BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi" din Iaşi
Volumul 68 (72), Numărul 4, 2022
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ
DOI:10.2478/bipie-2022-0019

A BRIEF OVERVIEW OF FEDERATED LEARNING - A NEW PERSPECTIVE ON DATA PRIVACY

BY

IULIANA-ALEXANDRA LIPOVANU*, CARLOS PASCAL and CONSTANTIN-FLORIN CĂRUNTU

"Gheorghe Asachi" Technical University of Iași, Faculty of Automatic Control and Computer Engineering, 27 Mangeron Blvd., 700050, Iași, Romania

Received: February 14, 2023
Accepted for publication: April 11, 2023

Abstract. While privacy concerns remain the main challenge starting with the promulgation of the General Data Protection Regulation (GDPR), for deep learning applications, Google introduced recently the Federated Learning (FL) technique to offer support for high privacy-sensitive data, which makes FL a hot research topic nowadays. Thus, it is a distributed machine learning technique in which multiple devices (clients) collaboratively train a global model to solve issues where the first concern is data privacy. This work provides a brief study of FL: an overview of this new topic, related works, a comparison with other machine learning techniques, an overview of algorithms that are currently used, and, in the end, some simulation results and new directions of research. The simulations show the distributed behavior of the FL algorithm and the way in which the Federated Averaging method can be applied. Through the performed analysis of the obtained results, it was figured out that approach would be beneficial for several applications in domains like automotive, 5G and others.

Keywords: Privacy protection, Federated Averaging, Deep Learning, Collaborative Artificial Intelligence, TensorFlow.

*Corresponding author; *e-mail*: iuliana-alexandra.bejenar@academic.tuiasi.ro

1. Introduction

Nowadays, real-world systems increasingly use machine learning techniques to detect anomalies (Pang *et al.*, 2021) and make real-time decisions (Zhang *et al.*, 2008). Various applications are developed in industrial engineering using deep learning (DL), machine learning (ML), and artificial intelligence (AI). The success of these perspectives, in particular for deep learning, was provided by the availability of vast amounts of data. Using these data, DL can execute different tasks that can sometimes exceed human performance. The procedure to collect data is time-consuming, and it is not easy to collect all the needed information about the problem under discussion.

Starting with the promulgation of the General Data Protection Regulation (GDPR) (EU, 2018), the organizations do not have access to users' data without agreement. Each user is the owner of their data. According to that, the applications, which were developed after GDPR, must have major attention to persevering the private character of raw data. Under this new promulgation, collecting and sharing data among different organizations became increasingly difficult. An idea to solve the problem of collecting all data in one place is to train a model at each location where the data is obtained, and then to communicate the respective model intending to create a new global model. Thus, to ensure user privacy and data confidentiality, the communication is made without sharing data between their sources. The global model is built as if the data sources were combined. This is the proposal to preserve the privacy character, and it is named "Federated Machine Learning" or "Federated Learning". Federated Learning (FL) is a ML technique that can handle privacy concerns and can improve the functionality of the applications (Konečný *et al.*, 2016). This approach fixes the issue of private data of deep learning applications using a global shared model.

Federated Learning means creating a ML model based on data located at multiple locations, and it includes two significant steps: model training and model inference. Model training assumes that the information (parameters of the training procedure for local data) can be exchanged between clients (organizations, companies that provide data), but not the raw data. At inference time, if the result is not the expected result, the model is applied again to a new data instance.

The most common and well-known application which uses the FL technique is an application for mobile devices which provides the next word prediction based on users' historical text data without leaking the private information (Hard *et al.*, 2018). Mobile devices are not the only ones that could use this concept; the approach is also applied in several applications like Internet of Things (IoT) applications (Wu *et al.*, 2020), automotive (Saputra *et al.*, 2019; Liu *et al.*, 2020), or smart healthcare (Rieke *et al.*, 2020).

Firstly, the purpose of this paper is to put forward a brief overview of FL using some related works and to do a short analysis between this perspective and other machine learning techniques. Another goal is to show the grouping of the FL method and to introduce the open-source frameworks which can be used for applications. Secondly, the target is to offer a good understanding of the FL algorithm, which is used to obtain some experimental results. The results were obtained using the MNIST dataset of handwritten digits to investigate the distributed behavior of the FL algorithm and the Federated Averaging method. In the end, some conclusions and new directions for research are presented.

2. Background of Federated Learning

This section contains a short overview of the concept, a discussion of some related works in which FL remains a new approach, and a presentation of the grouping of Federated Learning. After that, the section continues with a short description of the algorithm and some open-source frameworks that can be used.

2.1. Brief overview of Federated Learning

Federated Learning was introduced by Google (McMahan *et al.*, 2017) for supporting the data privacy-sensitive character. The initial work on this area was for mobile devices, but then it was extended and now FL could be included in various applications intended to improve their functionality, while keeping privacy and data confidentiality.

FL is a machine learning technique where multiple devices (clients) collaborate, under the coordination of a central server for solving a machine learning issue in which the first concern is about data privacy, by GDPR (Konečný *et al.*, 2016). Client-side data are independent and they are not distributed between devices (Li *et al.*, 2020). In Fig. 1, one can observe that the devices communicate with a central server to learn a global model. For simplicity, a typical communication round consists of the following steps:

1. **S1**: Model initialization - Each client receives the current model from the server;
2. **S2**: Local model update, training, and upload - Every client uses the local data to compute and update the received model; each updated model is sent to the server;
3. **S3**: Global model aggregation and update - The server receives the updated models and aggregates them; it improves the global model and prepares it for the next round of communication.

These 3 steps are repeated until the expected model is obtained or a stopping criterion is fulfilled. As can be seen according to this figure, the privacy between devices is preserved, because each device communicates only with the server and the information which is sent to the server are the parameters from every local training model (Li *et al*., 2020; Nilsson *et al*., 2018).

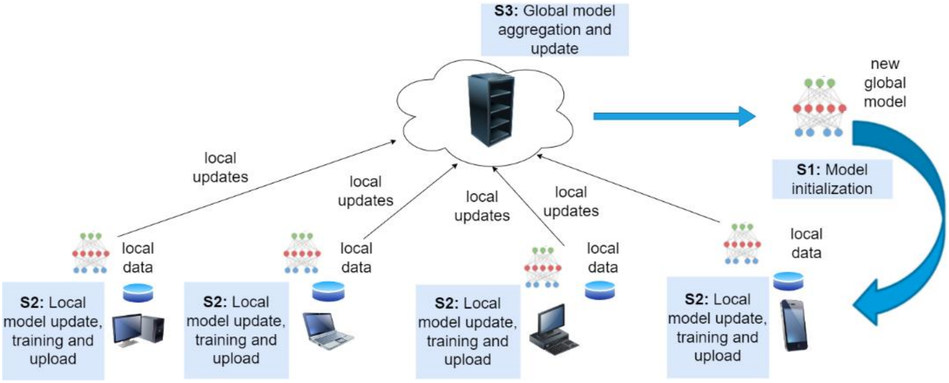

S3: Global model aggregation and update
new global model
S1: Model initialization
local updates
local updates
local updates
local updates
local data
local data
local data
local data
S2: Local model update, training and upload
S2: Local model update, training and upload
S2: Local model update, training and upload
S2: Local model update, training and upload

Fig. 1 – Illustration of FL process.

Meanwhile, the FL approach has the advantage over machine learning techniques because it respects data privacy. Another advantage is related to the communicated model from each client because they are immediately discarded after being merged into the global model.

FL is an active and ongoing area of research and often is compared with other ML techniques. Next, some papers are described to provide a good understanding of FL and also to present some applications which use this new approach. In (Niknam *et al*., 2020), the author tries to provide an accessible introduction to the general idea of FL, to introduce a short overview of possible applications in 5G networks, and finally to describe open problems and future research on Federated Learning in the context of wireless communications – the research on this new area is still in its early stage and it remains a new future direction for research. (Hsieh, 2019) and (Li *et al*., 2021) offers a large amount of information on this new approach, future directions, and also challenges in this domain. In (Liu *et al*., 2020), an application based on FL and 5G networks was developed. In this work, besides using FL for privacy and security, two critical threats about poisoning and membership inference attacks are presented. These attacks could be provided by malicious or unreliable participants, failing global models, or leakage of FL models. Model protection over the training process is another challenge and solutions from blockchain technologies are often proposed. Google reached out to train and deploy a logistic regression model for its keyboard query suggestion service without access to underlying user data (Yang *et al*., 2018). The authors of (Rieke *et al*., 2020) endorse a collaborative training approach in order to deal with the sensitive nature and diversity of medical data; healthcare data is highly sensitive, diverse, and raises other quality concerns. Mobile crowdsensing tasks have the potential to overtake data privacy, communication costs, and training efficiency in federated learning solutions for smart cities (Jiang *et al*., 2020).

Based on (Li *et al*., 2020), this section continues with a description of the grouping of Federated Learning. The FL approach is split into three groups: horizontally, vertically, and transfer Federated Learning, based on distribution characteristics of the data, according to Table 1.

Table 1
Categorization of FL approach

Name of FL group	**Description**
Horizontal FL (sample-based FL)	Datasets share the same space of characteristics, but each client has a different number of samples (Fig. 2 a)
Vertical FL (feature-based FL)	Datasets share the same sample number but differ in the space of characteristics (Fig. 2 b)
Transfer Learning	Datasets differ in the space of characteristics and also for samples (Fig. 2 c)

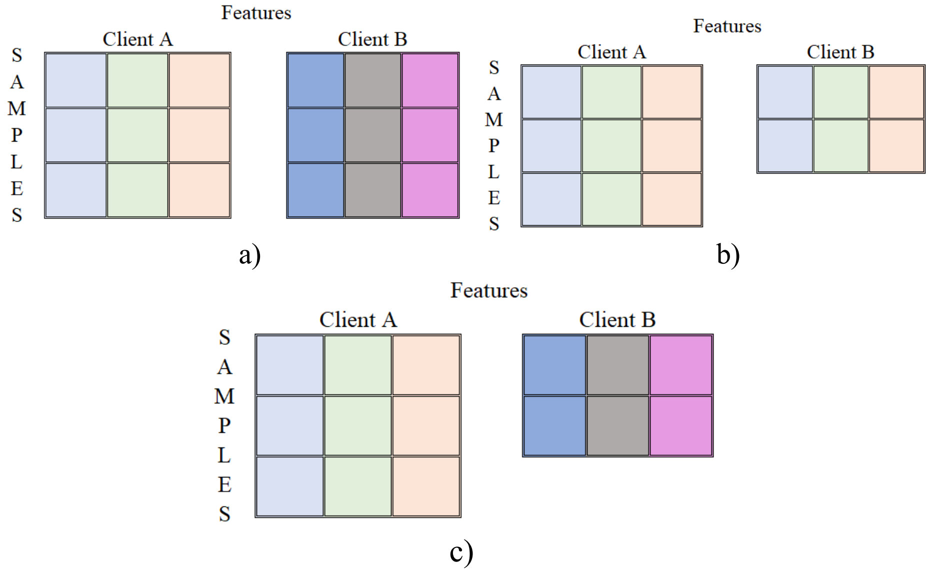


Fig. 2 – Categorization of FL: (a) horizontal FL, (b) vertical FL, (c) transfer learning.

2.2. Open-source Frameworks

To evaluate the FL approach, the most common framework used is Google's TensorFlow (Abadi *et al*., 2016), which is often used for developing machine learning algorithms. TensorFlow Federated (TFF) implements this new FL perspective and it enables multiple participating devices to train shared machine learning models while keeping their data locally.

Another open-source framework is Flower (Beutel *et al*., 2021), which supports heterogeneous environments and a large number of distributed devices. It is a framework for developing federated learning systems and it can also be used with any machine learning framework (e.g., TensorFlow). It is mainly used for research projects focusing on Federated Learning. The main goal of the Flower framework is to create a new framework and test the experiments from research projects using a large number of clients. The third framework to develop Federated Learning applications is LEAF (Caldas *et al*., 2018). It contains a suite of open-source datasets which was used to provide statistics. The plan for LEAF is to add datasets from different areas and to increase the variety of machine learning tasks.

All frameworks include also open-source federated datasets and are open to keeping frameworks up to date with new datasets and also for open-source solutions to make progress in the new area of Federated Learning. With current research projects, the Flower framework is the only one that can include heterogeneous clients. Having an existing implementation using different frameworks, the research progress is accelerated. The number of existing datasets included in an open-source framework is an advantage for research. If a comparison is done between these frameworks, LEAF comes with a greater number of different included datasets.

As can be seen, until now in the project's implementation of FL, each federated client has its own data, and data is not shared between clients. The first step in a FL developing algorithm, i.e., preparing the input data, is to define the clients and their data. After that, if it is needed, the next step is to pre-process the input data for training using special functions. For every client, a local model is created in order to train data. After the local model is updated for each device, the model is uploaded to the server where is done the global model aggregation. The most common algorithm to create the aggregation between the models received from clients is Federated Averaging (Zhang *et al*., 2020; Wahab *et al*., 2021).

2.3. Federated Learning algorithm

Federated Averaging (FedAvg) algorithm is described (McMahan *et al*., 2016) and (McMahan *et al*., 2017), and the author of these papers uses some experiments to show that models which are built using FedAvg can be trained using relatively few rounds of communication to obtain good models. The quality of the models obtained using FedAvg were demonstrated by results on multiple model architectures. The pseudocode of the Federated Averaging algorithm (McMahan *et al*., 2016; McMahan *et al*., 2017) is defined in Table 2, in which three main parameters exist: C - the fraction of clients used at each iteration; E - the number of training passes from each client using the local data on every round; B - local batch size used at each learning iteration.

Table 2

FL algorithm - Federated Averaging

Steps for Federated Averaging algorithm

```
Server executes:
    initialize $\omega_0$
    for each round $t = 1, 2, \ldots, T$ do
        $S_t \leftarrow$ (random set of [$K * C$] clients) // Select clients to compute updates and wait for updates from $K$ clients (indexed $1, \ldots, K$)
        for each client k ∈ $S_t$ in parallel do
            $\omega_{t+1}^{k} \leftarrow$ ClientUpdate($k$, $\omega_t$) // function client updates
        $\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^{k}$ // Average update

ClientUpdate($k$, $w$):
    $\beta \leftarrow$ (split $n_k$ into batches of size $B$) //local data is divided into minibatches
    for each local epoch $i$ from 1 to $E$ do
        for batch $b \in B$ do
            $\omega \leftarrow \omega - \eta \nabla \ell(\omega; b)$ // weighted update
    return $\omega$ to server
```

The first step in the FedAvg algorithm is the initialization step with initial weights (ω_0), which are randomly selected or are obtained by pretraining public data, as can be seen in Table 2. After the initialization step, the parameters of the server and client's devices communicate with each other during some communication rounds. When the number of the communication rounds (T) is too large, i.e., there are too many exchanges of information between clients and global server, the process of averaging may be influenced by the instability of the network. A round at time t in [1, . . . T] is described below:

- The global model is shared with a subset of clients (S_t) that are randomly selected from the group of clients (K), given the fraction of clients (C);
- Every client (k) has one or several training steps (E) on their local data (n_k), based on the division into minibatches (β) of the local data;

- For each minibatch of data (b), using a fixed learning rate (η), each client computes the average gradient on its local data at the current model ω, $\nabla\ell(\omega; b)$; in this way an updated local model is created, $\omega \leftarrow \omega - \eta\nabla\ell(\omega; b)$;
- The clients send back their model updates, once the local training is finished;
- The server computes an average model based on the client's updates $\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^{k}$, where during communication round t, n_k is defined as the number of samples used by client k, and n is defined as the total number of samples used by all clients.

The Federated Averaging algorithm started from the method of stochastic gradient descent (SGD) (McMahan *et al.*, 2016; McMahan *et al.*, 2017), which is an iterative method for optimization. The recent successful applications of deep learning have SGD as an optimization algorithm. As one can see in the steps of the FedAvg algorithm, each client uses the SGD algorithm on the mini-batches selected by the algorithm from the training dataset.

3. Experimental results

In this section, some simulation experiments using the MNIST digit dataset (Modified National Institute of Standards and Technology dataset) (LeCun, 1998) are defined. This dataset consists of images with size 28x28, and each class is kept in a different folder. The images are divided into 90% used for training and 10% used for testing the trained global model. All experiments were performed using Google's TensorFlow framework. The study uses a 3-layer MLP architecture, where activation functions are applied to the first two layers using a rectified linear unit (ReLU), and at the third layer using a SoftMax. The authors of (Tijani *et al.*, 2021) used a 2 -layer MLP architecture during experiments with the MNIST dataset, too. Based on the assumption that all clients are active, we conduct experiments with 10 clients during 100 rounds of communication.

Table 3 contains a short description of each considered scenario. The metrics to analyse the performances are accuracy and loss. Accuracy is the fraction of predictions our model got right and the loss (a value between 0 and 1), also known as a cost function, it is a method of evaluating how well specific algorithm models the given data. Accuracy and loss have different definition and are inversely proportional.

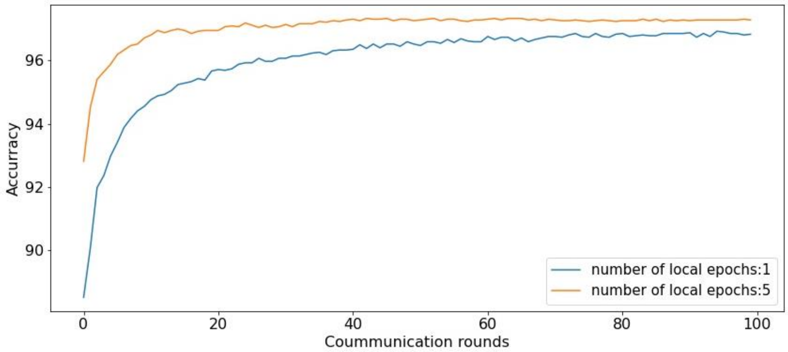
The experiments were performed to further validate the Federated Learning algorithm, which is able to connect machine learning models from different locations and using different data and more importantly, without leaking the privacy of clients' data.

Table 3
Description of considered experiments

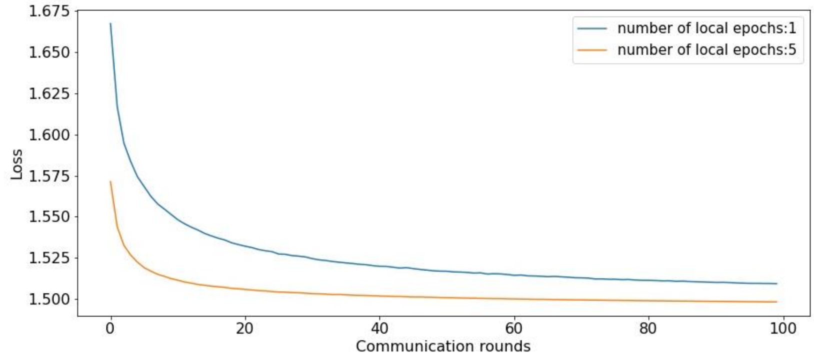
Id	Short description of the experiment	What is followed
1	Learning the local models using a different number of epochs	How does it influence the performances of the global model
2	Learning the models using a different number of epochs and different batch size	
3	The first two clients use less data than the rest of the clients	How does the amount of data used for training the local model could influence the performances of the global model and how could it influence the performances of the rest of the clients
4	Use different batch sizes for all clients	How does it influence the performances of the global model

3.1. Learning the local models using a different number of epochs

This experiment was performed to observe how the number of epochs from learning the local model could influence the performances of the global mode, see Fig. 3. The global model performance obtained during communication rounds (0, 25, 50, 75, 100) is shown in Table 4. With yellow color are presented the situations (number of local epochs is 1 and communication round is 0, number of local epochs is 1 and communication round is 100) when lower performance values for the global model are obtained, and the situations (number of local epochs is 5 and communication round is 0, number of local epochs is 5 and communication round is 100) when the best performance values are obtained are presented with green.



a)



b)

Fig. 3 – Performance values for the global model: a) Global model's accuracy, b) Global model's loss.

The performances of the global model are plotted in Fig. 3, and they illustrate that if the number of the local epochs is higher, i.e., 5, the obtained global model it's better than if the number of the local epochs is lower, i.e., 1. Thus, the performances of the global model are influenced by the number of the local epochs which is used for learning the local models.

Table 4

The performance values for the global model

Communication rounds	**Number of the local epochs**			
	1		**5**	
	Accuracy	**Loss**	**Accuracy**	**Loss**
0	88.5	1.69	92.80	1.58
25	95.92	1.52	97.11	1.50
50	96.47	1.51	97.28	1.50
75	96.76	1.51	97.26	1.49
100	96.83	1.50	97.28	1.49

3.2. Learning the models using a different number of epochs and different batch size

The experiment was performed using a different number of epochs (E) and a different batch size (B). Tables 5 and 6 show performance values based on different configurations of epochs and batch sizes.

The values for batch size were equal to 10, 30, and 50. The dataset which is considered in this experiment contains 42000 images and, from these, 90% are for training, that is, 37800 images. If the value for batch size for example is 10 and the number of clients is 10, each client uses into a communication round for learning (37800÷10)÷10 = 3780 images, which are randomly selected. In the tables, with yellow colour are represented the lowest values which demonstrate that if the value for batch size is increased the global model performance is decreased. This conclusion is given also by the graphic representation, from Fig. 4, where the global model performances during 100 communication rounds are plotted.

Table 5

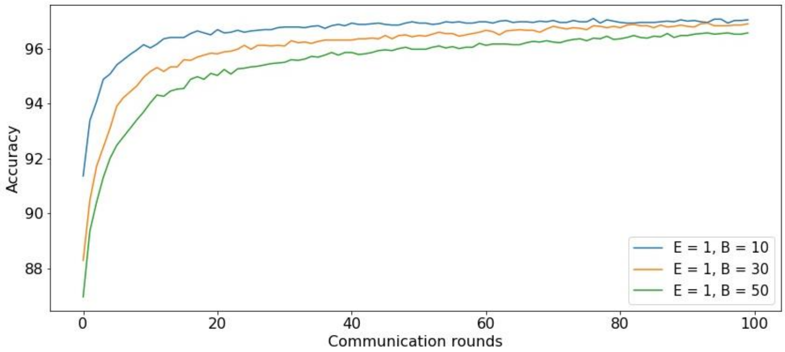
Accuracy – performance values from the global model

B	**E**	**Communication rounds**					
		10	**20**	**40**	**60**	**80**	**100**
10	1	96.14	96.50	96.83	96.97	96.97	97.04
10	5	**96.73**	97.02	97.09	97.19	97.14	**97.16**
30	1	94.95	95.83	96.30	96.59	96.80	96.90
30	5	96.33	96.64	97.00	97.07	97.21	97.21
50	1	**93.69**	95.09	95.85	96.19	96.33	**96.57**
50	5	95.80	96.54	96.92	96.92	97.11	97.26

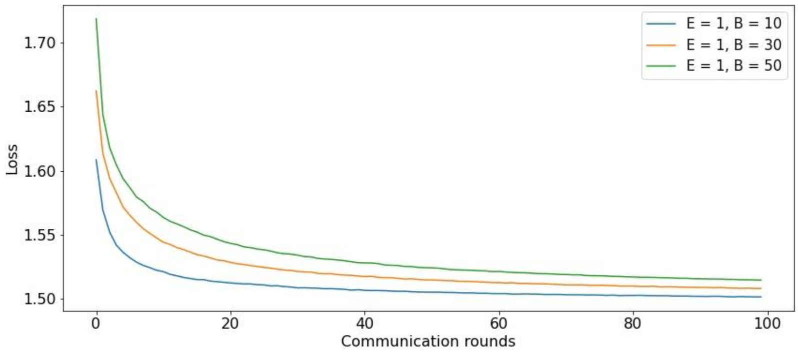
Table 6

Loss – performance values from the global model

B	**E**	**Communication rounds**					
		10	**20**	**40**	**60**	**80**	**100**
10	1	1.52	1.51	1.50	1.50	1.50	1.50
10	5	1.50	1.50	1.49	1.49	1.49	1.49
30	1	1.53	1.52	1.51	1.51	1.50	1.50
30	5	1.51	1.50	1.50	1.50	1.50	1.49
50	1	1.56	1.54	1.52	1.52	1.51	1.51
50	5	1.52	1.51	1.51	1.50	1.50	1.50



a)



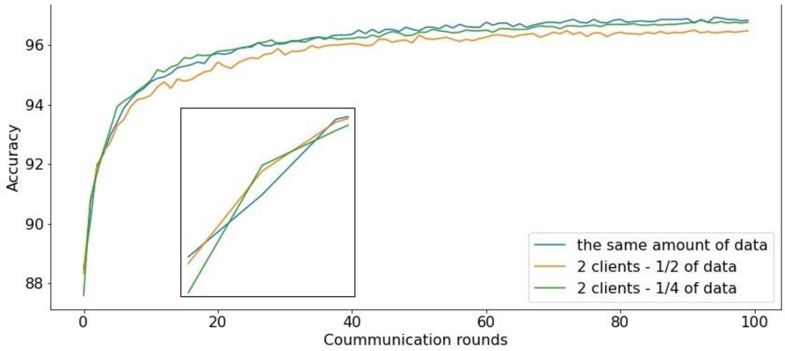
b)

Fig. 4 – Performance values for the global model: a) global model’s accuracy, b) global model’s loss.

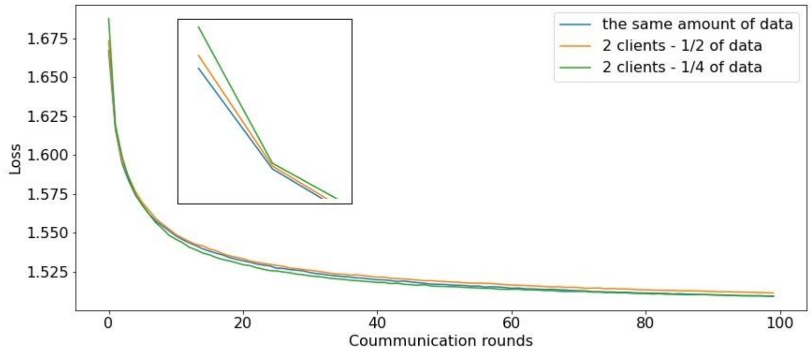
If a brief analysis is done using the information from the above tables, it can be said that a global model with good performances is obtained when the batch size is smaller, i.e., 10, and the number of epochs for local training is higher, i.e., equal to 5, the resulted accuracy for the model being 97.16 and the loss 1.49, which are the best values, at the end of the 100 communication rounds.

3.3. The first two clients use less data than the rest of the clients

This experiment was performed to see how the performances of the global model are influenced if some clients have less data than the rest of the clients, Fig. 5 and Fig. 6. Less data for training a model most often means low performance, and, in this experiment, we investigated if the global model is affected by two clients which don’t have good results.



a)



b)

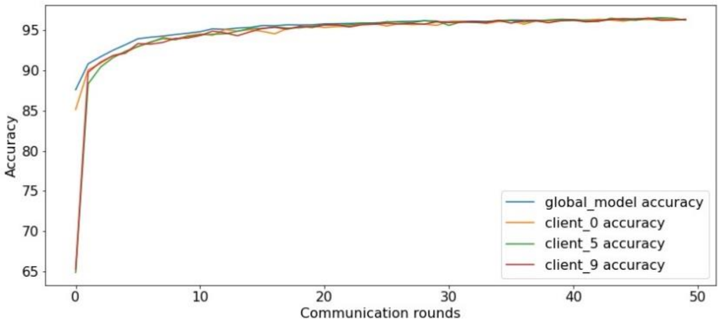
Fig. 5 – Global model performances where the first clients use less data: a) global model's accuracy, b) global model's loss.

Results from Table 7 indicate that the global model is affected in the first round of communication, and the performances are increased after the first aggregation of local models. Additionally, the small data used by some clients have a similar effect over the global model. As Figs. 5 and 6 illustrate, the aggregation method contributes to the re-balancing of local and global models, too.

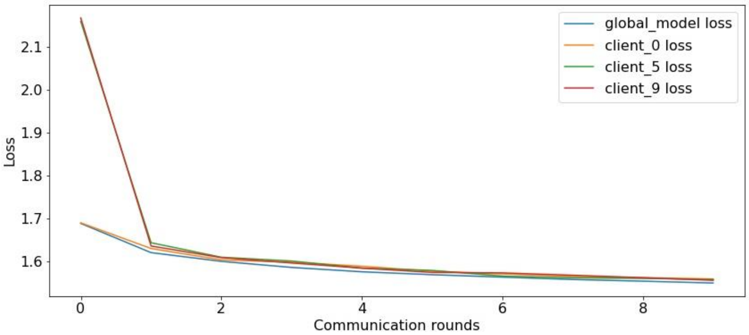
Table 7

Performance values using different amounts of data

Communication rounds	Performances analysis using different amounts of data					
	Equal data		½ of data		¼ of data	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
0	88.5	1.66	88.33	1.67	87.59	1.68
25	95.92	1.52	95.57	1.52	95.97	1.52
50	96.47	1.51	96.33	1.51	96.42	1.51
75	96.73	1.51	96.28	1.51	96.64	1.51
100	96.83	1.50	96.47	1.51	96.76	1.50



a)



b)

Fig. 6 – Local model performances from the clients which use less data:
a) accuracy, b) loss.

3.4. Use different batch sizes for all clients

In this section, the experiment was performed using different batch sizes for all clients. The value for batch size for client 0 is 10, for clients 1, 2 is 20, for clients 3 – 5 is 30 and for clients 5 – 9 is 50. This agreement means that each client uses into a communication round for learning a different number of images. Some performance values are extracted in Tables 8 and 9, in which it can be observed that in the first round of the communication the performance values are lower than the performance values from the next rounds.

Table 8

Accuracy performance values using different batch sizes for all clients

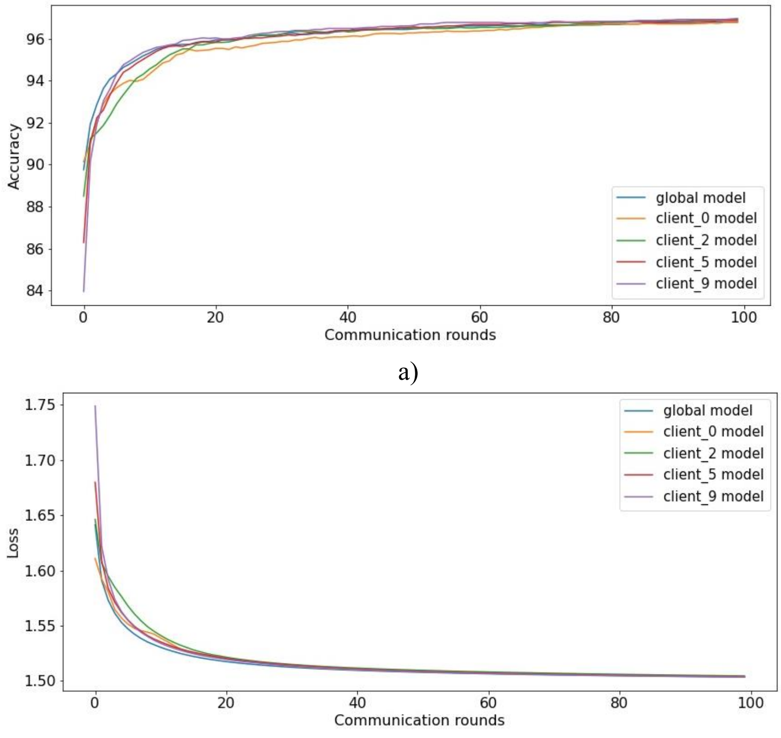
Client Id	**B**	**Accuracy performance on different communication rounds**					
		1	**20**	**40**	**60**	**80**	**100**
0	10	90.14	95.54	96.11	96.38	96.71	96.78
1 - **2**	20	88.5	95.83	96.33	96.54	96.78	96.92
3 - **5**	30	86.28	95.85	96.40	96.69	96.80	96.88
5 - **9**	50	83.95	96.04	96.50	96.78	96.83	96.97
Global model		89.76	95.95	96.38	96.61	96.69	96.80

Table 9

Loss performance values using different batch sizes for all clients

Client Id	**B**	**Accuracy loss performance on different communication rounds**					
		1	**20**	**40**	**60**	**80**	**100**
0	10	1.61	1.52	1.51	1.50	1.50	1.50
1 - **2**	20	1.64	1.52	1.51	1.50	1.50	1.50
3 - **5**	30	1.67	1.52	1.51	1.50	1.50	1.50
5 - **9**	50	1.74	1.51	1.51	1.50	1.50	1.50
Global model		1.64	1.51	1.50	1.50	1.50	1.50

Thus, the aggregation method which is used helps all clients in order to obtain good performances even if the amount of data is not enough to obtain good results from the beginning. On top of that, the same conclusion which is drawn out from Tables 8 and 9, it can also be extracted from Fig. 7 where the performance values of the models are plotted.



a)

b)

Fig. 7 – Performance models using different batch sizes: a) accuracy, b) loss.

4. Discussions and Conclusions

In this work, the proposal was to present an overview of what is federated learning, a three-level classification of this new concept, and a presentation of the open-source framework that can be used to develop FL applications. In the last part of the work some experimented results are explained, trying to highlight the strong side of the federated learning algorithm.

Inspired by the previous federated works, it is provided a complete overview for existing FL systems, a comprehensive categorization of FL, and a description of FedAvg algorithm.

Starting from the introduction of the section 3, Table 3, where the experimental use cases were described, in this part of the work, Table 10 is developed to offer a short overview about extracted conclusions which are obtained from the experimental results.

Table 10

Conclusion after experimental results

Id	Short description of the experiment	Conclusions
1	Learning the local models using a different number of epochs	The number of local epochs influences the performances of the global model
2	Learning the models using a different number of epochs and different batch size	Good performances are obtained when data for training is sufficient (appropriate batch size)
3	The first two clients use less data than the rest of the clients	Aggregation method helps to obtain the desired results
4	Use different batch sizes for all clients	

The experimental results were performed using the MNIST dataset to evaluate the performances of the local models and global models in different situations. The experiments show that the best results were obtained when was used a corresponding batch size and number of epochs for the local dataset, from each client. Otherwise, it is observed that some clients could have a negative impact on the learning process. In these situations, a protection mechanism for the global model seems to be helpful. Furthermore, the experimental results communicate that the aggregation method helps to achieve the expected results at the end of the communication rounds, no matter if some clients do not have good results in the first communication rounds.

The interest in future work is on new aggregation methods by using protection mechanism based on the accuracy of locally trained models. A series of test scenarios that incorporate configuration properties from horizontal, vertical, and transfer learning will be used to evaluate the effectiveness of the proposals. The desired for research in this domain is supported by the fact that this new concept guarantees the security and privacy of data, which nowadays is an issue, since the GDPR promulgation.

REFERENCES

Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M., Kudlur M., Levenberg J., Monga R., Moore S., Murray D. G., Steiner B., Tucker P., Vasudevan V., Warden P., Wicke M., Yu Y., Zheng X., *TensorFlow: a system for large-scale machine learning,* In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16), USENIX Association, USA, pp. 265-283, 2016.

Beutel D.J., Topal T., Mathur A., Qiu X., Fernandez-Marques J., Gao Y., Sani L., Li KH., Parcollet T., de Gusmão P.P., Lane ND., *Flower: A friendly federated learning framework*, preprint arXiv:2007.14390, 2021.

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

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

Hsieh K., *Machine learning systems for highly-distributed and rapidly-growing data*, Ph.D. Dissertation, Carnegie Mellon University, 2019.

Jiang J. C., Kantarci B., Oktug S., Soyata T., *Federated learning in smart city sensing: Challenges and opportunities*, Sensors, 20(21), 6230, 2020.

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

LeCun Y., *The MNIST database of handwritten digits*, http://yann.lecun.com/exdb/mnist/, 1998.

Li L., Fan Y., Tse M., Lin KY., *A review of applications in federated learning*, Computers & Industrial Engineering, pp. 106854, 2020.

Li Q., Wen Z., Wu Z., Hu S., Wang N., Li Y., Liu X., He B., *A survey on federated learning systems: vision, hype, and reality for data privacy and protection*, IEEE Transactions on Knowledge and Data Engineering, 2021.

Li T., Sahu A.K., Talwalkar A., Smith V., *Federated learning: Challenges, methods, and future directions*, IEEE Signal Processing Magazine, vol 37, no. 3, pp. 50-60, 2020.

Liu Y., James J.Q., Kang J., Niyato D., Zhang S., *Privacy-preserving traffic flow prediction: A federated learning approach*, IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7751-7763, 2020.

Liu Y., Peng J., Kang J., Iliyasu A.M., Niyato D., Abd El-Latif A.A., *A secure federated learning framework for 5G networks*, IEEE Wireless Communications, vol. 27, no. 4, pp. 24-31, 2020.

McMahan H.B., Moore E., Ramage D., Arcas B.A., *Federated learning of deep networks using model averaging*, arXiv, vol. abs/1602.05629, 2016.

McMahan H.B., Moore E., Ramage D., Hampson S., y Arcas B.A., *Communication-efficient learning of deep networks from decentralized data*, In Artificial intelligence and statistics, pp. 1273-1282, 2017.

Nilsson A., Smith S., Ulm G., Gustavsson E., Jirstrand M., *A performance evaluation of federated learning algorithms*, InProceedings of the Second Workshop on Distributed Infrastructures for Deep Learning, pp. 1-8, 2018.

Niknam S., Dhillon HS., Reed JH., *Federated learning for wireless communications: Motivation, opportunities, and challenges*, IEEE Communications Magazine, vol. 58, no. 6, pp. 46-51, 2020.

Pang G., Shen C., Cao L., Hengel A.V., *Deep learning for anomaly detection: A review*, ACM Computing Surveys (CSUR), vol. 54, no. 2, pp. 1-38, 2021.

Rieke N., Hancox J., Li W., Milletari F., Roth H.R., Albarqouni S., Bakas S., Galtier M.N., Landman B.A., Maier-Hein K., Ourselin S., *The future of digital health with federated learning*, NPJ digital medicine, vol. 3, no. 1, pp. 1-7, 2020.

Saputra Y.M., Hoang D.T., Nguyen D.N., Dutkiewicz E., Mueck M.D., Srikanteswara S., *Energy demand prediction with federated learning for electric vehicle networks*, in 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, pp. 1-6, 2019.

Tijani S.A., Ma X., Zhang R., Jiang F., Doss R., *Federated Learning with Extreme Label Skew: A Data Extension Approach*, In 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2021.

Wahab O.A., Mourad A., Otrok H., Taleb T., *Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems*, IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1342-1397, 2021.

Wu Q., He K., Chen X., *Personalized federated learning for intelligent IoT applications: A cloud-edge based framework*, IEEE Open Journal of the Computer Society, vol. 1, pp. 35-44, 2020.

Yang T., Andrew G., Eichner H., Sun H., Li W., Kong N., Beaufays F., *Applied federated learning: Improving google keyboard query suggestions*, arXiv preprint arXiv:1812.02903, 2018.

Zhang N., Wang FY., Zhu F., Zhao D., Tang S., *DynaCAS:* Co*mputational experiments and decision support for ITS*, IEEE Intelligent Systems, vol. 23, no. 6, pp. 19-23, 2008.

Zhang W., Lu Q., Yu Q., Li Z., Liu Y., Lo S.K., Chen S., Xu X., Zhu L., *Blockchain-based federated learning for device failure detection in industrial IoT*, IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5926-5937, 2020.

O SCURTĂ PREZENTARE DESPRE ÎNVĂȚAREA FEDERALIZATĂ – O NOUĂ METODĂ PENTRU CONFIDENȚIALITATEA DATELOR

(Rezumat)

Începând cu promulgarea legii privind protecția datelor cu caracter personal (GDPR), pentru cele mai multe aplicații principala problemă rămâne cea legată de caracterul privat al datelor. Pentru această problemă, Google a introdus recent un nou concept numit Învățarea Federalizată (FL), fiind o tehnică care oferă suport în păstrarea confidențialității datelor, de unde rezultă că acest domeniu prezintă un interes ridicat în zilele noastre. FL este o tehnică de învățare automată distribuită unde mai mulți clienți colaborează pentru a obține un model global, unde prima grijă este caracterul privat al datelor. Această lucrare oferă o scurtă introducere în acest domeniu nou: o idee generală despre ceea ce înseamnă FL, prezentarea câtorva lucrări scrise în acest domeniu, o comparație cu alte tehnici de învățare automată, prezentarea algoritmului folosit, și în final câteva rezultate experimentale și noi direcții de cercetare. Simulările realizate evidențiază comportamentul distribuit al algoritmului FL și modul în care algoritmul de mediere poate fi folosit, pentru a agregarea modelelor locale. Prin intermediul studiului realizat în cadrul acestei lucrări, s-a observat că acest concept aduce beneficii în mai multe aplicații precum cele din domeniul auto, 5G și altele.