







# Data sharing-based approach for Federated Learning tasks on Edge Devices

Renan R. de Oliveira   [ Institute of Informatics (INF) — Federal University of Goiás (UFG) and Federal Institute of Goiás (IFG) | [renan.rodrigues@ifg.edu.br](mailto:renan.rodrigues@ifg.edu.br) ]  
Leandro A. Freitas  [ Federal Institute of Goiás (IFG) | [leandro.freitas@ifg.edu.br](mailto:leandro.freitas@ifg.edu.br) ]  
Waldir Moreira  [ Fraunhofer Portugal AICOS | [waldir.junior@fraunhofer.pt](mailto:waldir.junior@fraunhofer.pt) ]  
Maria Ribeiro  [ Institute for Systems and Computer Engineering, Technology and Science (INESC-TEC) | [maria.r.ribeiro@inesctec.pt](mailto:maria.r.ribeiro@inesctec.pt) ]  
Antonio Oliveira-Jr  [ Institute of Informatics (INF) — Federal University of Goiás (UFG) and Fraunhofer Portugal AICOS | [antoniojr@ufg.br](mailto:antoniojr@ufg.br) ]

 Institute of Informatics (INF) — Federal University of Goiás (UFG), Campus Samambaia, Alameda Palmeiras, s/n - Chácara Califórnia, Goiânia, GO, 74690-900, Brazil.

Received: 16 September 2023 • Accepted: 12 March 2025 • Published: 19 May 2025

**Abstract** Federated Learning (FL) enables edge devices to collaboratively train a global machine learning model. In this paradigm, the data is maintained on the devices themselves and a server is responsible for aggregating the parameters of the local models. However, the aggregated model may present convergence difficulties when the device data are non-independent and identically distributed (non-IID), that is, when they present a heterogeneous distribution. This work proposes an algorithm that extends data sharing-based solutions from the literature by considering privacy-flexible environment, where users agree to share a small percentage of their private, and privacy-sensitive environment, where it is assumed that the aggregator server has a set of public global data that is shared with users in the initial phase of the FL process. The proposed algorithm is evaluated in a distributed and centralized way considering a Human Activity Recognition (HAR) application. The results show that data sharing strategies indicate improved global model performance in non-IID scenarios.

**Keywords:** Federated Learning, non-IID data, distributed datasets, privacy-flexible environment, privacy-sensitive environment, edge devices, training and convergence.

## 1 Introduction

The evolution of mobile devices has enabled Machine Learning (ML) models to run on heterogeneous edge devices and with limited resources. However, when it comes to training, one assumption is that these models should be centrally trained in the cloud, using aggregated training data from multiple users [Beutel *et al.*, 2020].

Training ML models on edge device has attracted increasing interest in recent years due to the need to process a large amount of private data, continuously generated through devices such as smartphones, wearable devices and autonomous vehicles. In this context, Federated Learning (FL) [McMahan *et al.*, 2016] was introduced as a decentralized ML approach that allows users mobile devices (or distributed edge devices) to collaboratively train a shared model keeping private data on the device. In this approach, only the parameters of locally trained models are shared with the aggregator server.

FL has some advantages in wireless communications. According to Yang *et al.* [2022], to send the ML model parameters instead of the training data can save energy, wireless network resources and communication latency. Furthermore, FL contributes to preserving data privacy as the training data remains on the devices. Furthermore, edge computing offers an effective approach

to providing the computing resources required by FL in close proximity to data sources, which allows computing and networking resources dispersed across multiple devices to be fully utilized [Duan *et al.*, 2023]. That way, edge computing has been widely deployed in recent years as a strategy to reduce costly data transfer by bringing computation closer to data sources than conventional cloud computing [Wu *et al.*, 2024].

In traditional ML, the central server needs to access all datasets from devices. In this way, the server can obtain a statistical sampling that represents all the variability of the population in a balanced way, making the training data distribution independent and identically distributed (IID) [Lim *et al.*, 2020]. However, this approach is impractical for the original FL proposal as the local dataset is only accessible by data owners. Furthermore, customer datasets are typically heterogeneous and vary in size, that is, the datasets are distributed in a non-IID way [Sirohi *et al.*, 2023].

According to Brecko *et al.* [2022], data from mobile devices tends to be based on the behavior pattern of a specific user and therefore in some situations it may not be able to represent the pattern of all users. In cases where devices have non-IID data, the FL aggregate model may have difficulty in convergence, causing low performance of the global model [Asad *et al.*, 2020].

Approaches to address the non-IID data challenge

in FL are categorized into four main types [Ma *et al.*, 2022]: data-based, model-based, algorithmic, and framework-driven strategies. Data-based approaches primarily aim to equalize local data distributions across clients. These strategies include techniques like data sharing, where clients exchange specific data points to balance distributions, and data selection, which involves curating data subsets to minimize statistical disparities. Model-based approaches focus on adaptive updating and aggregation of the model to adjust for data variations. Within the scope of algorithms, techniques are used that aim to adapt the model to the individual characteristics of each client. Finally, framework-based strategies include knowledge grouping and distillation methods, facilitating cooperation between heterogeneous models. According to Nakayama and Jenő [2022], there are several security measures in place to enable FL collaboration without forcing participants to trust each other. For example, Li *et al.* [2021] present cryptographic methods and differential privacy as the main approaches adopted in FL for data protection.

In Zhao *et al.* [2022] it is demonstrated that the reduction in FL accuracy on non-IID data can be explained by the divergence of weights of the models trained on the devices. As a solution, the authors propose the creation of a subset of data that is shared globally among edge devices. Although this study demonstrated the feasibility of data sharing in FL, this approach has attracted only limited research interest because such data sharing could lead to violations of data privacy requirements. In Seo *et al.* [2023], the authors argue that sharing datasets may be an economic issue rather than a privacy issue if only a very small amount of data is shared and the resulting benefits in performance and communication costs are substantial. Then, the authors propose a marginal-shared-data approach, where the edge node is responsible for sharing data with the devices. In identical label set, it creates a dataset with labels identical to the global set. In extended label set, it provides a set with additional labels, allowing devices to train with extra classes.

The paper by Shao *et al.* [2024] presents a systematic review on what to share in FL, focusing on model utility, privacy leakage, and communication efficiency. The authors categorize FL approaches into three main types of sharing: models, synthetic data, and knowledge. The paper discusses the generation and sharing of synthetic data in FL to address data heterogeneity among clients and improve training efficiency. According to the authors, instead of sharing real data, which may threaten privacy, synthetic data is used as a secure alternative.

In this context, our work proposes a FL algorithm that considers data sharing as a strategy to address the challenge of non-IID data for privacy-flexible and privacy-sensitive environments in edge devices<sup>1</sup>. Our work differs from Zhao *et al.* [2022] by considering data sharing as a strategy beyond privacy-sensitive environments, where convincing users to share part of their data is challenging. Furthermore, our work differs from Seo *et al.* [2023] in that instead of always considering that the shared set must be provided by an edge node, we consider for the sensitive environment in terms of

privacy that the shared data is generated by the aggregator server and shared with the devices only in the initialization phase of the FL process. Our work proposes an algorithm that considers two strategies: (i) considers a privacy-flexible environment, where users agree to share a small percentage of their private data to build a Global Shared Dataset (GSD); (ii) considers a privacy-sensitive environment in terms of privacy, assuming the existence of a Public Global Share Dataset (PGSD) that characterizes the global dataset. In this case, as pointed out by Shao *et al.* [2024], PGSD data must be generated reflecting users' behavior using generative models or simulation techniques.

Our experiments combine the proposed algorithm considering different distributions of non-IID data in the context of a Human Activity Recognition (HAR) application. HAR applications have been driven by the variety of mobile applications and smart devices that collect large amounts of data from sensors. HAR is a classification problem, whose objective is to predict an activity performed by a person in a given period of time. According to Sozinov *et al.* [2018], FL is suitable for this problem, considering that data from sensors in mobile and wearable devices can reveal private information about users. We implemented a realistic FL simulation using containers, prioritizing distributed evaluation of the global model. This framework orchestrates the FL process within a Docker container environment, enabling replication of real deployment conditions and seamless transition of experiments to production. By evaluating the model in the simulated environment where it will actually be deployed, this approach offers a practical and detailed view of global model performance on each device's local data, capturing data variability and client-specific conditions. Unlike other studies, this method considers the real usage scenario of the model, providing more accurate insights into FL's effectiveness and applicability in heterogeneous environments.

The main contributions of the article are as follows. First, we address the problem of non-IID data in the context of FL and propose an extension of existing solutions in the literature by considering a data sharing-based approach that balances privacy considerations and the need for model efficiency. Second, we present an FL algorithm that considers data sharing as a strategy to address the challenge of non-IID data for a privacy-flexible and privacy-sensitive environment in edge devices. Third, implemented and made available the FL simulation code based on Docker containers, highlighting the portability and adaptability of the research to real scenarios, allowing replication on different platforms. Finally, we present results that show the contribution of our algorithm to improving performance in scenarios with non-IID data. Unlike most FL works, our work focuses on FL distributed evaluation that allows participants to evaluate global model performance directly on their own data, where the models are actually deployed.

In addition to this introductory section, the remainder of this article is organized into sections as described below. Section 2 presents related works. Section 3 deals FL concepts and challenges. Section 4 describes the elements that make up the experimental scenario. Section 5 discusses the evaluation results of our proposed algorithm. Finally,

<sup>1</sup>Code is available at <https://github.com/LABORA-INF-UFG/DS-FL>

**Table 1.** Summary of Related Works

Ref.	What to Share		Privacy		Evaluation Type		Metrics		FL Tasks	
	Model	Data	Sensitive	Flexible	Centralized	Distributed	Accuracy	F1-Score	Generic	HAR
[McMahan <i>et al.</i> , 2016]	✓		✓		✓		✓		✓	
[Ma <i>et al.</i> , 2022]	✓	✓	✓	✓						
[Li <i>et al.</i> , 2020]	✓		✓		✓		✓		✓	
[Asad <i>et al.</i> , 2020]	✓	✓	✓		✓		✓		✓	
[Zhao <i>et al.</i> , 2022]	✓	✓	✓		✓		✓		✓	
[Seo <i>et al.</i> , 2023]	✓	✓		✓	✓		✓		✓	
[Amannejad, 2020]	✓		✓		✓		✓		✓	
[Quan <i>et al.</i> , 2023]	✓		✓		✓		✓		✓	
[Sozinov <i>et al.</i> , 2018]	✓		✓		✓		✓			✓
This work	✓	✓	✓	✓	✓	✓	✓	✓		✓

Section 6 presents the final considerations and indicates guidelines for future work.

## 2 Related Works

In this section, studies are presented that discuss, evaluate and indicate possible solutions for the implementation of FL in edge computing environments. These studies contribute to position the present work within this research context. Table 1 presents a summary of related work compared to our article.

In McMahan *et al.* [2016] the authors introduce the concept of FL and point out research directions for the problem of training ML models with decentralized data on mobile devices. The article presents an analysis of the performance of FedAvg with different parameter settings, ML models and datasets. However, the paper did not present a comprehensive discussion of FL in non-IID data scenarios.

In this context, there are several studies that have identified the distribution of non-IID data as an important challenge for FL. In Ma *et al.* [2022] the authors present the problems and potential solutions for FL in non-IID data. Li *et al.* [2020] presents a study that examines the properties of the FedAvg optimization process under non-IID data conditions. The authors present a formal analysis with examples and technical illustrations on how FedAvg is affected by data heterogeneity in the distributed training process.

FL research has emphasized the analysis of communication costs and the convergence of distributed models. In Asad *et al.* [2020] different FL strategies are evaluated in terms of communication accuracy and efficiency. The authors propose a strategy based on data sharing to deal with the distribution of non-IID data but did not present results that could demonstrate the usefulness of the proposal. In Zhao *et al.* [2022] it is demonstrated that the reduction in FL accuracy on non-IID data can be explained by the divergence of weights of the models trained on the devices. As a solution, the authors propose the creation of a subset of data that is shared globally among edge devices. Experiments show that FL accuracy for non-IID data can be increased when data is shared between devices. Seo *et al.* [2023] presents a study that uses small amounts of shared data (e.g., a single data input) to accelerate model

convergence and reduce communication costs in FL. The authors state that sharing small portions of data can be an attractive and practical solution in flexible environments in terms of privacy.

The development of automated solutions are useful for the implementation and evaluation of FL tasks. In Amannejad [2020] a framework is proposed to compare centralized ML models with federated solutions. The tool allows analyzing the accuracy of federated models considering the effect of different hyperparameters. However, the solution is not based on containers or virtualized environments, making unfeasible the portability to different devices and platforms. Quan *et al.* [2023] presents the architecture and implementation of a FL framework based on KubeEdge. The solution is evaluated through experiments that consider the FL in different configurations of edge devices, covering several types of heterogeneity, such as system heterogeneity, statistical heterogeneity and bandwidth rate.

FL can be used for a variety of HAR scenarios, such as monitoring health or predicting activities performed by an individual. In Sozinov *et al.* [2018] FL is used to train a HAR classifier, where the global model is compared with the performance of a centralized model. The FL assessment considers different ML models and the influence of IID, non-IID and skewed distributions. However, the study does not explore FL strategies to address the problem of heterogeneous data. The authors conclude that FL in a HAR task is capable of producing models with slightly worse, but acceptable, accuracy compared to centralized models.

In this work, we explore the complexities of FL with non-IID data in the context of HAR applications with highly heterogeneous data to accelerate convergence and improve the FL model by sharing small amounts of data. Our work proposes an algorithm that extends solutions data sharing-based from the literature, considering both privacy-flexible and privacy-sensitive environments. Unlike previous studies, our method suggests that the aggregator server generates and shares the necessary data only in the initial phase, increasing the privacy and effectiveness of the model. Our work is based on the context of a HAR application deployed in a Docker container-based environment, which provides insights into how FL models can behave in real-world scenarios. Our work differs from the studies mentioned above by employing centralized

and distributed approaches to evaluate the global model, jointly using accuracy and F1-Score metrics. FL distributed evaluation tests the global model directly on each device, where it will be deployed.

### 3 Federated Learning

FL is an ML setup that allows multiple clients (eg mobile devices) to collaboratively train a model without sharing local data with the coordination of a central server [Amannejad, 2020]. There are two main entities in the FL process: the data owner, i.e. the clients, and the global model owner, i.e. the server.

For a conceptualization of the FL environment, consider  $S = \{k_1, k_2, \dots, k_n\}$  as the set of  $K$  clients connected to a server before the start of a FL training round. Each customer has a  $\mathcal{P}_k$  dataset with  $n_k = |\mathcal{P}_k|$  samples stored on their respective local devices. In classic ML approaches, all clients send their data to a server and the server trains a conventional model, using all the data  $\mathcal{P} = \cup_{k=1}^n \mathcal{P}_k$ . However, this is not possible for all edge device scenarios due to privacy concerns and network bandwidth limitations. In this way, the FL training process can be expressed as

$$\begin{aligned} \min_{w_1, \dots, w_K} f(w_{global}) &\triangleq \min_{w_1, \dots, w_K} \frac{1}{|\mathcal{P}|} \sum_{k=1}^K \sum_{n=1}^{n_k} f(w_k, x_{kn}) \\ &= \min_{w_1, \dots, w_K} \sum_{k=1}^K \frac{n_k}{m_t} f_k(w_k) \\ \text{s.t. } w_1 &= w_2 = \dots = w_K = w_{global}, \end{aligned} \quad (1a)$$

where  $f(w_k, x_{kn})$  is the loss function that evaluates the performance of the local model  $w_k$  by observing the output produced by training with  $x_{kn}$  samples of data and  $m_t = \sum_{k \in S} n_k$ . For example, for supervised tasks, Equation (1) aims to find the parameters that result in predictions that come as close as possible to the real labels in  $x_{kn}$  according to a defined metric by the loss function. Constraint (1a) guarantees that the models shared between the devices and the aggregator server must be identical for a given FL task.

In FL original proposal, clients do not send their raw data to the aggregator server. In this case, what is sent are just the parameters of each locally trained  $w$  model. Local models received by the server are aggregated to create a global model  $w_{t+1}$  which is passed to clients for a new round of training. This process continues for  $t$  rounds of communication [Li et al., 2021].

#### 3.1 FedAvg Algorithm

Federated Averaging (FedAvg) [McMahan et al., 2016] was the first proposed algorithm for aggregating FL models. The main idea of FedAvg is to perform the aggregation of a global model over several rounds of communication based on the parameters of models trained locally on each device.

The pseudocode of the FedAvg algorithm is shown in Algorithm 1. Initially, a  $w_0$  global model is created or loaded from a pre-trained model. At the beginning of each round of

communication  $t$ , the server selects a random fraction  $f_t$  of  $K$  clients, generating a set  $\mathcal{S}_t$  of  $m$  clients. Then the current state of the  $w_t$  model is sent to each participant.

---

#### Algorithm 1: Federated Averaging (FedAvg)

---

**Server:**  
initialize  $w_0$   
**for** each round  $t = 1, 2, \dots$  **do**  
     $m \leftarrow \max(f_t \cdot K, 1)$   
     $\mathcal{S}_t \leftarrow$  (random set of  $m$  clients)  
    **for** each client  $k \in \mathcal{S}_t$  **in parallel do**  
         $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$   
     $m_t \leftarrow \sum_{k \in \mathcal{S}_t} n_k$   
     $w_{t+1} \leftarrow \sum_{k \in \mathcal{S}_t} \frac{n_k}{m_t} w_{t+1}^k$   
  
**ClientUpdate**( $k, w$ ):  $\triangleright$  Executed on client  $k$   
     $\mathcal{B} \leftarrow$  (data  $\mathcal{P}_k$  split into batches of size  $B$ )  
    **for** each local epoch  $i$  from 1 to  $E$  **do**  
        **for** each  $b \in \mathcal{B}$  **do**  
             $w \leftarrow w - \eta \nabla \ell(w; b)$   
    **return**  $w$  to server

---

Each participant trains the local model using an optimization algorithm based on Stochastic Gradient Descent (SGD) for each  $b \in \mathcal{B}$  mini-batches of  $\mathcal{P}_k$  (customer data  $k$ ) during  $E$  local epochs, where  $\nabla \ell$  represents the gradient from  $\ell$  into  $b$ . After the last update  $w \leftarrow w - \eta \nabla \ell(w; b)$ , the client sends the parameters of the local model  $w$  to the aggregator server.

In this way, the models  $w_{t+1}^k$  are received and aggregated by the server, generating the current state  $w_{t+1}$  of the global model applying the update  $w_{t+1} \leftarrow \sum_{k \in \mathcal{S}_t} \frac{n_k}{m_t} w_{t+1}^k$ . The aggregation is the weighted average of the clients parameters, whose weights are defined based on the amount of data used in the local training, where  $n_k = |\mathcal{P}_k|$  and  $m_t = \sum_{k \in \mathcal{S}_t} n_k$ .

The distributed training process can be repeated for several rounds of communication until convergence is achieved or model performance reaches a stopping criterion [Li et al., 2021]. Although FedAvg has been specifically proposed as a FL strategy, the accuracy of this strategy can be reduced substantially in highly distributed non-IID environments due to the statistical challenge of each participants datasets [Asad et al., 2020].

#### 3.2 Distribution of Heterogeneous Data in FL

The non-IID data distribution problem is related to the data heterogeneity problem. The empirical evidence presented in the literature highlights the significant impact of non-IID data on model performance within the FL context. This phenomenon introduces unique challenges to the FL landscape [Haller et al., 2023]. According to Zhao et al. [2022], traditional ML algorithms are usually trained on balanced data, under the assumption that the data is IID. However, datasets generated by mobile devices are based on the usage pattern of a given user and show a very different distribution across devices [Brecko et al., 2022]. Therefore, the dataset for each user will not be a representation of the population distribution [McMahan et al., 2016]. In this case,

the local training of a FL participant tends to specialize in the data characteristics of each user, which may lead to a low performance of the global model.

Zhao *et al.* [2022] presents experiments that demonstrate the extent to which non-IID data can affect the accuracy of the global model. The heterogeneous distribution of data in the FL can be expressed by asymmetry of the data distribution in different ways. According to Ma *et al.* [2022], the asymmetry can be caused by the skewed distribution of local data, varying from client to client. Furthermore, the distribution of the labels of the customers local data may be different, even though the probability distribution of the label considering the other customers is the same. The asymmetry can also be caused by different characteristics of the local data of each client that can indicate the same label, as well as, different labels can indicate the same characteristics of the local data for each client. Finally, the amount of local data for each customer can be significantly different, causing quantitative data heterogeneity.

### 3.3 FL Convergence Analysis with non-IID and Approaches based on Data Sharing

According to Ma *et al.* [2022], knowing the SGD is fundamental to understanding the challenges of data heterogeneity in FL. The SGD is an optimization algorithm often used to adjust parameters of ML models according to a loss function. Parameters are updated for each sample (random sampling), following the opposite direction of the loss function gradient. Thus, considering that the gradient must be calculated in each round of communication, the SGD can be naturally applied to the federated optimization problem.

In a FL task, each device must apply SGD to its own data to calculate local gradients. Then the local parameters are sent to a server that uses an aggregation strategy (for example, the FedAvg algorithm) to update the global model. In this context, based on Zhao *et al.* [2022], below we analytically demonstrate how the reduction in the accuracy of the FL can be explained by the divergence of weights of the local models, which is caused by the asymmetry of the data distribution.

Consider  $w_t^{(c)}$  as the  $t_{th}$  update of weights resulting from training a centralized model, where centralized SGD performs the following update

$$w_t^{(c)} = w_{t-1}^{(c)} - \eta \nabla \ell(w_{t-1}^{(c)}, \mathcal{P}), \quad (2)$$

where the convergence of the centralized model occurs as the SGD adjusts the respective parameters using samples  $\mathcal{P}$  of all customer data.

Next, let  $w_t^{(k)}$  be the  $t_{th}$  update of the weights resulting from training a device  $k$  on a FL task. In this scenario, the SGD runs on each device using only samples of local data  $\mathcal{P}_k$  from each client. At iteration  $t$  on client  $k \in K$ , local SGD performs

$$w_t^{(k)} = w_{t-1}^{(k)} - \eta \nabla \ell(w_{t-1}^{(k)}, \mathcal{P}_k). \quad (3)$$

Finally, consider  $w_T^{(f)}$  as the aggregation of the distributed model using the FedAvg algorithm every  $T$  steps. Based on

Equation (1), the aggregation of local models can be defined as

$$w_T^{(f)} = \sum_{k=1}^K \frac{n_k}{m_{t=T}} w_{t=T}^{(k)}. \quad (4)$$

Thus, when the data distribution is IID, the divergence  $w_t^{(k)}$  and  $w_t^{(c)}$  is small after the  $m_{th}$  aggregation of the model distributed, since  $w_T^{(f)}$  is still close to  $w_T^{(c)}$ . When the data distribution is non-IID, the divergence between  $w_t^{(k)}$  and  $w_t^{(c)}$  increases rapidly, causing the distancing of  $w_T^{(f)}$  and  $w_T^{(c)}$ . Therefore, this behavior does not allow the aggregate model to have a convergence that allows an effective generalization in all devices.

Data-driven approaches to the non-IID problem in FL include several [Ma *et al.*, 2022] strategies. Data sharing works to minimize data heterogeneity in FL. Similarly, data enhancement is aimed at sharing a small number of samples, where data privacy is controlled by introducing noise or using data anonymization techniques. Finally, data selection allows you to select specific customers that should participate in each round of communication.

Based on Zhao *et al.* [2022], consider  $K$  customers, each with  $n_k$  samples following the  $p^{(k)}$  distribution for customer  $k \in K$ . Given that synchronization is performed every  $T$  steps, then we have the following inequality for the weight divergence after the  $m$ -th synchronization

$$\|w_{mT}^{(f)} - w_{mT}^{(c)}\| \leq A + B, \quad (5)$$

$$A = \alpha \|w_{(m-1)T}^{(f)} - w_{(m-1)T}^{(c)}\|, \quad (5a)$$

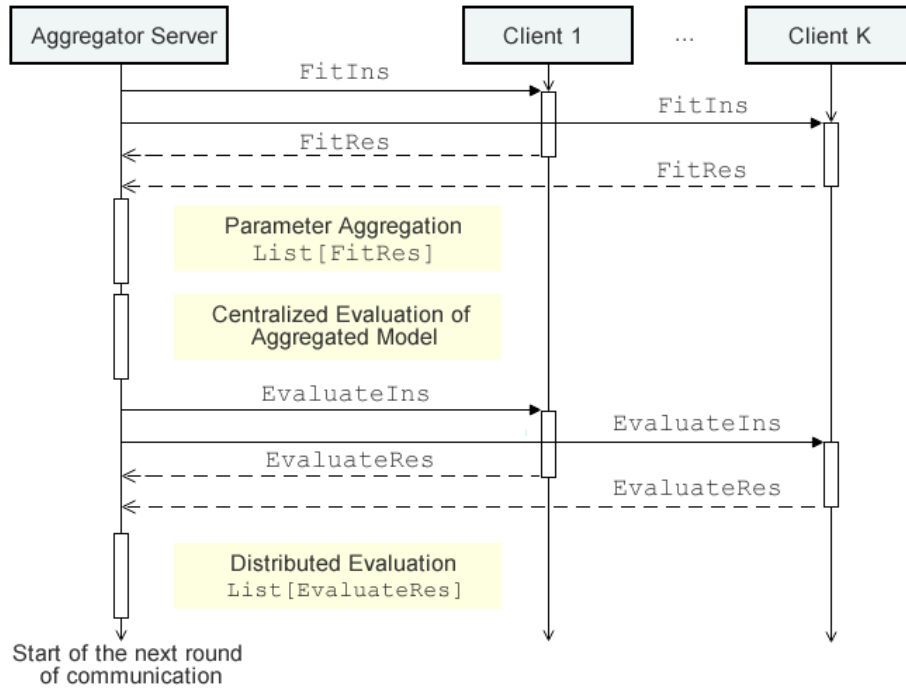
$$B = \eta \beta g_{\max}(w_{mT-1}^{(c)}), \quad (5b)$$

where  $A$  represents the accumulation of weight divergence between the global model and the local models over the iterations. The bigger  $\alpha$ , the greater the spread of divergence.  $B$  represents the direct impact of differences between local and global distributions. The term  $\beta$  captures the average difference between local class distributions  $p^{(k)}(y = i)$  and the global distribution  $p(y = i)$ .

By sharing a small amount of data that represents all classes (or, more generally, the data distributions of other clients), each client  $k$  now has access to samples that are more representative of the global distribution. This reduces the difference  $\|p^{(k)}(y = i) - p(y = i)\|$  for each class  $i$ , minimizing the contribution of this term to the weight divergence. Thus, the term  $g_{\max}(w_{mT-1}^{(c)})$  which represents the largest gradient between devices, is effectively reduced, leading to less weight divergence after each synchronization step. As a result, the propagation of divergence is controlled and  $\|w_{(m-1)T}^{(f)} - w_{(m-1)T}^{(c)}\|$  remains smaller at each step.

### 3.4 Evaluation of FL Tasks

The FL process is based on interactions between clients and servers that perform local and global computations [McMahan *et al.*, 2016]. In short, the global computations performed on the server side are responsible for orchestrating the FL process and aggregating edge device models. Clients,



**Figure 1.** Evaluation of FL Tasks. FL centralized evaluation uses a test dataset and metrics to provide a consistent view of the aggregated model’s performance on a unified dataset. FL distributed evaluation tests the global model directly on each device, where it will be deployed, aggregates results, and provides performance insights using the device’s own test data.

on the other hand, perform local calculations on each device, using their private data to perform local model training or server-aggregated model evaluation.

FL task assessment strategies must consider the decentralized nature of FL to measure global model performance. In this work, the aggregated model was evaluated in a distributed and centralized way, as shown in the diagram flow of Figure 1.

At the start of each training round, the aggregator server sends training instructions (*FitIns*) to each client, including the state of the aggregated model as a basis for the next round of local training. Other information guides the devices on how to perform local training, such as defining the number of training epochs and the learning rate.

After training is complete, each client sends a response (*FitRes*) to the server, including the locally trained model parameters and local performance metrics. Next, the server performs the aggregation of local models from all clients (*List[FitRes]*) to create an updated global model.

The centralized evaluation of aggregated model uses a test dataset and any ML model evaluation metrics. This evaluation provides a unified view of model performance, considering that all evaluations occur directly on the same dataset.

For distributed evaluation of aggregated model, the server sends instructions (*EvaluateIns*) to each device to evaluate the model using its local test data. Each device then sends a response (*EvaluateRes*) to the server as a result of evaluating the global model on each device. Finally, the server performs the distributed evaluation of the model by calculating the aggregation of the metrics for each client (*List[EvaluateRes]*). Distributed evaluation allows FL participants to see aggregated model performance of the

global model on their own data. Furthermore, it allows the server to verify that the global model is benefiting from stakeholder contributions, without the need to share the raw data between devices and the aggregator server.

### 3.5 Evaluation Metrics

In this work, the centralized evaluation of aggregated model use the same test dataset as the centralized training. Accuracy is used as a general evaluation metric for the distributed and centralized model. F1-Score is used as a centralized performance metric for each individual class.

As shown in Equation 6, accuracy is calculated by the ratio between the number of correct predictions and the total number of input samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

The number of correct predictions is defined by the sum of the terms *TP* (True Positive) and *TN* (True Negative), which describe situations where the model makes correct predictions, i.e., it correctly classifies an instance as positive or negative. To find the total number of samples (denominator of Equation 6), in addition to the sum of *TP* and *TN*, the terms *FP* (False Positive) and *FN* (False Negative) are added, which describe situations where the model makes incorrect predictions, i.e., failure to identify a positive or negative example.

F1-Score combines precision and recall into a single metric, taking into account both FP and FN. The range for F1-Score is  $[0, 1]$ . According to Equation 7, the F1-Score is the harmonic mean between precision (*P*) and recall (*R*). It is important to note that a high value for the F1-Score

normally indicates that the model is performing well in the classification task.

$$F1\text{-Score} = 2 \times \frac{P \times R}{P + R} \quad (7)$$

As observed in Equations 8 and 9, precision evaluates the proportion of positive examples correctly predicted by the model and recall evaluates the model's ability to correctly identify the positive examples present in the dataset.

$$P = \frac{VP}{VP + FP} \quad (8) \quad R = \frac{VP}{VP + FN} \quad (9)$$

For the distributed evaluation of the aggregated model, this work considers the FL evaluation pattern, where edge devices evaluate the global model locally with their own test data. Then, the devices send the evaluate accuracies to the server to perform distributed evaluate by calculating the arithmetic mean of respective local evaluations.

## 4 Experiment Setup

This section presents discussion of evaluating HAR applications as a means of evaluating our proposed FL algorithm, which incorporates data sharing as a strategy to address the challenges posed by non-IID data in flexible and privacy-sensitive edge computing environments. Initially, a preliminary analysis of the MotionSense, StresSense and HAR70+ datasets is presented. Next, we present the training of a centralized model used as a basis for comparing the convergence of FL tasks. Next, we present the distributed evaluation of the global FL model with non-IID data, as well as the strategies based on data sharing proposed by our algorithm. Finally, we present the results of the centralized evaluation of the distributed models.

### 4.1 DataSet

The diverse factors influencing human movement patterns in HAR applications result in non-IID data, offering a realistic and varied representation of real-world scenarios. This characteristic makes the dataset especially valuable for simulating practical applications, extending beyond purely theoretical frameworks. The inherent heterogeneity of non-IID data mirrors the complexity of real-world conditions, providing a robust foundation for testing and refining FL algorithms in challenging, practical settings.

This study explores the application of HAR as a means to evaluate our proposed FL algorithm, which incorporates data sharing as a strategy to address the challenges posed by non-IID data in flexible and privacy-sensitive edge computing environments. To evaluate our proposal, we use the following datasets: MotionSense [Malekzadeh *et al.*, 2019], StresSense [Saddaf Khan *et al.*, 2024] and HAR70+ [Ustad *et al.*, 2023]. In some parts of this article, we will refer to the respective datasets as MS, SS and H70+.

MotionSense data was generated by 24 participants with varying gender, age, weight, and height using an iPhone 6s put in a front pants pocket. Data collection from the Core Motion platform on iOS devices was performed using

the SensingKit library at a sampling rate of 50 Hz. While performing 6 activities in 15 trials under the same conditions and in the same environment, participants performed the following activities: downstairs (*dws*), upstairs (*ups*), walking (*wlk*), jogging (*jog*), sitting (*sit*) and standing (*std*).

StresSense data was generated by 40 participants (20 men and 20 women) between the ages of 20 and 25, wearing a Samsung Galaxy S5 on the wrist of each participant's dominant arm. Data collection was done via an Android application with a sampling rate of 50 Hz. Data resources include only time series data corresponding to the accelerometer, gyroscope and magnetometer, which have been found to be good predictors of stress-associated activities. However, our work was limited to data from the first 20 participants, using exclusively accelerometer and gyroscope information, allowing an analysis of the ML/FL model gradually and with slower convergence. Participants performed 5 activities with the following labels: staying still (*sts*), nail-biting (*nab*), smoking (*smk*), eating (*eat*) and face touching (*fst*).

HAR70+ data was generated by 18 older adults ages 70 to 95 performing multiple repetitions of daily activities. Participants were equipped with a chest-mounted camera and two Axivity AX3 accelerometers positioned on their lower back and right thigh. AX3 GUI software was used to configure the accelerometers to record at a sampling frequency of 50 Hz. Participants performed 7 activities with the following labels: walking (*wlk*), shuffling (*shf*), ascending stairs (*asc*), descending stairs (*dsc*), standing (*std*), sitting (*sit*) and lying (*lyg*). According to Ustad *et al.* [2023], we combined classes for the analyses so that stair walking categories were integrated with the walking category and shuffling was integrated with standing.

### 4.2 ML Model

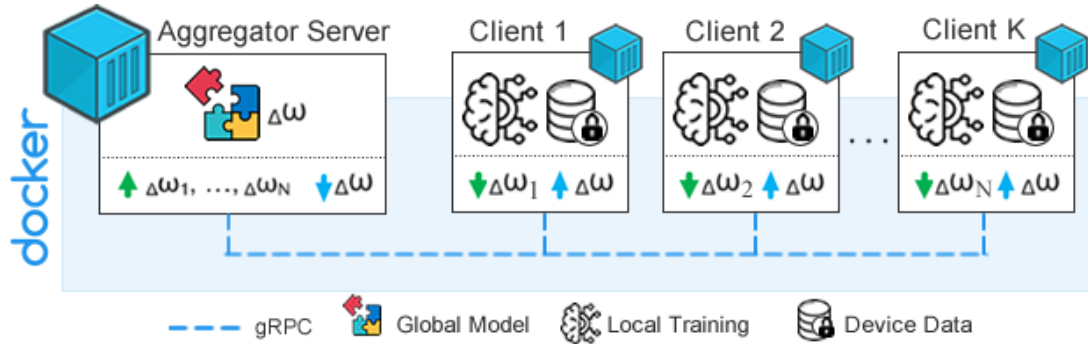
The experiments in this work used a Multi-Layer Perceptron (MLP) neural network architecture with three hidden layers, with 128, 64 and 32 neurons per layer and ReLU activation function. Furthermore, ADAM (Adaptive Moment Estimation) is used as a model optimization method, Sparse Categorical Crossentropy as a loss function. The federated training process took place over 250 training rounds, with all devices being selected in each communication mode. Locally, each device trained the local model for 1, 5 and 10 local training epochs, using a batch size of 128 and a learning rate of  $lr = 0.001$ . Other information can be found in the repository available on GitHub.

It is noteworthy that it is possible to use different architectures of deeper neural networks in order to achieve better model performance. However, according to Zhao *et al.* [2022], even if the model's accuracy does not reach the state of the art, this may still be sufficient for evaluating the behavior of FL strategies in different scenarios.

### 4.3 Simulation Environment

Figure 2 presents the simulation environment of this work. The FL process orchestration is implemented using the Flower framework [Beutel *et al.*, 2020] in an environment





**Figure 2.** FL simulation environment is implemented using the Flower framework in an environment based on Docker containers.

based on Docker containers. Flower provides a stable implementation of the core components of an FL system, independent of programming language and ML framework. In Figure 2, we can see the existence of several devices capable of training an ML model using its local data and an FL aggregator server, both implemented in containers. Communication between devices and the aggregator server is facilitated by Flower using the gRPC protocol.

Use of containers allows the isolation of the aggregator server and the clients participating in the FL process. Furthermore, it is possible to define heterogeneous clients by customizing computing and memory limitations of each container, as well as restricting network bandwidth using traffic control tools. Furthermore, using Flower with container technology allows FL simulations to be quickly transferred to production environments without the need for complex adjustment.

## 5 Discussion of Results

This section presents discussion of evaluating HAR applications as a means of evaluating our proposed FL algorithm, which incorporates data sharing as a strategy to address the challenges posed by non-IID data in flexible and privacy-sensitive edge computing environments. Initially, a preliminary analysis of the MotionSense, StresSense and HAR70+ datasets is presented. Next, we present the training of a centralized model used as a basis for comparing the convergence of FL tasks. Next, we present the distributed evaluation of the global FL model with non-IID data, as well as the strategies based on data sharing proposed by our algorithm. Finally, we present the results of the centralized evaluation of the distributed models.

### 5.1 Dataset Analysis

This section presents a preliminary analysis of each data set with the aim of getting an intuition about the characteristics of the sample distribution of each user's data. Consider  $\mathcal{S} = \{k_1, k_2, \dots, k_K\}$  as the set of  $K$  users that have a set  $\mathcal{P}_k$  of local data. Initially, each dataset was classified according to the label that identifies each user, generating non-IID datasets, with 75% of data reserved for training and 25% of data for testing. Then, each user device trained a centralized  $w_k$  model using their respective local data. Finally, the

accuracy of the  $w_k$  model was evaluated using local test data from other  $\mathcal{S}$  users.

Figures 3a, 3b and 3c present the graphs of the evaluation of the  $w_k$  model using  $P_k$  non-IID for each dataset. Each element on the main diagonal represents the accuracy of the  $w_k$  model trained on its dataset. The remaining elements of each column indicate the accuracy of  $w_k$  tested with local data from other users. Looking at this graph, we can observe the non-IID nature of this dataset, as the  $w_k$  model fits very well to the unique patterns of the local dataset, however it has difficulty generalizing to other datasets. This occurs because the  $w_k$  model was trained with data that do not represent the characteristics of other users.

For generating IID data, the samples from each dataset were shuffled and divided into  $K$  datasets with the same number of examples, reserving 75% of data for training and 25% of data for testing. As previously done, each user device trained a centralized model  $w_k$  using their respective local data and the accuracy of the model  $w_k$  was evaluated based on the local test data of the other users.

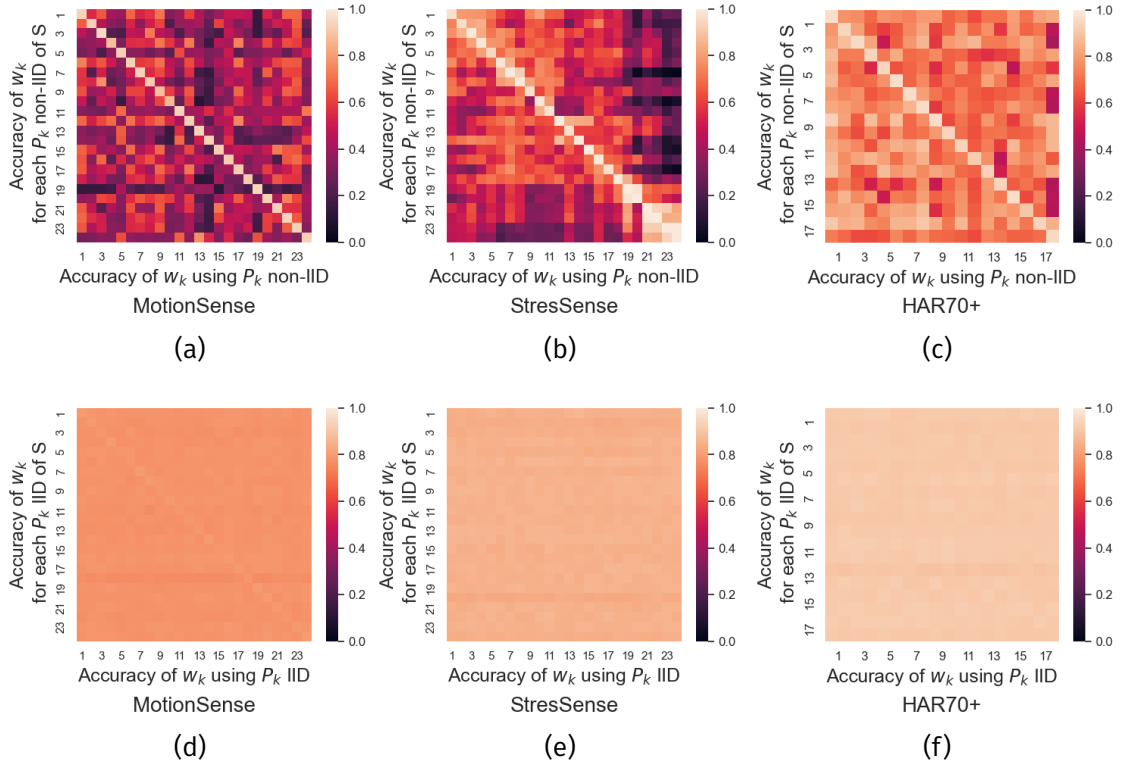
The evaluation of the model  $w_k$  using the  $P_k$  IID for each dataset is presented in the graphs of the Figures 3d, 3e and 3f. In this scenario, each  $P_k$  dataset has a similar distribution. In this way, it is possible to observe the IID nature of this dataset, as each  $w_k$  model manages to learn patterns and relationships that are consistent for all users. Therefore, the  $w_k$  model trained with its local data is able to generalize its learning to the other datasets.

According to Ma *et al.* [2022], data heterogeneity is among the main challenges faced by FL. In this context, it is important to find out the IID or non-IID characteristics of the distribution of customer data in order to make initial assumptions about the performance of the distributed model in FL tasks.

### 5.2 Centralized Model Training

Centralized model training to refer to training a model in a centralized environment, where all data is in a single location and the model is trained with full access to the complete dataset. This work uses the accuracy of a centralized model training as a way to establish a reference point to qualitatively evaluate the convergence of FL tasks in different scenarios. However, it is important to highlight that the objective of FL is not necessarily to replicate the results of centralized





**Figure 3.** Evaluation of the  $w_k$  model using  $P_k$  with non-IID and IID data for MotionSense, StresSense and HAR70+ datasets.

training. FL must be evaluated in terms of its ability to achieve a high quality solution in distributed scenarios, with a focus on privacy preservation and communication efficiency.

By establishing the accuracy of a centralized model as a reference point, it is possible to show that FL models can achieve or approach the accuracy of centralized models. In this case, the centralized model serves as an essential comparative basis, indicating how far FL can evolve in terms of convergence, showing the feasibility of adopting FL in critical applications where precision is a crucial factor.

Figure 4 illustrates the evolution of accuracy and loss during the training phase of the centralized model. This model was trained for 250 local epochs, utilizing 75% of the samples from each dataset for training and the remaining 25% for testing. In order to obtain the best possible performance, the state of the model was saved after each training epoch, allowing later restoration for use and evaluation. Thus, the best performing model achieved an accuracy value of 82.84, 89.84, and 94.28, respectively, for the MotionSense, StresSense, and HAR70+ datasets. Even in non-IID conditions, it is observed that the  $w_t^{(c)}$  model presents good accuracy as it is trained using the set  $\mathcal{P}$  that includes data variations from all devices. In this case, the gradient updates are more representative, enabling a good generalization of the centralized model.

It is noteworthy that it is possible to achieve a better performance of the centralized model using ML techniques to expand the characteristics of the dataset and using different architectures of deeper neural networks. According to Zhao *et al.* [2022], even if the accuracy of the model does not reach the state of the art, this can still be enough for the evaluation

of FL behavior in different scenarios.

Table 2 shows the performance of the centralized model using the F1-Score metric for each activity performed by the participants. In this way, it is possible to obtain a slightly more detailed view of the performance of the model considering the particularities of each activity.

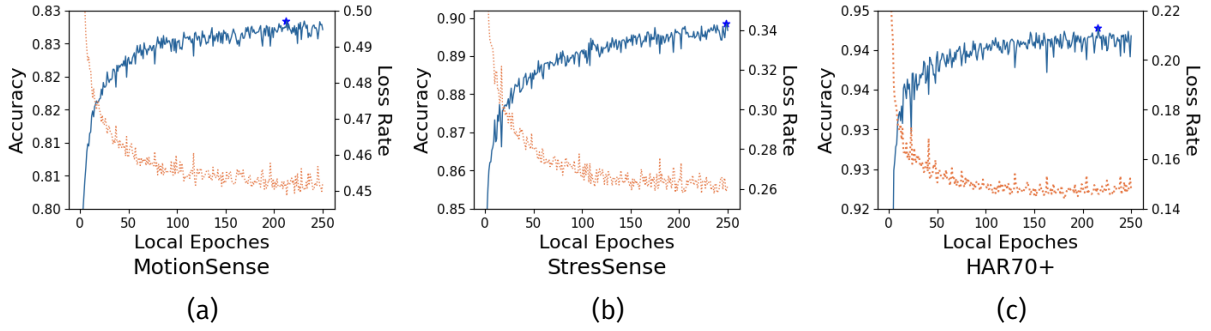
**Table 2.** Centralized model performance using the F1-Score metric for each activity from MotionSense, StresSense, and HAR70+ datasets.

Dataset	F1-Score					
MotionSense	<b>dws</b>	<b>jog</b>	<b>sit</b>	<b>std</b>	<b>ups</b>	<b>wlk</b>
	.523	.742	.997	.984	.546	.777
StresSense	<b>sts</b>	<b>nab</b>	<b>smk</b>	<b>eat</b>	<b>fct</b>	
	.886	.874	.899	.801	.999	
HAR70+	<b>wlk</b>	<b>sit</b>	<b>std</b>	<b>lyg</b>		
	.940	.998	.863	.999		

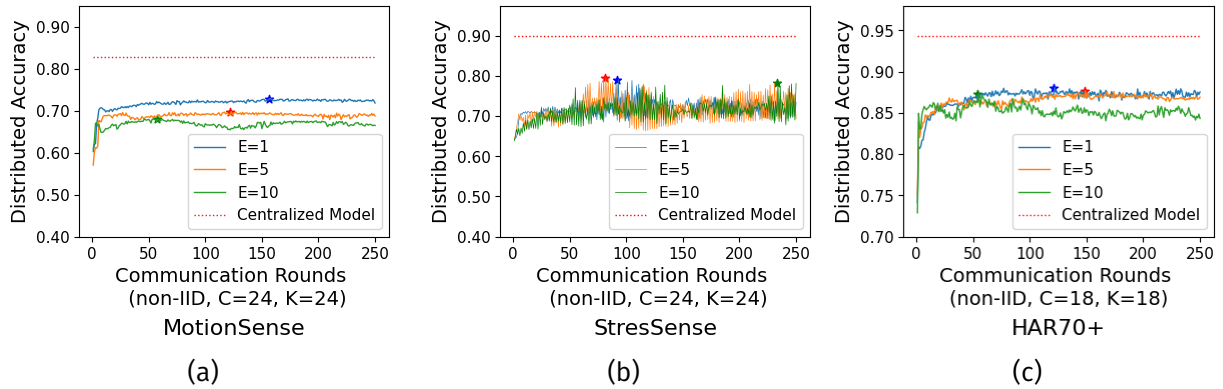
The F1-Score metric indicates that the model managed to achieve a good performance for activities with a greater variation in the dynamics of movements. On the other hand, activities with similar temporal patterns or that become similar due to the detection capabilities of the sensors (e.g. upstairs and downstairs) achieved a low performance.

### 5.3 FL with non-IID Data

This section presents FL results for datasets with non-IID distributions, for 250 communication rounds and different values  $E$  for amounts of local training epochs. The non-IID



**Figure 4.** Centralized model training for MotionSense, StresSense, and HAR70+ datasets. ML model training is performed in a centralized environment where all data is in a single location and the model is trained with full access to the complete dataset.



**Figure 5.** FL distributed accuracy with non-IID data for MotionSense, StresSense, and HAR70+ datasets. For the MotionSense and StresSense datasets, all  $K=24$  clients were selected in each communication round, while for the HAR70+ dataset, all  $K=18$  clients were chosen in each round.

data distribution pattern is more realistic for FL tasks in the context of edge devices. In these experiments, the server uses the FedAvg algorithm strategy for parameter aggregation and  $f_t = 1$ , i.e, all  $C = K$  clients participate in each communication round using the same neural network architecture of centralized training.

Figure 5 presents the FL challenges in non-IID scenarios for MotionSense, StresSense, and HAR70+ datasets. We present FL distributed evaluation, which allows participants to evaluate aggregate model performance based on their own data. Furthermore, it serves as a metric that captures the performance of the model in the environment where it will be deployed, that is, on the participants' own devices. Figure 5 shows that the distributed model reaches a low accuracy compared to the centralized model. This is due to the heterogeneity of data from each device, which causes local models to be updated in different directions. In this case, the disparity in data distribution between devices means that gradients  $\nabla \ell(w_T^{(k)}, \mathcal{P}_k)$  vary significantly between devices. This leads to greater divergence in  $w_T^{(k)}$ , as each local gradient reflects a different direction for optimizing the loss function. Therefore, the aggregation  $w_T^{(f)} = \sum_{k=1}^K \frac{n_k}{m_T} w_{t=T}^{(k)}$  does not represent the distribution of the entire dataset.

In Table 3 it can be seen that after the completion of the 250 communication rounds, the best performances of distributed models tend to occur with fewer local training epochs, that is, when  $E = 1$ . In non-IID scenarios, excessive local iterations can hinder distributed model convergence due to significant

parameter discrepancies across devices. Reducing local epochs minimizes these divergences, facilitating parameter alignment and improving global performance. Formally, the disparity in data distribution between devices means that gradients  $\nabla \ell(w_T^{(k)}, \mathcal{P}_k)$  vary significantly between devices. This leads to greater divergence in  $w_T^{(k)}$ , as each local gradient reflects a different direction for optimizing the loss function. Therefore, the aggregation  $w_T^{(f)} = \sum_{k=1}^K \frac{n_k}{m_T} w_{t=T}^{(k)}$  does not represent the distribution of the entire dataset.

## 5.4 Data-driven approach for FL with non-IID data distribution

The previous section presented experiments that demonstrate the low performance of FL for non-IID data, which is caused by the divergence of the weights of the local models of each device. This section presents our strategies in one dimension based on data sharing to reduce the asymmetry of devices data distribution. The impact of convergence and model effectiveness for FL data-driven solution with non-IID data distribution may not be trivial. Data heterogeneity can result in local models that are highly specialized for data patterns from a small group of devices. In some cases, adding small portions of global data can result in significant performance improvements. However, in other situations, data sharing may not justify the model performance gains. Our two strategies presented in Algorithm 2, as follows:

**Table 3.** FL distributed evaluation for non-IID data, where  $\text{Acc}_D$  is the distributed accuracy and  $\text{DIF}_{DC}$  is the difference of  $\text{Acc}_D$  with respect to the centralized model. For the MotionSense and StresSense datasets, all  $K=24$  clients were selected in each communication round, while for the HAR70+ dataset, all  $K=18$  clients were chosen in each round.

Dataset	E	$\text{Acc}_D$ (%)	$\text{DIF}_{DC}$
MS <sub>NIID</sub>	1	72.86	-9.98
	5	69.73	-13.11
	10	67.98	-14.86
SS <sub>NIID</sub>	1	79.08	-10.76
	5	79.02	-10.82
	10	78.23	-11.60
H70+ <sub>NIID</sub>	1	88.05	-6.22
	5	87.65	-6.62
	10	87.29	-6.99

1. The first strategy considers a privacy-flexible environment, where users agree to share a small percentage of their private data to build a global shared dataset (GSD).
2. The second strategy considers a privacy-sensitive environment emphasizing the protection of sensitive information, assuming the existence of a public global share dataset (PGSD) that characterizes the global dataset.

Considering that the biggest challenges of traditional FL refer to the computation and communication costs in the various communication rounds, the cost of the Algorithm 2 does not overload the network infrastructure, as the global data sharing is not a recurring event as it only occurs in the algorithm initialization phase. Thus, the trade-off related to communication cost and model performance can be balanced by the limit that controls the amount of shared data, restricting the network load to the minimum necessary for the initial configuration. In the Privacy-Flexible Environment strategy, the Algorithm 2 is initialized with the devices uploading a percentage of their data to the FL server. Then, the server concatenates the received data to form the global share dataset (GSD). Finally, each device downloads GSD, updating its local data through the  $\mathcal{P}'_k \leftarrow \mathcal{P}_k + \text{GSD}$  operation. On the other hand, in the Privacy-Sensitive Environment strategy, there is no communication between devices and the server when the algorithm is initialized. In this case, assuming the existence of a publicly shared global dataset (PGSD) generated by the server through generative models or simulation techniques that must reflect the behavior of users, updating their local data by the  $\mathcal{P}'_k \leftarrow \mathcal{P}_k + \text{PGSD}$ . Likewise, the integration of global data between devices does not add significant computational costs to the local model training process, as the devices make efficient use of mini-batches in each local training epoch, which already guarantees control of the processing load so that training is computationally economical. According to the 2 Algorithm, minibatch data is generated from  $\mathcal{P}'_k$ , using private data concatenation from  $\mathcal{P}_k$  more data shared in the algorithm initialization phase.

In the first strategy, convincing users to share part of

#### Algorithm 2: Data-driven solution for FL

##### Server:

▷ *Initialization: Privacy-Flexible Environment*

**for** each client  $k \in \mathcal{S}$  **in parallel do**

$\text{GSD} \leftarrow \text{ClientUpload}(k, \text{pct}_{\mathcal{P}_k})$

**for** each client  $k \in \mathcal{S}$  **in parallel do**

$\text{ClientDownload}(k, \text{GSD}) \triangleright \mathcal{P}'_k \leftarrow \mathcal{P}_k + \text{GSD}$

▷ *Initialization: Privacy-Sensitive Environment*

$\text{PGSD} \leftarrow \text{public global share dataset}$

**for** each client  $k \in \mathcal{S}$  **in parallel do**

$\text{ClientDownload}(k, \text{PGSD}) \triangleright \mathcal{P}'_k \leftarrow \mathcal{P}_k + \text{PGSD}$

initialize  $w_0$

**for** each round  $t = 1, 2, \dots$  **do**

$m \leftarrow \max(f_t \cdot K, 1)$

$\mathcal{S}_t \leftarrow (\text{random set of } m \text{ clients})$

$C \leftarrow |\mathcal{S}_t|$

**for** each client  $k \in \mathcal{S}_t$  **in parallel do**

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

$m_t \leftarrow \sum_{k \in \mathcal{S}_t} n_k$

$w_{t+1} \leftarrow \sum_{k \in \mathcal{S}_t} \frac{n_k}{m_t} w_{t+1}^k$

**ClientUpdate**( $k, w$ ): ▷ *Executed on client k*

$\mathcal{B} \leftarrow (\text{data } \mathcal{P}'_k \text{ split into batches of size } B)$

**for** each local epoch  $i$  from 1 to  $E$  **do**

**for** each  $b \in \mathcal{B}$  **do**

$w \leftarrow w - \eta \nabla \ell(w; b)$

**return**  $w$  to server

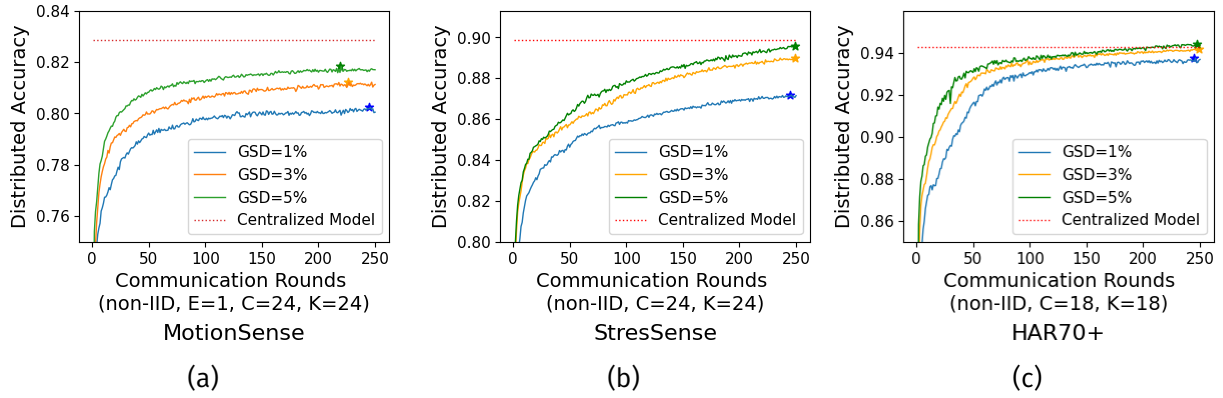
their data can be a challenge. To be more likely to agree, users must have a clear sense of the tangible benefits of sharing data. Furthermore, data sharing must not include the label that identifies users and must respect the data protection laws of each region. In this context, it is necessary to implement privacy preservation techniques, such as cryptographic methods and differential privacy. In the second strategy, the challenge is related to the process of generating data that reflect user behavior, such as the use of generative models or simulation techniques.

## 5.5 FL Distributed Evaluation

This section presents results of the analysis of our strategies in one dimension based on data sharing to reduce the asymmetry of devices data distribution. This means that the difference between the local parameters  $w_T^{(k)}$  is minimized favoring a better convergence of  $w_T^{(f)}$  compared to  $w_T^{(c)}$ .

To simulate the first data sharing strategy, the percentage of 1%, 3% and 5% of data from each client were removed to define the GSD. Then, the global data were respectively incorporated into each dataset  $\mathcal{P}_k$ , i.e., for each client from the MotionSense, StresSense, and HAR70+ datasets.

Figure 6 shows the evolution of accuracies of distributed FL models for GSD datasets. To allow stably aggregation, the devices trained each local model for only one epoch. In this scenario, it is observed that the aggregation process of model updates converges a little more smoothly and towards better accuracy when compared to the results where the data distribution is non-IID without data sharing. Furthermore, it



**Figure 6.** FL distributed accuracy for non-IID data with  $E=1$  using a global dataset. Each client  $k$  contributed 1%, 3% and 5% of  $\mathcal{P}_k$  to define the GSD. For the MotionSense and StresSense datasets, all  $K=24$  clients were selected in each communication round, while for the HAR70+ dataset, all  $K=18$  clients were chosen in each round.

**Table 4.** FL distributed accuracy with global data sharing with  $E=1$ , where  $P$  is the percentage of global data,  $\text{Acc}_D$  is the distributed accuracy,  $\text{DIF}_{DC}$  is the difference of  $\text{Acc}_D$  with respect to the centralized model and  $\text{DIF}_{DD}$  is the difference of  $\text{Acc}_D$  with respect to the non-IID distributed model without data sharing with  $E=1$ .

Dataset	P(%)	$\text{Acc}_D$ (%)	$\text{DIF}_{DC}$	$\text{DIF}_{DD}$
MS <sub>GSD</sub>	1	80.22	-2.62	+7.36
	3	81.21	-1.63	+8.35
	5	81.82	-1.02	+8.96
MS <sub>PGSD</sub>	1	76.61	-6.23	+3.75
	3	79.67	-3.17	+6.81
	5	81.45	-1.39	+8.59
SS <sub>GSD</sub>	1	87.17	-2.67	+8.08
	3	88.99	-0.85	+9.91
	5	89.55	-0.29	+10.47
SS <sub>PGSD</sub>	1	81.77	-8.07	+2.69
	3	86.90	-2.94	+7.81
	5	88.58	-1.26	+9.49
H70+ <sub>GSD</sub>	1	93.73	-0.54	+5.67
	3	94.15	-0.12	+6.09
	5	94.40	+0.11	+6.34
H70+ <sub>PGSD</sub>	1	90.22	-4.06	+2.16
	3	91.49	-2.78	+3.44
	5	92.45	-1.83	+4.39

is observed that models trained with a higher percentage of global data achieved better FL distributed accuracy. In Table 4 it is possible to observe that the distributed models that used 1%, 2% and 5% of global data brought their accuracies closer to the performance of the centralized model, as well as obtaining an increase in its accuracies compared to the non-IID model without data sharing.

To simulate the second data sharing strategy,  $\mathcal{P}_{23}$  and  $\mathcal{P}_{24}$  of the MotionSense and StresSense datasets were made public in their entirety. Similarly, for the HAR7+ dataset,  $\mathcal{P}_{17}$  and  $\mathcal{P}_{18}$  were made public in their entirety. Then a percentage of 1%, 3% and 5% was removed from  $\mathcal{P}_{(K-1)} \cup \mathcal{P}_{(K)}$  proportional to  $\sum_{k=1}^{(K-2)} |\mathcal{P}_k|$  to define the PGSD for each datasets. Subsequently, the global data was incorporated respectively into each  $\mathcal{P}_k$  dataset, i.e., for each

**Table 5.** FL centralized accuracy for non-IID data.  $\text{Acc}_{CM_D}$  is the FL centralized accuracy,  $\text{DIF}_{CM_DC}$  is the difference of  $\text{Acc}_{CM_D}$  with relation to the accuracy of centralized model.

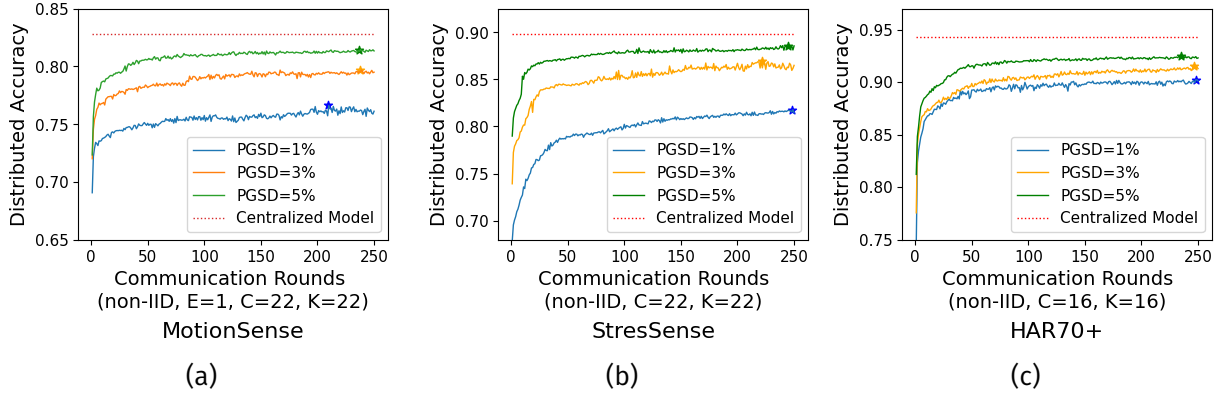
Dataset	E	$\text{Acc}_{CM_D}$ (%)	$\text{DIF}_{CM_DC}$
MS <sub>NIID</sub>	1	72.80	-10.04
	5	69.76	-13.08
	10	68.12	-14.72
SS <sub>NIID</sub>	1	78.77	-11.07
	5	79.41	-10.43
	10	78.76	-11.07
H70+ <sub>NIID</sub>	1	87.78	-6.49
	5	87.57	-6.70
	10	87.29	-6.98

client of the MotionSense, StresSense and HAR70+ datasets.

Figure 7 presents the evolution of accuracies of distributed FL models for PGSD datasets trained on each device for a local epoch. In this scenario, it is observed that the results are inferior to the FL process for the GSD datasets, but still showed better accuracies when compared to the results where the data distribution is non-IID without data sharing. In Table 6, it can be observed that the models with GSD and PGSD data obtained better performance when they used 5% of data sharing. From the perspective of the distributed evaluation, it is possible to conclude that the data sharing of the GSD and PGSD sets reduced the heterogeneity of the data of the devices participating in the FL tasks.

## 5.6 FL Centralized Evaluation

This section presents the centralized evaluation of distributed FL models, which refers to the evaluation of the global FL model using the aggregator server using a standard dataset. In this way, it is possible to validate whether an aggregated model is able to adequately generalize to examples of the same set of validation data. In this work, the FL centralized evaluation of the distributed models uses the same test dataset as the centralized training. The accuracy and the F1-Score are used, respectively, as metrics for the general evaluation of the model and the specific performance of each class.



**Figure 7.** FL distributed accuracy for non-IID data with  $E=1$  using a public global dataset.  $\mathcal{P}_{(K-1)}$  and  $\mathcal{P}_K$  were made public and 1%, 3% and 5% were withdrawn to define the PGSD. For the MotionSense and StresSense datasets, all  $K=22$  clients were selected in each communication round, while for the HAR70+ dataset, all  $K=16$  clients were chosen in each round.

Table 5 presents the FL centralized evaluation for non-IID data. As expected, the performance of the centralized evaluation of the models for non-IID data is not satisfactory. Table 6 presents the FL centralized evaluation with our data sharing strategies. In the strategy with GSD data, it is observed that the centralized accuracy value of the distributed model increases as the percentage of global data increases. Furthermore, it is possible to observe that the accuracy approaches the performance of the centralized model. This means that the distributed model was able to generalize reasonably well to the centralized validation set, indicating that the model is not simply memorizing local device data.

On the other hand, in the strategy with PGSD data, the characteristics of the global data were not sufficiently representative for the aggregated model to obtain a good performance in the centralized evaluation. In this case, it is still possible to observe an important degree of heterogeneity in the distribution of customer data that contributed to the aggregation of the global model, since in general, the centralized accuracy of the distributed model decreases as the percentage of global data increases.

Figure 8 summarizes the performance of the FL distributed model for some scenarios, indicating distributed and centralized evaluation. The FL scenario with GSD data stands out, where the performance of the distributed model reached good centralized and distributed accuracy compared to the centralized model. Likewise, it is observed that the FL scenarios with PGSD data, where the use of 1% and 5% of global data resulted in an increase in distributed accuracy and little variation in centralized accuracy compared to the centralized model.

Tables 7 and 8 show the performance of FL centralized evaluation using the F1-Score metric for each activity performed by the participants of the MotionSense, StresSense and HAR70+ datasets. The F1-Score metric in the different scenarios indicated that the distributed model achieved a low performance for the same activities pointed out in the centralized evaluation that are naturally difficult to be distinguished based on the data captured by the sensors.

For the non-IID and PGSD data distribution scenarios of Tables 7 and 8, it is observed that the degree of heterogeneity

**Table 6.** FL centralized evaluation with global data sharing with  $E=1$ , where  $P$  is percentage of global data,  $\text{Acc}_{CM_D}$  is centralized accuracy of distributed model,  $\text{DIF}_{CM_D C}$  is the difference of  $\text{Acc}_{CM_D}$  with respect to the centralized model

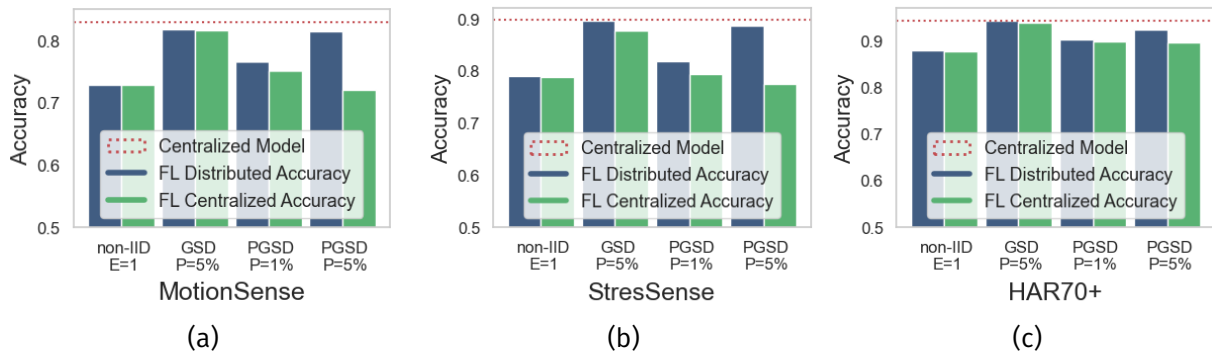
Dataset	P(%)	$\text{Acc}_{CM_D}$ (%)	$\text{DIF}_{CM_D C}$
MS <sub>GSD</sub>	1	80.30	-2.54
	3	81.25	-1.59
	5	81.67	-1.17
MS <sub>PGSD</sub>	1	75.16	-7.68
	3	73.95	-8.89
	5	72.11	-10.73
SS <sub>GSD</sub>	1	86.13	-3.70
	3	87.26	-2.57
	5	87.80	-2.03
SS <sub>PGSD</sub>	1	79.36	-10.48
	3	78.77	-11.06
	5	77.55	-12.28
H70+ <sub>GSD</sub>	1	93.40	-0.87
	3	93.84	-0.44
	5	93.88	-0.40
H70+ <sub>PGSD</sub>	1	89.86	-4.41
	3	89.63	-4.64
	5	89.66	-4.62

of the data impacts especially the performance activities with similar time patterns. In addition, for the non-IID data scenario, the performance of all activities decreases as the number of local epochs increases. In PGSD data scenario, the performance of all activities decreases as the percentage of global data whose shared data was not generated by the clients participating in the FL task increases. This behavior indicates that in these scenarios there is a significant degree of heterogeneity in the distribution of data from these clients that impacts the better performance of the global FL model.

## 6 Conclusions and Future Work

This work presents FL challenges based on the simulation of a HAR application. The results showed that FL performance tends to be low when data distribution is heterogeneous.





**Figure 8.** FL distributed accuracy results from evaluation where the global FL model is evaluated using a centralized dataset. FL centralized accuracy is the result of distributed evaluation, where the average value of the aggregated model is generated for each client using their own test local data.

**Table 7.** FL centralized evaluation for non-IID data using the F1-Score metric for each activity in the MotionSense, StresSense and HAR70+ datasets.

Dataset	F1-Score						
MS <sub>NIID</sub>	E	dws	jog	sit	std	ups	wlk
	1	.430	.589	.975	.962	.439	.561
	5	.411	.576	.951	.964	.417	.516
	10	.406	.538	.888	.962	.404	.475
SS <sub>NIID</sub>	E	sts	nab	smk	eat	fct	
	1	.571	.801	.750	.553	.991	
	5	.450	.818	.724	.531	.992	
	10	.669	.804	.800	.579	.995	
H70 <sub>NIID</sub>	E	wlk	sit	std	lyg		
	1	.884	.960	.714	.926		
	5	.880	.966	.701	.931		
	10	.861	.958	.632	.890		

As a solution, we propose an algorithm with two data sharing strategies to reduce the asymmetry of device data distribution. The results showed that the aggregation process of the global models that used strategies in a dimension based on data converge to a better performance when compared to the scenario where the data distribution is more heterogeneous. As a future work, we intend to analyze the communication costs of our data sharing strategies, as well as, implement security measures to enable customer collaboration and utilize other strategies to deal with the non-IID data issue. As an example, we mention the use of techniques based on ML models that are a little more complex and adaptable to the heterogeneous scenario, the implementation of algorithms that customize the aggregation strategy based on local tasks, the use of generative models to generate synthetic global data, and the development of FL frameworks that use various approaches to handle data heterogeneity on the same platform.

## Declarations

## Authors' Contributions

All authors contributed to the writing of this article, read and approved the final manuscript.

**Table 8.** FL centralized evaluation for GSD and PGSD data using the F1-Score metric for each activity in the MotionSense, StresSense and HAR70+ datasets.

Dataset	F1-Score						
MS <sub>GSD</sub>	P(%)	dws	jog	sit	std	ups	wlk
	1	.494	.694	.996	.983	.517	.729
	3	.496	.716	.997	.983	.533	.752
	5	.502	.722	.997	.984	.542	.762
MS <sub>PGSD</sub>	1	.441	.632	.977	.963	.467	.629
	3	.429	.614	.971	.958	.436	.595
	5	.421	.599	.969	.957	.440	.599
SS <sub>GSD</sub>	P(%)	sts	nab	smk	eat	fct	
	1	.853	.855	.851	.707	.998	
	3	.870	.863	.860	.728	.998	
	5	.873	.866	.869	.754	.998	
SS <sub>PGSD</sub>	1	.641	.815	.793	.597	.991	
	3	.679	.819	.783	.601	.993	
	5	.646	.811	.768	.590	.993	
H70 <sub>GSD</sub>	P(%)	wlk	sit	std	lyg		
	1	.930	.997	.848	.997		
	3	.936	.997	.857	.997		
	5	.936	.997	.856	.998		
H70 <sub>PGSD</sub>	1	.906	.968	.786	.941		
	3	.902	.969	.778	.942		
	5	.900	.967	.773	.938		

## Competing interests

The authors declare that they have no competing interests.

## Availability of data and materials

Codes and data can be made available upon request or through the repository available at <https://github.com/LABORA-INF-UFG/DS-FL>.

## References

- Amannejad, Y. (2020). Building and Evaluating Federated Models for Edge Computing. In *2020 16th International Conference on Network and Service Management (CNSM)*, pages 1–5. DOI: 10.23919/CNSM50824.2020.9269105.



- Asad, M., Moustafa, A., Ito, T., and Aslam, M. (2020). Evaluating the Communication Efficiency in Federated Learning Algorithms. *CoRR*, abs/2004.02738. DOI: 10.48550/arXiv.2004.02738.
- Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., and Lane, N. D. (2020). Flower: A Friendly Federated Learning Research Framework. *CoRR*, abs/2007.14390. DOI: 10.48550/arXiv.2007.14390.
- Brecko, A., Kajati, E., Koziorek, J., and Zolotova, I. (2022). Federated Learning for Edge Computing: A Survey. *Applied Sciences*, 12(18). DOI: 10.3390/app12189124.
- 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*, 25(4):2892–2950. DOI: 10.1109/COMST.2023.3316615.
- Haller, M., Lenz, C., Nachtigall, R., Awayshehl, F. M., and Alawadi, S. (2023). Handling Non-IID Data in Federated Learning: An Experimental Evaluation Towards Unified Metrics. In *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing*, pages 0762–0770. DOI: 10.1109/DASC/PiCom/CBDCCom/Cy59711.2023.10361408.
- Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., and He, B. (2021). A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1. DOI: 10.1109/TKDE.2021.3124599.
- Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. (2020). On the Convergence of Fedavg on Non-IID Data. *arXiv*. DOI: 10.48550/arXiv.1907.02189.
- Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., and Miao, C. (2020). Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Communications Surveys Tutorials*, 22(3):2031–2063. DOI: 10.1109/COMST.2020.2986024.
- Ma, X., Zhu, J., Lin, Z., Chen, S., and Qin, Y. (2022). A state-of-the-art Survey on Solving non-IID Data in Federated Learning. *Future Generation Computer Systems*, 135:244–258. DOI: 10.1016/j.future.2022.05.003.
- Malekzadeh, M., Clegg, R. G., Cavallaro, A., and Haddadi, H. (2019). Mobile Sensor Data Anonymization. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, pages 49–58, New York, NY, USA. ACM. DOI: 10.48550/arXiv.1810.11546.
- McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2016). Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv*. DOI: 10.48550/arXiv.1602.05629.
- Nakayama, K. and Jeno, G. (2022). *Federated Learning with Python: Design and implement a federated learning system and develop applications using existing frameworks*. Packt Publishing. Book.
- Quan, P. K., Kundroo, M., and Kim, T. (2023). Experimental Evaluation and Analysis of Federated Learning in Edge Computing Environments. *IEEE Access*, 11:33628–33639. DOI: 10.1109/ACCESS.2023.3262945.
- Saddaf Khan, N., Qadir, S., Anjum, G., and Uddin, N. (2024). Stressense: Real-time detection of stress-displaying behaviors. *International Journal of Medical Informatics*, 185:105401. DOI: <https://doi.org/10.1016/j.ijmedinf.2024.105401>.
- Seo, E., Pham, V., and Elmroth, E. (2023). Accelerating Convergence in Wireless Federated Learning by Sharing Marginal Data. In *2023 International Conference on Information Networking (ICOIN)*. DOI: 10.1109/ICOIN56518.2023.10048937.
- Shao, J., Li, Z., Sun, W., Zhou, T., Sun, Y., Liu, L., Lin, Z., Mao, Y., and Zhang, J. (2024). A Survey of What to Share in Federated Learning: Perspectives on Model Utility, Privacy Leakage, and Communication Efficiency. DOI: 10.48550/arXiv.2307.10655.
- Sirohi, D., Kumar, N., Rana, P. S., Tanwar, S., Iqbal, R., and Hijjii, M. (2023). Federated Learning for 6G-enabled Secure Communication Systems: A Comprehensive Survey. *Artificial Intelligence Review*. DOI: 10.1007/s10462-023-10417-3.
- Sozinov, K., Vlassov, V., and Girdzijauskas, S. (2018). Human Activity Recognition using Federated Learning. In *2018 IEEE Intl Conf on Parallel Distributed Processing with Applications*, pages 1103–1111. DOI: 10.1109/BDCLOUD.2018.00164.
- Ustad, A., Logacjov, A., Trollebo, S. O., Thingstad, P., Vereijken, B., Bach, K., and Maroni, N. S. (2023). Validation of an activity type recognition model classifying daily physical behavior in older adults: The har70+ model. *Sensors*, 23(5). DOI: 10.3390/s23052368.
- Wu, J., Dong, F., Leung, H., Zhu, Z., Zhou, J., and Drew, S. (2024). Topology-aware Federated Learning in Edge Computing: A Comprehensive Survey. *ACM Comput. Surv.*, 56(10). DOI: 10.1145/3659205.
- Yang, Z., Chen, M., Wong, K.-K., Poor, H. V., and Cui, S. (2022). Federated Learning for 6G: Applications, Challenges, and Opportunities. *Engineering*, 8:33–41. DOI: 10.1016/j.eng.2021.12.002.
- Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2022). Federated Learning with Non-IID Data. *arXiv*. DOI: 10.48550/ARXIV.1806.00582.