advances in Neural Information Processing Systems*, 34:4778–4791. DOI: 10.48550/arXiv.2106.05956.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. DOI: 10.48550/arXiv.1602.05629.
- Qiao, S., Wang, H., Liu, C., Shen, W., and Yuille, A. (2019). Micro-batch training with batch-channel normalization and weight standardization. *arXiv preprint arXiv:1903.10520*. DOI: 10.48550/arXiv.1903.10520.
- Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M. V., Herrera, F., and Martínez-Cámara, E. (2023). Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90:148–173. DOI: 10.1016/j.inffus.2022.09.011.
- Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? *Advances in neural information processing systems*, 31. DOI: 10.48550/arXiv.1805.11604.
- Shome, D., Waqar, O., and Khan, W. U. (2022). Federated learning and next generation wireless communications: A survey on bidirectional relationship. *Transactions on Emerging Telecommunications Technologies*, 33(7):e4458. DOI: 10.1002/ett.4458.
- Vieira, F. and Campos, C. A. V. (2023). Fedws: Uma nova abordagem para aprendizado federado usando dados heterogêneos. In *Anais do XXII Workshop em Desempenho de Sistemas Computacionais e de Comunicação*, pages 1–12. SBC. DOI: 10.5753/wperformance.2023.230814.
- Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., and Zhang, W. (2023). A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14(2):513–535. DOI: 10.1007/s13042-022-01647-y.
- Wu, Y. and He, K. (2018). Group normalization. In

- European Conference on Computer Vision (ECCV)*, pages 3–19. Available at: https://openaccess.thecvf.com/content_ECCV_2018/html/Yuxin_Wu_Group_Normalization_ECCV_2018_paper.html.
- Xu, Y., Liao, Y., Xu, H., Ma, Z., Wang, L., and Liu, J. (2022). Adaptive control of local updating and model compression for efficient federated learning. *IEEE Transactions on Mobile Computing*. DOI: 10.1109/TMC.2022.3186936.
- Yazid, Y., Ez-Zazi, I., Guerrero-Gonzalez, A., El Oualkadi, A., and Arioua, M. (2021). Uav-enabled mobile edge-computing for iot based on ai: A comprehensive review. *Drones*, 5(4):148. DOI: 10.3390/drones5040148.
- Yu, F., Zhang, W., Qin, Z., Xu, Z., Wang, D., Liu, C., Tian, Z., and Chen, X. (2021). Fed2: Feature-aligned federated learning. In *27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2066–2074. DOI: 10.1145/3447548.3467309.
- Zhang, Z., Yang, Y., Yao, Z., Yan, Y., Gonzalez, J. E., Ramchandran, K., and Mahoney, M. W. (2021). Improving semi-supervised federated learning by reducing the gradient diversity of models. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1214–1225. IEEE. DOI: 10.1109/BigData52589.2021.9671693.
- Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*. DOI: 10.48550/arXiv.1806.00582.
- Zhu, H., Xu, J., Liu, S., and Jin, Y. (2021). Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390. DOI: 10.1016/j.neucom.2021.07.098.