Dissertation for The Degree of Doctor of Philosophy

Privacy-preserving distributed artificial intelligence in connectionism-based models

Irina Arévalo

Supervisor: Prof. Dr. Jose L. Salmeron

Programa de Doctorado en Biotecnología, Ingeniería y Tecnología Química
Universidad Pablo de Olavide
Sevilla, Spain, 2024

"I don't think we have any alternative other than remaining optimistic. Optimism is an absolute necessity, even if it's only optimism of the will, as Gramsci said, and pessimism of the intellect"
- Angela Davis

"A mis padres
y a Marcos"
- 2017

"... y a Nuria y Víctor"
- 2024

Privacy-preserving distributed artificial intelligence in connectionism-based models

Irina Arévalo

Universidad Pablo de Olavide

Abstract

Federated learning is an emerging machine learning approach that allows the construction of a model between several participants who hold their own private data. In the initial proposal of federated learning the architecture was a centralized neural network, and the aggregation was done with federated averaging, meaning that a central server will orchestrate the federation using the most straightforward averaging strategy. In this thesis we discuss several advances in aggregation methods, encryption methods to ensure the privacy of the system, study of non-iid datasets, and federation of Fuzzy Cognitive Maps.

Keywords

Distributed Machine Learning, Federated Learning, Private Machine Learning

List of Publications

Appended publications

This thesis is based on the following publications:

[**Paper I**] J.L. Salmeron, **I. Arévalo**, A. Ruiz-Celma, *Benchmarking federated strategies in Peer-to-Peer Federated learning for biomedical data Heliyon, Volume 9, Issue 6 (2023), e16925.* 2022 Journal Impact Factor: 4.0. Rank 23/73 in MULTIDISCIPLINARY SCIENCES (Q2).

[**Paper II**] **I. Arévalo**, J.L. Salmeron, *A Chaotic Maps-based Privacy-preserving Distributed Deep Learning for Incomplete and Non-IID Datasets IEEE Transactions on Emerging Topics in Computing, ISSN 2168-6750.* 2022 Journal Impact Factor: 5.9. Rank 35/158 in COMPUTER SCIENCE, INFORMATION SYSTEMS (Q1).

[**Paper III**] J.L. Salmeron, **I. Arévalo**, *A Privacy-Preserving, Distributed and Cooperative FCM-Based Learning Approach for Cancer Research. In: Bello, R., Miao, D., Falcon, R., Nakata, M., Rosete, A., Ciucci, D. (eds) Rough Sets. IJCRS 2020. Lecture Notes in Computer Science, vol 12179, 477–487. Springer, Cham. Online ISBN 978-3-030-52705-1.* Conference with Rank C CORE.

Other publications

The following publications, related to the contents of this thesis, are currently in submission/under revision but weren't included due to time constraints.

[**a**] J.L. Salmeron, **I. Arévalo**, *Blind Federated Learning without initial model.*
Sent to publication.

[**b**] J.L. Salmeron, **I. Arévalo**, *Vertical and Horizontal Federated Learning with Square FL.*
Sent to publication.

Agradecimientos

Deseo expresar mi más profundo agradecimiento a todas las personas que me han apoyado durante la elaboración de esta tesis doctoral:

En primer lugar, gracias a mi director, el Profesor José Luis Salmerón, no solo por sus grandes conocimientos y su excelente orientación, sino también por su apoyo, en particular en algunos momentos en los que vi muy lejos el final de la tesis. Eres el mayor referente para mí.

Gracias también a todas las personas que me han ayudado en la Universidad Pablo de Olavide y en CUNEF.

En especial quiero agradecer a mi familia sus ánimos durante estos años. Gracias a mis padres, Pilar y Luciano, por todo lo que han hecho para que yo llegue hasta aquí. Gracias a mis suegros, Janet y José, por tratarme como su hija. Gracias a Ian, por ser uno de mis mejores amigos además de mi medio hermano. Gracias a todos mis amigos, a los que también considero de mi familia.

Y por último, gracias a los amores de mi vida, Marcos, Nuria y Víctor. Sois lo mejor que me ha pasado nunca.

Contents

List of Figures

List of Tables

List of Algorithms

Part I

Preliminaries

Chapter 1

Introduction

Federated learning is a distributed machine learning approach that allows several participants to train collaboratively a machine learning model in a private and secure way, and more importantly, without sharing any private data among them or with a central server. This proposal is particularly useful in highly regulated and private industries such as healthcare and finance given the raise of legal regulations all around the world such as the General Data Protection Regulation (GDPR) in the European Union [1], the Cybersecurity Law of the People's Republic (CLPR) of China [2], the Personal Data Protection Act (PDP) in Singapore [3], the California Consumer Privacy Act (CCPA) [4] and the Consumer Privacy Bill of Rights (CPBR) in the United States [5].

The initial process to train a federated learning model was developed by researchers at Google in [6] and is formed by a central server and several participants with private data who want to train a shared neural network model. The steps are as follows:

1. The central server defines the architecture of the neural network that will be trained by all participants and sends it.

2. The participants train the model using their private data, obtaining a local model, and then sending the parameters (weights or gradients of the model) to the server.

3. The central server aggregates the parameters of the model using an arithmetic mean. With this process the server builds a federated model using those aggregated parameters.

4. The central server checks the termination condition, for instance in terms of number of iterations or performance of the global model. If it is accomplished the federated model is finished, otherwise the server sends the federated model back to the participants in order to be retrained in the local data and the process is iterated.

Therefore, federated learning brings the code to the data, instead of other common solutions for distributed machine learning that brings the data to

the code, and addresses the fundamental problems of privacy, ownership and locality of data.

Federated learning also exploits parallelization techniques designed for distributed machine learning, such as data parallelism, which trains multiple instances of the same model on different subsets of the training dataset, or model parallelism, to distribute parallel paths of a single model to multiple devices in order to handle different features.

The first application of federated learning was to create collaborative predictive models using private data in Android mobile phones [7]. Since then it has been applied in many fields, such as personalized medicine [8], recommender systems [9], smart cities [10], finance and insurance [11], edge computing [12] and IOT [13] among others.

Nevertheless, there is always a risk associated with the data transmission, such as the possibility of the reconstruction of the model or the training data from the model parameters. Due to these risks, there is an increasing interest in the use of an additional layer of privacy to this information, and there are many studies that use privacy-preserving methods in federated learning such as Differential Privacy [14], Secure Multi-Party Computation [15] or Homomorphic Encryption [16].

Since its development in 2016, there have been many advances in every aspect of federated learning, such as its taxonomy, techniques, data privacy, efficiency, security, applications... Nevertheless, many authors conclude that research on current federated learning mainly faces four bottlenecks [17]:

1. settle the best techniques to optimize the performance of the federated model,
2. privacy and security threats, due to hidden dangers of parameter leakage and attacks by malicious operations,
3. heterogeneity challenges, since data distribution may vary widely between participants,
4. and huge communication overhead, since given large quantities of clients participating in a federated system the communication overhead is far greater than the computational overhead.

The research line this thesis has followed closely the first three items in the list above for connectionism-based models, a class of machine learning models often used to model aspects of human perception, cognition, and behavior, the learning processes underlying such behavior, and the storage and retrieval of information from memory, mainly neural networks and fuzzy cognitive maps.

1.1 Aggregation strategies in federated learning

Related to the technical setting of federated learning, we studied the impact of the aggregation strategy on the federated model. In the initial definition of

the federated learning approach, the aggregation step is done by averaging the model parameters using an arithmetic mean. Nevertheless, other aggregation methods may be of more interest since they can improve the performance of the model by giving more weight to different agents depending on their size or the performance of the local models in their data.

The main contributions of this research are two-fold:

1. Several aggregation strategies are proposed, such as weighted averaging aggregation using the dataset size, weighted average using the normalized inverse of the local test accuracy, weighted averaging aggregation using the dataset size and accuracy, weighted average using the contribution of the participant (difference between the losses of the local and the global models), and weighted sum using the inverse contribution of the participant. Federated averaging is included for comparison.

2. The strategies are tested with different data sizes on each participant. This allows analyzing the strategies under different circumstances and identifying those that are more robust.

Assuming that the parameters of the model at iteration j are,

$$\Phi_j = [\Phi_{j1}, \Phi_{j2}, \cdots, \Phi_{jn}] \tag{1.1}$$

where n is the number of participants, $\mathcal{D}_i$ is the dataset of the participant i, and Φ'_j is the parameters (weights or gradients) of the federated model, the functions of the parameters that we will discuss are the following:

- Average of the parameters (weights or gradients):

$$\Phi'_j = \frac{1}{n} \sum_{i=1}^{n} \Phi_{ji} \tag{1.2}$$

  where every participant contributes the same to the global model.This is the classical setting.

- Weighted averaging aggregation using the normalized size of each participants' dataset:

$$\Phi'_j = \sum_{i=1}^{n} \frac{|\mathcal{D}_i|}{\sum_{k=1}^{n} |\mathcal{D}_k|} \cdot \Phi_{ji} \tag{1.3}$$

  where every participant contributes to the global model proportionally to the size of their data, and agents with less information will affect less to the final model.

- Weighted average using the normalized inverse accuracy of the model in a test set of each participants:

$$\Phi'_j = \sum_{i=1}^{n} \frac{1/\mathrm{acc}_{ji}}{\sum_{k=1}^{n} 1/\mathrm{acc}_{jk}} \cdot \Phi_{ji} \tag{1.4}$$

  where the individual models add to the global model inversely to their performance metric, trying to give more weight to the less accurate models in order to improve their metric in their datasets.

- Weighted average using the accuracy and the size of the dataset:

$$\Phi'_j = \sum_{i=1}^{n} \frac{\text{acc}_{ji}|\mathcal{D}_i|}{\sum_{k=1}^{n}|\mathcal{D}_k|} \cdot \Phi_{ji} \tag{1.5}$$

  where the contribution of each participant's model depends on both the accuracy of the model and the size of the dataset.

- Weighted average using the contribution ($\mathcal{C}$) of the participant, that is, the normalized absolute difference between the loss of the participant's model and the loss of the global model when applied to the participants' data as shown in Equation 1.6

$$\mathcal{C}_{ji} = |\mathcal{L}^*_j(\mathcal{D}_i, \Phi) - \mathcal{L}_j(\mathcal{D}_i, \Phi)| \tag{1.6}$$

  and

$$\Phi'_j = \sum_{i=1}^{n} \frac{\mathcal{C}_{j,i}}{\sum_{k=1}^{n}\mathcal{C}_{jk}} \cdot \Phi_{ji} \tag{1.7}$$

- Weighted sum using the inverse contribution of the participant:

$$\Phi'_j = \sum_{i=1}^{n} \frac{1/\mathcal{C}_{ji}}{\sum_{k=1}^{n}1/\mathcal{C}_{jk}} \cdot \Phi_{ji} \tag{1.8}$$

In our experimental setup to test which aggregation method is optimal, we assume that several hospitals with their own private data wish to train a deep learning model (a dense neural network made of five layers followed by a non-linear ReLU function and a dropout layer for regularization with Binary Cross Entropy as loss function) for diagnosis of a disease, but the size of their data is not large enough for training an accurate model. The data of all the hospitals should not be combined due to data regulations given their special sensitivity, therefore they decide to participate in a federation process.

In a first test all of the hospitals will have the same amount of data. In the other three experiments they will have different amount of data, where in the last two we have forced that there would be several participants with a very small number of samples (less than 10%). The distribution of the variables, including the percentage of positive cases for the target, will also vary from one hospital to another to test the robustness of the aggregation methods.

We perform four different experiments (with four different datasets related to four different diseases). For each experiment the accuracy in the participant's test dataset is compared for each aggregation method discussed above. The results show that the classical federated averaging (arithmetic mean) is a reliable aggregation method that improves the performance of the local methods in 11 out of 16 cases that we have contemplated. Nevertheless, there are other aggregation methods with similar or even better behavior. The contribution-based aggregation, using the difference between the losses of the global and local model, increases the accuracy in 11 out of 16 cases as well, while the size-based and the inverse contribution-based perform better in one more

case. The weighted average using both the size of the participant's dataset and the accuracy of the local model increase the accuracy in only 10 cases out of 16. Finally, the weighted average using the accuracy outperforms all aggregation methods, improving the accuracy in 15 out of 16 cases, that is, in all experiments but one partition where all other methods failed to increase the performance as well.

With this results, we believe that an accuracy-based federated learning may perform better than the federated averaging classical approach. In subsequent research we want to investigate if this difference is related to the performance metric, the loss functions, or it is a general behavior.

1.2 Chaotic map encryption for non-IID federated learning

In this research we tackle the privacy problem in federated learning by developing a new encryption method based on chaotic maps. Then we compare the results of an unsecured federated model and a secured model built using differential privacy and chaotic maps as its encrypting layer. In each experiment this research proves that the federation process improves the averaged performance metrics of a deep neural network for the participants, with disregard of whether the data has been evenly split among them or there are differences between the amount of data each participant has, and that the performance with or without the privacy layer are similar, meaning that the additional security does not worsen the model's results thanks to the federation process.

Moreover, to ensure the simulation closely resembles a real-world implementation of federated learning, we have set a scenario where one of the participants possesses incomplete data with a distinct structure, leaning into another research interest: federated learning with data heterogeneity. This situation may arise, for example, when a variable is unavailable in the dataset of one participant. In such cases, this proposal enables other participants to privately share the distribution of the missing variable, allowing for the imputation of the missing data in the dataset of the participant who lacks that variable. The rational behind this proposal is that the federation process and its multiple iterations will average the model performance even in the case when one of the participant's features has been imputed.

The main contributions of this research are two fold:

- This proposed FL extension aims to handle datasets that are incomplete or contain missing values, as well as datasets that are non-IID (non-Independently and Identically Distributed). In certain scenarios, the data used in the FL process may have missing values, lack completeness or uneven distribution across the participating devices or nodes.

- An efficient and secure method for encrypting distributed models based on chaotic maps. Chaotic maps possess inherent complexity and unpredictability, which makes them resistant to conventional cryptographic attacks. Furthermore, their non-linear nature enhances their security and the

deterministic characteristics of chaotic maps make them an efficient encryption method.

The extension of federated learning proposed to combine the use of an additional encryption layer with non-IID datasets is as follows:

1. As a first step, a central server will send an untrained deep learning model to the participants.

2. If one of them does not have a complete dataset, meaning that one of the features is missing (and therefore the features are non-IID), the server will also send, in a encrypted fashion, the distribution of the feature for any other participant so that the lacking feature can be imputed.

3. Then all the participants will split their data into train/test/validation datasets and train the model in their training data. Then, they will send the parameters of the model back to the central server, using one of the three possible different approaches to this step:

   (a) the first one where the data is sent without any additional security measures,

   (b) the second where the data is encrypted using differential privacy to avoid privacy issues, in order to compare if the use of this privacy-preserving layer affects the results of the model,

   (c) and the third, where the data is encrypted using a chaotic map and the process ends with the decryption of the obfuscated data.

4. The server then aggregates the parameters of the local models to find a global model, and iterates this process to improve the global accuracy. As already discussed, the most common aggregation method is federated averaging, which is just a weighted average of the network weights across training sites.

The federated model is evaluated in each participant's test data. The convergence of the differential privacy process is guaranteed by the work in [18], where the authors apply differential privacy to federated learning.

For the experiments we will assume there are five different participants. In a first test all of them will have the same amount of data, obtained from an evenly split dataset, but we will also consider the case where each participant has a different dataset size.

In particular, we have performed three additional experiments with uneven splits: the first one will be a random split among the participants, but in the remaining two we have forced that there are several participants with a very small number of samples (less than 10% of the samples). The amount of positive cases will also vary from one participant to another. In any case, each participant's data will be split into a train and a test dataset. As it is customary, the models will be trained in each participant's train data, and the evaluation metrics will be obtained from each participant's test data.

The different experiments show that, even in the most imbalanced cases, the federated learning process improves the average metrics of the models, increasing their performance, both in accuracy and F1 score, and for both the private, non-private and encrypted approaches. We can also conclude that using an additional layer of encryption and ensuring the privacy of the process does not affect the performance metrics of the model, when compared with the non-private federated learning.

1.3 Federated Fuzzy Cognitive Maps

In recent years, there have been several attempts to create a federated version of conventional machine learning algorithms, such as federated linear regression [19]–[21], federated logistic regression [22], federated random forest [23], federated XGBoost [24]–[26], and federated support vector machines [27], [28].

The third research interest of this thesis is to develop a framework to federate Fuzzy Cognitive Maps. FCMs are an special case of Cognitive Maps [29] where the relationships between nodes are modeled with fuzzy cause-effect relations instead of the crisp cause-effect relations proposed by the original Cognitive Map [30]. The FCM are neuro-fuzzy dynamical systems built from expert knowledge and/or historical raw data

FCMs are composed by nodes modeling concepts or variables, edges representing relationships between them, and the weights quantifying the influence of those relations [31], [32], that is, the value of a FCM's fuzzy weight ϖ_{ij} describe the influence of node c_i over the node c_j.

They can be trained using optimization techniques, in our research Particle Swarm Optimization, to obtain a machine learning model that are inherently interpretable methods [33], standing in contrast to other techniques that require the application of Explainable Artificial Intelligence (XAIs) approaches for their explanation [34].

Our proposed methodology combines federated learning with learning FCMs using particle swarm optimization. The process is explained as follows.

1. Triggering of the federated learning process. The central server triggers the process in the participants machines.

2. Training FCM in the local dataset. Each participant trains a local FCM with their own dataset. We apply PSO but this methodology is agnostic to the learning approach. The FCM dynamics is considered steady when the difference between two consecutive vector states is under $tol = 0.00001$

3. Sending the trained adjacency matrices and local accuracy for this stage to the central server. The local FCM is stored in the participant devices.

4. Weighting local FCMs using accuracy. The central server aggregates the local FCMs weighting by the accuracy.

5. Aggregating federated and local FCMs. The participants aggregate the federated FCM from the central server and their own local FCM.

6. Sending adjacency matrices and accuracy. Participants send again the local adjacency matrices and the new local accuracy.

7. Checking termination condition. The central server checks if the federated process has been run 20 iterations as termination condition. If it is not accomplished then it goes back to the step 4.

8. If the termination condition is accomplished then a federated FCM is achieved.

The main contribution of our research in this topic is the application of federation learning paradigm for privacy-preserving FCM distributed and cooperative learning.

Our experimental results show that the Fuzzy Cognitive Map-based classifier is able to improve the accuracy of a single Fuzzy Cognitive Map trained in the whole data, and the accuracy in each participant before the federation.

The goal of this work is not the accuracy of the proposal but a distributed and privacy-preserving approach. Nevertheless, our performance results for this problem are similar to the ones found in literature [35].

Chapter 2

Introducción

Federated learning o aprendizaje federado es un nuevo enfoque de aprendizaje automático distribuido que permite a varios participantes entrenar colaborativamente un modelo de aprendizaje automático de una manera privada y segura, y más importante, sin compartir datos privados entre los participantes o con un servidor centrar. Esta propuesta es particularmente útil en industrias muy reguladas y privadas, como la salud o las finanzas, dado el aumento de regulaciones legales en todo el mundo, como el Reglamento General de Protección de Datos (GDPR) en la Unión Europea [1], la Ley de Ciberseguridad de la República Popular (CLPR) en China [2], la Ley de Protección de Datos Personales (PDP) en Singapur [3], la Ley de Privacidad del Consumidor de California (CCPA) [4] y la Ley de Derechos de Privacidad del Consumidor (CPBR) en Estados Unidos [5].

El proceso inicial para entrenar un modelo de aprendizaje federado fue desarrollado por investigadores en Google [6] y está formado por un servidor central y varios participantes con datos privados que quieren entrenar una red neuronal común. Los pasos son los siguientes:

1. El servidor central define la arquitectura de la red neuronal que será entrenada por todos los participantes y la envía.

2. Los participantes entrenan el modelo usando sus datos privados, con lo que obtienen un modelo local, y después envían los parámetros (pesos o gradientes de la red) al servidor.

3. El servidor central agrega los parámetros del modelo usando una media aritmética. Con este proceso el servidor construye un modelo federado usando esos parámetros agregados.

4. El servidor central comprueba la condición de terminación, por ejemplo en términos de número de iteraciones o rendimiento del modelo global. Si se consigue, el modelo federado ha sido completado, si no, el servidor envía el modelo federado de vuelta a los participantes para que lo reentrenen en sus datos locales y se itera el proceso.

Por lo tanto, el aprendizaje federado lleva el código a los datos, en lugar de otras soluciones habituales para el aprendizaje automático distribuido en el que se mandan los datos al código, y así responde a los problemas fundamentales de privacidad, propiedad y localización de los datos.

El aprendizaje federado también aprovecha las técnicas de paralelización diseñadas para el aprendizaje automático distribuido, como paralelización de datos, que entrena múltiples instancias del miso modelo en distintos subconjuntos del conjunto de entrenamiento, o paralelización de modelos, para distribuir trayectorias paralelas del modelo a múltiples dispositivos para trabajar con características de los datos.

La primera aplicación del aprendizaje federado fue la creación de modelos predictivos colaborativos utilizando datos privados en teléfonos móviles Android [7]. Desde entonces, se ha aplicado ampliamente en diversos campos, como la medicina personalizada [8], sistemas de recomendación [9], ciudades inteligentes [10], finanzas y seguros [11], edge computing [12] e Internet de las cosas (IoT) [13] entre otros.

Sin embargo, siempre existen riesgos asociados con la transmisión de datos, como la posibilidad de reconstruir el modelo o los datos de entrenamiento a partir de los parámetros del modelo. Debido a estos riesgos, hay un interés creciente en el uso de una capa adicional de privacidad para esta información, y existen muchos estudios que emplean métodos de preservación de la privacidad en el aprendizaje federado, como la Privacidad Diferencial (Differential Privacy) [14], la Computación Segura entre Partes (Secure Multi-Party Computation) [15] o la encriptación homomórfica (homomorphic encryption) [16].

Desde su desarrollo en 2016 ha habido muchos avances en todos los aspectos del aprendizaje federado, como en su taxonomía, técnicas, privacidad de datos, eficiencia, seguridad y aplicaciones, entre otros. Sin embargo, muchos autores concluyen que la investigación en el aprendizaje federado actual enfrenta principalmente cuatro cuellos de botella [17]:

1. establecer las mejores técnicas para optimizar el rendimiento del modelo federado,
2. amenazas de privacidad y seguridad debido a los peligros ocultos de la fuga de parámetros y ataques de operaciones maliciosas,
3. desafíos de heterogeneidad de los datos, ya que su distribución puede variar ampliamente entre los participantes,
4. enorme sobrecarga de comunicación, ya que, dada la gran cantidad de clientes participantes en un sistema federado, la sobrecarga de comunicación es mucho mayor que la sobrecarga computacional.

La línea de investigación de esta tesis ha abordado de cerca los tres primeros puntos mencionados en la lista anterior para modelos basados en conexionismo, una clase de modelos de aprendizaje automático a menudo usados para modelar aspectos de la percepción humana, la cognición y el comportamiento, los procesos de aprendizaje en los que se basan esos comportamientos, y el guardado y la recuperación de información de la memoria, en particular redes neuronales y mapas cognitivos difusos.

2.1 Estrategias de agregación en aprendizaje federado

Relacionado con las técnicas algorítmicas óptimas del aprendizaje federado, hemos estudiado el impacto de la estrategia de agregación en el modelo federado. En la definición inicial de aprendizaje federado, el paso de agregación se realiza mediante la media aritmética de los parámetros del modelo. Sin embargo, otros métodos de agregación podrían ser de mayor interés, ya que pueden mejorar el rendimiento del modelo al dar más peso a diferentes agentes según su tamaño o el rendimiento de los modelos locales en sus datos.

Las principales contribuciones de esta investigación son las siguientes:

1. Se proponen varias estrategias de agregación, como la agregación ponderada utilizando el tamaño del conjunto de datos, el promedio ponderado utilizando la inversa normalizada de la precisión local en la prueba, la agregación ponderada utilizando el tamaño del conjunto de datos y la precisión, el promedio ponderado utilizando la contribución del participante (diferencia entre las pérdidas del modelo local y el modelo global), y la suma ponderada utilizando la contribución inversa del participante. La agregación federada se incluye para fines de comparación.

2. Se prueban las estrategias con diferentes tamaños de datos en cada participante. Esto permite analizar las estrategias en diferentes circunstancias e identificar aquellas que son más robustas.

Asumiendo que los parámetros del modelo en la iteración j son

$$\Phi_j = [\Phi_{j1}, \Phi_{j2}, \cdots, \Phi_{jn}] \tag{2.1}$$

donde n es el número de participantes, $\mathcal{D}_i$ es el dataset del participantes i y Φ'_j es el conjunto de parámetros (pesos o gradientes) del modelo federado, las funciones de los parámetros que vamos a discutir son las siguientes:

- Media de los parámetros (pesos o gradientes):

$$\Phi'_j = \frac{1}{n} \sum_{i=1}^{n} \Phi_{ji} \tag{2.2}$$

  donde cada participante contribuye igualmente al modelo global. Este es el método clásico.

- Media ponderada usando el tamaño normalizado del dataset de cada participante:

$$\Phi'_j = \sum_{i=1}^{n} \frac{|\mathcal{D}_i|}{\sum_{k=1}^{n} |\mathcal{D}_k|} \cdot \Phi_{ji} \tag{2.3}$$

  donde cada participante contribuye al modelo global de manera proporcional al tamaño de sus datos, y los agentes con menos información impactarán menos al modelo final.

- Media ponderada usando la precisión inversa normalizada del modelo global, calculado en el conjunto de test de cada participante:

$$\Phi'_j = \sum_{i=1}^{n} \frac{1/\text{acc}_{ji}}{\sum_{k=1}^{n} 1/\text{acc}_{jk}} \cdot \Phi_{ji} \tag{2.4}$$

  donde los modelos individuales añaden al modelo global de manera inversa a su métrica de rendimiento, intentando dar más peso a los modelos menos precisos para mejorar las métricas en sus datasets.

- Media ponderada usando la precisión y el tamaño del dataset:

$$\Phi'_j = \sum_{i=1}^{n} \frac{\text{acc}_{ji}|\mathcal{D}_i|}{\sum_{k=1}^{n} |\mathcal{D}_k|} \cdot \Phi_{ji} \tag{2.5}$$

  donde la contribución del modelo de cada participante depende tanto de la precisión del modelo como del tamaño de los datos.

- Media ponderada usando la contribución ($\mathcal{C}$) del participante, es decir, la diferencia en valor absoluto normalizada de la pérdida del modelo del participante y la pérdida del modelo global, aplicado a los datos del participante, como se muestra en la Ecuación 2.6

$$\mathcal{C}_{ji} = |\mathcal{L}_j^*(\mathcal{D}_i, \Phi) - \mathcal{L}_j(\mathcal{D}_i, \Phi)| \tag{2.6}$$

  y

$$\Phi'_j = \sum_{i=1}^{n} \frac{\mathcal{C}_{j,i}}{\sum_{k=1}^{n} \mathcal{C}_{jk}} \cdot \Phi_{ji} \tag{2.7}$$

- Media ponderada usando el inverso de la contribución del participante:

$$\Phi'_j = \sum_{i=1}^{n} \frac{1/\mathcal{C}_{ji}}{\sum_{k=1}^{n} 1/\mathcal{C}_{jk}} \cdot \Phi_{ji} \tag{2.8}$$

En nuestra configuración experimental para probar qué método es óptimo suponemos que varios hospitales con sus datos privados desean entrenar un modelo de aprendizaje profundo (una red neuronal densa compuesta por cinco capas seguidas de una función no lineal ReLU y una capa de dropout para la regularización, con la Binary Cross Entropy como función de pérdida) para el diagnóstico de una enfermedad, pero el tamaño de sus datos no es lo suficientemente grande para entrenar un modelo preciso. Los datos de todos los hospitales no deben combinarse debido a las regulaciones de datos dada su especial sensibilidad, por lo que deciden unirse en un sistema federado.

En una primera prueba todos los hospitales tendrán la misma cantidad de datos. Los otros tres experimentos tendrán distintos tamaños del conjunto de datos, donde en los dos últimos hemos forzado que haya varios participantes con un número muy pequeño de muestras (menos del 10%). La distribución de las variables, incluido el porcentaje de casos positivos para el objetivo,

también variará de un hospital a otro para probar la robustez de los métodos de agregación.

Realizamos cuatro experimentos diferentes (con cuatro conjuntos de datos diferentes relacionados con cuatro enfermedades diferentes). Para cada experimento, se compara la precisión en el conjunto de datos de prueba del participante para cada método de agregación mencionado anteriormente. Los resultados muestran que el media aritmética clásica es un método de agregación confiable que mejora el rendimiento de los métodos locales en 11 de los 16 casos que hemos contemplado. Sin embargo, hay otros métodos de agregación con un comportamiento similar o incluso mejor. La agregación basada en la contribución, utilizando la diferencia entre las pérdidas del modelo global y local, aumenta la precisión en 11 de los 16 casos, mientras que la basada en el tamaño y la contribución inversa funcionan mejor en un caso adicional. El promedio ponderado utilizando tanto el tamaño del conjunto de datos del participante como la precisión del modelo local aumenta la precisión en solo 10 de los 16 casos. Finalmente, el promedio ponderado utilizando la precisión supera a todos los métodos de agregación, mejorando la precisión en 15 de los 16 casos, es decir, en todos los experimentos excepto en una partición donde todos los demás métodos no lograron aumentar el rendimiento también.

Con estos resultados, creemos que un enfoque de aprendizaje federado basado en la precisión puede funcionar mejor que el enfoque clásico de Federated Averaging. En investigaciones posteriores indagaremos si esta diferencia está relacionada con la métrica de rendimiento, las funciones de pérdida o si es un comportamiento general.

2.2 Encriptación con mapas caóticos para aprendizaje federado con datos no idénticamente distribuidos

En esta investigación abordamos el problema de privacidad en el aprendizaje federado desarrollando un nuevo método de cifrado basado en mapas caóticos. Después comparamos los resultados de un modelo federado no seguro y un modelo seguro construido utilizando la privacidad diferencial y mapas caóticos como su capa de cifrado. En cada experimento esta investigación demuestra que el proceso de federación mejora las métricas de rendimiento promedio de una red neuronal profunda para los participantes, independientemente de si los datos se han dividido uniformemente entre ellos o si hay diferencias en la cantidad de datos que tiene cada participante, y que el rendimiento con o sin la capa de privacidad es similar, lo que significa que la seguridad adicional no empeora los resultados del modelo gracias al proceso de federación.

Además, para asegurar que la simulación se asemeje lo máximo posible a una implementación en el mundo real de aprendizaje federado, hemos tenido en cuenta un escenario donde uno de los participantes posee datos incompletos con una estructura distinta, tocando otros de los intereses de la investigación en este tema: el aprendizaje federado con heterogeneidad de los datos. Esta

situación puede surgir, por ejemplo, cuando una variable no está disponible en el conjunto de datos de un participante. En tales casos, esta propuesta permite que otros participantes compartan de manera privada la distribución de la variable faltante, lo que permite la imputación de esos datos en el conjunto de datos del participante que carece de esa variable. La lógica detrás de esta propuesta es que el proceso de federación y sus múltiples iteraciones promediarán el rendimiento del modelo incluso en el caso de que se haya imputado una característica del participante.

Las principales contribuciones de esta investigación son dos:

- Esta extensión de aprendizaje federado tiene como objetivo manejar conjuntos de datos que son incompletos o contienen valores faltantes, así como conjuntos de datos que no son IID (Independientes e Idénticamente Distribuidos). En ciertos escenarios, los datos utilizados en el proceso de federación pueden tener valores faltantes, carecer de completitud o tener una distribución desigual entre los dispositivos o nodos participantes.

- Un método eficiente y seguro para cifrar modelos distribuidos basados en mapas caóticos. Los mapas caóticos poseen una complejidad e imprevisibilidad que los hace resistentes a ataques criptográficos convencionales. Además, su naturaleza no lineal mejora su seguridad. Finalmente, las características determinísticas de los mapas caóticos los convierten en un método eficiente de cifrado.

La extensión de Federated Learning propuesta para combinar el uso de una capa adicional de cifrado con atributos no IID (no Independientes e Idénticamente Distribuidos) es la siguiente:

1. Como primer paso, un servidor central enviará un modelo de aprendizaje profundo no entrenado a los participantes.

2. Si alguno de ellos no tiene un conjunto de datos completo, lo que significa que falta una de las características (y, por lo tanto, las características no son IID), el servidor también enviará, de manera cifrada, la distribución de la característica para cualquier otro participante para que la característica faltante pueda ser imputada.

3. Después todos los participantes dividirán sus datos en conjuntos de entrenamiento/prueba/validación y entrenarán el modelo en sus datos de entrenamiento. Luego, enviarán los parámetros del modelo de regreso al servidor central, utilizando uno de los tres enfoques posibles para este paso:

   (a) El primero, donde los datos se envían sin medidas de seguridad adicionales,

   (b) El segundo, donde los datos se cifran utilizando privacidad diferencial para evitar problemas de privacidad, con el fin de comparar si el uso de esta capa de preservación de la privacidad afecta los resultados del modelo,

   (c) Y el tercero, donde los datos se cifran utilizando un mapa caótico y el proceso termina con la descifrado de los datos ofuscados.

4. El servidor después agrega los parámetros de los modelos locales para encontrar un modelo global e itera este proceso para mejorar la precisión global. Como ya hemos comentado, el método de agregación más común es la media aritmética, que es simplemente un promedio ponderado de los pesos de la red en los sitios de entrenamiento.

El modelo federado se evalúa en los datos de prueba de cada participante. La convergencia del proceso de privacidad diferencial está garantizada por el trabajo en [18], donde los autores aplican la privacidad diferencial al aprendizaje federado.

Para los experimentos, asumiremos que hay cinco participantes diferentes. En una primera prueba, todos tendrán la misma cantidad de datos, obtenidos de un conjunto de datos dividido uniformemente. Pero también consideraremos el caso en el que cada participante tiene un tamaño de conjunto de datos diferente. En particular, hemos realizado tres experimentos adicionales con divisiones desiguales: el primero será una división aleatoria entre los participantes, pero en los dos restantes hemos forzado que haya varios participantes con un número muy pequeño de muestras (menos del 10% de las muestras). La cantidad de casos positivos también variará de un participante a otro. En cualquier caso, los datos de cada participante se dividirán en conjuntos de entrenamiento y prueba. Como es habitual, los modelos se entrenarán en los datos de entrenamiento de cada participante y las métricas de evaluación se obtendrán a partir de los datos de prueba de cada participante. Las métricas de rendimiento finales se promediarán.

También simulamos el caso en el que uno de los participantes no tiene datos sobre una de las variables. En ese caso, utilizamos el intercambio cifrado de datos para enviar la distribución de esa variable a uno de los otros participantes de manera privada y luego procedemos a imputar la media del valor. La distribución de la característica faltante se calcula en los datos de otro participante y se envía a través del cifrado utilizado para compartir los datos del modelo al participante sin la característica. Luego, aplicamos la imputación L^2 utilizando la distribución de las características.

Para cada experimento realizamos dos enfoques diferentes de FL: uno con una capa de privacidad diferencial y otro sin ella. Las diferencias en los resultados entre los dos tipos de experimentos se comparan para comprender si la versión cifrada ofrece resultados similares. Para comparar el rendimiento de diferentes modelos y métodos, calculamos métricas (precisión y puntuación F1) en un conjunto de pruebas para cada participante.

Los diferentes experimentos muestran que, incluso en los casos más desequilibrados, el proceso de aprendizaje federado mejora las métricas promedio de los modelos, aumentando su rendimiento, tanto en precisión como en puntuación F1, y para los enfoques privados, no privados y cifrados. También podemos concluir que el uso de una capa adicional de cifrado y garantizar la privacidad del proceso no afecta las métricas de rendimiento del modelo, en comparación con el Aprendizaje Federado no privado.

2.3 Mapas Cognitivos Difusos federados

En los últimos años ha habido varios intentos de crear una versión federada de algoritmos convencionales de aprendizaje automático, como la regresión lineal federada [19]–[21], la regresión logística federada [22], el bosque aleatorio federado [23], XGBoost federado [24]–[26], y las máquinas de soporte vectorial federadas [27], [28].

La tercera línea de investigación de esta tesis es desarrollar un marco para federar Mapas Cognitivos Difusos. Los Mapas Cognitivos Difusos son un caso especial de Mapas Cognitivos [29], donde las relaciones entre nodos se modelan con relaciones de causa-efecto difusas en lugar de las relaciones de causa-efecto nítidas propuestas por el Mapa Cognitivo original [30]. Los MCD son sistemas dinámicos neurodifusos construidos a partir de conocimiento experto y/o datos históricos sin procesar.

Los MCD están compuestos por nodos que modelan conceptos o variables, aristas que representan relaciones entre ellos y pesos que cuantifican la influencia de esas relaciones [31], [32]. Es decir, el valor de un peso difuso ϖ_{ij} de un FCM describe la influencia del nodo c_i sobre el nodo c_j.

Estos mapas pueden entrenarse utilizando técnicas de optimización, en el caso de nuestra investigación, Optimización por Enjambre de Partículas (PSO), para obtener un modelo de aprendizaje automático que sea inherentemente interpretable [33], a diferencia de otras técnicas que requieren la aplicación de enfoques de Inteligencia Artificial Explicable (XAIs) para su explicación [34].

Nuestra metodología propuesta combina el aprendizaje federado con el aprendizaje de MCD mediante la optimización por enjambre de partículas. El proceso se explica de la siguiente manera:

1. Inicio del proceso de aprendizaje federado. El servidor central inicia el proceso en las máquinas de los participantes.

2. Entrenamiento del MCD en el conjunto de datos local. Cada participante entrena un MCD local con su propio conjunto de datos. Aplicamos PSO, pero esta metodología es agnóstica al método de aprendizaje. La dinámica del MCD se considera estable cuando la diferencia entre dos estados vectoriales consecutivos es menor a $tol = 0.00001$.

3. Envío de las matrices de adyacencia entrenadas y la precisión local a la etapa actual al servidor central. El MCD local se almacena en los dispositivos de los participantes.

4. Agregación de los MCD locales mediante la precisión. El servidor central agrega los MCD locales ponderados por la precisión.

5. Agregación de MCD federados y locales. Los participantes agregan el MCD federado del servidor central y su propio MCD local.

6. Envío de las matrices de adyacencia y precisión. Los participantes envían nuevamente las matrices de adyacencia locales y la nueva precisión local.

7. Comprobación de la condición de terminación. El servidor central verifica si el proceso federado se ha ejecutado durante 20 iteraciones como condición de terminación. Si no se cumple, vuelve al paso 4.

8. Si se cumple la condición de terminación, se logra un MCD federado.

La principal contribución de nuestra investigación en este tema es la aplicación del paradigma de aprendizaje federado para el aprendizaje distribuido y cooperativo de Mapas Cognitivos Difusos.

Nuestros resultados experimentales muestran que el clasificador basado en Mapas Cognitivos Difusos mejora la precisión de un solo Mapa Cognitivo Difuso entrenado en todos los datos y la precisión en cada participante antes de la federación.

El objetivo de este trabajo no es la precisión de la propuesta, sino un enfoque distribuido y preservador de la privacidad. No obstante, nuestros resultados de rendimiento para este problema son similares a los encontrados en la literatura [35].

Chapter 3

Objective

In this thesis we attempt to advance the research of federated learning in

- algorithmic basics: how the principles of Federated Learning can be improved to increase the general performance of the federated model,
- security and privacy,
- data heterogeneity

for connectionism-based models, that is, neural networks and fuzzy cognitive maps.

Focusing on the algorithmic basics of federated learning, our main research question was how much the aggregation strategy chosen for the federated model can affect the performance of the model. We answer that question in our paper

[**Paper I**] J.L. Salmeron, **I. Arévalo**, A. Ruiz-Celma, *Benchmarking federated strategies in Peer-to-Peer Federated learning for biomedical data Heliyon, Volume 9, Issue 6 (2023), e16925.* 2022 Journal Impact Factor: 4.0. Rank 23/73 in MULTIDISCIPLINARY SCIENCES (Q2)

that can be found in Chapter 9 and [36].

On security and privacy and data heterogeinity, we aimed to create a new encryption method based on chaotic maps that did not decrease the performance of the model when compared with other common privacy-preserving techniques like differential privacy or even the federated model without an additional encryption layer. Moreover, we were interested on the real world scenario where participants may not have data with the same distributions (the non-IID case). We developed a framework for federated learning with non-IID datasets when one or more features of one or more participants are missing. Our methodology and experimental approach is contained in

[**Paper II**] **I. Arévalo**, J.L. Salmeron, *A Chaotic Maps-based Privacy-preserving Distributed Deep Learning for Incomplete and Non-IID Datasets*

IEEE Transactions on Emerging Topics in Computing, ISSN 2168-6750. 2022 Journal Impact Factor: 5.9. Rank 35/158 in COMPUTER SCIENCE, INFORMATION SYSTEMS (Q1)

in this thesis in Chapter 10 and [37].

Finally, related again with the algorithmic basics of federated learning, we developed a framework for federating Fuzzy Cognitive Maps, cognitive maps where concepts, variables or features are represented as nodes, the relationships between them as arcs, and the strengths of those relations as weights, therefore supporting causal knowledge. Paired with an optimization algorithm, such as particle swarm optimization, fuzzy cognitive maps can be trained as a classification machine learning model with interesting characteristics. Our framework is published in

[**Paper III**] J.L. Salmeron, **I. Arévalo**, *A Privacy-Preserving, Distributed and Cooperative FCM-Based Learning Approach for Cancer Research. In: Bello, R., Miao, D., Falcon, R., Nakata, M., Rosete, A., Ciucci, D. (eds) Rough Sets. IJCRS 2020. Lecture Notes in Computer Science, vol 12179, 477–487. Springer, Cham. Online ISBN 978-3-030-52705-1.* Conference with Rank C CORE

appended in Chapter 11 and [38].

Part II

Theoretical Background

Chapter 4

Fuzzy Cognitive Maps

Machine learning is the field of artificial intelligence that focuses on the development of computer algorithms that improve automatically through experience and by the use of data. In this work we will focus on supervised machine learning, a category of machine learning that uses labeled datasets to train algorithms to predict outcomes and recognize patterns. Formally, the task of supervised learning is as follows:

Given a training set of n input-output pairs:

$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n),$$

(referred to as features in the case of x_k and labels in the case of y_k) where each y_k was generated by an unknown function $y = f(x)$, discover a function h that approximates the true function f. The function h is called a hypothesis, and is learnt based on the discrepancy between predictions (the result of $h(x_k)$) and observations (the corresponding y_k). In order to find this difference we need a measure for the quality of the predictions obtained from a hypothesis, called loss functions. The space of possible hypothesis maps, from which a machine learning method can choose from, is called a hypothesis space or model.

Different machine learning methods use different choices for data points, the hypothesis space, and loss function, and these design aspects are guided by computational and statistical aspects. There are several monographs on this topic, see for instance [39].

In this chapter we present the two main families of models (or hypothesis spaces) that have been studied in our research: neural networks and Fuzzy Cognitive Maps. Neural networks are a well-known algorithm family and many techniques in the learning process of neural networks has been used in the training processes of Fuzzy Cognitive Maps.

4.1 Background: Connectionism-based models and neural networks

Connectionism, coined by Edward Thorndike in the 1930s, is an approach of cognitive science that is based on the idea that our understanding of behavior and of mental states should be informed and constrained by our knowledge of the neural processes that underpin cognition. The prevailing theory over the past four decades in cognitive science has been dominated by the notion that human cognition, especially at higher levels, bears resemblance to symbolic computation in computers. According to the classical viewpoint, information is encoded through strings of symbols, mirroring the way we store data in computer memory or on physical documents. In contrast, proponents of connectionism argue that information is stored in a non-symbolic manner within the weights or connection strengths among the units of a neural network. Classicists contend that cognition operates akin to digital processing, where sequences of strings are generated based on the instructions of a symbolic program. On the other hand, connectionists perceive mental processing as a dynamic and graded evolution of activity within a neural network, with each unit's activation being influenced by the connection strengths and activity of its neighboring units.

Connectionist networks are made up of interconnected processing units which can take on a range of numerical activation levels (for example, a value ranging from 0 – 1). A given unit may have incoming connections from, or outgoing connections to many other units. The excitatory or inhibitory strength (or weight) of each connection is determined by its positive or negative numerical value.This knowledge representation is completely opposed to traditional approaches to knowledge representation all make the formalist assumption that knowledge can be represented by finite structures composed of discrete atomic symbols arranged in accordance with a finite number of syntactic relations.

Therefore, connectionism-based models are a class of machine learning models often used to model aspects of human perception, cognition, and behavior, the learning processes underlying such behavior, and the storage and retrieval of information from memory [40]–[44].

A clear example of a connectionsim-based model is the McCulloch-Pitts perceptron. Perceptrons are binary classifiers that assign a weight to all inputs and then sums over the products of these weights and their input. The outcome of this process is then labeled as active or inactive depending on a threshold.

Mathematically, given the input space $X = \{x_1, x_2, \cdots, x_n\}$, the weights $W = \{w_1, w_2, \cdots, w_n\}$, and the threshold b, the output of the perceptron is given by the following equation:

$$f(x) = \begin{cases} 1, & \text{if } \sum_{k=1}^{n} x_k * w_k \geq b \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

The training of the perceptron, that is, the process to find the best parameters or weights $W = \{w_1, w_2, \cdots, w_n\}$ based on the target information for the input and a cost function to compute the error with respect to the target,

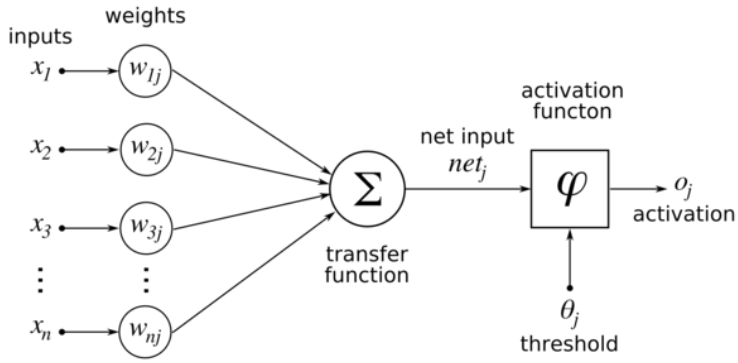


Figure 4.1: Example of the McCulloch-Pitts perceptron, adapted from [45]

is usually done in an iterative way that begins by randomly initializing them, obtaining the prediction of the inputs and computing the error. Then the weights are updated using the backpropagation algorithm.

Algorithm 1: Backpropagation

Data: Learning rate η, dataset $\{(x_k, y)\}_{k=1}^{n}$.
Random initialization of the parameters (W, b)
Define cost function
Compute output and error E given by cost function
while $E > \varepsilon$ *(desired error criteria)* **do**
  $\forall w \in W : \Delta w = -\frac{\partial E}{\partial w}$;
  $w_{new} \leftarrow w_{old} + \eta \Delta w$;
  Compute output and error E.

A neural network is a perceptron-based system formed by multiple layers: an input layer, one or more hidden layers, and an output layer. Each layer has nodes connected to the previous and following layers through edges with associated weights. Finally the nodes usually apply an activation function on the output to introduce non-linearity. In principle, if a neural network has many hidden layers (usually called deep neural networks) it is able to learn hierarchical feature abstractions of the data, with increasing abstraction through the network.

One of the main advantages of neural networks is that they perform feature extraction in an automated way, which allows researchers to extract discriminative features with minimal domain knowledge and human effort thanks to their layered architecture of data representation, where the high-level features can be extracted from the last layers of the networks while the low-level features are extracted from the lower layers.

There are several monographs on neural networks and deep learning, see for instance [47], [48] or [49].

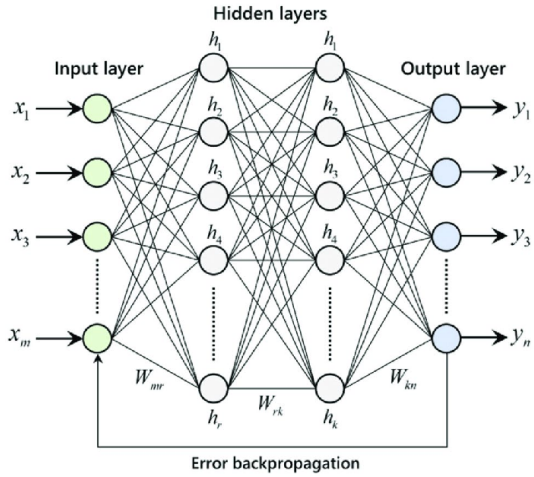


Figure 4.2: Example of a deep neural network, adapted from [46]

4.2 FCM Fundamentals

Fuzzy cognitive maps (FCMs) are fuzzy-graph structures for representing causal reasoning. Its nodes represent concepts, variables or features, the edges model relationships between them, and the weights are the influence of those relations [31], meaning that the value of a weight ϖ_{ij} shows how much node c_i impacts over the node c_j. The fuzzy weights between edges are normalized so that $\xi = \{[0, +1]|[-1, +1]\}$, depending if it includes only positive values or both positive and negative. Fuzzy cognitive models are represented by the adjacency matrix which contains all edges' weights between the nodes, see image 4.3.

Furthermore, FCMs are a kind of dynamic system with feedback, where the influence of the change in the state of one node can affect the state of other nodes, which in turn can affect the previous node [51].

The FCM dynamical analysis begins with an initial state vector $c(0) = [c_1(0), \ldots, c_n(0)]$, that models the initial situation of each node. The state of the nodes is updated in an iterative process. Thus, it includes a activation function [52] for mapping the state of the node into a normalized range between $[0, +1]$ or $[-1, +1]$. If the range is $[0, +1]$, the sigmoid is the most used transformation function, while hyperbolic tangent is the most used when the nodes' range is $[-1, +1]$ [53].

If the selected activation function f is the unipolar sigmoid, then the component i of the vector state c_i at the instant t is computed as shown in Equation 4.2:

$$c_i(t) = \frac{1}{1 + e^{-\lambda \cdot \sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)}} \tag{4.2}$$

where λ represents the slope of the unipolar sigmoid function. On the contrary, if the selected activation function f is the hyperbolic tangent, then the node's

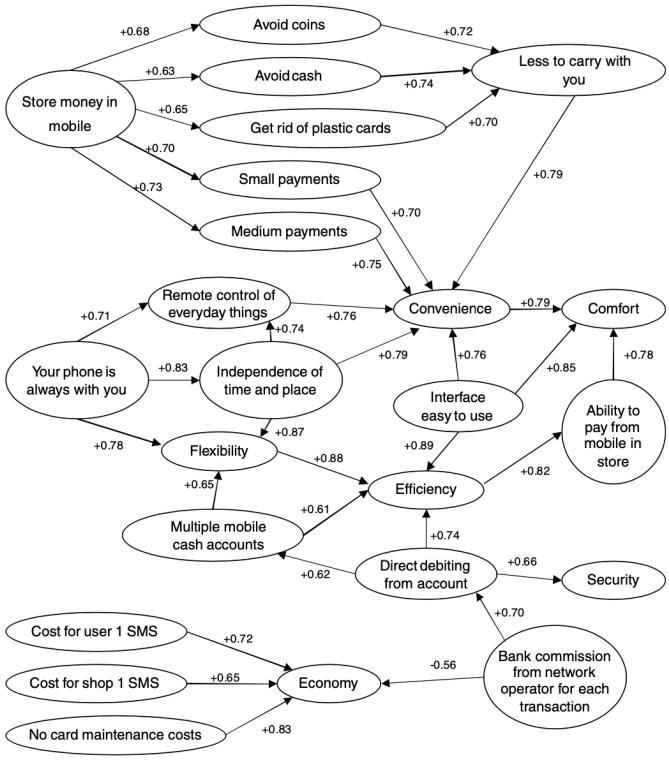


Figure 4.3: Example of a FCM, adapted from [50]

state c_i at the instant t is computed as Equation 4.3 shows:

$$c_i(t) = \frac{\sinh\left(\lambda \cdot \sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)\right)}{\cosh\left(\lambda \cdot \sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)\right)} \tag{4.3}$$

The state vector $c = [c_1, c_2, \ldots, c_N]$ shows the state of the nodes at any iteration in the FCM dynamics [51], that is, the state of the node i at time t is computed as shown in Equation 4.4:

$$c_i(t) = f\left(\sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)\right) \tag{4.4}$$

where c_j are the nodes and ϖ_{ji} is the weight of the edge from c_j to c_i. Formally, a FCM is defined as a 4-tuple $\Phi = \langle c, \mathcal{W}, f, \xi \rangle$, where $c = \{c_i\}_{i=1}^{n}$ is the nodes' state with n number of nodes, $\mathcal{W} = [\varpi_{ij}]_{n \times n} \,|-1, 0 \leq i, j \leq +1$ is the adjacency matrix, f is the activation function, and ξ is the nodes' range [54].

After the dynamics, the FCM reaches one of three possible states after a number of iterations: it settles down to either a fixed pattern of node values (the so-called hidden pattern), to a limited cycle, or to a fixed-point attractor [55], [56].

4.3 Augmented FCMs

There are two main approaches to build FCMs. The first involves having a group of experts who would individually design a FCM model with their knowledge on the system [55].

The second approach is the automatic construction from raw data [51], [54], [57]–[60], which is the focus of this research. According to the literature [52], this automatic construction can be done by building an augmented adjacency matrix by aggregating the adjacency matrix of each FCM. If the adjacency matrices have common nodes, the states ϖ_{jk} in the augmented matrix are computed by adding the adjacency matrix of each FCM model ($\mathcal{W}_i$). If they do not have common nodes, a direct sum of matrices is used, and the augmented matrix is denoted as $\odot_{i=1}^{N}\mathcal{W}_i$. Given two of FCMs with no common nodes with adjacency matrices $\varpi_{n\times n}^{A}$ and $\varpi_{m\times m}^{B}$, the resulting augmented adjacency matrix can be computed as in Equation 4.5:

$$\bigodot_{i=1}^{N}\mathcal{W}_i = \begin{bmatrix} 0 & \mathcal{W}_{n\times n}^{A} \\ \mathcal{W}_{m\times m}^{B} & 0 \end{bmatrix} \tag{4.5}$$

where N is the number of adjacency matrices to join, zeros are actually zero matrices, and the dimension of $\odot_{i=1}^{N}\mathcal{W}_i$ is $[\,\cdot\,]_{(m+n)\times(m+n)}$. In the case of common nodes, they would be computed as the average of the nodes' states in each adjacency matrix $\mathcal{W}^i$.

4.4 Pattern recognition with FCMs

FCMs are a neuro-fuzzy technique and learning algorithms have been proposed for training and updating FCMs weights mostly based on ideas coming from the field of artificial neural networks. FCMs have been applied both in classification and regression tasks.

The literature has analyzed pattern recognition tasks using Fuzzy Cognitive Maps. Papakostas et al. [61] and Papakostas and Koulouriotis [62] propose several FCM architectures for pattern recognition. Swzed [63] proposed a FCM based classifier with a fully connected architecture. Wu et al. [64] applied broad learning systems for time series classification with FCMs. Ramirez-Bautista et al. [65] applies FCMs for classification of human plantar foot alterations. Baykasoglu and Golcuk [66] proposed alpha-cut based FCM methods are tested on several case studies. Papakostas et al. [67] applied unsupervised hebbian learning for pattern recognition problems.

Our research focuses on classification tasks. In general terms, the main goal of a conventional classifier is the mapping of an input to a specific binary output according to a pattern. The input concepts are represented by the features of the dataset, while the output are the classes' labels where the patterns belong. Figure 4.4 shows an example topology of a Fuzzy Cognitive Map classifier, where the state of the concepts c_1 and c_2 defines the class where the input vector state belongs.

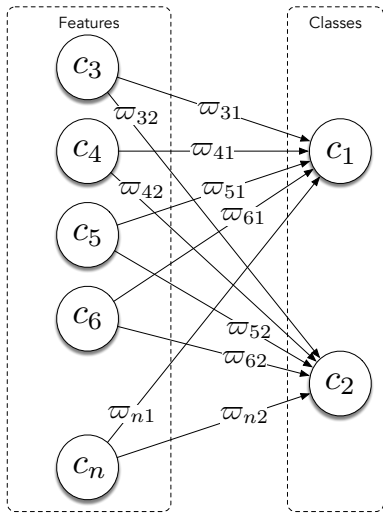


Figure 4.4: FCM binary classifier example [36]

In that sense, if $c_1 > c_2$ the input vector state belongs to class 1, while if $c_1 < c_2$ the input vector state belongs to class 2.

4.5 FCM Learning

FCM learning approaches can be categorized into three main groups [57], [68]: Hebbian, population-based, and hybrid, combining key aspects of both Hebbian-based and population-based learning algorithms. Table 4.1 shows the main FCM learning approaches.

Hebbian-based FCM learning approaches aim to adjust adjacency matrices to guide the FCM model toward either reaching a stable state or converging to an acceptable region for the target system. However, this approach has not proven successful for FCM extensions, such as Fuzzy Grey Cognitive Maps [56].

Population-based methods operate without human intervention. They calculate adjacency matrices from historical raw data that best align with the sequence of input state vectors (instances in the dataset). The objective of FCM evolutionary learning is to produce an optimal adjacency matrix for modeling system behavior.

In this sense, Particle Swarm Optimization (PSO) is a bio-inspired, population-based, and stochastic optimization algorithm. The PSO algorithm creates a swarm of particles moving in an n-dimensional search space that encompasses all possible candidate solutions. To train the adjacency matrices of FCMs, we consider the position of the k^{th} particle, which represents a candidate solution or adjacency matrix, denoted as $\varpi_k = [\varpi_{k_1}, \ldots, \varpi_{k_j}]$ and its velocity, $v_k = [v_{k_1}, \ldots, v_{k_j}]$. It's important to note that each particle is a potential solution or candidate for the FCM, and its position ϖ_k corresponds to the adjacency matrix of the k-th FCM candidate [32], [54]. Each particle's velocity

Table 4.1: FCM Learning Algorithms

Category	Learning approach	Author(s)
Hebbian	Differential Hebbian Learning (DHL/DDNHL)	[56], [69]–[71]
	Nonlinear Hebbian Learning (NHL)	[56], [72]
	Balance Differential Algorithm (BDA)	[56], [73]
	Petri Nets	[74]
	Active Hebbian Learning (AHL)	[75]
Population	Evolutionary Strategies (ES)	[76]
	Particle Swarm Optimization (PSO)	[32], [77]
	Genetic Algorithm (GA/RCGA/SOGA)	[78]–[80]
	Memetic Particle Swarm Optimization (MPSO)	[68], [81]
	Simulated Annealing (SA/CSA)	[82], [83]
	Tabu Search (TS)	[84]
	Game-based learning	[85]
	Differential Evolution (DE)	[86]
	Immune Algorithm	[87]
	Big-Bang Big-Crunch (BB-BC)	[88]
	Self-Organized Migration Algorithm (SOMA)	[89]
	Ant Colony Optimization (ACO)	[90]
	Extended Great Deluge Algorithm (EGDA)	[91]
	Artificial Bee Colony (ABC)	[92]
	Cultural Algorithm (CA)	[93]
	Imperialist Competitive Algorithm (ICLA)	[94]
	Modified Asexual Reproduction	[59]
	Multiobjective Evolutionary Algorithm (MOEA-FCM)	[95]
Hybrid	NHL-DE	[96]
	NHL-RCGA	[97]
	NHL-EGDA	[98]
Emerging	Pseudoinverse learning	[60]
	Federated Learning	[38]
	Quantum-based learning	[33]

and position are updated at each time step. The position and the velocity of each particle are computed as shown in Equations 4.6 and 4.7 ([36]):

$$\varpi_k(t+1) = \varpi_k(t) + v_k(t) \tag{4.6}$$

$$\begin{aligned} v_k(t+1) &= v_k(t) + U(0,\phi_1) \otimes (\dot{\varpi}_k - \varpi_k(t)) \\ &+ U(0,\phi_2) \otimes (\ddot{\varpi}_k - \varpi_k(t)) \end{aligned} \tag{4.7}$$

Here $U(0,\phi_i)$ is a vector of random numbers drawn from a uniform distribution within $[0,\phi_i]$ generated at each iteration and for each particle. Also, $\dot{\varpi}_k$ represents the best position of particle k in all previous iterations, $\ddot{\varpi}_k$ denotes the best position of the entire population in all past iterations, and $\otimes$ signifies component-wise multiplication.

The objective of the PSO algorithm is to position all particles in the global optima within a multidimensional hyper-volume. The fitness function employed in this study is the complement of the Jaccard similarity coefficient ($\overline{J} = (Y \times \hat{Y}) \backslash J$). The Jaccard score calculates the average similarity coefficients between pairs of the i-th samples, considering a ground truth label set and a predicted label set. The complement operation is necessary for minimizing the fitness function. The Jaccard similarity coefficient's complement is computed

as follows in Equation 4.8 ([36]):

$$\overline{J}(y_i, \hat{y}_i) = 1 - \frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|} \tag{4.8}$$

The fitness function is evaluated after each particle position update and serves as the objective function to determine the proximity of a given particle towards achieving the global optimum.

For more information about Fuzzy Coginitive Maps, see [99], [100], [101], [102], [103], [50].

Chapter 5

Federated Learning

5.1 Fundamentals

Conventional machine learning requires all data collected on local devices to be stored centrally on a data silo. Distributed machine learning is the subfield of artificial intelligence that studies the sharing of knowledge between agents in order to solve complex problems, classically via the distribution of tasks or data. Such processes may not be of interest in fields where the characteristics of the data and the regulations make it impossible to share it, such as finance or health.

The goal of federated learning is building a global model that can be trained on distributed data while ensuring data privacy [104]. It was proposed by McMahan et al. [6] and further developed in Konecny et al. [105] and McMahan and Ramage [7], and the initial proposal has the following steps: the participants collaborate to train a model with their private data by updating that model in their infrastructure and then sending the parameters to an aggregation node. The participants own the data and train the partial models. The aggregation node then federates the participant's models to obtain a global model trained with private data. This method can be iterated as many times as desired.

The main advantage of federated learning is the training of a model in the private data of several participants that wish to maintain avoid data-sharing while improving their models [106]. This approach allows the use of heterogeneous data among the participants. It also allows the use of more accurate models with low latency, ensuring privacy and less power consumption. Its first application was to create collaborative predictive models using private data in Android mobile phones [7]. In particular, a model in Gboard on Android, the Google Keyboard, in order to predict the following word or phrase that the user is going to write based on the former text and other users (private) data. In this set-up the central server manages the federated model and the communications with the agents, while the participants own their data and train the partial models. In this way, a federated learning system ensures that the distributed model is built in a private environment, since the private data

never leaves the local agent.

Nevertheless, there are always risks associated with the data transmission, such as the possibility of the reconstruction of the model or the training data from the model parameters. Due to these risks, there is an increasing interest in the use of an additional layer of privacy to this information, and there are many studies that use privacy-preserving methods in federated learning such as Differential Privacy [14], Secure Multi-Party Computation [15] or Homomorphic Encryption [16].

Many surveys and literature reviews have thoroughly explored the extensive body of academic work concerning architectures, methodologies, applications, and utilization of federated learning. [107] provides an overview of the common solutions to address statistical, system, and privacy challenges in federated learning. They also highlight the potential implications and opportunities that federated learning holds for the healthcare sector. Furthermore, numerous studies have sought to optimize federated learning and broaden its practical applications, with research efforts spanning areas such as computation fusion [108], data transmission [109], [110], as well as privacy and security-related concerns [111].

In recent years, there have been several attempts to create a federated version of conventional machine learning algorithms, such as federated linear regression [19]–[21], federated logistic regression [22], federated random forest [23], federated XGBoost [24]–[26], and federated support vector machines [27], [28].

Federated learning represents a significant step forward in the privacy-preserving machine learning field. Its practical managerial significance lies in its potential to address the balance between utilizing valuable data for business insights, respecting privacy regulations and customer trust. By allowing model distributed training on decentralized data sources while preserving privacy, federated learning offers several managerial benefits:

- Collaborative business insights: FL can facilitate collaboration between different business units or partners without sharing sensitive data directly. This fosters knowledge sharing and cross-functional collaboration while maintaining data privacy.
- Enhanced data privacy compliance: FL enables organisations to comply with strict data protection regulations such as GDPR. This approach avoids reputational damage that may result from non-compliance of data leaks .
- Cost-Efficient AI training: Since data remains on local devices or servers, it reduces the need for extensive data transfer and centralized storage infrastructure.
- Customer trust and brand loyalty: Companies can build trust with their customers by demonstrating a strong commitment to data privacy. This trust can lead to increased customer loyalty and positive brand perception.

In this sense, a practical real-world healthcare FL application would involve a consortium of healthcare institutions or health data owners working together

to improve patient care and disease prediction while preserving data privacy. In this scenario, each institution would retain control of its patient data, ensuring compliance with strict privacy regulations like HIPAA and GDPR.

5.2 Physical Architecture

The architecture of a Federated Learning system is a critical design choice in the process that will affect the system scalability, communication efficiency and security.

There are two main federated learning architectures [112]:

1. Coordinated or centralized (client–server): It consists of a central server, that delivers the model architecture, performs the aggregation tasks, manages the communications, and delivers the model architecture, and a set of data silos or participants.

2. Swarm learning (Peer-2-Peer): This architecture does not need any central server because one or all of the nodes play the role simultaneously of central server and data silos. In this architecture, the federated learning process is triggered by one of the nodes.

A centralized federated learning system can be described as follows (see figure 5.1:

1. The central server delivers a model to each agent. In the initial iteration of this process, the server has built an empty model.

2. The participants train the model with their own private data.

3. Each participant sends the parameters of the model or its gradients to the central server in a private way, usually encrypted.

4. The central server builds a federated model by aggregating the parameters of the individual models.

5. The central server checks if the termination condition is accomplished in which case the federated model is finished, otherwise the process goes back to step 1.

The development of a Peer-2-Peer federated learning process is similar with one or all of the nodes taking the role of the central server, see figure 5.2.

In any case, the target of the federated model is to minimize the total loss for all participants, computed as follows:

$$\mathcal{L}^* = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\mathcal{D}_i, \Phi) \tag{5.1}$$

where n is the number of participants, Φ is the federated model parameters, $\mathcal{D}_i$ is the dataset of the participant i, $\mathcal{L}^*$ is the loss function for the federated model, and $\mathcal{L}_i$ is the loss function for each participant in the federation.

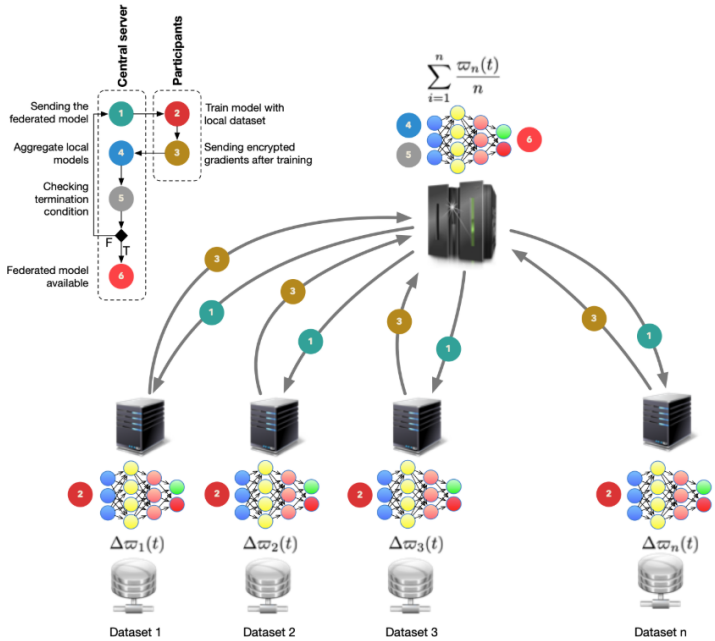


Figure 5.1: A centralized federated learning system [36]

Algorithm 2: Federated learning

Data: The K clients are indexed by k; B is the local mini-batch size, E is the number of local epochs, T is the maximum iteration number, and η is the learning rate.

Federation process
initialize Φ_0
for *each round* $t = 1, 2, \ldots, T$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  for *each client* $k \in S_t \mid S_t \sim U(m)$ *in parallel* **do**
    $\Phi_{t+1}^{k} \leftarrow client\_update(k, \Phi_t)$
  $\Phi_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \cdot \Phi_{t+1}^{k}$
Training process
client_update(k, Φ): $B \leftarrow$(split $\mathcal{P}_k$ into batches of size B);
for *each local epoch* i *from 1 to* E **do**
  for *batch* $b \in B$ **do**
    $w \leftarrow \Phi - \eta \nabla l(\Phi; b)$
return Φ to server

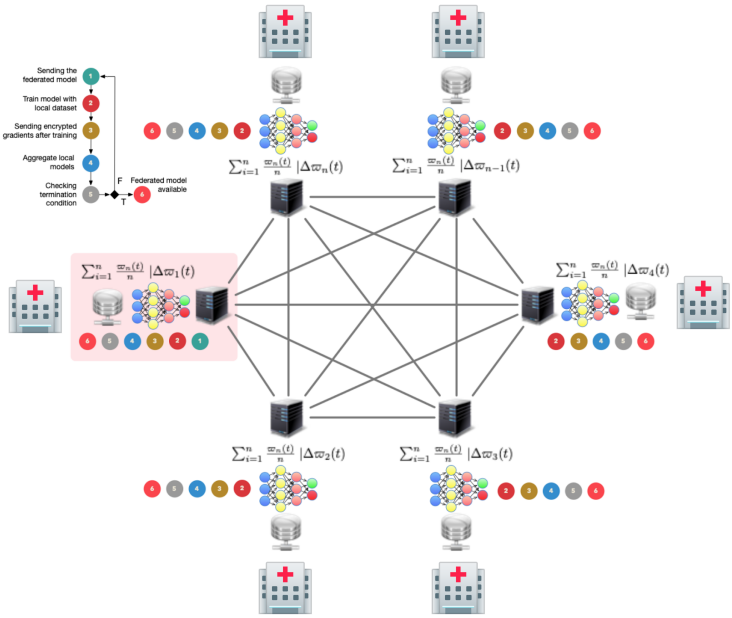


Figure 5.2: A Peer-2-Peer federated learning system [38]

5.3 Taxonomy

Regarding the nature of the data, federated learning can be categorized [112], [113] into three sets: horizontal federated learning, vertical federated learning and federated transfer learning.

In horizontal federated learning the features space is overlapped across data silos, but the samples space is different in data locations. This approach is the original federated learning proposal but it still presents challenges. For instance, an innovative approach named hierarchical heterogeneous horizontal federated learning faces limited labeled entities in horizontal federated learning [113]. In this research, the lack of labeled instances is mitigated by adapting the heterogeneous domain multiple times by using each participant as the target domain each time.

Vertical federated learning is needed when the features space has a partial or low overlap across data silos, but the samples space is nearly the same across those data locations. Unlike the case of horizontal federated learning, the aggregation of the entire data set in a single data silo to train a global model would not work in vertical federated learning. Some vertical proposals have been developed in [114], [115].

Moreover, the data does not share a sample space or a feature space in most cases. Federated transfer learning approach proposed by [116] generalize federated learning when it comes to common parties with small intersection. This proposal can be easily adapted to various secure, machine learning endeavors with minimal modification to the existing model and provides the same level

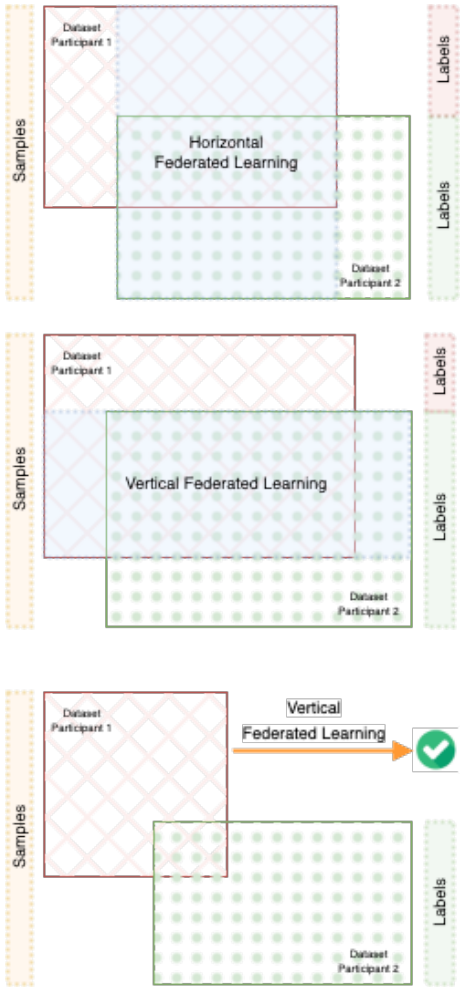


Figure 5.3: FL categories [37]

of accuracy as non-privacy-preserving transfer learning.

5.4 Data nature

The original proposal of federated learning implicitly asumes IID data in order to ensure that the stochastic gradient is an unbiased estimate of the full gradient. But in practice it is unrealistic to assume that the local data on each participant is always IID.

Since each participant may have obtained their data from different sources, the distributions of the data could be quite dissimilar from each other, a phenomenon known as Non-IID [117]. This fact is another challenge for the training of a federated model. Zhe et al. [104] propose the following categories

of Non-IID data:

- Attribute skew. This category includes several subcategories: *Non-overlapping attribute skew*: It means that data attributes across the clients are mutually exclusive. *Partial overlapping attribute skew*: In this case, some portions of the data attributes can be shared with each other. *Full overlapping attribute skew*: Data attributes are the same in all participants but the attributes distributions can be different.

- Label skew. This category includes several subcategories: *Label distribution skew*: Label distributions on the clients are different. *Label preference skew*: The label distribution is different on the client data, although the distribution of the attributes is the same. *Temporal skew*: The focus is on addressing distribution skewness in temporal data, which encompasses spatio-temporal data as well as time series data.

- Attribute and label skew. Different clients hold data with different labels and different attributes.

- Quantity skew. The number of training data varies across different clients.

In general, non-IID datasets can be challenging to analyze because they often contain a high degree of variance and may not be representative of the overall population.

Chapter 6

Security and Privacy

Federated learning is related to two main data protection mechanisms: security and privacy. Security guarantees the confidentiality and integrity of the data, and privacy further refines the secrecy of personal information, therefore privacy prevails when dealing with private data, while security means protecting data from unauthorized access.

Given that federated learning is a distributed approach, its attack range is larger than that of centralized learning, and the concealment is stronger. During the training and testing phases any internal dishonest participant or external malicious attacker can have a very serious impact on the final performance of the federated global model.

6.1 Security attacks

There are various types of attacks that can compromise the security of the federated model and the local participants' models. In this section we introduce the two most common types of attacks that FL studies: Poisoning and Byzantine attacks

6.1.1 Poisoning attacks

Poisoning attacks are built to tamper, destroy or pollute the clients' local training datasets (data poisoning) or local models (model poisoning) to affect the security of FL system.

Data poisoning attackers maliciously interfere with local client data's source tags or characteristics to influence the training result of federated models [118]. Because of the distributed nature of FL deployment, it is difficult to determine whether a client is participating in FL in good faith, so detecting data poisoning is a challenging task. In recent years, some defense mechanisms have been studied to protect the system prom data poison attacks by detecting and suppressing data outliers [119], [120] or creating federated learning frameworks using blockchain [121], [122].

Another type of poisoning attack is caused by model poisoning, where some malicious attackers can bring in hidden backdoor functionalities to the local client model or global federated model, causing the model to produce wrong output. Model poisoning can take advantage of the fact that malicious participants in federated learning can directly influence the performance of the federated model, and therefore being more powerful than data poisoning attacks.

In order to defend the models against this type of attack, Zhao et al. [123] created a mechanism that generates audit data through generative adversarial network during training [124] and then eliminates adversaries who upload malicious models through the accuracy of audit models. Shi et al. [125] designed a framework where clients and server cooperate to analyze model exceptions and ensure the security and validity of the global model.

6.1.2 Byzantine attacks

Poisoning attacks mainly consider the attack on a single user or a server, while Byzantine attacks mainly focus on the collusion of multiple users in a distributed learning environment. In federated learning a malicious attacker may control multiple clients, known as Byzantine users, which in turn can manipulate the global model, for instance by harming the normal communication between clients and server. To defend against these attacks, Ma et al. [126] designed reliability indicators to evaluate the authenticity of knowledge transmitted by clients. Zhai et al. [127] designed a Byzantine robust model for FL by evaluating the reliability of non-IID data using transfer learning anomaly detection [128].

Li et al. [129] focused on detecting the Byzantine model and identify the attacker and created a Byzantine resistant secure blockchained FL framework.

6.2 Privacy-preserving methods

There are three main techniques to protect data privacy: Trusted Execution Environment (TEEs), anti-Generative Adversarial Network (anti-GAN), and secure multi-party computation, encryption and differential privacy, [130]–[134].

A TEE is an environment where the execution is secured and no information can be leaked to unauthorized users. Federated learning has been used in combination with TEEs by containing the entire training process in the TEE of each distributed computing resource or to check a small part of the distributed training while exploiting insecure computing resources such as GPUs, see [135], [136].

A machine learning model can leak information about the training data based on the parameters of the models such as the gradients of a neural network [137], [138], [139] since GANs can be used to generate data similar to the training data. The adversary can then reconstruct other participant's private data, even if it has no knowledge of the label information.

Secure multi-party computation is another usual privacy-preserving method, in which multiple parties collaborate to compute a common function of interest

without revealing their private inputs to other parties. Therefore, the parties learn only the final result and no other information. Nevertheless, a SMPC system tends to have significant communication and computation overhead when implemented on a large-scale decentralized federated learning system. Several works have tried to develop an efficient SMPC FL system, see for instance [140].

In this work we will focus on two privacy-preserving techniques: differential privacy and chaotic maps-based encryption.

6.2.1 Differential Privacy

Differential privacy is a widely-used standard for privacy guarantee of algorithms operating on aggregated data. In general, a randomized algorithm $A(D)$ satisfies ε, δ-differential privacy if for all datasets D and D' that differ in a single record, and for all sets $S \in R$, where R is the range of A,

$$P(A(D) \in S) \leq \exp(\varepsilon) P(A(D') \in S) + \delta \tag{6.1}$$

where the probability P is taken over the coin tosses of A and ε and δ are non-negative numbers.

In general, the mathematical idea behind differential privacy is that the presence or absence of a single individual in a data set should not significantly affect the overall results of a query. This is accomplished by bounding the privacy loss of the query using the concept of epsilon, which ensures that the query's output will not be significantly affected by the presence or absence of any single individual in the data set. This allows for the protection of individual privacy while still allowing for the analysis of large data sets.

In [130] the authors add a differential privacy layer to a deep network using the Differentially Private Stochastic Gradient Descent algorithm (see Algorithm 3) that modifies the optimization process in a deep network adding some noise. This algorithm trains the model by obtaining the parameters θ via minimizing the empirical loss function $\mathcal{L}$.

Here we assume that the gradient of the loss function has a bounded L^2 norm, therefore we ask for the loss function to be a Sobolev function, $\mathcal{L} \in \mathcal{W}^{1,2}$, which is a weaker condition than being a Lipschitz function.

This additional privacy layer is expected to lower the model performance, both in accuracy and in the training time, due to the extra computations and the necessity of finding the privacy cost ε, δ.

The use of differential privacy in FL has also been studied in works such as [18], where they compare the accuracy for a multi layer perceptron trained on MNIST data for Different Privacy values ε, number of participants and iterations to experimentally evaluate their algorithm. The application of differential privacy in FL with non-IID dataset is not a novelty either. Zhao et al. [141] have previously applied differential privacy to non-IID datasets, including in cases where participants only received data from a single class.

Algorithm 3: Differentially private SGD

Data: Sample points $\{x_i\}_{i=1}^{N}$, loss function $\mathcal{L}(\theta)$, learning rate η_t, noise scale σ, group size L, gradient norm bound C

Result: Model parameters θ_T, and privacy cost ε, δ

begin
  $\theta_0 \leftarrow X \sim \mathcal{N}(\mu, \sigma^2)$
end
for $t \in [T]$ **do**
  Random sample L_t with sampling probability $\frac{L}{N}$
  for $i \in L_t$ **do**
    `/* Compute gradient */`
    $g_t(x_i) = \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$
    `/* Clip gradient */`
    $\overline{g}_t(x_i) = g_t(x_i) / \max(1, \frac{1}{C} \cdot \|g_t(x_i)\|_2)$
    `/* Noise addition */`
    $\tilde{g}_t = \frac{1}{L} \cdot (\sum_i \overline{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$
    `/* Descent */`
    $\theta_{t+1} = \theta_t - \eta_t \tilde{g}_t$
  end
end

6.2.2 Chaotic maps-based encryption

Chaotic maps are a branch of mathematics that investigates dynamic systems capable of generating highly randomized states. These states exhibit complete disorder and apparent irregularity, yet their evolution is determined by the initial conditions of the system. This unpredictability can be harnessed for encryption purposes.

Chaotic maps algorithms for encryption are highly regarded for their ability to deliver a combination of high speed, reasonable computation, and strong security. It's worth noting that the specific implementation and design choices for encryption with chaotic maps can vary. Different chaotic maps can be used, such as the logistic map, Henon map, or Lorenz system, depending on the desired properties and security requirements. In this research, logistic map is the selected map for testing our proposal. The contents of this section are a novel research topic developed by the author and her advisor in [37].

The logistic map, a recurrence relation of degree 2 or polynomial mapping, is widely recognized as an archetypal instance that demonstrates the emergence of complex and chaotic behavior from simple nonlinear dynamical equations. The logistic map is defined as

$$x_{i+1} = r \cdot x_i \cdot (1 - x_i) \tag{6.2}$$

where the parameter r fall within the interval $[0, 4]$ in order to ensure that x_n remains bounded on $[0, 1]$. When $r \in [3.57, 4]$ the logistic map is chaotic [142] (see figure 6.1). In this research, the value of r is assigned as 3.8 to ensure

chaotic behavior.

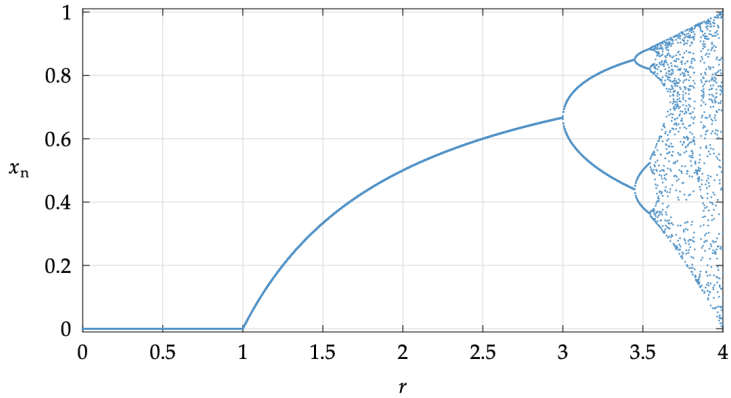


Figure 6.1: Dynamics of the logistic map

Algorithm 4: Logistic map-based encryption

Data: Original plain data ($\mathcal{D}$)
Result: Cipher data (Γ)
Key Generation: Generate the r parameter
Initialization: Choose an initial value x_0
Encryption:
for $i = 1$ **to** n **do**
  `/* Calculate the chaotic value */`
  ; $x_{i+1} = r \cdot x_i \cdot (1 - x_{i-1})$
  $\Gamma[i] = \mathcal{D}[i] \oplus \text{Frac}(X[i])$
end
Output: The cipher data obtained from the encryption process

The distinct characteristics exhibited by chaotic systems, including determinism, ergodicity, and sensitivity to initial conditions, make them a compelling option for constructing cryptographic systems. These properties share similarities with the desirable properties of a robust cryptosystem, such as confusion and diffusion. One of the advantadges of chaos-based encryption techniques is their computational efficiency [143]. The encryption process with chaotic maps is as follows:

1. Key Generation: Chaotic maps require a secret key to initialize the map. The key should be kept secret, as it determines the encryption/decryption process. In the case of the logistic map, the parameter r determines the chaotic behavior of the map.

2. Chaotic Map Iteration: The chaotic map takes a value and generates iteratively a new value based on its mathematical definition and the

Algorithm 5: Logistic map-based decryption

Data: Cipher data (Γ)
Result: Original plain data ($\mathcal{D}$)
Key Generation: Generate the r parameter
Initialization: Choose an initial value x_0
Decryption:
for $i = 1$ **to** n **do**
    `/* Calculate the chaotic value */`
    ; $x_{i+1} = r \cdot x_i \cdot (1 - x_{i-1})$
    $\text{Frac}[i] = \mathcal{D}[i] \oplus \Gamma(X[i])$
end
Output: The plain data obtained from the decryption process

previous value. The inherent chaotic nature of the map guarantees that even a slight alteration in the initial value can yield a significatively different output.

3. Obfuscation: The chaotic map's output could be combined with the original data through obfuscation operations. The aim of this stage is to make the relationship between the original and the encrypted data as complex and nonlinear as possible.

4. Iterations and key updating: During the encryption process, it is common to employ multiple iterations of the chaotic map along with key updates. Following each iteration, the key may undergo changes to introduce additional randomness and strengthen the security of the encryption.

5. Output: The final output of the encryption process is the cipher data, which is the encrypted form of the original plain data. It should appear random and be statistically independent of the original data.

6. Decryption: The same chaotic map is applied iteratively to the cipher data using identical initial conditions, parameters and key as in the encryption phase to retrieve the original plain data.

One possible obfuscation operation is shown in Algorithm 4), where XOR ($\oplus$) is applied between each element of the data set (whether it is plain data or cipher data) and the fractional part of the chaotic value generated by the logistic map at that moment. XOR the i-th element of the plain data with the fractional part of x_i to obtain the i-th element of the cipher data. It is important to note that XOR ($\oplus$) is a bitwise operation, which means that it is applied independently to each corresponding pair of bits in the data elements and chaotic values. This allows for a reversible operation (Algorithm 5), as performing XOR between the encrypted data and the same encryption key (or parameter) will yield the original data.

Encryption with chaotic maps offers certain advantages, such as a high degree of randomness, sensitivity to initial conditions, and resistance to various

attacks. However, it also poses challenges in terms of stability, security analysis, and the need for efficient chaotic map implementations. it's important to note that in the context of chaotic maps, the terms *encryption* and *decryption* might not be the most accurate. Chaotic maps are primarily used for generating pseudorandom sequences or for generating chaotic behaviour, rather than encryption and decryption.

Part III

Discussion

Chapter 7

Conclusions

Federated learning is a new approach to distributed machine learning that is suitable for highly regulated industries due to its promise of security and privacy. In this thesis our aim was to advance the research field in three different aspects:

- federated learning principles: which algorithmic methods are optimal for the federated model,
- privacy,
- and data heterogeneity

for connectionism-based models, that is, models that aim to model the human brain, such as neural networks and fuzzy cognitive maps.

Regarding the first point, we have worked in two different advances: aggregation methods and federated Fuzzy Cognitive Maps.

The original proposal of federated learning contemplated aggregating the local participants' model via an arithmetic mean of their parameters. We defined several new aggregation methods based on weighted average and characteristics of the federated system (size of participants, accuracy of the local models, contribution of each participant...) and compared them with the arithmetic mean. Our experimental results show that a weigthed average based on the normalized inverse accuracy of the local models outperform all other aggregation strategies. Our conclusion is that the aggregation method in a federated learning system is another parameter that has to be investigated while setting the process, and that this weighted average may be a better choice than the usual arithmetic mean.

Another element in the original proposal of federated learning was the algorithm used to train the local models: neural networks. This family of connectionism-based models is of interest due to the recent advances made by deep learning, its ability to perform feature extraction in an automated way, the capacity to learn hierarchical feature abstractions of the data... Nevertheless

there are other algorithms that may outperform or have more appealing characteristics than neural networks, such as better suited models for low dimensional data, or interpretable models.

In recent years, there have been several attempts to create a federated version of conventional machine learning algorithms, but to the best of our knowledge there wasn't a federated Fuzzy Cognitive Maps, a connectionism-based model represented by a fuzzy-graph structure that describes causal reasoning, and that trained becomes an explainable machine learning model. We have developed a new methodology to train a federated FCM across several participants without sharing their local data that, according to our experimental results, improves the accuracy of a single Fuzzy Cognitive Map trained in the whole data, and the accuracy in each participant before the federation. Given the characteristics of FCMs related to explainability and causal reasoning, we think this new approach could be used by practitioners in highly regulated industries to share an explainable model in a secure way.

Regarding privacy and data heterogeneity, our research has developed a new encryption layer for federated learning processes based on chaotic maps, that offers a combination of high speed, reasonable computation, and strong security. Our experimental results, comparing the performance of a federated learning model without an additional privacy layer, with differential privacy, and with chaotic map encryption, show that our privacy method does not worsen the performance metrics of the model when compared with the non-private of differentially private system.

Moreover, to ensure the simulation closely resembles a real-world implementation of Federated Learning we assume one or more of the participants has incomplete data with a different distribution. Our proposal contemplates that, if one or more of the participants does not have a complete dataset, meaning that one of the features is missing (and therefore the features are non-IID), the server will also send, in a encrypted fashion, the distribution of the feature for any other participant so that the lacking feature can be imputed with an L^2 norm. Again, our experimental results show that our imputation proposal is valid and of interest for the applications of federated learning, since the non-iid scenario is a natural occurrence in real-world problems.

Chapter 8

Future research

This thesis has opened several research fronts that we would like to attack in the next years, including:

- Different aggregation methods for federated learning that do not involve a weighted average,
- deeper research on the explanations behind the outperformance of the accuracy-based aggregation method,
- use of different privacy methods, such as Secure Multi-Party Computation,
- deeper research on non-iid datasets and vertical federated learning,
- our research has focused usually on classification problems, we would like to understand if our results follow readily in the case of regression problems,
- FCMs have appealing characteristics when trained as machine learning algorithms, we would like to understand their use to federated systems,
- explainability for federated learning,
- in terms of security and privacy, our research has dealt with privacy, we would like to understand the processes behind security in federated learning,
- finally, our work has focused on three out of four research bottlenecks in federated learning (algorithmic principles, security and privacy and data heterogeneity), we would like to also understand the problems behind communication overhead in federated learning.

Part IV

Appended Papers

Chapter 9

Benchmarking federated strategies in Peer-to-Peer Federated learning for biomedical data

Heliyon 9 (2023) e16925

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Benchmarking federated strategies in Peer-to-Peer Federated learning for biomedical data

Jose L. Salmeron [a,b], Irina Arévalo [c,*], Antonio Ruiz-Celma [d]

[a] CUNEF Universidad, Madrid, Spain
[b] Universidad Autónoma de Chile, Chile
[c] Universidad Pablo de Olavide, Seville, Spain
[d] Universidad de Extremadura, Badajoz, Spain

ARTICLE INFO

Keywords:
Federated learning
Privacy-preserving machine learning

ABSTRACT

The increasing requirements for data protection and privacy have attracted a huge research interest on distributed artificial intelligence and specifically on federated learning, an emerging machine learning approach that allows the construction of a model between several participants who hold their own private data. In the initial proposal of federated learning the architecture was centralised and the aggregation was done with federated averaging, meaning that a central server will orchestrate the federation using the most straightforward averaging strategy. This research is focused on testing different federated strategies in a peer-to-peer environment. The authors propose various aggregation strategies for federated learning, including weighted averaging aggregation, using different factors and strategies based on participant contribution. The strategies are tested with varying data sizes to identify the most robust ones. This research tests the strategies with several biomedical datasets and the results of the experiments show that the accuracy-based weighted average outperforms the classical federated averaging method.

1. Introduction

Artificial intelligence applications in healthcare are increasing every day. These applications have the ability to advance the healthcare industry by, for instance, supporting clinical decision making, risk prediction, developing early warning systems for patients, increasing the accuracy and timeliness of diagnosis, improving patient–physician interaction, and optimizing operations and resource allocation [21].

Federated learning is a new approach for distributed artificial intelligence that aims to have several agents train a deep learning model in a collaborative and secure way, without sharing any private data. This training is done the following way: a central server defines a deep learning model and sends it to the agents, who train the model in their private data. Then, they send the parameters of the model (weights or gradients) back to the server, who aggregates these data in order to find a global federated model, which in turn is delivered back to the agents to be retrained in their data. This process is iterated until convergence.

In the initial definition of the federated learning approach, the aggregation step is done by averaging the model parameters. Nevertheless, other aggregation methods may be of more interest since they can improve the performance of the model by giving more weight to different agents depending on their size or the performance of the local models in their data.

The main contributions of this research are two-fold:

* Corresponding author.
E-mail addresses: joseluis.salmeron@cunef.edu (J.L. Salmeron), iarebar@alu.upo.es (I. Arévalo), aruiz@unex.es (A. Ruiz-Celma).

https://doi.org/10.1016/j.heliyon.2023.e16925
Received 5 December 2022; Received in revised form 31 May 2023; Accepted 1 June 2023

Available online 2 June 2023

1. Several aggregation strategies are proposed, such as weighted averaging aggregation using the dataset size, weighted average using the normalized inverse of the local test accuracy, weighted averaging aggregation using the dataset size and accuracy, weighted average using the contribution of the participant, and weighted sum using the inverse contribution of the participant. Federated averaging is included for comparison.
2. The strategies are tested with different data sizes on each participant. This allows analyzing the strategies under different circumstances and identifying those that are more robust.

The rest of this paper is organized as follows. We discuss the theoretical background of federated learning in section 2. The different federated strategies are described in section 2.3. The methodological proposal can be found in section 3, while section 4 shows the results of the experimental approach, serving as a benchmark. Finally, the authors draw a conclusion in section 5.

2. Fundamentals

2.1. Related work

Federated learning is an emerging approach for distributed artificial intelligence in which the different data owners (or participants) train collaboratively a machine learning model [17,22]. The model is updated (trained) in the own private data of each participant and then the trained model is sent for aggregation to a central server or one of the participants. It was first proposed by McMahan et al. [18] and further developed in Konecny et al. [14] and McMahan and Ramage [19]. The main advantage of federated learning is the training of a model with the private data of each participant keeping the security and compliance requirements while improving their models [3]. It also allows the use of more accurate models with low latency, ensuring privacy and less power consumption [31].

Numerous surveys and literature reviews have extensively examined the body of work documented in academic literature pertaining to architectures, approaches, utilization, and applications of federated learning. In the healthcare domain, Hoyos et al. [10] present federated learning approaches (horizontal, vertical and transfer learning) for FCMs for the prediction of mortality and the prescription of treatment of severe dengue. Antunes et al. [4] outline a broad architecture for federated learning applied to healthcare data, drawing upon key insights derived from the literature review. Li et al. [15] analyse recent literature on the utilization of federated learning, outlining various federated learning architectures and classification models. Nguyen et al. [20] provide a comprehensive and up-to-date review of the latest advancements in federated learning within crucial healthcare domains, encompassing health data management, remote health monitoring, medical imaging, and COVID-19 detection. Xu et al. [30] provide an overview of the common solutions to address statistical challenges, system challenges, and privacy issues in federated learning. They also highlight the potential implications and opportunities that federated learning holds for the healthcare sector. Furthermore, numerous studies have sought to optimize federated learning and broaden its practical applications, with research efforts spanning areas such as computation fusion [32], data transmission [23,26], as well as privacy and security-related concerns [9].

2.2. Architecture

Fully decentralised learning aims to replace server-based communication with peer-to-peer communication among individual clients as its core concept. Each round in fully decentralized algorithms involves a client performing a local update and sharing information with their neighbours in the graph. In this research, clients share the model with all participants, making it a fully connected peer-to-peer architecture (see Fig. 1).

Peer-to-peer architectures can have a significant impact on federated learning. With peer-to-peer communication, clients can collaborate and share locally trained models directly with each other, which can result in faster and more efficient learning compared to a centralized architecture. Peer-to-peer architectures can also provide better privacy and security as the clients can keep their data locally and only share the necessary information with their peers.

Recently, a fully decentralised solution where participants collaborate asynchronously and communicate in a peer-to-peer fashion, without any central server to orchestrate a global state of the system or even to coordinate the protocol, is proposed in [28]. In this scenario, peer-to-peer architectures can also have some challenges. For example, it can be more difficult to coordinate and manage the communication between clients, and it may require additional mechanisms to ensure the consistency of the model across all clients. To overcome this challenge, this paper focuses on a fully connected peer-to-peer architecture. Moreover, this kind of systems may not be suitable for large-scale federated learning scenarios due to the high communication overhead between clients. However, the goal of this research is to present a fault-tolerant architecture in case of failure of a central server and/or in multiple participants.

In this case, the group of autonomous peers run iteratively multiple training rounds to update the federated model [28,29]. Indeed, peer-to-peer algorithms include scalability-by-design to large sets of devices thanks to the locality of their updates [13]. In addition, a decentralised peer-to-peer architecture intrinsically provides an additional some security guarantee as it becomes much more tough for any third party to get the full state information of the system [29].

A peer-to-peer federated learning process needs a minimum of two participants. In this case, there is not a central server managing the federated model and the communications with the participants. In a peer-to-peer architecture, each participant receives the trained models of the remainder participants and then averages the participants' models to obtain a federated model trained with the private data. The participants own the data and train the partial models. This process can be iterated as many times as needed. The process is shown in Fig. 1 and is as follows:

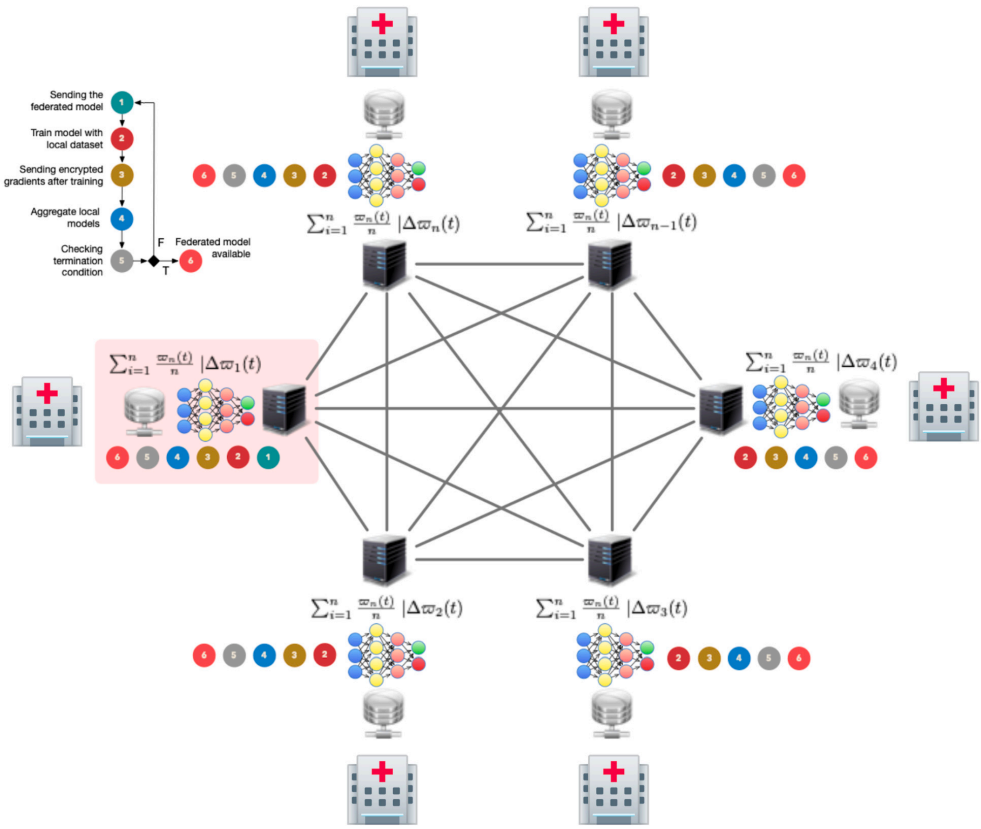


Fig. 1. Fully connected peer-to-peer federated learning architecture.

1. A participant initiates the federated learning process by sending an initial model to all the other participants. If this is the initial iteration, the federated model is dispatched by the participant triggering the process.
2. Each participant trains the received model using their own private data.
3. Each participant sends the parameters of the model in a private way (usually encrypting the data to be sent) to the remaining participants.
4. Every participant aggregates the partial models with the updated parameters and builds the federated model according to a federation strategy.
5. Every participant checks a termination condition in either accuracy of the model in a test dataset or number of iterations. If it is accomplished, the federated learning process ends, otherwise the process iterate again from step 1.

The target of the federated learning process is to minimize the total loss for all participants, computed as in Equation (1),

$$\mathcal{L}^* = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i(D_i, \Phi) \tag{1}$$

where $\mathcal{L}^*$ is the loss function for the federated model, $\mathcal{L}_i$ is the loss function for each participant in the federation, D_i is the dataset of the participant i and Φ is the federated model parameters, n is the number of participants.

The federated learning process trains a model between different participants without the sharing of private data. Nevertheless, there are other possible risks, like model poisoning, potential attacks to reconstruct the model or the training data from the parameters that the participants send to the central server, or the use of attack models [5,27]. As a consequence, there have been several

advances in the use of privacy-preserving methods such as Differential Privacy or Homomorphic Encryption in federated learning, see [1,2,12,11,6].

Algorithm 1: Federated learning.

Data: The K clients are indexed by k; B is the local mini-batch size, E is the number of local epochs, T is the maximum iteration number, and η is the learning rate.
Federation process
initialize Φ_0
for *each round* $t = 1, 2, \dots, T$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  for *each client* $k \in S_t \mid S_t \sim U(m)$ *in parallel* **do**
    $\Phi_{t+1}^k \leftarrow client\_update(k, \Phi_t)$
  end
  $\Phi_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \cdot \Phi_{t+1}^k$
end
Training process
client_update(k, Φ): $B \leftarrow$(split $\mathcal{P}_k$ into batches of size B);
for *each local epoch* i *from 1 to* E **do**
  for *batch* $b \in B$ **do**
    $w \leftarrow \Phi - \eta \nabla l(\Phi; b)$
  end
end
return Φ to server

2.3. Federation strategies

As mentioned before, the aggregation method for federated learning is an important parameter of the process. The original definition contemplated the arithmetic mean of the parameters of the model to obtain the federated model.

In this research, the authors propose a series of different aggregations and compare them in our experimental section (see section 4). Assuming that the parameters of the model at iteration j are as shown in Equation (2),

$$\Phi_j = [\Phi_{j1}, \Phi_{j2}, \cdots, \Phi_{jn}] \tag{2}$$

where n is the number of participants, D_i is the dataset of the participant i, and Φ'_j is the parameters (weights or gradients) of the federated model, the functions of the parameters that we will discuss are the following:

- Average of the parameters (weights or gradients):

$$\Phi'_j = \frac{1}{n} \sum_{i=1}^{n} \Phi_{ji} \tag{3}$$

  where every participant contributes the same to the global model.
- Weighted averaging aggregation using the normalized size of each participants' dataset:

$$\Phi'_j = \sum_{i=1}^{n} \frac{|D_i|}{\sum_{k=1}^{n} |D_k|} \cdot \Phi_{ji} \tag{4}$$

  where every participant contributes to the global model proportionally to the size of their data, and agents with less information will affect less to the final model.
- Weighted average using the normalized inverse accuracy of the model in a test set of each participant:

$$\Phi'_j = \sum_{i=1}^{n} \frac{1/\mathrm{acc}_{ji}}{\sum_{k=1}^{n} 1/\mathrm{acc}_{jk}} \cdot \Phi_{ji} \tag{5}$$

  where the individual models add to the global model inversely to their performance metric, trying to give more weight to the less accurate models in order to improve their metric in their datasets.
- Weighted average using the accuracy and the size of the dataset:

$$\Phi'_j = \sum_{i=1}^{n} \frac{\mathrm{acc}_{ji}|D_i|}{\sum_{k=1}^{n} |D_k|} \cdot \Phi_{ji} \tag{6}$$

  where the contribution of each participant's model depends on both the accuracy of the model and the size of the dataset.
- Weighted average using the contribution (C) of the participant, that is, the normalized inverse of the absolute difference between the loss of the participant's model and the loss of the global model when applied to the participants' data as shown in Equation (7)

$$C_{ji} = |\mathcal{L}_j^*(D_i, \Phi) - \mathcal{L}_j(D_i, \Phi)| \tag{7}$$

and

$$\Phi'_j = \sum_{i=1}^{n} \frac{C_{j,i}}{\sum_{k=1}^{n} C_{jk}} \cdot \Phi_{ji} \tag{8}$$

- Weighted sum using the inverse contribution of the participant:

$$\Phi'_j = \sum_{i=1}^{n} \frac{1/C_{ji}}{\sum_{k=1}^{n} 1/C_{jk}} \cdot \Phi_{ji} \tag{9}$$

3. Methodological proposal

The federated learning proposal starts with a central server sending an untrained model to the participants. As a first step, each hospital trains this model with their training data, evaluates it in their test data, and sends the parameters of the trained model back to the server. After receiving all the parameters from all the participants, in this proposal the server aggregates the parameters using one of the aggregation methods described in Section 2.3 to obtain the global method. This process is iterated until convergence. In this use case the authors have proposed a peer-to-peer architecture, and have not included an additional encryption layer, but it is possible and desirable to do so in real-life applications in healthcare.

In the following experiments, the initial model will be a dense neural network made of five layers followed by a non-linear ReLU function and a dropout layer for regularization (Fig. 2). For training, the loss function will be computed using the Binary Cross Entropy.

As a setup for the experiments, the authors assume that several hospitals with their own private data wish to train a deep learning model for diagnosis of a disease, but the size of their data is not large enough for training an accurate model. The data of all the hospitals should not be combined due to data regulations given their special sensitivity. Therefore, for each one of the experiments the corresponding dataset will be randomly split in five different subsets, to simulate five hospitals. In a first test all of them will have the same amount of data, obtained by splitting evenly the dataset. The other three tests will be random splits among the participants, where in the last two we have forced that there would be several participants with a very small number of samples (less than 10%). The distribution of the variables, including the percentage of positive cases for the target, will also vary from one hospital to another.

In every case, each participant's data will be split into a train and a test dataset for training and evaluation.

The metric we will use to compare the performance of the different models and methods is the accuracy on the test set. Given a classification model, its performance can be summarized with a confusion matrix, a table that shows the number of real positive and negative samples in our dataset versus the number of positive and negative values that our classifier predicts.

The accuracy of a model is the ratio of number of correct predictions to the total number of input samples, that is, the sum of the main diagonal of the confusion matrix divided by the sum of all values in the table, which is the amount of times the classifier got the prediction right. This metric varies between 0 and 1, where 0 is the worst case scenario, and 1 is associated to a perfect classifier. Moreover, a random binary classifier such as a coin toss has an accuracy of 0.5, and therefore this is a first baseline for any balanced binary classifier, understanding that an accuracy close to 0.5 is as bad as a random guess.

The results of the experiments will be shown as follows: for each experiment, each line will represent one of the five participant hospitals. The columns represent the number of the partition, its size, and the percentage of positives in the partition, and the accuracy of a model evaluated in each participant's test set. The first one is the local model in every participant without any federation, and the following are the results for each aggregation method described in Section 2.3: Federated Average (Equation (3)) and weighted sums with size (Eq. (4)), inverse accuracy (Eq. (5)), size and accuracy (Eq. (6)), contribution (Eq. (8)), and inverse contribution (Eq. (9)). The average accuracy for all participants in each experiment will also be included and used for comparison of the aggregation methods.

Table 1 summarizes how the results of the experiments will be shown, and what federation strategies are being evaluated.

These results will serve as a benchmark for which averaging strategy improves the most the performance of the federated learning process, by comparing the accuracy in different datasets of the competing approaches. The benchmarking methodology is as follows:

1. Define the problem: Firstly, define the problem that the AI model is intended to solve. In this research, we have selected four medical problems (breast cancer, chronic kidney disease, Parkinson's disease, and heart disease) to better validate our proposals.
2. Identify the relevant data: We have selected well-known datasets for each of these medical problems to ensure reproducibility.
3. Preprocess and split the data: The datasets have already been preprocessed and are ready for use. The next step is to split the data into training and testing sets to evaluate the performance of the model. The training set will be used to train the model, while the testing set will be used to measure its accuracy on unseen data.
4. Train the model: Train the Federated FCM model using different aggregation strategies.
5. Evaluate the model: Evaluate the performance of the model using the test set using the accuracy metric.
6. Compare the results: Conduct a comparison of the outcomes achieved by each aggregation strategy with the remaining ones to ascertain whether there are any significant performance differences among the strategies.

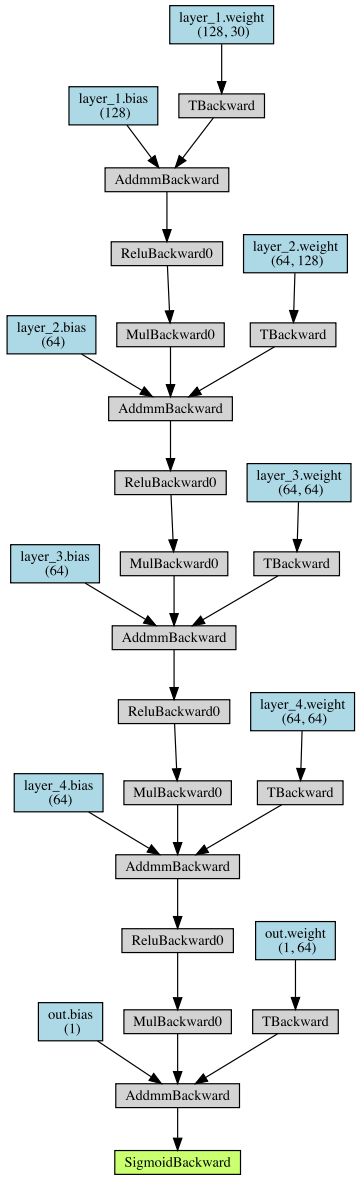


Fig. 2. Deep Neural Network topology for the experiments.

Table 1
Column names and federation strategies.

Column name	Definition	Federation strategy
Part	Partition number	
Size	Percentage of the total samples in this agent	
Pos (%)	Percentage of positives in the partition	
Acc (local)	Accuracy in the non-federated case	
Acc Federated Averaging (Fed)	Accuracy of the federated model using fed averaging	Equation (3)
Acc. Size (Fed)	Accuracy of the model using size-based averaging	Equation (4)
Acc. Accuracy (Fed)	Accuracy of the federated model using accuracy-based averaging	Equation (5)
Acc. Size & Acc. (Fed)	Accuracy of the federated model using both accuracy and size in the averaging	Equation (6)
Acc. Cont (Fed)	Accuracy of the federated model using the contribution of the participants	Equation (8)
Acc. Inv. Cont. (Fed)	Accuracy of the federated model using the inverse contribution of the participants	Equation (9)

4. Experiments

4.1. Experiment 1 - Breast Cancer

The Breast Cancer Wisconsin Dataset contains descriptions of features of the nucleus of a breast mass, obtained from digitized images of the fine-needle aspirate, for 569 patients, where 212 are malignant and 357 are benign tumors. This dataset is publicly available [8]. More details can be found in [24], [25].

The results of the federated learning process for this dataset are shown in Table 2. As we can see, the federation improves the accuracy metrics in general, but there are differences between the aggregation methods used. In the first case of the even dataset all accuracies are improved, but in the first uneven split, the contribution aggregation of the difference of the losses does not induce an increase in the performance. In the second uneven split, more extreme than the previous one, with one participant with only 3% of the data, the Federated Averaging does not improve the performance of the local models, and in the last one, with two participants with 6% and 7% of the data, the weighted aggregation using the size of each participant and the size and the accuracy show a lower accuracy than the first iteration of local models.

Summing up, for this experiment the accuracy-based aggregation and the inverse contribution are the only aggregation methods that improve the performance of the local models after the federation process.

4.2. Experiment 2 - chronic kidney disease

The term chronic kidney disease (CKD) describes all degrees of decreased renal function. It is more prevalent in the elderly population and it is estimated that affects 10–15% of the world population. CKD is not often identified in premature stages.

The Chronic Kidney Dataset contains 25 features and a target that represents whether the patient has the Chronic Kidney Disease, for 400 patients, one third of which did not have the disease and two thirds that did. It is publicly available at the UC Irvine Machine Learning Repository [8].

The results of the experiments for this dataset are shown in Table 3. In this case, the aggregation using the inverse contribution does not improve the accuracy for all participants, since this metric worsens for the first uneven split. The size-based and contribution-base aggregation do not increase the accuracy of the local models for this split as well, while the Federated Averaging fails for the last uneven split and the size and accuracy weighted average for the second one. As in the previous experiment, the only aggregation method that impoves the accuracy for all cases is the accuracy-based one.

4.3. Experiment 3 - Parkinson's

Parkinson's is a neurodegenerative disease that produces alterations in gait and posture that may increase the risk of falls and leads to mobility disabilities. Parkinson's affects about 1% of the world population over the age of 55.

The symptoms generally develop over years and their progressions are very diverse, making the diagnosis of the disease in the early stages extremely difficult.

Table 2
Experiment 1 - Breast Cancer.

Part	Size	Pos (%)	Acc (local)	Acc Federated Averaging (Fed)	Acc Size (Fed)	Acc Accuracy (Fed)	Acc Size & acc (Fed)	Acc Cont (Fed)	Acc Inv Cont (Fed)
1	20%	47%	0.5454	0.6363	0.6363	0.3636	0.7272	0.6363	0.7272
2	20%	41%	0.4166	0.5000	0.5833	0.7500	0.7500	0.6666	0.7500
3	20%	31%	0.5454	0.7272	0.8181	0.5454	0.8181	0.6363	0.8181
4	20%	35%	0.5833	0.5833	0.6666	0.5833	0.7500	0.4166	0.8333
5	20%	32%	0.5833	0.6666	0.6666	0.7500	0.4166	0.5833	0.5833
Avg	–	–	0.5348	**0.6227**	**0.6742**	**0.5984**	**0.6924**	**0.5879**	**0.7424**
1	20%	12%	0.9090	0.9090	0.8000	0.8181	0.8888	0.8181	0.8181
2	22%	30%	0.6153	0.5384	0.7692	0.5333	0.5714	0.7058	0.5000
3	18%	47%	0.5000	0.6000	0.3636	0.4444	0.5454	0.4285	0.4000
4	17%	17%	0.7000	0.7000	0.9000	0.7500	0.7500	0.6363	0.8000
5	24%	39%	0.4285	0.5714	0.6666	0.6666	0.4285	0.3571	0.7142
Avg	–	–	0.6306	**0.6637**	**0.6999**	**0.6425**	**0.6368**	0.5892	**0.6465**
1	49%	28%	0.5714	0.6428	0.6333	0.6896	0.7142	0.7096	0.6896
2	3%	40%	0.5000	0.0000	1.0000	0.6666	1.0000	0.5000	0.6666
3	15%	40%	0.8888	1.0000	0.5000	0.8333	0.6250	0.5000	0.8333
4	5%	63%	0.3333	0.3333	1.0000	0.6666	0.0000	0.6666	0.6666
5	29%	20%	0.4117	0.5882	0.7500	0.5789	0.7368	0.8666	0.5789
Avg	–	–	0.5410	0.5128	**0.7766**	**0.6870**	**0.6152**	**0.6486**	**0.6870**
1	48%	30%	0.6666	0.7333	0.7419	0.6969	0.6774	0.7000	0.8000
2	7%	33%	0.6666	0.6666	0.7500	0.7500	0.5000	1.0000	0.6666
3	6%	27%	0.6666	0.6666	0.8000	1.0000	0.5000	0.7500	0.3333
4	16%	15%	1.0000	1.0000	0.3750	0.8000	0.8750	0.8750	1.0000
5	23%	38%	0.4545	0.4545	0.5000	0.3636	0.5000	0.5454	0.7272
Avg	–	–	0.6970	**0.7042**	0.6334	**0.7221**	0.6104	**0.7741**	**0.7055**

Table 3
Experiment 2 - Chronic Kidney Disease.

Part	Size	Pos (%)	Acc (local)	Acc Federated Averaging (Fed)	Acc Size (Fed)	Acc Accuracy (Fed)	Acc Size & acc (Fed)	Acc Cont (Fed)	Acc Inv Cont (Fed)
1	20%	73%	1.0000	0.9375	1.0000	1.0000	1.0000	0.8750	0.8583
2	20%	58%	0.9375	1.0000	0.9375	1.0000	1.0000	1.0000	1.0000
3	20%	56%	0.8125	0.8750	0.9375	0.8750	0.9375	0.9375	0.8750
4	20%	65%	0.9375	1.0000	1.0000	0.8750	1.0000	1.0000	1.0000
5	20%	60%	1.0000	1.0000	0.8750	1.0000	0.8750	1.0000	1.0000
Avg	–	–	0.9375	**0.9625**	**0.9500**	**0.9500**	**0.9625**	**0.9625**	**0.9467**
1	14%	36%	1.0000	1.0000	1.0000	0.9166	0.9000	0.9285	0.9166
2	29%	67%	0.9583	0.9583	0.9583	1.0000	0.9545	0.8947	0.9565
3	20%	77%	0.8750	0.9375	0.9285	1.0000	1.0000	0.9411	0.8571
4	13%	48%	1.0000	1.0000	0.9000	1.0000	1.0000	1.0000	1.0000
5	24%	68%	0.9500	0.9000	0.9583	0.9200	0.9545	1.0000	0.9523
Avg	–	–	0.9567	**0.9592**	0.9490	**0.9673**	**0.9618**	0.9528	0.9365
1	52%	64%	1.0000	0.9761	0.9756	0.9250	0.9767	0.9750	0.9500
2	3%	50%	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000	1.0000
3	14%	76%	1.0000	1.0000	1.0000	1.0000	0.9285	1.0000	0.9285
4	5%	83%	0.7500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	26%	51%	0.9523	0.9523	0.9090	0.9523	0.8333	0.9500	0.9130
Avg	–	–	0.9404	**0.9857**	**0.9769**	**0.9755**	0.8477	**0.9855**	**0.9583**
1	44%	63%	0.9428	0.9428	0.9743	0.9750	0.9736	0.9047	0.9756
2	8%	72%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	6%	43%	0.8000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000
4	13%	43%	0.9090	0.9090	1.0000	1.0000	1.0000	1.0000	1.0000
5	30%	72%	1.0000	1.0000	0.8571	0.9444	1.0000	1.0000	1.0000
Avg	–	–	0.9304	0.9304	**0.9663**	**0.9839**	**0.9947**	**0.9809**	**0.9951**

Table 4
Experiment 3 - Parkinson's.

Part	Size	Pos (%)	Acc (local)	Acc Federated Averaging (Fed)	Acc Size (Fed)	Acc Accuracy (Fed)	Acc Size & acc (Fed)	Acc Cont (Fed)	Acc Inv Cont (Fed)
1	20%	65%	0.4285	0.7142	0.8571	1.0000	1.0000	1.0000	0.8571
2	20%	78%	0.7500	0.8750	1.0000	1.0000	0.8750	1.0000	1.0000
3	20%	75%	0.5000	0.8750	0.8750	0.8750	0.8750	0.8750	0.8750
4	20%	78%	0.7500	0.7500	0.8750	0.8750	0.8750	0.5000	0.7500
5	20%	79%	1.0000	1.0000	0.6250	0.8750	0.8750	0.8750	0.8750
Avg	–	–	0.6857	**0.8429**	**0.8464**	**0.9250**	**0.9000**	**0.8500**	**0.8714**
1	11%	61%	0.6000	0.8000	0.6000	0.7142	0.8000	0.8000	0.6666
2	25%	71%	0.9000	0.9000	0.9000	0.7272	0.8461	0.8000	0.7777
3	27%	87%	0.8181	0.9090	1.0000	0.8750	1.0000	1.0000	0.8000
4	10%	45%	0.5000	0.7500	0.6000	0.7500	1.0000	0.7500	0.8000
5	27%	84%	0.9090	0.9090	0.8000	0.9090	0.8181	0.8333	1.0000
Avg	–	–	0.7454	**0.8536**	**0.7800**	**0.7951**	**0.8929**	**0.8367**	**0.8089**
1	50%	78%	0.8500	0.8500	0.8181	0.8500	0.7894	0.6666	0.8500
2	6%	57%	0.6666	0.3333	1.0000	0.7500	0.6666	0.8000	0.7500
3	13%	84%	0.7142	0.8571	0.7500	1.0000	0.3333	0.7142	0.8571
4	4%	57%	1.0000	0.6666	0.6666	0.6666	0.7500	0.6666	0.3333
5	21%	79%	0.8750	0.8750	0.7777	0.6250	0.7777	0.6250	0.4285
Avg	–	–	0.8212	0.7164	0.8025	0.7783	0.6634	0.6945	0.6438
1	44%	63%	0.9428	0.9428	0.9743	0.9750	0.9736	0.9047	0.9756
2	8%	72%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	6%	43%	0.8000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000
4	13%	43%	0.9090	0.9090	1.0000	1.0000	1.0000	1.0000	1.0000
5	30%	72%	1.0000	1.0000	0.8571	0.9444	1.0000	1.0000	1.0000
Avg	–	–	0.9304	0.9304	**0.9663**	**0.9839**	**0.9947**	**0.9810**	**0.9951**

This dataset was created by Max Little of the University of Oxford [16], in collaboration with the National Centre for Voice and Speech, in Denver, Colorado and is composed by a range of biomedical voice measurements from patients. Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals.

Table 4 shows the results of the different federation processes in this dataset. In this case, the splits are more polarized, finding that with the even split and the first uneven split all aggregation methods improve the performance of the local models, while for the second uneven split, with two participants with 4% and 6% of the data, no aggregation increases the accuracy. In the last uneven split, all aggregation methods improve the local models except the Federated Averaging.

4.4. Experiment 4 - heart disease

Heart disease describes a range of conditions that affect the heart, including blood vessel diseases, such as coronary artery disease, arrhythmia and congenital heart defects among others.

Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data analysis.

The Heart Disease dataset includes data from noninvasive test results of patients undergoing angiographies in order to study the possibility of angiographic coronary disease in them. The data collected include 14 attributes of the patients. For more information about this dataset, see [7].

In Table 5 we find the results of the different federation processes changing the aggregation method. With the even split we see that all aggregation methods improve the accuracy of the local models but for the aggregation based in a weighted average of the size of the participants and the accuracy of the local model. In the case of the first uneven split, the only aggregation methods that are able to improve the performance are the size-based and the accuracy-based. On the other hand, for the second uneven split, with two participants with 2% and 4% of the data, all aggregation methods increase the accuracy. Finally, for the last uneven split, only the Federated Averaging and the accuracy-based aggregation improve the performance of the models, resulting on the accuracy-based aggregation being the only aggregation that improves in all cases for this dataset.

5. Conclusions

Federated learning is a distributed artificial intelligence approach that can be very useful for hospitals to build collaboratively a machine learning model in a secure way, without sharing their private data. Nevertheless, the aggregation method is a decisive parameter that can change the performance of the final federated model.

In this research we have proved that the classical Federated Averaging is a reliable aggregation method that improves the performance of the local methods in 11 out of 16 cases that we have contemplated. Nevertheless, there are other aggregation

Table 5
Experiment 4 - Heart Disease.

Part	Size	Pos (%)	Acc (local)	Acc Federated Averaging (Fed)	Acc Size (Fed)	Acc Accuracy (Fed)	Acc Size & acc (Fed)	Acc Cont (Fed)	Acc Inv Cont (Fed)
1	20%	53%	0.7500	0.7500	1.0000	0.7500	0.8333	0.7500	0.9166
2	20%	64%	0.6666	0.8333	0.8333	0.8333	0.7500	0.8333	1.0000
3	20%	44%	0.5833	0.8333	1.0000	0.9166	0.5833	0.8333	0.8333
4	20%	58%	0.9166	0.8333	0.7500	0.8333	0.5833	0.9166	0.9166
5	20%	54%	0.9230	1.0000	0.7692	0.7692	0.6153	0.6923	0.6923
Avg	–	–	0.7679	**0.8500**	**0.8705**	**0.8205**	0.6731	**0.8051**	**0.8718**
1	15%	30%	0.8000	0.7000	1.0000	0.7777	0.9000	0.7142	0.6000
2	26%	63%	0.8750	0.7500	0.7500	0.7142	0.9375	0.9444	0.7647
3	18%	78%	0.9090	0.9090	1.0000	0.8333	0.7692	0.8888	0.7500
4	13%	39%	0.6250	0.8750	0.7777	0.8888	0.7000	1.0000	1.0000
5	28%	51%	0.8888	0.8333	0.9230	0.8888	0.5714	0.9047	0.8666
Avg	–	–	0.8195	0.8134	**0.8902**	**0.8206**	0.7756	0.7963	0.7963
1	50%	55%	0.6666	0.7333	0.7333	0.7500	0.7241	0.8387	0.7096
2	2%	75%	0.5000	1.0000	0.6666	1.0000	0.5000	0.6666	0.6666
3	14%	72%	0.8888	0.8888	0.8000	0.8750	0.8181	0.7272	0.7777
4	4%	71%	0.3333	0.6666	0.3333	1.0000	0.6666	0.7500	0.5000
5	30%	43%	0.6842	0.6315	0.8125	0.8235	0.5882	0.7142	0.7500
Avg	–	–	0.6146	**0.7841**	**0.6692**	**0.8897**	**0.6594**	**0.7394**	**0.6808**
1	47%	58%	0.7500	0.6875	0.8387	0.7500	0.7666	0.8928	0.8787
2	7%	38%	0.8000	0.8000	0.6000	0.8000	0.6000	0.6666	0.6000
3	10%	39%	0.4000	0.4000	0.2500	1.0000	0.2500	0.5000	0.7500
4	9%	31%	0.8750	1.0000	1.0000	0.7142	0.6666	0.8000	0.6000
5	26%	68%	0.9285	0.9285	0.8750	0.7857	0.9333	0.7777	0.8235
Avg	–	–	0.7507	**0.7632**	0.7127	**0.8100**	0.6433	0.7274	0.7305

methods with similar or even better behaviour. The contribution-based aggregation, using the difference between the losses of the global and local model, increases the accuracy in 11 out of 16 cases as well, while the size-based and the inverse contribution-based perform better in one more case. The weighted average using both the size of the participant's dataset and the accuracy of the local model increase the accuracy in only 10 cases out of 16. Finally, the weighted average using the accuracy outperforms all aggregation methods, improving the accuracy in 15 out of 16 cases, that is, in all experiments but one partition where all other methods failed to increase the performance as well.

With these results, the authors believe that an accuracy-based federated learning may perform better than the Federated Averaging classical approach. A fully connected peer-to-peer architecture has been used to show a resilient architecture against different points of failures, including the central server. As limitations of this study, using open-source datasets instead of real-world data in a study on federated learning with medical data may limit the realism, generalization, diversity, quality, and ethical considerations of the research findings.

CRediT authorship contribution statement

Jose L. Salmeron; Irina Arevalo; Antonio Ruiz-Celma: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data included in article/supp. material/referenced in article.

Acknowledgements

Prof. Salmeron research was kindly supported by the project Artificial Intelligence for Healthy Aging (Convocatoria 2021 – Misiones de I+D en Inteligencia Artificial: Inteligencia Artificial distribuida para el diagnóstico y tratamiento temprano de enfermedades con gran prevalencia en el envejecimiento, exp.: MIA.2021.M02.0007) lead by Capgemini Engineering.

References

[1] M. Abadi, A. Chu, I. Goodfellow, B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: 23rd ACM Conference on Computer and Communications Security (ACM CCS), 2016, pp. 308–318, https://arxiv.org/abs/1607.00133.

[2] A. Acar, H. Aksu, S. Uluagac, M. Conti, A survey on homomorphic encryption schemes: theory and implementation, ACM Comput. Surv. 51 (04) (2017), https://doi.org/10.1145/3214303.

[3] K.M. Ahmed, A. Imteaj, M.H. Amini, Federated deep learning for heterogeneous edge computing, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1146–1152.

[4] R.S. Antunes, C.A. da Costa, A. Küderle, I.A. Yari, B. Eskofie, Federated learning for healthcare: systematic review and architecture proposal, ACM Trans. Intell. Syst. Technol. 13 (4) (2022) 1–23, https://doi.org/10.1145/3501813.

[5] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: S. Chiappa, R. Calandra (Eds.), Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 108, PMLR, 26–28 Aug 2020, pp. 2938–2948.

[6] Y. Cheng, Y. Liu, T. Chen, Q. Yang, Federated learning for privacy-preserving ai, Commun. ACM 63 (12) (Nov 2020) 33–36, https://doi.org/10.1145/3387107.

[7] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.J. Schmid, S. Sandhu, K.H. Guppy, S. Lee, V. Froelicher, International application of a new probability algorithm for the diagnosis of coronary artery disease, Am. J. Cardiol. 64 (5) (1989) 304–310, https://doi.org/10.1016/0002-9149(89)90524-9, https://www.sciencedirect.com/science/article/pii/0002914989905249.

[8] D. Dua, C. Graff, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2017.

[9] R. Hou, S. Ai, Q. Chen, H. Yan, T. Huang, K. Chen, Similarity-based integrity protection for deep learning systems, Inf. Sci. 601 (2022) 255–267, https://doi.org/10.1016/j.ins.2022.04.003, https://www.sciencedirect.com/science/article/pii/S0020025522003279.

[10] W. Hoyos, J. Aguilar, M. Toro, Federated learning approaches for fuzzy cognitive maps to support clinical decision-making in dengue, Eng. Appl. Artif. Intell. 123 Part B (August 2023) 106371.

[11] R. Hu, Y. Guo, H. Li, Q. Pei, Y. Gong, Privacy-preserving personalized federated learning, in: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–6.

[12] G. Kaissis, M. Makowski, D. Rückert, R. Braren, Secure, privacy-preserving and federated machine learning in medical imaging, Nat. Mach. Intell. 2 (2020) 305–311.

[13] A.M. Kermarrec, F. Taïani, Want to scale in centralized systems? Think p2p, J. Internet Serv. Appl. 6 (1) (2015) 16.

[14] J. Konecný, B. McMahan, D. Ramage, P. Richtárik, Federated optimization: distributed machine learning for on-device intelligence, arXiv:1610.02527 [abs], 2016.

[15] H. Li, Chengcheng Li, J. Wang, A. Yang, Z. Ma, Zunqian Zhang, D. Hua, Review on security of federated learning and its application in healthcare, Future Gener. Comput. Syst. 144 (July 2023) 271–290, https://doi.org/10.1016/j.future.2023.02.021.

[16] M. Little, P. Mcsharry, S. Roberts, D. Costello, I. Moroz, Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection, Biomed. Eng. Online 6 (23) (02 2007), https://doi.org/10.1186/1475-925X-6-23.

[17] Y. Liu, Y. Kang, C. Xing, T. Chen, Q. Yang, A secure federated transfer learning framework, IEEE Intell. Syst. 35 (4) (2020) 70–82, https://doi.org/10.1109/MIS.2020.2988525.

[18] B. McMahan, E. Moore, D. Ramage, B.A. y Arcas, Federated learning of deep networks using model averaging, arXiv:1602.05629 [abs], 2016.

[19] B. McMahan, D. Ramage, Google ai blog, https://ai.googleblog.com/2017/04/federated-learning-collaborative.html, Apr 2017.

[20] D.C. Nguyen, Q.V. Pham, P.N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, W.J. Hwang, Federated learning for smart healthcare: a survey, ACM Comput. Surv. 55 (3) (February 2022) 1–37, https://doi.org/10.1145/3501296.

[21] S.A. Rahimi, F. Légaré, G. Sharma, P. Archambault, H.T.V. Zomahoun, S. Chandavong, N. Rheault, S.T. Wong, L. Langlois, Y. Couturier, J.L. Salmeron, M.P. Gagnon, J. Légaré, Application of artificial intelligence in community-based primary health care: systematic scoping review and critical appraisal, J. Med. Internet Res. 23 (9) (2021) 1–19.

[22] J.L. Salmeron, I. Arévalo, A privacy-preserving, distributed and cooperative fcm-based learning approach for cancer research, in: R. Bello, D. Miao, R. Falcon, M. Nakata, A. Rosete, D. Ciucci (Eds.), Rough Sets, Springer International Publishing, Cham, 2020, pp. 477–487.

[23] F. Sattler, S. Wiedemann, K.R. Müller, W. Samek, Robust and communication-efficient federated learning from non-iid data, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2020) 3400–3413, https://doi.org/10.1109/TNNLS.2019.2944481.

[24] W.N. Street, W.H. Wolberg, O.L. Mangasarian, Nuclear feature extraction for breast tumor diagnosis, in: R.S. Acharya, D.B. Goldgof (Eds.), Biomedical Image Processing and Biomedical Visualization, in: International Society for Optics and Photonics, vol. 1905, SPIE, 1993, pp. 861–870.

[25] W. Street, W. Wolberg, O. Mangasarian, Breast cancer diagnosis and prognosis via linear programming, Oper. Res. 43 (4) (Aug 1995) 570–577, https://doi.org/10.1287/opre.43.4.570.

[26] L. Su, V.K.N. Lau, Hierarchical federated learning for hybrid data partitioning across multitype sensors, IEEE Int. Things J. 8 (13) (July 2021) 10922–10939.

[27] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: user-level privacy leakage from federated learning, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019, pp. 2512–2520.

[28] T. Wink, Z. Nochta, An approach for peer-to-peer federated learning, in: 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2021, pp. 150–157.

[29] T. Wink, Z. Nochta, An approach for peer-to-peer federated learning, in: 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2021, pp. 150–157.

[30] J. Xu, B.S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, J. Healthc. Inform. Res. 5 (November 2021) 1–19, https://doi.org/10.1007/s41666-020-00082-4.

[31] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (March 2019) 1–19, https://doi.org/10.1145/3298981.

[32] Z. Zhao, C. Feng, W. Hong, J. Jiang, C. Jia, T.Q.S. Quek, M. Peng, Federated learning with non-iid data in wireless networks, IEEE Trans. Wirel. Commun. 21 (3) (March 2022) 1927–1942, https://doi.org/10.1109/TWC.2021.3108197.

Chapter 10

A Chaotic Maps-based Privacy-preserving Distributed Deep Learning for Incomplete and Non-IID Datasets

A chaotic maps-based privacy-preserving distributed deep learning for incomplete and Non-IID datasets

Irina Arévalo
Universidad Pablo de Olavide, Sevilla (Spain)
Email: iarebar@alu.upo.es

Jose L. Salmeron
CUNEF Universidad, Madrid (Spain)
Universidad Autónoma de Chile (Chile)
Email: joseluis.salmeron@cunef.edu

Abstract—Federated Learning is a machine learning approach that enables the training of a deep learning model among several participants with sensitive data that wish to share their own knowledge without compromising the privacy of their data. In this research, the authors employ a secured Federated Learning method with an additional layer of privacy and proposes a method for addressing the non-IID challenge. Moreover, differential privacy is compared with chaotic-based encryption as layer of privacy. The experimental approach assesses the performance of the federated deep learning model with differential privacy using both IID and non-IID data. In each experiment, the Federated Learning process improves the average performance metrics of the deep neural network, even in the case of non-IID data.

Index Terms—Federated Learning, Privacy-preserving machine learning, Non-IID datasets

1 Introduction

Privacy-preserving machine learning models are designed to protect the privacy of individuals whose data is used to train the model. This can be achieved through various techniques, such as using Federated Learning (FL) to train a model on decentralized datasets without sharing the raw data, or using secure multiparty computation to allow multiple parties to collaboratively train a model without revealing their individual data. Privacy-preserving models are particularly important in sensitive applications, such as healthcare or finance, where protecting the personal information of individuals is a top priority. Nevertheless, such additional privacy layer could lower the model performance both in accuracy and time.

In this research, the authors compare the results of an unsecured federated model and a secured model built using differential privacy and chaotic maps as its encrypting layer. In each experiment this research proves that the federation process improves the averaged performance metrics of a deep neural network for the participants, with disregard of whether the data has been evenly split among them or there are differences between the amount of data each participant has, and that the performance with or without the privacy layer are similar, meaning that the additional security does not worsen the model's results thanks to the federation process.

Moreover, to ensure the simulation closely resembles a real-world implementation of Federated Learning, the authors have taken into account a scenario where one of the participants or clinical centers possesses incomplete data with a distinct structure. This situation may arise, for example, when a variable is unavailable in the dataset of one participant. In such cases, this proposal enables other participants to privately share the distribution of the missing variable, allowing for the imputation of the missing data in the dataset of the participant who lacks that variable. The rational behind this proposal is that the federation process and its multiple iterations will average the model performance even in the case when one of the participant's features has been imputed.

The main contributions of this paper are two fold:

- This proposed FL extension aims to handle datasets that are incomplete or contain missing values, as well as datasets that are non-IID (non-Independently and Identically Distributed). In certain scenarios, the data used in the FL process may have missing values, lack completeness or uneven distribution across the participating devices or nodes.
- An efficient and secure method for encrypting distributed models based on chaotic maps. Chaotic maps possess inherent complexity and unpredicta-

bility, which makes them resistant to conventional cryptographic attacks. Furthermore, their non-linear nature enhances their security. Additionally, the deterministic characteristics of chaotic maps make them an efficient encryption method.

The proposed extension aims to address these issues and enable the use of FL on several categories of non-IID datasets. As best of our knowledge, this approach is a novelty. The specific details of the extension, as well as its performance and effectiveness, are described in the paper. The categories of non-IID data analysed are the following ones:

- Partial overlapping attribute skew
- Full overlapping attribute skew
- Label distribution skew
- Attribute and label skew
- Quantity skew.

The rest of this paper is organized as follows. We discuss the theoretical background and related work in section 2. The methodological proposal is described in section 4. Section 5 outlines the details of the experimental approach and the results. Finally, the authors draw a conclusion in section 6.

2 Related work

Recently, deep learning has achieved remarkable results in different domains, such as object classification [9] and self-driving cars [23]. Deep learning is a subset of machine learning algorithms that models high level abstraction using computational architectures that allow non linear transformations in the data in the form of a neural network [12].

A deep neural network for a supervised problem learns from processing many labeled examples through its layers [11], [17]. Layers are composed of a number of interconnected nodes which contain an activation function that polarizes network activity. This function includes nonlinear behaviour and helps it to become steady. A common activation funcion is ReLU (Rectified Linear Unit), which both the function and its derivative are monotonic. The function returns 0 if it receives any negative input, but for any positive value x, it returns that value back, and thus it gives an output that has a range from 0 to infinity.

The labeled examples to train the network are supplied via the input layer, which communicates to one or more hidden layers where the actual processing is done via a system of weighted connections. The hidden layers then link to the output layer. The training of the neural network is done via backpropagation, an algorithm that modifies the weights of the network by computing the gradient of the loss function with respect to those weights for a single input–output example.

2.1 Non-IID data

The data used to train a model on each client in Federated Learning often depends on the usage patterns of specific local devices, resulting in data distributions among connected clients that can vary significantly from one another. This phenomenon is known as Non-Independent and Identically Distributed (Non-IID) data [22]. Zhe et al. [32] propose several categories of non-IID data:

- Attribute skew. This category includes several subcategories: *Non-overlapping attribute skew*: It means that data attributes across the clients are mutually exclusive. *Partial overlapping attribute skew*: In this case, some portions of the data attributes can be shared with each other. *Full overlapping attribute skew*: Data attributes are the same in all participants but the attributes distributions can be different.
- Label skew. This category includes several subcategories: *Label distribution skew*: Label distributions on the clients are different. *Label preference skew*: The label distribution is different on the client data, although the distribution of the attributes is the same. *Temporal skew*: The focus is on addressing distribution skewness in temporal data, which encompasses spatio-temporal data as well as time series data.
- Attribute and label skew. Different clients hold data with different labels and different attributes.
- Quantity skew. The number of training data varies across different clients.

In general, non-IID datasets can be challenging to analyse because they often contain a high degree of variance and may not be representative of the overall population. This paper is focused on the performance analysis of differential privacy on FL in the IID data and several flavours of non-IID data. This is a challenging issue because the features of connected clients are different from each other [32].

2.2 Federated Learning

Federated Learning, proposed by McMahan et al. [20] and further developed in Konecny et al. [16] and McMahan and Ramage [21], is a distributed Machine Learning approach in which the participants collaborate to train a model with their private data by updating that model in their infrastructure and then sending the parameters to an aggregation node. The participants own the data and train the partial models. The aggregation node then federates the participant's models to obtain a global model trained with private data. This method can be iterated as many times as desired.

The use of FL with non-IID datasets has been studied in the literature [32]. For instance Zhao et al. [31] train convolutional neural networks (CNNs) on MNIST, CIFAR-10 and Speech commands datasets and find the reduction in the test accuracy of the federated averaging for non-IID data. Also, Wang et al. [26] optimise Federated Learning on Non-IID Data with Reinforcement Learning. Chen et al. [4] proposed an asynchronous online FL framework, where the edge devices perform online learning with continuous streaming local non-IID data and a central server aggregates model parameters from clients. None of the previously mentioned research focus the analysis of FL's performance with non-IID data and Differential Privacy or Chaotic Maps, as proposed in this paper.

2.2.1 Federated Learning architectures

There are two main Federated Learning architectures [29]:

1) Coordinated or centralised (client–server): It consists of a central server, that delivers the model architecture, performs the aggregation tasks, manages the communications, and delivers the model architectured , and a set of data silos or participants.
2) Swarm Learning (Peer-2-Peer): This architecture does not need any central server because all the nodes play the role simultaneously of central server and data silos. In this architecture, the Federated Learning process is triggered by one of the nodes.

The main advantage of Federated Learning is the training of a model in the private data of several participants that wish to maintain avoid data-sharing while improving their models [8]. This approach allows the use of heterogeneous data among the participants. It also allows the use of more accurate models with low latency, ensuring privacy and less power consumption. The process of a coordinated Federated Learning process is as follows:

1) The central server sends a model to each participant. If this is the initial iteration the federated model is proposed by the central server.
2) Each participant trains the received model using their own private data.
3) Each participant sends the parameters of the model in a private way (usually encrypting the data to be sent, see next subsection) to the central server.
4) The central server aggregates the partial models through their parameters and builds the federated model.
5) The central server checks a termination condition in either accuracy of the model in a test dataset or number of iterations. If it is accomplished, the FL process ends, otherwise we iterate from step 1.

The development of a Peer-2-Peer Federated Learning process is similar with one of the nodes taking the role of the central server. In any case, the target of the federated model is to minimize the total loss for all participants, computed as follows:

$$\mathcal{L}^* = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\mathcal{D}_i, \Phi) \tag{1}$$

where n is the number of participants, Φ is the federated model parameters, $\mathcal{D}_i$ is the dataset of the participant i, $\mathcal{L}^*$ is the loss function for the federated model, and $\mathcal{L}_i(\cdot)$ is the loss function for each participant in the federation.

2.2.2 Federated Learning categories

Regarding the nature of the data, Federated Learning can be categorised [10], [29] into three sets (Horizontal Federated Learning, Vertical Federated Learning and Federated Transfer Learning).

In Horizontal Federated Learning, the features space is overlapped across data silos, but the samples space is different in data locations. This approach is the original Federated Learning proposal but it still presents challenges. For instance, an innovative approach named hierarchical heterogeneous horizontal Federated Learning faces limited labeled entities in Horizontal Federated Learning [10]. In this research, the lack of labeled instances is mitigated by adapting the heterogeneous domain multiple times by using each participant as the target domain each time.

Vertical Federated Learning is needed when the features space has a partial or low overlap across data silos, but the samples space is nearly the same across those data locations. Unlike the case of horizontal Federated Learning, the aggregation of the entire data set in a single data silo to train a global model would not work in vertical Federated Learning. Some vertical proposals have been developed in [5], [18].

Moreover, the data does not share a sample space or a feature space in most cases. Federated Transfer Learning approach proposed by [19] generalise Federated Learning when it comes to common parties with small intersection. This proposal can be easily adapted to various secure, Machine Learning endeavours with minimal modification to the existing model and provides the same level of accuracy as non-privacy-preserving transfer learning.

2.2.3 Federated Learning challenges

There are several challenges that must be addressed in Federated Learning in order to effectively protect the privacy of enterprises and users [30]. These include:

1) *Ensuring privacy protection*: Federated Learning was designed to protect the privacy of data in machine learning, and it is important to ensure that the training model does not reveal users' private information or that the model itself is altered.
2) *Overcoming the lack of sufficient data*: In conventional machine learning, a large amount of data is typically needed to train a model with optimum performance. However, in a distributed environment, the amount of data on each device may be insufficient and collecting all the data in a centralized manner can be too expensive or legally prohibited. Federated Learning allows each device to use its own local data to train a local model, which is then aggregated with the models of other devices to create a global model.
3) *Dealing with statistical heterogeneity*: There are many edge devices in the federated environment, and the data held by these devices may not be evenly distributed or similar in structure (i.e., it may exhibit skew).

This paper addresses all of these challenges simultaneously. The authors test the use of differential privacy and chaotic maps for improving privacy protection, conduct experiments with participants that have very limited amounts of data, and examine the impact of different skew and overlap of attributes among participants.

3 PRIVACY-PRESERVING TECHNIQUES

In this paper the authors are testing two privacy-preserving techniques: differential privacy and chaotic maps-based encryption.

3.1 Differential Privacy

The Federated Learning process guarantees that the sharing of private data is not needed to train the federated model. However, there are still risks associated with the transmission of such information, like model poisoning, potential attacks to reconstruct the model or the training data from the parameters that the participants send to the central server, or the use of attack models [3], [27]. Therefore, there have been several advances in the use of privacy-preserving methods [1], [2], [6], [13], [15] in Federated Learning.

In this research the authors have applied differential privacy to ensure the security of the system. Differential privacy is a widely-used standard for privacy guarantee of algorithms operating on aggregated data. In general, a randomised algorithm $A(D)$ satisfies ε, δ-differential privacy if for all datasets D and D' that differ in a single record, and for all sets $S \in R$, where R is the range of A,

$$P(A(D) \in S) \leq \exp(\varepsilon) P(A(D') \in S) + \delta \tag{2}$$

where the probability P is taken over the coin tosses of A and ε and δ are non-negative numbers. This means that no single record in the dataset has a significant impact on the output of the algorithm.

The authors add a differential privacy layer to a deep network using the Differentially Private Stochastic Gradient Descent algorithm (see Algorithm 1) that modifies the optimization process in a deep network adding some noise [1]. This algorithm trains the model by obtaining the parameters θ via minimizing the empirical loss function $\mathcal{L}$.

Here we assume that the gradient of the loss function has a bounded L^2 norm, therefore we ask for the loss function to be a Sobolev function, $\mathcal{L} \in \mathcal{W}^{1,2}$, which is a weaker condition than being a Lipschitz function. Nevertheless, in order to ensure the convergence of the algorithm with non-IID data, we will additionally ask for the gradient of the loss, $\nabla\mathcal{L}$, to be a Lipschitz function. The inputs of the algorithm are the examples $\{x_i\}_{i=1}^N$, the loss function $\mathcal{L}(\theta)$, the learning rate η_t, the noise scale σ, the group size L, and the Sobolev norm of the loss function C.

This additional privacy layer is expected to lower the model performance, both in accuracy and in the training time, due to the extra computations and the necessity of finding the privacy cost ε, δ.

The use of differential privacy in FL has already been studied in works such as [28], where they compare the accuracy for an MLP trained on MNIST data for Different Privacy values ε, number of participants and iterations to experimentally evaluate their algorithm. The application of differential privacy in FL with non-IID dataset is not a novelty either. Zhao et al. [31] have previously applied differential privacy to non-IID datasets, including in cases where participants only received data from a single class. In contrast, this paper addresses the challenge of missing features, rather than just a single class of the target, in the context of incomplete and non-IID datasets.

3.2 Chaotic maps-based privacy-preserving

Chaotic maps are a branch of mathematics that investigates dynamic systems capable of generating highly randomized states. These states exhibit complete disorder and apparent irregularity, yet their evolution is determined by the initial conditions of the system. Chaotic maps exhibit sensitive dependence on initial conditions and generate complex, unpredictable behavior. This unpredictability can be harnessed for encryption purposes.

Algorithm 1: Differentially private SGD

Data: Sample points $\{x_i\}_{i=1}^N$, loss function $\mathcal{L}(\theta)$, learning rate η_t, noise scale σ, group size L, gradient norm bound C

Result: Model parameters θ_T, and privacy cost ε, δ

begin
  $\theta_0 \leftarrow X \sim \mathcal{N}(\mu, \sigma^2)$
end
for $t \in [T]$ **do**
  Random sample L_t with sampling probability $\frac{L}{N}$
  for $i \in L_t$ **do**
    /* Compute gradient */
    $g_t(x_i) = \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$
    /* Clip gradient */
    $\overline{g}_t(x_i) = g_t(x_i) / \max(1, \frac{1}{C} \cdot \|g_t(x_i)\|_2)$
    /* Noise addition */
    $\tilde{g}_t = \frac{1}{L} \cdot (\sum_i \overline{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$
    /* Descent */
    $\theta_{t+1} = \theta_t - \eta_t \tilde{g}_t$
  end
end

Chaotic maps algorithms for encryption are highly regarded for their ability to deliver a combination of high speed, reasonable computation, and strong security. It's worth noting that the specific implementation and design choices for encryption with chaotic maps can vary. Different chaotic maps can be used, such as the logistic map, Henon map, or Lorenz system, depending on the desired properties and security requirements. In this research, logistic map is the selected map for testing our proposal.

The logistic map, a recurrence relation of degree 2 or polynomial mapping, is widely recognized as an archetypal instance that demonstrates the emergence of complex and chaotic behavior from simple nonlinear dynamical equations. The logistic map is defined as

$$x_{i+1} = r \cdot x_i \cdot (1 - x_i) \tag{3}$$

where the parameter r fall within the interval $[0, 4]$ in order to ensure that x_n remains bounded on $[0, 1]$. When $r \in [3.57, 4]$ the logistic map is chaotic [14]. In this research, the value of r is assigned as 3.8 to ensure chaotic behaviour.

Encryption with chaotic maps is a method of encrypting data using chaotic dynamics (Algorithm 2). In the case of the encryption algorithm with the logistic map, XOR ($\oplus$) is applied between each element of the data set (whether it is plain data or cipher data) and the fractional part of the chaotic value generated by the logistic map at that moment. XOR the i-th element of the plain data with the fractional part of x_i to obtain the i-th element of the cipher data. It is important to note that XOR ($\oplus$) is a bitwise operation, which means that it is applied independently to each corresponding pair of bits in the data elements and chaotic values. This allows for a reversible operation

Algorithm 2: Logistic map-based encryption

Data: Original plain data ($\mathcal{D}$)
Result: Cipher data (Γ)
Key Generation: Generate the r parameter
Initialization: Choose an initial value x_0
Encryption:
for $i = 1$ **to** n **do**
  /* Calculate the chaotic value */
  ; $x_{i+1} = r \cdot x_i \cdot (1 - x_{i-1})$
  $\Gamma[i] = \mathcal{D}[i] \oplus \text{Frac}(X[i])$
end
Output: The cipher data obtained from the encryption process

Algorithm 3: Logistic map-based decryption

Data: Cipher data (Γ)
Result: Original plain data ($\mathcal{D}$)
Key Generation: Generate the r parameter
Initialization: Choose an initial value x_0
Decryption:
for $i = 1$ **to** n **do**
  /* Calculate the chaotic value */
  ; $x_{i+1} = r \cdot x_i \cdot (1 - x_{i-1})$
  $\text{Frac}[i] = \mathcal{D}[i] \oplus \Gamma(X[i])$
end
Output: The plain data obtained from the decryption process

(Algorithm 3), as performing XOR between the encrypted data and the same encryption key (or parameter) will yield the original data.

The distinct characteristics exhibited by chaotic systems, including determinism, ergodicity, and sensitivity to initial conditions, make them a compelling option for constructing cryptographic systems. These properties share similarities with the desirable properties of a robust cryptosystem, such as confusion and diffusion. One of the advantadges of Chaos-based encryption techniques is their computational efficiency [33]. The encryption process with chaotic maps is as follows:

1) Key Generation: Chaotic maps require a secret key to initialize the map. The key should be kept secret, as it determines the encryption/decryption process. In the case of the logistic map, the parameter r determines the chaotic behavior of the map.
2) Chaotic Map Iteration: The chaotic map takes a value and generates iteratively a new value based on its mathematical definition and the previous value. The inherent chaotic nature of the map guarantees that even a slight alteration in the initial value can yield a significatively different output.
3) Offuscation: The chaotic map's output could be combined with the original data through offuscation operations. The aim of this stage is to make the relationship between the original and the encrypted data as complex and nonlinear as possible.
4) Iterations and key updating: During the encryption process, it is common to employ multiple iterations of the chaotic map along with key updates. Following each iteration, the key may undergo changes to introduce additional randomness and strengthen the security of the encryption..
5) Output: The final output of the encryption process is the cipher data, which is the encrypted form of the original plain data. It should appear random and be statistically independent of the original data.
6) Decryption: The same chaotic map is applied iteratively to the cipher data using identical initial conditions, parameters and key as in the encryption phase to retrieve the original plain data.

Encryption with chaotic maps offers certain advantages, such as a high degree of randomness, sensitivity to initial conditions, and resistance to various attacks. However, it also poses challenges in terms of stability, security analysis, and the need for efficient chaotic map implementations. it's important to note that in the context of chaotic maps, the terms *encryption* and *decryption* might not be the most accurate. Chaotic maps are primarily used for generating pseudorandom sequences or for generating chaotic behaviour, rather than encryption and decryption.

4 METHODOLOGICAL PROPOSAL

The extension of Federated Learning proposed to combine the use of an additional encryption layer with partial and full overlapping attribute, differents skews and non-IID datasets is as follows:

1) As a first step, a central server will send an untrained deep learning model to the participants.
2) If one of them does not have a complete dataset, meaning that one of the features is missing (and therefore the features are non-IID), the server will also send, in a encrypted fashion, the distribution of the feature for any other participant so that the lacking feature can be imputed.
3) Then all the participants will split their data into train/test/validation datasets and train the model in their training data. Then, they will send the parameters of the model back to the central server, using one of the three possible different approaches to this step:
   a) the first one where the data is sent without any additional security measures,
   b) the second where the data is encrypted using either differential privacy to avoid privacy issues, in order to compare if the use of this privacy-preserving layer affects the results of the model,
   c) and the third, where the data is encrypted using a chaotic map and the process ends with the decryption of the offuscated data, as described in subsection 3.2.
4) The server then aggregates the parameters of the local models to find a global model, and iterates this process to improve the global accuracy. The most common aggregation method is Federated Averaging, which is just a weighted average of the network weights across training sites.

The federated model is evaluated in each participant's test data. The convergence of the differential privacy process is guaranteed by the work in [28], where the authors apply Differential Privacy to Federated Learning.

The convergence of the chaotic map encryption process is ensured by the encryption-decryption process as described in subsection 3.2.

5 Experimental approach

The scenario for the experiments is as follows. The authors are assuming there are several participant facilities that have private health data from their patients. This information is used to train a deep learning-based classification model to diagnose some specific disease with high security standards.

Nevertheless, the amount of available data is inadequate in some cases for training a robust model. One naive solution is to share their data with others participant facilities or an intermediary to train a model with all their shared data. But since the participants are dealing with extremely sensitive information, it is unlikely they would accept that solution, and it even raises the question of whether it is compliant with data regulations. Moreover, there exists the possibility that the distribution of data is not uniform among all the participants. In this simulation, the authors consider the case where one of the participants does not have one of the features the others have.

The selected initial model is a dense neural network, made of five dense layers, each of them followed by a ReLU function and a dropout layer of 0.3 for regularization, with a learning rate 0.01 and the loss function is binary cross-entropy.

For the experiments we will assume there are five different participants. In a first test all of them will have the same amount of data, obtained from a evenly split dataset, but we will also consider the case where each participant has a different dataset size.

In particular, the authors have performed three additional experiments with uneven splits: the first one will be a random split among the participants, but in the remaining two we have forced that there are several participants with a very small number of samples (less than 10% of the samples). The amount of positive cases will also vary from one participant to another. In any case, each participant's data will be split into a train and a test dataset. As it is customary, the models will be trained in each participant's train data, and the evaluation metrics will be obtained from each participant's test data. The final performance metrics will be averaged.

We also simulate the case where one of the participant does not have data on one of the variables. In that case, we use the encrypted sharing of data to send the distribution of that variable in one of the other participants in a private way, and then proceed to impute the mean of the value. The distribution of the missing feature is computed in another participant's data and is sent via the encryption used for sharing the model data to the participant without the feature. Then, the authors apply the $L2$ imputation using the features' distribution.

For each experiment the authors will perform two different FL approaches, as detailed in Section 4: one with a layer of differential privacy, and another one without it. The differences in results between the two types of experiments are compared to understand whether the encrypted version offers similar results. To compare the performance of different models and methods, the authors applied metrics (accuracy and F1 score) in a set of tests for each participant.

The results are presented in tables with the metrics for each participant, and the columns are as follows: the size or percentage of the original dataset that each participant has the percentage of positives in that participant's dataset, and the accuracy and the F1 score on a test set before and after the Federated Learning process, for the non-private, the differentially private, and the chaotic map-based privacy approaches. For each experiment there will be two tables, one for the case when there are no missing features, and one for the case when the fifth participant does have a missing column.

5.1 Experiment 1

The dataset for the first experiment is the Breast Cancer Wisconsin. It was created by the University of Wisconsin Hospital at Madison, and is publicly available [7]. More details can be found in [25], [24].

The results of the Federated Learning process for this dataset are shown in Tables 1 and 2. As previously mentioned, the authors have tested two different scenarios:

- All of the participants have data with a consistent structure.
- One participant is missing one of the features in their data.

The authors have also considered four different data partitions: the first one is an evenly split dataset for every Participant, were each of them has a 20% of the Brest Cancer dataset, while the remaining three tests include uneven sets, the first one a random partition and the remaining two with sharp differences where several participants have very small datasets, like 2% and 6% in the case of the third test and 5% and 5% in the case of the fourth test. There also are an uneven percentage of positives in these splits, including some participants with more than a 65% of positive samples and others with less than a 25%. With this setting, it is possible to test some hypothetical cases where a group of participants want to share secure information and a private model even in the case where one or more of the participants have much less information to share than the rest.

As the results show, the Federated Learning process improves the averaged performance metrics for all participants in every case, both the even and the uneven partitions, and the private and non-private approaches. In particular, when there is no missing data, the Federated Learning process improves the metrics for the experiments without an additional privacy layer from 0.9735 to 0.9826 in average accuracy for all participants. In average F1 the score goes from 0.9538 to 0.9667 for all participants for the evenly split case. In average accuracy from 0.9609 to 0.9672, and from 0.9259 to 0.9333 in average F1 in the first case of uneven splits. From 0.9097 to 0.9536 in average accuracy and from 0.8946 to 0.9397 in average F1 in the second case of uneven splits. From 0.9679 to 0.9778 in average accuracy and from

0.9533 to 0.9778 in average F1 for the last uneven split. This improvement is to be expected due to the convergence of the federation process and the use of iterative local models.

In the case of the experiments with an additional differential privacy layer added, the average accuracy increases from 0.9470 to 0.9648 while the average F1 goes from 0.9051 to 0.9444 with even splits. The average accuracy goes from 0.9424 to 0.9593 and the average F1 from 0.9180 to 0.9451 for the first uneven split. For the second uneven split the average accuracy increases from 0.9034 to 0.9401 and the average F1 from 0.8513 to 0.8897. In the last uneven split the average accuracy goes from 0.9230 to 0.9848 and the average F1 from 0.8287 to 0.9287.

Moreover, the results for the private approach, where differential privacy is used, are in general terms very similar to the non-private approach, with a small decrease in accuracy and F1 in the case of the balanced datasets, and both decreases and increases in the performance metrics in the imbalanced examples. This outcome is more surprising, since given the additional privacy layer used in the models one could expect worse accuracy metrics. Nevertheless, the federation process, iterating the averaging of all the local models, is able to maintain the performance metrics of the non-differentially private model.

The results for the experiments with missing data are also very similar to the previous outcomes, with a small decrease in the performance metrics with respect with non-missing data, but still achieving high performance metrics. Once again this result is surprising, not because of the lower performance metrics, which is understandable given the missing data, but for its limited nature, since one could anticipate a larger decrease. Nevertheless, the iterative federation process averages the metrics and reduces its diminution after several repetitions.

More explicitly, the average accuracy for the experiments without an additional privacy layer increase from 0.9391 to 0.9826 in the case of even splits. In the first uneven split from 0.9321 to 0.9397. From 0.9345 to 0.9470 in the second, and from 0.8961 to 0.9815 in the last one. The average F1 score goes from 0.9136 to 0.9825 in the even split, from 0.9016 to 0.9407 in the first uneven split, from 0.9286 to 0.9331 in the second, and from 0.7766 to 0.9698 in the third.

In the case of the additional differential privacy layer, the model improves the average accuracy from 0.9478 to 0.9652 in the evenly split datasets, from 0.9353 to 0.9536 in the first uneven split datasets. From 0.9157 to 0.9913 in the second split, and from 0.9294 to 0.9657 in the last one, and the increment of the average F1 score goes from 0.9339 to 0.9697 in the split with even data for every participant. From 0.9213 to 0.9365 in the first split with uneven data, from 0.9123 to 0.9818 in the second, and from 0.9095 to 0.9409 in the last one.

Finally, in the case of the chaotic map approach with no missing data, for the even split datasets the accuracy improves from 0.9561 to 0.9652 while the F1 score goes from 0.9331 to 0.9381, in the first uneven split the accuracy increases from 0.9424 to 0.9778 and the F1 from 0.9228 to 0.9750, in the second uneven split the accuracy goes from 0.9301 to 0.9679 and the F1 from 0.9190 to 0.9618, and in the third uneven split the metrics improve from 0.9571 to 0.9655 in the case of the accuracy and from 0.9244 to 0.9655 in the case of the F1 score.

When dealing with missing data in the chaotic map experiment, the accuracy increases from 0.9301 to 0.9478 when the data is split evenly, from 0.9614 to 0.9637 with the first uneven split, from 0.9413 to 0.9736 in the second uneven split, and from 0.9773 to 0.9909 in the third uneven split, whereas the F1 score goes from 0.9123 to 0.9228, from 0.9252 to 0.9267, from 0.9157 to 0.9554, and from 0.9706 to 0.9913 in the even split and first, second and third uneven split respectively.

5.2 Experiment 2

The Chronic Kidney Dataset is the dataset for this experiment. It is publicly available at the UC Irvine Machine Learning Repository [7].

Tables 3 and 4 show the results of the experiments made with this dataset and with the same partitions as the previous experiment.

According to the results, the federation process improves the averaged accuracy and the F1 score over all participants with no missing data for every test. Both the private and non-private approaches result in similar performance metrics, with a small decrease in the case of the balanced datasets, and both increases and decreases for the imbalanced datasets, as in the previous experiment: for the case with no additional privacy layer, for the even split the average accuracy improves from 0.9625 to 0.9750 and the average F1 score from 0.9690 to 0.9787, for the first uneven split the average accuracy goes from 0.9388 to 0.9738 and the average F1 score from 0.9478 to 0.9802, for the second uneven split the average accuracy increases from 0.9331 to 0.9857 and the average F1 score from 0.9493 to 0.9913, and for the last uneven split the average accuracy goes from 0.8506 to 0.9294 and the average F1 score from 0.8793 to 0.9428; and for the case with differential privacy, for the even split the average accuracy improves from 0.9375 to 0.9625 and the average F1 score from 0.9480 to 0.9659, for the first uneven split the average accuracy goes from 0.9473 to 0.9713 and the average F1 score from 0.9509 to 0.9777, for the second uneven split the average accuracy increases from 0.8446 to 0.9192 and the average F1 score from 0.8505 to 0.9400, and for the last uneven split the average accuracy goes from 0.9316 to 0.9770 and the average F1 score from 0.9468 to 0.9830.

In the case of missing data, we find a similar outcome, with the federation process improving the performance metrics in all cases: in the first case without a layer of differential privacy, for the even split the average accuracy increases from 0.9375 to 0.9875 and the average F1 score from 0.9491 to 0.9913, for the first uneven split the average accuracy goes from 0.9326 to 0.9538 and the average F1 score from 0.9425 to 0.9590, for the second uneven split the average accuracy improves from 0.9533 to 0.9867 and the average F1 score from 0.9578 to 0.9869, and for the third uneven split the average accuracy goes from 0.9179 to 0.9439 and the average F1 score from 0.8971 to 0.9294. In the second case with the layer of differential Privacy, for the even split the average accuracy goes from 0.9375 to 0.9625 and the average F1 score from 0.9473 to 0.9672, for the first uneven split the average accuracy improves from 0.9678 to 0.9895 and the average F1 score from 0.9744 to 0.9913, for the second uneven split

TABLE 1
Experiment 1 (no missing data)

Participant	Size	pos (%)	Accuracy pre-FL	Accuracy post-FL	F1 pre-FL	F1 post-FL	Accuracy pre Private FL	Accuracy post Private FL	F1 pre Private FL	F1 post Private FL	Accuracy pre encr. FL	Accuracy post encr. FL	F1 pre encr. FL	F1 post encr. FL
1	20%	47%	0.9546	1.0000	0.9231	1.0000	0.9091	0.9545	0.8750	0.9231	0.9545	1.0000	0.9333	1.0000
2	20%	41%	1.0000	1.0000	1.0000	1.0000	1.0000	0.9565	1.0000	0.9655	0.9565	1.0000	0.9655	1.0000
3	20%	31%	0.9565	0.9130	0.9231	0.8333	0.8696	0.9130	0.7273	0.8333	0.9130	0.9130	0.8333	0.8333
4	20%	35%	1.0000	1.0000	1.0000	1.0000	0.9565	1.0000	0.9231	1.0000	1.0000	1.0000	1.0000	1.0000
5	20%	32%	0.9565	1.0000	0.9231	1.0000	1.0000	1.0000	1.0000	1.0000	0.9565	0.9130	0.9333	0.8571
Avg	-	-	0.9735	**0.9826**	0.9538	**0.9667**	0.9470	**0.9648**	0.9051	**0.9444**	0.9561	**0.9652**	0.9331	**0.9381**
1	19%	28%	1.0000	1.0000	1.0000	1.0000	0.9000	0.9500	0.8000	0.8889	1.0000	1.0000	1.0000	1.0000
2	22%	47%	0.9310	0.9310	0.9167	0.9167	0.8621	0.8966	0.8333	0.8800	1.0000	1.0000	1.0000	1.0000
3	14%	54%	1.0000	1.0000	1.0000	1.0000	0.9500	0.9500	0.9565	0.9565	0.8333	0.8889	0.8235	0.8750
4	17%	14%	0.9048	0.9048	0.7500	0.7500	1.0000	1.0000	1.0000	1.0000	0.9500	1.0000	0.8571	1.0000
5	28%	41%	0.9688	1.0000	0.9630	1.0000	1.0000	1.0000	1.0000	1.0000	0.9286	1.0000	0.9333	1.0000
Avg	-	-	0.9609	**0.9672**	0.9259	**0.9333**	0.9424	**0.9593**	0.9180	**0.9451**	0.9424	**0.9778**	0.9228	**0.9750**
1	50%	40%	0.9608	0.9804	0.9474	0.9730	0.9836	0.9672	0.9787	0.9583	0.9649	0.9825	0.9524	0.9756
2	2%	67%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	11%	42%	0.8125	0.8125	0.7692	0.7692	1.0000	1.0000	1.0000	1.0000	0.8571	0.8571	0.8333	0.8333
4	6%	65%	0.8000	1.0000	0.8000	1.0000	0.6667	0.8333	0.5000	0.6667	0.8571	1.0000	0.8571	1.0000
5	32%	25%	0.9750	0.9750	0.9565	0.9565	0.8667	0.9000	0.7778	0.8235	0.9714	1.0000	0.9524	1.0000
Avg	-	-	0.9097	**0.9536**	0.8946	**0.9397**	0.9034	**0.9401**	0.8513	**0.8897**	0.9301	**0.9679**	0.9190	**0.9618**
1	52%	39%	0.9661	1.0000	0.9545	1.0000	0.9828	0.9828	0.9767	0.9767	0.9821	1.0000	0.9756	1.0000
2	5%	28%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	5%	38%	1.0000	1.0000	1.0000	1.0000	0.7500	1.0000	0.6667	1.0000	1.0000	1.0000	1.0000	1.0000
4	14%	22%	0.9474	1.0000	0.8889	1.0000	0.8824	0.9412	0.5000	0.6667	0.9412	1.0000	0.8000	1.0000
5	25%	44%	0.9259	0.8889	0.9231	0.8889	1.0000	1.0000	1.0000	1.0000	0.8621	0.8276	0.8462	0.8276
Avg	-	-	0.9679	**0.9778**	0.9533	**0.9778**	0.9230	**0.9848**	0.8287	**0.9287**	0.9571	**0.9655**	0.9244	**0.9655**

TABLE 2
Experiment 1 (with missing data)

Participant	Size	pos (%)	Accuracy pre-FL	Accuracy post-FL	F1 pre-FL	F1 post-FL	Accuracy pre Private FL	Accuracy post Private FL	F1 pre Private FL	F1 post Private FL	Accuracy pre encr. FL	Accuracy post encr. FL	F1 pre encr. FL	F1 post encr. FL
1	20%	47%	0.9130	0.9565	0.9167	0.9600	0.9130	1.0000	0.8750	1.0000	0.9565	0.9565	0.9474	0.9474
2	20%	41%	0.9130	0.9565	0.9000	0.9524	0.9565	0.9565	0.9474	0.9412	0.8696	0.8696	0.8235	0.8000
3	20%	31%	0.9565	1.0000	0.9333	1.0000	0.9565	1.0000	0.9474	1.0000	0.9130	0.9565	0.8571	0.9333
4	20%	35%	0.9565	1.0000	0.9091	1.0000	0.9130	0.8696	0.9000	0.8571	0.9565	0.9565	0.9333	0.9333
5	20%	32%	0.9565	1.0000	0.9091	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Avg	-	-	0.9391	**0.9826**	0.9136	**0.9825**	0.9478	**0.9652**	0.9339	**0.9697**	0.9391	**0.9478**	0.9123	**0.9228**
1	19%	28%	0.9583	1.0000	0.9474	1.0000	0.8846	0.9615	0.8696	0.9524	0.9565	1.0000	0.9565	1.0000
2	22%	47%	0.8889	0.8889	0.8571	0.8571	1.0000	0.9615	1.0000	0.9565	0.9032	0.8710	0.8696	0.8333
3	14%	54%	0.9048	0.8095	0.9231	0.8462	0.8421	0.8947	0.8800	0.9167	1.0000	1.0000	1.0000	1.0000
4	17%	14%	0.9500	1.0000	0.8571	1.0000	0.9500	0.9500	0.8571	0.8571	0.9474	0.9474	0.8000	0.8000
5	28%	41%	0.9583	1.0000	0.9231	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Avg	-	-	0.9321	**0.9397**	0.9016	**0.9407**	0.9333	**0.9536**	0.9213	**0.9365**	0.9614	**0.9637**	0.9252	**0.9267**
1	50%	40%	0.9565	0.9783	0.9545	0.9767	1.0000	1.0000	1.0000	1.0000	0.9783	0.9783	0.9787	0.9787
2	2%	67%	1.0000	1.0000	1.0000	1.0000	0.6667	1.0000	0.6667	1.0000	0.7500	1.0000	0.6667	1.0000
3	11%	42%	0.9375	1.0000	0.9412	1.0000	0.9333	1.0000	0.9474	1.0000	1.0000	0.9333	1.0000	0.9412
4	6%	65%	0.8000	0.8000	0.8000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	32%	25%	0.9783	0.9565	0.9474	0.8889	0.9783	0.9565	0.9474	0.9091	0.9783	0.9565	0.9333	0.8571
Avg	-	-	0.9345	**0.9470**	0.9286	**0.9331**	0.9157	**0.9913**	0.9123	**0.9818**	0.9413	**0.9736**	0.9157	**0.9554**
1	52%	39%	0.9545	0.9773	0.9600	0.9796	0.9524	0.9286	0.9474	0.9189	0.9318	0.9545	0.9362	0.9565
2	5%	28%	0.8333	1.0000	0.6667	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	5%	38%	0.8333	1.0000	0.8000	1.0000	0.8000	1.0000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000
4	14%	22%	0.8824	1.0000	0.5000	1.0000	0.8947	0.9474	0.8000	0.8571	1.0000	1.0000	1.0000	1.0000
5	25%	44%	0.9767	0.9302	0.9565	0.8696	1.0000	0.9524	1.0000	0.9286	0.9545	1.0000	0.9167	1.0000
Avg	-	-	0.8961	**0.9815**	0.7766	**0.9698**	0.9294	**0.9657**	0.9095	**0.9409**	0.9773	**0.9909**	0.9706	**0.9913**

the average accuracy increases from 0.9438 to 0.9875 and the average F1 score from 0.9505 to 0.9818, and for the third uneven split the average accuracy goes from 0.8443 to 0.9572 and the average F1 score from 0.8385 to 0.9560.

Adding an encryption layer using chaotic maps, the results are also positive after the federated learning process both in accuracy and in F1 score. Firstly, if there is no missing data, the accuracy in the even split case goes from 0.95 to 0.9625, and the F1 score from 0.9543 to 0.9682. In the first uneven split, the accuracy increases from 0.9255 to 0.9455, and the F1 score from 0.8961 to 0.9143, in the second uneven split the accuracy improves from 0.9752 to 0.9857 and the F1 score from 0.9799 to 0.9895, and in the last uneven split the accuracy goes from 0.9543 to 0.9907 and the F1 score from 0.9426 to 0.9926.

When there is missing data and the data is split evenly, the accuracy increases from 0.9625 to 0.9750 and the F1 score from 0.9645 to 0.9750. With the first uneven split the accuracy goes from 0.9689 to 0.9875 and the F1 score from 0.9246 to 0.9920, with the second uneven split the accuracy improves from 0.9204 to 0.9935 and the F1 score from 0.9148 to 0.9959, and in the third uneven split the accuracy increases from 0.9596 to 0.9939 and the F1 score from 0.9543 to 0.9943.

Summarizing these results, in every case the average accuracy and the average F1 score show a performance close to 0.9 in every case except for the first uneven split for the chaotic map encryption, and the third uneven split, where two participants have less than 6% of the data, that we see that the average accuracy is bigger than 0.85 before the federation process, and bigger than 0.9 after. Also, the Federation Learning process improves the performance metrics in every case, and that the final results are very similar in every experiment, with or without an additional layer of differential privacy or chaotic map encryption, and with or without the imputing of missing data.

5.3 Experiment 3

The Parkinson's dataset is the dataset for the third experiment. It is publicly available at the UC Irvine Machine Learning Repository [7].

The results for the experiments with the Parkinson's disease dataset are shown in Tables 5 and 6. As in the previous experiments, the averaged accuracy and F1 score

TABLE 3
Experiment 2 (no missing data)

Participant	Size	pos (%)	Accuracy pre-FL	Accuracy post-FL	F1 pre-FL	F1 post-FL	Accuracy pre Private FL	Accuracy post Private FL	F1 pre Private FL	F1 post Private FL	Accuracy pre encr. FL	Accuracy post encr. FL	F1 pre encr. FL	F1 post encr. FL
1	20%	73%	0.9375	0.9375	0.9474	0.9524	0.9375	0.9375	0.9474	0.9474	0.8750	0.8750	0.8889	0.9000
2	20%	58%	0.9375	1.0000	0.9412	1.0000	0.9375	0.9375	0.9412	0.9412	0.9375	1.0000	0.9412	1.0000
3	20%	56%	0.9375	1.0000	0.9565	1.0000	0.9375	1.0000	0.9565	1.0000	1.0000	1.0000	1.0000	1.0000
4	20%	65%	1.0000	0.9375	1.0000	0.9412	0.9375	0.9375	0.9474	0.9412	0.9375	0.9375	0.9412	0.9412
5	20%	60%	1.0000	1.0000	1.0000	1.0000	0.9375	1.0000	0.9474	1.0000	1.0000	1.0000	1.0000	1.0000
Avg	-	-	0.9625	**0.9750**	0.9690	**0.9787**	0.9375	**0.9625**	0.9480	**0.9659**	0.9500	**0.9625**	0.9543	**0.9682**
1	14%	43%	0.9167	1.0000	0.9091	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	27%	71%	0.9552	0.9524	0.9600	0.9600	0.8571	0.9524	0.8800	0.9600	0.9000	1.0000	0.9091	1.0000
3	21%	76%	0.8667	1.0000	0.9000	1.0000	1.0000	0.9474	1.0000	0.9630	1.0000	1.0000	1.0000	1.0000
4	14%	39%	1.0000	1.0000	1.0000	1.0000	0.9231	1.0000	0.9091	1.0000	0.7273	0.7273	0.5714	0.5714
5	24%	66%	0.9583	0.9167	0.9697	0.9412	0.9565	0.9565	0.9655	0.9655	1.0000	1.0000	1.0000	1.0000
Avg	-	-	0.9388	**0.9738**	0.9478	**0.9802**	0.9473	**0.9713**	0.9509	**0.9777**	0.9255	**0.9455**	0.8961	**0.9143**
1	49%	67%	0.9750	1.0000	0.9804	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	4%	60%	1.0000	1.0000	1.0000	1.0000	0.8000	1.0000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000
3	13%	68%	0.8571	0.9286	0.9091	0.9565	0.9231	0.8462	0.9524	0.9000	0.9286	0.9286	0.9474	0.9474
4	3%	63%	0.8333	1.0000	0.8571	1.0000	0.5000	0.7500	0.5000	0.8000	1.0000	1.0000	1.0000	1.0000
5	30%	54%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9474	1.0000	0.9524	1.0000
Avg	-	-	0.9331	**0.9857**	0.9493	**0.9913**	0.8446	**0.9192**	0.8505	**0.9400**	0.9752	**0.9857**	0.9799	**0.9895**
1	52%	66%	1.0000	1.0000	1.0000	1.0000	0.9535	0.9302	0.9643	0.9454	0.9535	0.9535	0.9630	0.9630
2	5%	33%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	3%	29%	0.6000	0.8000	0.7500	0.8571	0.7500	1.0000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000
4	15%	34%	0.7778	0.8889	0.7500	0.8889	1.0000	1.0000	1.0000	1.0000	0.8182	1.0000	0.7500	1.0000
5	25%	82%	0.8750	0.9583	0.8966	0.9677	0.9545	0.9545	0.9697	0.9697	1.0000	1.0000	1.0000	1.0000
Avg	-	-	0.8506	**0.9294**	0.8793	**0.9428**	0.9316	**0.9770**	0.9468	**0.9830**	0.9543	**0.9907**	0.9426	**0.9926**

TABLE 4
Experiment 2 (with missing data)

Participant	Size	pos (%)	Accuracy pre-FL	Accuracy post-FL	F1 pre-FL	F1 post-FL	Accuracy pre Private FL	Accuracy post Private FL	F1 pre Private FL	F1 post Private FL	Accuracy pre encr. FL	Accuracy post encr. FL	F1 pre encr. FL	F1 post encr. FL
1	20%	73%	0.8750	1.0000	0.8889	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	20%	58%	0.8750	0.9375	0.9091	0.9565	1.0000	0.9375	1.0000	0.9474	0.8750	0.8750	0.8750	0.8750
3	20%	56%	1.0000	1.0000	1.0000	1.0000	0.8750	0.9375	0.9000	0.9412	1.0000	1.0000	1.0000	1.0000
4	20%	65%	0.9375	1.0000	0.9474	1.0000	0.9375	0.9375	0.9474	0.9474	1.0000	1.0000	1.0000	1.0000
5	20%	60%	1.0000	1.0000	1.0000	1.0000	0.8750	1.0000	0.9333	1.0000	0.9375	1.0000	0.9474	1.0000
Avg	-	-	0.9375	**0.9875**	0.9491	**0.9913**	0.9375	**0.9625**	0.9473	**0.9672**	0.9625	**0.9750**	0.9645	**0.9750**
1	14%	43%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9444	1.0000	0.9565	1.0000
2	27%	71%	0.9130	0.9565	0.9286	0.9677	0.8947	0.9474	0.9091	0.9565	1.0000	1.0000	1.0000	1.0000
3	21%	76%	0.8750	0.9375	0.9091	0.9524	0.9444	1.0000	0.9630	1.0000	1.0000	0.9375	1.0000	0.9600
4	14%	39%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9000	1.0000	0.6667	1.0000
5	24%	66%	0.8750	0.8750	0.8750	0.8750	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Avg	-	-	0.9326	**0.9538**	0.9425	**0.9590**	0.9678	**0.9895**	0.9744	**0.9913**	0.9689	**0.9875**	0.9246	**0.9920**
1	49%	67%	0.86667	0.9667	0.9091	0.9778	1.0000	1.0000	1.0000	1.0000	0.9355	0.9677	0.9600	0.9796
2	4%	60%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.7500	1.0000	0.6667	1.0000
3	13%	68%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9167	1.0000	0.9474	1.0000
4	3%	63%	1.0000	1.0000	1.0000	1.0000	0.7500	1.0000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000
5	30%	54%	0.9000	0.9667	0.8800	0.9565	0.9688	0.9375	0.9524	0.9091	1.0000	1.0000	1.0000	1.0000
Avg	-	-	0.9533	**0.9867**	0.9578	**0.9869**	0.9438	**0.9875**	0.9505	**0.9818**	0.9204	**0.9935**	0.9148	**0.9959**
1	52%	66%	0.9394	0.9697	0.9615	0.9804	0.9355	0.9677	0.9583	0.9796	1.0000	1.0000	1.0000	1.0000
2	5%	33%	0.7500	0.7500	0.6667	0.6667	0.7500	1.0000	0.6667	1.0000	1.0000	1.0000	1.0000	1.0000
3	3%	29%	1.0000	1.0000	1.0000	1.0000	0.7500	1.0000	0.8000	1.0000	1.0000	1.0000	1.0000	1.0000
4	15%	34%	0.9000	1.0000	0.8571	1.0000	0.8182	0.8182	0.8000	0.8000	0.8889	1.0000	0.8571	1.0000
5	25%	82%	1.0000	1.0000	1.0000	1.0000	0.9677	1.0000	0.9677	1.0000	0.9091	0.9697	0.9143	0.9714
Avg	-	-	0.9179	**0.9439**	0.8971	**0.9294**	0.8443	**0.9572**	0.8385	**0.9560**	0.9596	**0.9939**	0.9543	**0.9943**

of the models are improved after the Federated Learning procedure in every case, and both the traditional, the differential privacy, and the chaotic map approaches, and the experiments with or without missing data, reach similar performance metrics.

As a summary of the results: when dealing without missing data and the experiments without the additional layer of differential privacy, the average accuracy goes from 0.6893 to 0.7929 in the even split, from 0.8288 to 0.8429 in the first uneven split, from 0.6065 to 0.8104 in the second uneven split, and from 0.7714 to 0.8254 in the last uneven split, and the average F1 score increases from 0.7738 to 0.8598 for the even split, from 0.8894 to 0.9049 for the first uneven split, from 0.6495 to 0.8678 for the second uneven split, and from 0.8345 to 0.8646 for the third uneven split. With the differential privacy layer, the average accuracy improves from 0.6643 to 0.7964, from 0.7429 to 0.8596, from 0.7389 to 0.9500, and from 0.8181 to 0.9219, and the average F1 score increases from 0.7506 to 0.8444, from 0.7961 to 0.8958, from 0.8149 to 0.9636, and from 0.8701 to 0.9492 for the even split, the first, the second and the third uneven split respectively. Including the chaotic map encryption, the accuracy improves from 0.6964 to 0.7464 and the F1 score from 0.7957 to 0.8078 in the case of the even split dataset, with the first uneven split the accuracy goes from 0.8267 to 0.9022 and the F1 score from 0.8814 to 0.9330, for the second uneven split the accuracy increases from 0.7793 to 0.8079 and the F1 score from 0.7160 to 0.7338, and for the third case, the accuracy goes from 0.8556 to 0.8667 and the F1 score from 0.9 to 0.9091.

When imputing missing data, when there is no additional privacy layer the average accuracy increases from 0.7250 to 0.7750, from 0.7405 to 0.7690, from 0.8122 to 0.8344, and from 0.8271 to 0.8857, and the average F1 score goes from 0.8220 to 0.8513, from 0.8133 to 0.8352, from 0.8631 to 0.8898, and from 0.8616 to 0.9095 for the even split, the first, the second and the third uneven split respectively. With the additional differential privacy layer the performance metrics improve from 0.7500 to 0.8000 for the average accuracy and from 0.8248 to 0.8609 for the average F1 score for the even split, from 0.7419 to 0.9262 for the average accuracy and from 0.8305 to 0.9559 for the average F1 score for the first uneven split, from 0.8159 to 0.9270 for the average accuracy and from 0.8500 to 0.9548 for the average F1 score for the

second, and from 0.6886 to 0.8076 for the average accuracy and from 0.6933 to 0.8574 for the average F1 score for the third. Finally, when adding the chaotic map encryption, the accuracy improves from 0.75, 0.8133, 0.7044, and 0.6933 to 0.8750, 0.8483, 0.8467, and 0.72 in the even split, and first, second and third uneven split respectively, and the F1 score goes from 0.8238, 0.8425, 0.7814, and 0.6698 to 0.9198, 0.9006, 0.8987, and 0.685 in the same cases.

Again we can see that in every case the Federated Learning process improves the performance metrics, not only in the traditional case but also in the experiments when we improve the security of the system by adding the additional layer of differential privacy or the chaotic map encryption, and when we impute missing data for one of the participants.

6 Conclusions

The different experiments show that, even in the most imbalanced cases, the Federated Learning process improves the average metrics of the models, increasing their performance, both in accuracy and F1 score, and for both the private, non-private and encrypted approaches. We can also conclude that using an additional layer of encryption and ensuring the privacy of the process does not affect the performance metrics of the model, when compared with the non-private Federated Learning.

This research proofs that the averaged accuracy and F1 score improves not only in the case where every participants has the same amount of data, but also in cases where there are sharp differences between the volume of data that the participants have. This way, this manuscript includes the hypothetical case where several participants want to train an accurate deep learning model and share it among them, even although some of those participants have much less available data.

In every experiment, an additional layer of differential privacy and chaotic map encryption was added to ensure the privacy and encryption of the data and compared with the Federated Learning approach without this layer, finding that the performance results of both models are extremely similar.

Moreover, in order to simulate real cases, the authors also test the same experiments in the event that one of the participants has a missing feature. As in the previous experiments, the performance of the models is improved by the federation process in all cases, for the accuracy and F1 metric and the private, encrypted and non-private cases.

This approach could be adopted to improve the models used to diagnose diseases, such as breast cancer, chronic kidney disease, Parkinson's, or potentially anyone else, as this paper have shown in the experiments.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: 23rd ACM Conference on Computer and Communications Security (ACM CCS). pp. 308–318 (2016), https://arxiv.org/abs/1607.00133

[2] Acar, A., Aksu, H., Uluagac, S., Conti, M.: A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys **51** (04 2017). https://doi.org/10.1145/3214303

[3] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: Chiappa, S., Calandra, R. (eds.) Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 108, pp. 2938–2948. PMLR (26–28 Aug 2020)

[4] Chen, Y., Ning, Y., Slawski, M., Rangwala, H.: Asynchronous online federated learning for edge devices with non-iid data. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 15–24. IEEE Computer Society, Los Alamitos, CA, USA (dec 2020). https://doi.org/10.1109/BigData50022.2020.9378161, https://doi.ieeecomputersociety.org/10.1109/BigData50022.2020.9378161

[5] Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D., Yang, Q.: Secureboost: A lossless federated learning framework (2019). https://doi.org/10.48550/ARXIV.1901.08755, https://arxiv.org/abs/1901.08755

[6] Cheng, Y., Liu, Y., Chen, T., Yang, Q.: Federated learning for privacy-preserving ai. Commun. ACM **63**(12), 33–36 (Nov 2020). https://doi.org/10.1145/3387107, https://doi.org/10.1145/3387107

[7] Dua, D., Graff, C.: Uci machine learning repository (2017), http://archive.ics.uci.edu/ml

[8] Duan, J., Zhou, J., Li, Y., Huang, C.: Privacy-preserving and verifiable deep learning inference based on secret sharing. Neurocomputing **483**, 221–234 (04 2022). https://doi.org/10.1016/j.neucom.2022.01.061

[9] Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature **542**(7639), 115–118 (2017). https://doi.org/10.1038/nature21056, https://doi.org/10.1038/nature21056

[10] Gao, D., Ju, C., Wei, X., Liu, Y., Chen, T., Yang, Q.: Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography (2019). https://doi.org/10.48550/ARXIV.1909.05784, https://arxiv.org/abs/1909.05784

[11] Goodfellow, I.J., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA, USA (2016), http://www.deeplearningbook.org

[12] Guerrero-Gomez-Olmedo, R., Salmeron, J.L., Kuchkovsky, C.: Lrp-based path relevances for global explanation of deep architectures. Neurocomputing **381**, 252–260 (2020). https://doi.org/10.1016/j.neucom.2019.11.059

[13] Hu, R., Guo, Y., Li, H., Pei, Q., Gong, Y.: Privacy-preserving personalized federated learning. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC). pp. 1–6 (2020). https://doi.org/10.1109/ICC40277.2020.9149207

[14] Ibitoye, O., Abou-Khamis, R., el Shehaby, M., Matrawy, A., Shafiq, M.O.: The threat of adversarial attacks on machine learning in network security – a survey (2023). https://doi.org/10.48550/ARXIV.1901.08755, https://arxiv.org/abs/1901.08755

[15] Kaissis, G., Makowski, M., Rückert, D., Braren, R.: Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence **2**, 305–311 (2020)

[16] Konecný, J., McMahan, B., Ramage, D., Richtárik, P.: Federated optimization: Distributed machine learning for on-device intelligence. ArXiv **abs/1610.02527** (2016)

[17] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–44 (05 2015). https://doi.org/10.1038/nature14539

[18] Lee, S., Lacy, M.E., Jankowich, M., Correa, A., Wu, W.C.: Association between obesity phenotypes of insulin resistance and risk of type 2 diabetes in african americans: The jackson heart study. Journal of Clinical & Translational Endocrinology **19**(3), 100210 (2020)

[19] Liu, Y., Kang, Y., Xing, C., Chen, T., Yang, Q.: A secure federated transfer learning framework. IEEE Intelligent Systems **35**(4), 70–82 (jul 2020). https://doi.org/10.1109/mis.2020.2988525, https://doi.org/10.1109%2Fmis.2020.2988525

[20] McMahan, B., Moore, E., Ramage, D., Agüera, B.: Federated learning of deep networks using model averaging. ArXiv **abs/1602.05629** (2016)

[21] McMahan, B., Ramage, D.: Google ai blog (Apr 2017), https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[22] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from

TABLE 5
Experiment 3 (no missing data)

Participant	Size	pos (%)	Accuracy pre-FL	Accuracy post-FL	F1 pre-FL	F1 post-FL	Accuracy pre Private FL	Accuracy post Private FL	F1 pre Private FL	F1 post Private FL	Accuracy pre encr. FL	Accuracy post encr. FL	F1 pre encr. FL	F1 post encr. FL
1	20%	65%	0.5714	0.7143	0.6667	0.8000	0.5714	0.8571	0.6667	0.9091	0.8571	0.8571	0.9091	0.8889
2	20%	78%	1.0000	1.0000	1.0000	1.0000	0.8750	0.8750	0.9231	0.9231	0.7500	0.8750	0.8333	0.9231
3	20%	75%	0.5000	0.8750	0.6000	0.9091	0.6250	0.6250	0.6667	0.6667	0.7500	0.6250	0.8000	0.6667
4	20%	78%	0.7500	0.500	0.8333	0.6667	0.6250	0.7500	0.7692	0.8000	0.6250	0.6250	0.7692	0.7273
5	20%	79%	0.6250	0.8750	0.7692	0.9231	0.6250	0.8750	0.7273	0.9231	0.5000	0.7500	0.6667	0.83333
Avg	-	-	0.6893	**0.7929**	0.7738	**0.8598**	0.6643	**0.7964**	0.7506	**0.8444**	0.6964	**0.7464**	0.7957	**0.8078**
1	14%	69%	0.6667	0.6667	0.8000	0.8000	0.2500	0.7500	0.4000	0.8000	0.6667	0.8333	0.7500	0.8571
2	24%	75%	0.7500	0.7500	0.8235	0.8421	0.8889	0.8889	0.9333	0.9333	1.0000	1.0000	1.0000	1.0000
3	19%	86%	1.0000	0.8889	1.0000	0.9412	0.9091	0.9091	0.9474	0.9474	0.6667	0.7778	0.8000	0.8750
4	10%	50%	1.0000	1.0000	1.0000	1.0000	0.8333	0.8333	0.8000	0.8571	1.0000	1.0000	1.0000	1.0000
5	32%	78%	0.7273	0.9091	0.8235	0.9412	0.8333	0.9167	0.9000	0.9412	0.8000	0.9000	0.8571	0.9333
Avg	-	-	0.8288	**0.8429**	0.8894	**0.9049**	0.7429	**0.8596**	0.7961	**0.8958**	0.8267	**0.9022**	0.8814	**0.9330**
1	21%	75%	0.8182	0.9091	0.8750	0.9412	0.8889	1.0000	0.9231	1.0000	1.0000	1.0000	1.0000	1.0000
2	41%	79%	0.6429	0.7857	0.7059	0.8421	0.7500	0.7500	0.8182	0.8182	0.8824	0.8824	0.9231	0.9231
3	16%	83%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8000	0.8000	0.8571	0.8571
4	8%	67%	0.0000	0.5000	0.0000	0.6667	0.5000	1.0000	0.6667	1.0000	0.5000	0.5000	0.0000	0.0000
5	16%	67%	0.5714	0.8571	0.6667	0.8889	0.5556	1.0000	0.6667	1.0000	0.7143	0.8571	0.8000	0.8889
Avg	-	-	0.6065	**0.8104**	0.6495	**0.8678**	0.7389	**0.9500**	0.8149	**0.9636**	0.7793	**0.8079**	0.7160	**0.7338**
1	55%	81%	0.8571	0.9048	0.9032	0.9333	0.8571	0.8095	0.9032	0.8571	1.0000	0.9444	1.0000	0.9655
2	10%	75%	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5000	0.7500	0.6667	0.8571
3	8%	67%	0.3333	0.6667	0.5000	0.8000	0.3333	1.0000	0.5000	1.0000	1.0000	1.0000	1.0000	1.0000
4	9%	50%	1.0000	0.6667	1.0000	0.6667	1.0000	1.0000	1.0000	1.0000	1.0000	0.7500	1.0000	0.8000
5	18%	76%	0.6667	0.8889	0.7692	0.9231	0.9000	0.8000	0.9474	0.8889	0.7778	0.8889	0.8333	0.9231
Avg	-	-	0.7714	**0.8254**	0.8345	**0.8646**	0.8181	**0.9219**	0.8701	**0.9492**	0.8556	**0.8667**	0.9000	**0.9091**

TABLE 6
Experiment 3 (with missing data)

Participant	Size	pos (%)	Accuracy pre-FL	Accuracy post-FL	F1 pre-FL	F1 post-FL	Accuracy pre Private FL	Accuracy post Private FL	F1 pre Private FL	F1 post Private FL	Accuracy pre encr. FL	Accuracy post encr. FL	F1 pre encr. FL	F1 post encr. FL
1	20%	65%	0.6250	0.7500	0.7692	0.8333	0.7500	0.7500	0.8333	0.8000	0.7500	0.8750	0.8333	0.9091
2	20%	78%	0.7500	0.7500	0.8571	0.8333	0.7500	0.6250	0.8571	0.7692	0.6250	0.7500	0.7692	0.8333
3	20%	75%	0.8750	0.7500	0.9231	0.8333	0.7500	0.8750	0.8333	0.9231	0.7500	0.8750	0.8333	0.9231
4	20%	78%	0.6250	0.7500	0.7273	0.8333	0.5000	0.8750	0.6000	0.8889	0.8750	0.8750	0.9333	0.9333
5	20%	79%	0.7500	0.8750	0.8333	0.9231	1.0000	0.8750	1.0000	0.9231	0.7500	1.0000	0.7500	1.0000
Avg	-	-	0.7250	**0.7750**	0.8220	**0.8513**	0.7500	**0.8000**	0.8248	**0.8609**	0.7500	**0.8750**	0.8238	**0.9198**
1	14%	69%	0.7143	0.8571	0.8000	0.9091	0.4286	0.8571	0.6000	0.9231	0.7500	0.7500	0.8571	0.8571
2	24%	75%	0.6667	0.6667	0.8000	0.8000	0.6667	0.9167	0.7778	0.9474	0.9167	0.9167	0.9412	0.9333
3	19%	86%	1.0000	1.0000	1.0000	1.0000	0.9000	1.0000	0.9412	1.0000	0.9000	0.7000	0.9474	0.8235
4	10%	50%	0.7500	0.7500	0.8000	0.8000	1.0000	1.0000	1.0000	1.0000	0.7500	1.0000	0.6667	1.0000
5	32%	78%	0.5714	0.5714	0.6667	0.6667	0.7143	0.8571	0.8333	0.9091	0.7500	0.8750	0.8000	0.8889
Avg	-	-	0.7405	**0.7690**	0.8133	**0.8352**	0.7419	**0.9262**	0.8305	**0.9559**	0.8133	**0.8483**	0.8425	**0.9006**
1	21%	75%	0.7778	0.7778	0.8750	0.8750	1.0000	1.0000	1.0000	1.0000	0.7778	0.7778	0.8333	0.8571
2	41%	79%	0.8667	0.8667	0.9167	0.9167	0.8571	0.8571	0.9167	0.9167	0.8000	0.7333	0.8571	0.8182
3	16%	83%	0.7500	0.7500	0.8571	0.8571	1.0000	1.0000	10000.	1.0000	0.6667	0.8333	0.7500	0.9091
4	8%	67%	1.0000	1.0000	1.0000	1.0000	0.6667	1.0000	0.6667	1.0000	0.5000	1.0000	0.6667	1.0000
5	16%	67%	0.6667	0.7778	0.6667	0.8000	0.5556	0.7778	0.6667	0.8571	0.7778	0.8889	0.8000	0.9091
Avg	-	-	0.8122	**0.8344**	0.8631	**0.8898**	0.8159	**0.9270**	0.8500	**0.9548**	0.7044	**0.8467**	0.7814	**0.8987**
1	55%	81%	0.9286	1.0000	0.9524	1.0000	1.0000	0.9286	1.0000	0.9630	1.0000	1.0000	1.0000	1.0000
2	10%	75%	1.0000	0.7500	1.0000	0.8000	1.0000	1.0000	1.0000	1.0000	0.4000	0.8000	0.5714	0.8889
3	8%	67%	0.6000	1.0000	0.6667	1.0000	0.8000	0.8000	0.8000	0.8571	1.0000	0.6667	1.0000	0.6667
4	9%	50%	0.7500	0.7500	0.8000	0.8000	0.0000	0.6667	0.0000	0.8000	0.3333	0.3333	0.0000	0.0000
5	18%	76%	0.8571	0.9286	0.8889	0.9474	0.6429	0.6429	0.6667	0.6667	0.7333	0.8000	0.7778	0.8696
Avg	-	-	0.8271	**0.8857**	0.8616	**0.9095**	0.6886	**0.8076**	0.6933	**0.8574**	0.6933	**0.7200**	0.6698	**0.6850**

decentralized data. In: International Conference on Artificial Intelligence and Statistics (2016)

[23] Ramos, S., Gehrig, S., Pinggera, P., Franke, U., Rother, C.: Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling. In: 2017 IEEE Intelligent Vehicles Symposium (IV). pp. 1025–1032 (2017). https://doi.org/10.1145/1553374.1553486

[24] Street, W., Wolberg, W., Mangasarian, O.: Breast cancer diagnosis and prognosis via linear programming. Oper. Res. **43**(4), 570–577 (Aug 1995). https://doi.org/10.1287/opre.43.4.570, https://doi.org/10.1287/opre.43.4.570

[25] Street, W., Wolberg, W., Mangasarian, O.: Nuclear feature extraction for breast tumor diagnosis. In: Electronic imaging. vol. 1993 (01 1999). https://doi.org/10.1117/12.148698

[26] Wang, H., Kaplan, Z., Niu, D., Li, B.: Optimizing federated learning on non-iid data with reinforcement learning. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications pp. 1698–1707 (2020)

[27] Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., Qi, H.: Beyond inferring class representatives: User-level privacy leakage from federated learning. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. pp. 2512–2520 (2019). https://doi.org/10.1109/INFOCOM.2019.8737416

[28] Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q.S., Vincent Poor, H.: Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security **15**, 3454–3469 (2020). https://doi.org/10.1109/TIFS.2020.2988575

[29] Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H.: Federated Learning. Morgan & Claypool (2019), vol. 13, (3) , 2019, pp. 1–207

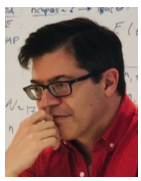
[30] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., Gao, Y.: A survey on federated learning. Knowledge-Based Systems **216**, 106775 (2021). https://doi.org/10.1016/j.knosys.2021.106775

[31] Zhao, Y., Suda, N., Li, M., Civin, D., Lai, L., Chandra, V.: Federated learning with non-iid data: A metric learning approach. arXiv **1806.00582** (2018)

[32] Zhu, H., Xu, J., Liu, S., Jin, Y.: Federated learning on non-iid data: A survey. Neurocomputing **465**, 371–390 (2021)

[33] Zia, U., McCartney, M., Scotney, B., Martinez, J., AbuTair, M., Memon, J., Sajjad, A.: Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains. International Journal of Information Security **21**(4), 917–935 (2022). https://doi.org/10.1007/s10207-022-00588-5, https://doi.org/10.1007/s10207-022-00588-5

Irina Arévalo is a Data Scientist and researcher in Artificial Intelligence. She holds a PhD in Mathematics, and is currently a PhD candidate at the University Pablo de Olavide (Seville, Spain). Irina Arévalo also has several years of experience as a Data Scientist in several fields of expertise, including Finance, Consultancy, and Communications. Her current research interests include Distributed Artificial Intelligence, Explainable Artificial Intelligence, and Bias & Fairness.



Jose L. Salmeron is a Professor of Artificial Intelligence at CUNEF University and an AI Senior Research Associate at the University Autónoma of Chile (Chile). His research papers have been published in prestigious journals such as IEEE Transactions on Cybernetics, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Software Engineering, Expert Systems with Applications, Communications of the ACM, Journal of Systems and Software, Computer Standards & Interfaces, Applied Soft Computing, Engineering Applications of Artificial Intelligence, Neurocomputing, and Information Sciences, among others. Additionally, he is recognized as an ACM lifetime senior member. Currently, his research interests encompass privacy-preserving computing, Distributed Artificial Intelligence, explainable artificial intelligence, and quantum machine learning.

Chapter 11

A Privacy-Preserving, Distributed and Cooperative FCM-Based Learning Approach for Cancer Research

A Privacy-Preserving, Distributed and Cooperative FCM-Based Learning Approach for Cancer Research

Jose L. Salmeron[1,2( )] and Irina Arévalo[1]

[1] Universidad Pablo de Olavide, Km. 1 Utrera Road, 43013 Seville, Spain
salmeron@acm.org, iarebar@alu.upo.es
[2] Tessella, Altran World-Class Center for Analytics, c/ Campezo 1, 28022 Madrid, Spain

Abstract. Distributed Artificial Intelligence is attracting interest day by day. In this paper, the authors introduce an innovative methodology for distributed learning of Particle Swarm Optimization-based Fuzzy Cognitive Maps in a privacy-preserving way. The authors design a training scheme for collaborative FCM learning that offers data privacy compliant with the current regulation. This method is applied to a cancer detection problem, proving that the performance of the model is improved by the Federated Learning process, and obtaining similar results to the ones that can be found in the literature.

Keywords: Fuzzy Cognitive Maps · Federated Learning · Distributed Artificial Intelligence · Cancer diagnosis

1 Introduction

Distributed Artificial Intelligence is a subfield of Artificial Intelligence that studies the coordination among several semi-autonomous agents called participants. Such systems are able to solve more complex problems involving a large amount of data, but there are privacy concerns about sharing sensitive information.

Federated Learning is a novel approach to Distributed Artificial Intelligence that enables privacy-preserving communications by sharing the model (or gradients) instead of the data. A central server sends a model to be trained by the participants with their local data, who send the parameters of the model back to the server to be aggregated. After iterating this process, the output is a model that has been trained with the private information of all participants.

This method is especially useful when dealing with sensitive data, from domains such as finance or healthcare. In this paper, the authors propose a Federated Fuzzy Cognitive Map approach to help diagnose malignant breast tumor cells.

R. Bello et al. (Eds.): IJCRS 2020, LNAI 12179, pp. 477–487, 2020.
https://doi.org/10.1007/978-3-030-52705-1_35

The contributions of this paper can be summarized as follows:

- Distributed learning. The authors propose a PSO-based FCM learning in a distributed way.
- Privacy-preserving machine learning. The authors design a training scheme for collaborative FCM learning that offers data privacy. This proposal enables multiple participants to learn a FCM model on their own inputs, preserving the privacy of their own data and complying with data privacy regulations.
- Implementation. The authors evaluate the performance of the proposal with a well-known dataset of cancer diagnosis. The experimental results show that the proposal achieve a similar performance to other non-distributed methods and improves the performance of the non-collaborative approach.

The rest of this paper is organized as follows. We discuss existing fundamentals of FCM and the learning approach in Sect. 2. Distributed Artificial Intelligence is described in Sect. 3. Then, we present the methodological proposal in Sect. 3. Section 4 describes the details of the experimental approach and the results. Finally, we draw a conclusion in Sect. 5.

2 Fuzzy Cognitive Maps

2.1 Fundamentals

Fuzzy Cognitive Maps (FCMs) were initially proposed by Kosko [3]. FCMs represent concepts, variables or features as nodes, the relationships between them as arcs, and the strengths of those relations as weights. It means that a weight assesses how much node X causes node Y. The fuzzy weights for arcs are normalised on the range $\{[0, +1]|[-1, +1]\}$, depending if it includes negative values or not. The maximum negative influence is -1 and the maximum positive influence is $+1$. The value zero shows that there is no relationship between the concepts. For computational purposes, FCMs can be described via a weight matrix (connection or adjacency matrix) which contains all weight values of edges between the concepts.

The relationships between the nodes are expressed by their weights. That is, if there is a positive causality between two nodes, then $\varpi_{ij} > 0$. If there is a negative causality, then $\varpi_{ij} < 0$ and if there is no relationship between the two nodes, then $\varpi_{ij} = 0$. The state of the nodes together is shown in the state vector $c = [c_1, c_2, \ldots, c_N]$ that gives a snapshot of nodes at any point of the instant in the scenario.

From a formal point of view, it is possible to represent a FCM as a 4-tuple $\Phi = \langle c, \mathcal{W}, f, r \rangle$, where $c = \{c_i\}_{i=1}^{n}$ is the state of the nodes with n as the number of nodes, $\mathcal{W} = [\varpi_{ij}]_{n \times n}$ is the adjacency matrix representing the weights between the nodes, f is the activation function, and r is the nodes' range.

FCMs are dynamical systems involving feedback, where the effect of change in the state of a node may affect the state of other nodes, which in turn can affect the former node [7].

The dynamic starts with an initial vector state $c(0) = \big(c_1(0), \ldots, c_n(0)\big)$, which represents the initial state (value) of each node. The new state of the nodes is computed as an iterative process. It includes an activation function [1] for mapping monotonically the node state into a normalized range $\{[0, +1]||[-1, +1]\}$. If the range is $[0, +1]$, the unipolar sigmoid is the most used activation function, but hyperbolic tangent is the most used when the range is $[-1, +1]$.

The component i of the vector state at time t, $c_i(t)$, can be computed as

$$c_i(t) = f\left(\sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)\right). \tag{1}$$

Some systems include nodes whose states should be steady because their states are not related with the dynamics of the system but their state has some influence on the state of the other nodes (i.e. sun radiation, wind speed and so on). In such cases, the state of the node is the same along the dynamics $c_i(t) = c(t-1) \mid c_i \in \mathcal{O}$, where $\mathcal{O}$ is the set of output concepts.

If the activation function f is unipolar sigmoid, then the component i of the vector state $c_i(t)$ at the instant t is computed as follows

$$c_i(t) = \left(1 + e^{-\lambda \cdot \sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)}\right)^{-1} \tag{2}$$

If the activation function f is hyperbolic tangent, then the component i of the vector state $c_i(t)$ at the instant t is computed as follows

$$c_i(t) = \frac{e^{\lambda \cdot \sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)} - e^{-\lambda \cdot \sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)}}{e^{\lambda \cdot \sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)} + e^{-\lambda \cdot \sum_{j=1}^{n} \varpi_{ji} \cdot c_j(t-1)}} \tag{3}$$

After the dynamics, the FCM reaches one of the three following states after a number of iterations: it settles down to either a fixed pattern of node values (the so-called hidden pattern), a limited cycle, or a fixed-point attractor.

2.2 Augmented FCMs

According to the FCM literature [4], an augmented adjacency matrix is built by aggregating the adjacency matrix of each FCM. The elements' aggregation depends on whether there are common nodes. If the adjacency matrices had no common nodes, the elements ϖ_{ij} in the augmented matrix $(\otimes_{i=1}^{N})$ are computed by adding the adjacency matrix of each FCM model $(\mathcal{W}_i)$.

The addition method when the adjacency matrices have not common nodes is known as direct sum of matrices, and the augmented matrix is denoted as $\otimes_{i=1}^{N} \varpi_i$. Given a couple of FCMs with no common nodes and even different number of nodes with adjacency matrices $[\varpi_{ij}]_{n \times n}$ and $[\varpi_{kl}]_{m \times m}$, the resulting augmented adjacency matrix is as follows

$$\begin{aligned} \otimes_{i=1}^{N} \varpi_i &= \mathbf{diag}(\varpi_{jk}, \varpi_{lo}) \\ &= \begin{pmatrix} 0 & [\varpi_{jk}]_{r \times r} \\ [\varpi_{lo}]_{m \times m} & 0 \end{pmatrix} \end{aligned} \tag{4}$$

where N is the number or adjacency matrices to add, zeroes are actually zero matrices and the dimension of $\otimes_{i=1}^{N} \varpi_i$ is $[\cdot]_{m+r \times m+r}$. In the case of common nodes, they would be computed as the average or weighted average of the states of the nodes in each adjacency matrix.

2.3 FCM for Classification

FCMs classification capabilities have been analysed by [8]. In general terms, the main goal of a conventional classifier is the mapping of an input to a specific output according to a pattern. In this proposal, the input concepts represent the features of the dataset, while the output concepts are the classes' labels where the patterns belong.

Figure 1 shows the typical topology of a FCM classifier where the state of the concepts c_1 and c_2 defines the class where the input vector state belongs. In that sense, if $c_1 > c_2$ the input vector state belongs to class 1 but if $c_1 < c_2$ the input vector state belongs to class 2. Note that $c_i \in \{[-1, +1], [0, +1]\}$, therefore if $c_1 = 0.03$ and $c_2 = 0.1$, then the input vector state belong to class 2.

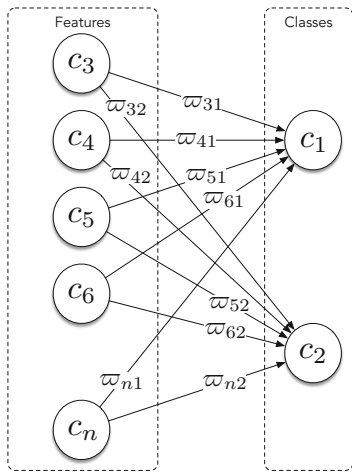


Fig. 1. Fuzzy Cognitive Maps classifier

2.4 PSO-Based FCM Learning

FCM learning endeavours are commonly focused on building the adjacency matrix based either on the available historical raw data or on expert knowledge. FCM learning approaches could be divided into three categories [10]: Hebbian,

population-based, and hybrid, mixing the main aspects of Hebbian-based and population-based learning algorithm.

The goal of the Hebbian-based FCM learning approaches is to modify adjacency matrices leading the FCM model to either achieve a steady state or converge into an acceptable region for the target system.

Population-based approaches do not need the human intervention. They compute adjacency matrices from historical raw data that best fit the sequence of input state vectors (the instances of the dataset). The learning goal of FCM evolutionary learning is to generate optimal adjacency matrix for modeling systems behaviour.

In this sense, Salmeron et al. [11] proposed an advanced decision support tool based on consultations with a group of experienced medical professionals using FCMs trained with Particle Swarm Optimization (PSO). Also, Salmeron and Froelich [9] apply PSO for time series forecasting.

PSO is a bio-inspired, population-based and stochastic optimization algorithm. The PSO algorithm generates a swarm of particles moving in an n-dimensional search space which must include all potential candidate solutions.

In order to train the FCM adjacency matrices we take into account the k^{th} particle's position (a candidate solution or adjacency matrix), denoted as $\varpi_k = (\varpi_{k_1}, \ldots, \varpi_{k_j})$ and its velocity, $v_k = (v_{k_1}, \ldots, v_{k_j})$. Note that each particle is a potential solution or FCM candidate and its position ϖ_k represents its adjacency matrix.

Each particle's velocity and position are updated at each time step. The position and the velocity of each particle is computed as follows

$$\varpi_k(t+1) = \varpi_k(t) + v_k(t) \tag{5a}$$

$$v_k(t+1) = v_k(t) + U(0,\phi_1) \otimes (\dot{\varpi}_k - \varpi_k(t)) + U(0,\phi_2) \otimes (\ddot{\varpi}_k - \varpi_k(t)) \tag{5b}$$

where $U(0,\phi_i)$ is a vector of random numbers generated from a uniform distribution within $[0,\phi_i]$, generated at each iteration and for each particle. Also, $\dot{\varpi}_k$ is the best position of particle k in all former iterations and $\ddot{\varpi}_k$ the best position of the whole population in all previous iterations and $\otimes$ is the component-wise multiplication.

The PSO algorithm's goal is to locate all the particles in the global optima to a multidimensional hyper-volume. The fitness function used in this research is the complement of the Jaccard similarity coefficient ($\overline{J} = (Y \times \hat{Y}) \setminus J$). The Jaccard score computes the average of Jaccard similarity coefficients between pairs of the sets of labels. The Jaccard similarity coefficient of the i-th samples, with a ground truth label set and a predicted label set. The complement operation is needed in terms of minimization of the fitness function. The Jaccard similarity coefficient's complement is computed as follows

$$\overline{J}(y_i, \hat{y}_i) = 1 - \frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|} \tag{6}$$

The fitness function is sampled after each particle position update and is the objective function used to compute how close a given particle is in order to be able to achieve the set aims.

3 Methodological Proposal

3.1 Fundamentals

Distributed Artificial Intelligence is a subset of Artificial Intelligence that allows the sharing of information among several agents or participants that interact by cooperation, by coexistence or by competition. Such system manages the distribution of tasks, being therefore more apt to solve complex problems, especially if they involve a large amount of data.

One of the methods available to construct a distributed artificial intelligence system is Federated Learning, proposed by McMahan et al. [5] and further developed in Konecny et al. [2] and McMahan and Ramage [6]. In such system, a central server constructs a model, usually a neural network, and sends it to the participants, who train the model in their private data. Their data never leaves their local devices, therefore ensuring privacy and security. The parameters of the participant's model are then averaged to obtain a global model. This process may be iterated till convergence.

Described in a formal way, a Federated Learning project is composed by a central server and the participants. The central server is responsible for managing the federated model and the communications with the participants. The participants own the datasets and train the partial models. The whole process is described in Fig. 2 and it is as follows:

1. The central server sends a federated model to each participant. If it is the initial iteration the federated model is proposed by the central server.
2. Each participant trains the received model with their own private dataset.
3. After the partial model is trained, each participant sends the parameters of the model or its gradients to the central server, encrypted to ensure privacy.
4. The central server aggregates the partial model and builds the federated model.
5. The central server checks the termination condition and if it is accomplished the federated model is finished, otherwise the process goes back to step 1.

When the researchers at Google first defined Federated Learning, their initial idea was to allow Android mobile phones to collaborative construct a prediction model without migrating the training data from the phone (see McMahan et al. [6] from the Google AI Blog). A first application they had was to use FL in Gboard on Android, the Google Keyboard, which predicts the most probable next phrase or word based on the user-generated preceding text. Recently, Federated Learning has improved this process, allowing the use of more accurate models with lower latency, ensuring privacy and less power consumption.

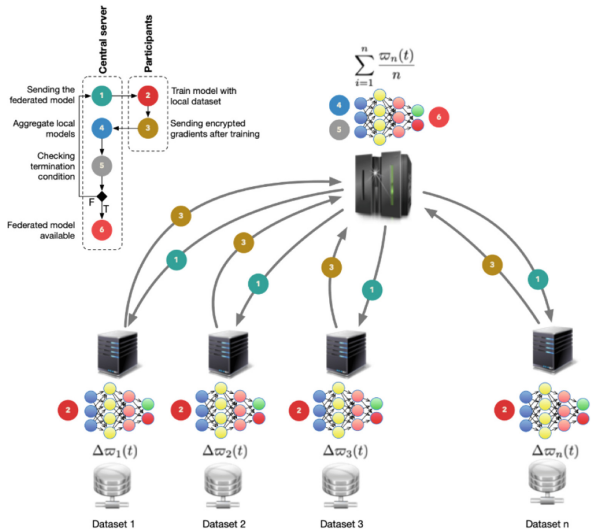


Fig. 2. Federated Learning process

One of the main advantages of Federated Learning is the promise of secure and private distributed machine learning, but there are risks associated with sharing data among several agents, such as the reconstruction of training examples from the neural network parameters, the uploading of private data from the agents to the central server, and the protection of the models as intellectual property of the companies. There is a large research interest in privacy-preserving methods applied to Federated Learning, such as the application of Differential Privacy, Secure Multi-Party Computation or Homomorphic encryption.

3.2 FCM Distributed Learning

The proposed methodology combines Federated Learning with learning FCMs using Particle Swarm Optimization. The process is shown in Fig. 3 and it is explained as follows.

1. Triggering the Federated Learning process. The central server triggers the process in the participants machines.
2. Training FCM in the local dataset. Each participant trains a local FCM with their own dataset. The authors apply PSO but this methodology is agnostic to the learning approach. The FCM dynamics is considered steady when the difference between two consecutive vector states is under $tol = 0.00001$
3. Sending the trained adjacency matrices and local accuracy for this stage to the central server. The local FCM is stored in the participant devices.

4. Weighting local FCMs using accuracy. The central server aggregates the local FCMs weighting by the accuracy. The aggregation method have been detailed as Sect. 2.2.
5. Aggregating Federated and Local FCMs. The participants aggregate the Federated FCM from the central server and their own local FCM.
6. Sending adjacency matrices and accuracy. Participants send again the local adjacency matrices and the new local accuracy.
7. Checking termination condition. The central server checks if the Federated process has been run 20 iterations as termination condition. If it is not accomplished then it goes back to the step 4.
8. If the termination condition is accomplished then a Federated FCM is achieved.

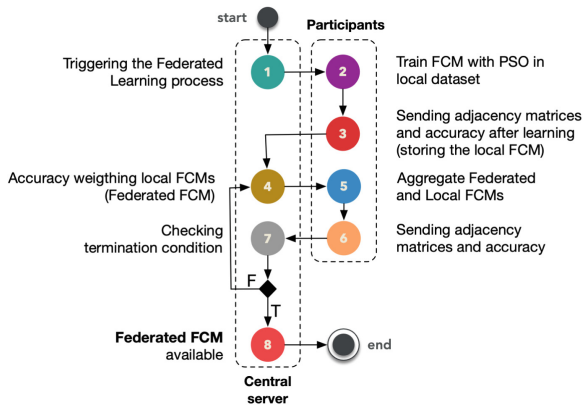


Fig. 3. Proposed methodology

The main contribution of this paper is the application of Federation Learning paradigm for privacy-preserving FCM distributed and coorperative learning.

4 Experimental Approach

4.1 Dataset

Breast cancer is one of the most common cancers among women, accounting for 25% of all cancer cases that affect women worldwide. According to the American Cancer Society, when breast cancer is detected early, and is in the localized stage, the 5-year relative survival rate is 99%, which makes the early diagnosis of breast cancer a main key in the prognosis and chance of survival of such types of cancer.

In recent years the use of Machine Learning algorithms in medicine has increased exponentially, with applications such as EEG analysis and Cancer

detection. For example, automatized algorithms have been use to examine biological data such as DNA methylation and RNA sequencing to infer which genes can cause cancer and which genes can instead be able to suppress its expression.

In this paper the authors will use the Breast Cancer Wisconsin Dataset, created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, and made publicly available at the UC Irvine Machine Learning Repository. The dataset comprises data from digitized images of the fine-needle aspirate of a breast mass that describes features of the nucleus of the current image of 569 patients, of which 212 are malignant and 357 are benign cases.

The first two features correspond to the identifier number and the diagnosis status (our target). The remaining attributes are thirty real attributes that measure the mean, the standard error, and the worst radius, texture, perimeter, area, smoothness, compactness, concave points, concavity, symmetry, and fractal dimension of the nucleus of the solid breast mass (see Table 1). These data were

Table 1. Dataset details

Id	Description	Mean	Std
Diagnosis	Target	0.3726	0.4839
radius_mean	Mean of distances from center to points on the perimeter	14.1273	3.524
texture_mean	Standard deviation of gray-scale values	19.2896	4.301
perimeter_mean		91.969	24.299
area_mean		654.8891	351.9141
smoothness_mean	Local variation in radius lengths	0.0964	0.0141
compactness_mean	Perimeter^2 / area - 1	0.1043	0.0528
concavity_mean	Severity of concave portions of the contour	0.0888	0.0797
concave points_mean	Number of concave portions of the contour	0.0489	0.0388
symmetry_mean		0.1812	0.0274
fractal_dimension_mean	Coastline approx. - 1	0.0628	0.0071
radius_se	Mean of distances from center to points on the perimeter	0.4052	0.2773
texture_se	Standard deviation of gray-scale values	1.2169	0.5516
perimeter_se		2.8661	2.0219
area_se		40.3371	45.491
smoothness_se	Local variation in radius lengths	0.007	0.003
compactness_se	Perimeter^2 / area - 1	0.0255	0.0179
concavity_se	Severity of concave portions of the contour	0.0319	0.0302
concave points_se	Number of concave portions of the contour	0.0118	0.0062
symmetry_se		0.0205	0.0083
fractal_dimension_se	Coastline approx. - 1	0.0038	0.0026
radius_worst	Mean of distances from center to points on the perimeter	16.2692	4.8332
texture_worst	Standard deviation of gray-scale values	25.6772	6.1463
perimeter_worst		107.2612	33.6025
area_worst		880.5831	569.357
smoothness_worst	Local variation in radius lengths	0.1324	0.0228
compactness_worst	perimeter^2 / area - 1	0.2543	0.1573
concavity_worst	Severity of concave portions of the contour	0.2722	0.2086
concave points_worst	Number of concave portions of the contour	0.1146	0.0657
symmetry_worst		0.2901	0.0619
fractal_dimension_worst	Coastline approx. - 1	0.0839	0.0181

obtained using a graphical computer program called Xcyt, which is capable of perform the analysis of cytological features based on a digital scan. More details can be found in [12,13].

4.2 Results

After 20 iterations of the Federated Learning process, the Fuzzy Cognitive Map-based classifier is able to predict whether the tumor is malignant with an average accuracy of 0.9383 across all participants, improving the accuracy of a single Fuzzy Cognitive Map trained in the whole data, and the accuracy in each participant before the federation.

The goal of this paper is not the accuracy of the proposal but a distributed and privacy-preserving approach. Nevertheless, our results are similar to the ones found in literature [14] (Table 2).

Table 2. Results of the experiments

Participant	Accuracy pre-federated learning	Accuracy post-federated learning
1	0.7727	0.9091
2	0.9130	0.9130
3	0.8696	0.8696
4	0.9565	1.0000
5	1.0000	1.0000

5 Conclusions

This paper proposes an innovative methodology for learning Fuzzy Cognitive Maps with Federated Learning. It is a step forward for Distributed Artificial Intelligence and accomplishes the privacy-preserving requirements of the society.

In addition, the authors have developed a method for distributed Fuzzy Cognitive Maps that improves the accuracy of both the algorithm trained in the whole dataset in a local node and the participant's algorithms before the Federated Learning process.

This method was applied to a cancer detection problem, obtaining an accuracy of 0.9383. The participants in this process do not share their private data, therefore forming a privacy-preserving distributed system.

References

1. Bueno, S., Salmeron, J.L.: Benchmarking main activation functions in fuzzy cognitive maps. Expert Syst. Appl. **36**(3 Part 1), 258–268 (2009)
2. Konecný, J., McMahan, B., Ramage, D., Richtárik, P.: Federated optimization: distributed machine learning for on-device intelligence. ArXiv abs/1610.02527 (2016)
3. Kosko, B.: Fuzzy cognitive maps. Int. J. Man Mach. Stud. **24**(1), 65–75 (1986)
4. Lopez, C., Salmeron, J.L.: Modeling maintenance projects risk effects on erp performance. Comput. Stand. Interfaces **36**(3), 545–553 (2014)
5. McMahan, B., Moore, E., Ramage, D., y Arcas, B.A.: Federated learning of deep networks using model averaging. ArXiv abs/1602.05629 (2016)
6. McMahan, B., Ramage, D.: Google ai blog, April 2017. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
7. Nápoles, G., Jastrzebska, A., Mosquera, C., Vanhoof, K., Homenda, W.: Deterministic learning of hybrid fuzzy cognitive maps and network reduction approaches. Neural Networks **124**, 258–268 (2020)
8. Papakostas, G., Koulouriotis, D.: Classifying patterns using fuzzy cognitive maps. In: Glykas, M. (ed.) Fuzzy Cognitive Maps. Studies in Fuzziness and Soft Computing, vol. 247, pp. 291–306. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-03220-2_12
9. Salmeron, J.L., Froelich, W.: Dynamic optimization of fuzzy cognitive maps for time series forecasting. Knowl.-Based Syst. **105**, 29–37 (2016)
10. Salmeron, J.L., Mansouri, T., Moghadam, M.R.S., Mardani, A.: Learning fuzzy cognitive maps with modified asexual reproduction optimisation algorithm. Knowl.-Based Syst. **163**, 723–735 (2019)
11. Salmeron, J.L., Rahimi, S.A., Navali, A.M., Sadeghpour, A.: Medical diagnosis of rheumatoid arthritis using data driven pso-fcm with scarce datasets. Neurocomputing **232**, 65–75 (2017)
12. Street, W., Wolberg, W., Mangasarian, O.: Breast cancer diagnosis and prognosis via linear programming. Oper. Res. **43**(4), 570–577 (1995). https://doi.org/10.1287/opre.43.4.570
13. Street, W., Wolberg, W., Mangasarian, O.: Nuclear feature extraction for breast tumor diagnosis, vol. 1993, January 1999. https://doi.org/10.1117/12.148698
14. Wang, S., Wang, Y., Wang, D., Yin, Y., Wang, Y., Jin, Y.: An improved random forest-based rule extraction method for breast cancer diagnosis. Appl. Soft Comput. **86**, 105941 (2020)

Bibliography

[1] European Parliament and Council of the European Union. (4th May 2016). 'Regulation (EU) 2016/679 of the European Parliament and of the Council,' of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), [Online]. Available: https://data.europa.eu/eli/reg/2016/679/oj (visited on 13/04/2023) (cit. on pp. 3, 11).

[2] R. Creemers, 'Cybersecurity law and regulation in China: Securing the smart state,' *China Law and Society Review*, vol. 6, no. 2, pp. 111 –145, 2023. DOI: 10.1163/25427466-06020001. [Online]. Available: https://brill.com/view/journals/clsr/6/2/article-p111_001.xml (cit. on pp. 3, 11).

[3] B. Wong YongQuan, 'Data privacy law in Singapore: the Personal Data Protection Act 2012,' *International Data Privacy Law*, vol. 7, no. 4, pp. 287–302, Sep. 2017, ISSN: 2044-3994. DOI: 10.1093/idpl/ipx016. eprint: https://academic.oup.com/idpl/article-pdf/7/4/287/22923057/ipx016.pdf. [Online]. Available: https://doi.org/10.1093/idpl/ipx016 (cit. on pp. 3, 11).

[4] P. Bukaty, *The California Consumer Privacy Act (CCPA): An implementation guide*. IT Governance Publishing, 2019, ISBN: 9781787781320. [Online]. Available: http://www.jstor.org/stable/j.ctvjghvnn (visited on 05/01/2024) (cit. on pp. 3, 11).

[5] L. A. Bygrave, 'Core Principles of Data Privacy Law,' in *Data Privacy Law: An International Perspective*, Oxford University Press, Jan. 2014, ISBN: 9780199675555. DOI: 10.1093/acprof:oso/9780199675555.003.0005. eprint: https://academic.oup.com/book/0/chapter/196492041/chapter-pdf/40280573/acprof-9780199675555-chapter-5.pdf. [Online]. Available: https://doi.org/10.1093/acprof:oso/9780199675555.003.0005 (cit. on pp. 3, 11).

[6] B. McMahan, E. Moore, D. Ramage and B. Agüera, 'Federated learning of deep networks using model averaging,' *ArXiv*, vol. abs/1602.05629, 2016 (cit. on pp. 3, 11, 35).

[7] B. McMahan and D. Ramage, *Google AI blog*, 2017. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html (cit. on pp. 4, 12, 35).

[8] B. Liang, J. Cai and H. Yang, 'A new cell group clustering algorithm based on validation and correction mechanism,' *Expert Systems with Applications*, vol. 193, p. 116 410, 2022, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.116410. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421016985 (cit. on pp. 4, 12).

[9] Z. Sun, Y. Xu, Y. Liu, W. He, L. Kong, F. Wu, Y. Jiang and L. Cui, *A survey on federated recommendation systems*, 2023. arXiv: 2301.00767 [cs.IR] (cit. on pp. 4, 12).

[10] T. Long and Q.-S. Jia, 'Matching uncertain renewable supply with electric vehicle charging demand—a bi-level event-based optimization method,' *Complex System Modeling and Simulation*, vol. 1, no. 1, pp. 33–44, 2021. DOI: 10.23919/CSMS.2021.0001 (cit. on pp. 4, 12).

[11] T. Liu, Z. Wang, H. He, W. Shi, L. Lin, W. Shi, R. An and C. Li, *Efficient and secure federated learning for financial applications*, 2023. arXiv: 2303.08355 [cs.LG] (cit. on pp. 4, 12).

[12] H. Zhou, G. Yang, H. Dai and G. Liu, 'Pflf: Privacy-preserving federated learning framework for edge computing,' *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1905–1918, 2022. DOI: 10.1109/TIFS.2022.3174394 (cit. on pp. 4, 12).

[13] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li and H. Vincent Poor, 'Federated learning for internet of things: A comprehensive survey,' *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, 1622–1658, 2021, ISSN: 2373-745X. DOI: 10.1109/comst.2021.3075439. [Online]. Available: http://dx.doi.org/10.1109/COMST.2021.3075439 (cit. on pp. 4, 12).

[14] L. T. Phong, Y. Aono, T. Hayashi, L. Wang and S. Moriai, 'Privacy-preserving deep learning via additively homomorphic encryption,' *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, 2017 (cit. on pp. 4, 12, 36).

[15] C. Berry and N. Komninos, 'Efficient optimisation framework for convolutional neural networks with secure multiparty computation,' *Computers & Security*, vol. 117, p. 102 679, 2022 (cit. on pp. 4, 12, 36).

[16] S. Halder and T. Newe, 'Enabling secure time-series data sharing via homomorphic encryption in cloud-assisted iiot,' *Future Generation Computer Systems*, vol. 133, pp. 351–363, 2022 (cit. on pp. 4, 12, 36).

[17] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai and W. Zhang, 'A survey on federated learning: Challenges and applications,' *International Journal of Machine Learning and Cybernetics*, vol. 14, pp. 1–23, Nov. 2022. DOI: 10.1007/s13042-022-01647-y (cit. on pp. 4, 12).

[18] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek and H. Vincent Poor, 'Federated learning with differential privacy: Algorithms and performance analysis,' *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020. DOI: 10.1109/TIFS.2020.2988575 (cit. on pp. 8, 17, 45).

[19] A. F. Karr, X. Lin, A. P. Sanil and J. P. Reiter, 'Privacy-preserving analysis of vertically partitioned data using secure matrix products,' *Journal of Official Statistics*, vol. 25, pp. 125–138, 2009 (cit. on pp. 9, 18, 36).

[20] A. Gascon, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur and D. Evans, 'Privacy-preserving distributed linear regression on high-dimensional data,' *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 345–364, 2017. DOI: doi:10.1515/popets-2017-0053. [Online]. Available: https://doi.org/10.1515/popets-2017-0053 (cit. on pp. 9, 18, 36).

[21] Y. Zhang, S. Wei, S. Liu, Y. Wang, Y. Xu, Y. Li and X. Shang, 'Graph-regularized federated learning with shareable side information,' *Knowledge-Based Systems*, vol. 257, p. 109 960, 2022 (cit. on pp. 9, 18, 36).

[22] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith and B. Thorne, 'Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption,' *ArXiv*, vol. abs/1711.10677, 2017 (cit. on pp. 9, 18, 36).

[23] Y. Liu, Y. Liu, Z. Liu, J. Zhang, C. Meng and Y. Zheng, 'Federated forest,' *ArXiv*, vol. abs/1905.10053, 2019 (cit. on pp. 9, 18, 36).

[24] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen and Q. Yang, 'Secureboost: A lossless federated learning framework,' *ArXiv*, vol. abs/1901.08755, 2019 (cit. on pp. 9, 18, 36).

[25] W. Fang, C. Chen, J. Tan, C. Yu, Y. Lu, L. xilinx Wang, L. Wang, J. Zhou and X Alex, 'A hybrid-domain framework for secure gradient tree boosting,' *ArXiv*, vol. abs/2005.08479, 2020 (cit. on pp. 9, 18, 36).

[26] L. Xie, J. Liu, S. Lu, T.-H. Chang and Q. Shi, 'An efficient learning framework for federated xgboost using secret sharing and distributed optimization,' *ArXiv*, vol. abs/2105.05717, 2021 (cit. on pp. 9, 18, 36).

[27] B. Gu, Z. Dang, X. Li and H. Huang, 'Federated doubly stochastic kernel learning for vertically partitioned data,' *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020 (cit. on pp. 9, 18, 36).

[28] H. Yu, J. Vaidya and X. Jiang, 'Privacy-preserving svm classification on vertically partitioned data,' in *PAKDD*, 2006 (cit. on pp. 9, 18, 36).

[29] R. Axelrod, *Structure of Decision: the Cognitive Maps of Political Elites*. Princeton University Press, Princeton, New Jersey, 1976 (cit. on pp. 9, 18).

[30] J. Salmeron and J. Hurtado, 'Modelling the reasons to establish b2c in the fashion industry,' *Technovation*, vol. 7, no. 26, pp. 865–872, 2006 (cit. on pp. 9, 18).

[31] B. Kosko, 'Fuzzy cognitive maps,' *International Journal of Man-Machine Studies*, vol. 24, no. 1, pp. 65–75, 1986 (cit. on pp. 9, 18, 28).

[32] J. L. Salmeron, S. A. Rahimi, A. M. Navali and A. Sadeghpour, 'Medical diagnosis of rheumatoid arthritis using data driven pso–fcm with scarce datasets,' *Neurocomputing*, vol. 232, pp. 65–75, 2017 (cit. on pp. 9, 18, 31, 32).

[33] S. Rahimi, M. Kolahdoozi, A. Mitra, J. Salmeron, A. Navali, A. Sadeghpour and A. Mohammadi, 'Quantum-inspired interpretable ai-empowered decision support system for detection of early-stage rheumatoid arthritis in primary care using scarce dataset,' *Mathematics*, no. 10, p. 496, 2022. [Online]. Available: `https://www.mdpi.com/2227-7390/10/3/496` (cit. on pp. 9, 18, 32).

[34] R. Guerrero-Gomez-Olmedo, J. L. Salmeron and C. Kuchkovsky, 'Lrp-based path relevances for global explanation of deep architectures,' *Neurocomputing*, vol. 381, pp. 252–260, 2020. DOI: `10.1016/j.neucom.2019.11.059` (cit. on pp. 9, 18).

[35] S. Wang, Y. Wang, D. Wang, Y. Yin, Y. Wang and Y. Jin, 'An improved random forest-based rule extraction method for breast cancer diagnosis,' *Applied Soft Computing*, vol. 86, 2020 (cit. on pp. 10, 19).

[36] J. L. Salmeron and I. Arevalo, 'A privacy-preserving, distributed and cooperative fcm-based learning approach for cancer research,' in *International Joint Conference on Rough Sets*, , La Habana (Cuba), June 29–July 3, 2020 (cit. on pp. 21, 31–33, 38).

[37] I. Arévalo and J. L. Salmeron, 'A chaotic maps-based privacy-preserving distributed deep learning for incomplete and non-iid datasets,' (to appear in IEEE Transactions on Emerging Topics in Computing) (cit. on pp. 22, 40, 46).

[38] J. L. Salmeron, I. Arevalo and A. Ruiz-Celma, 'Benchmarking federated strategies in peer-2-peer federated learning for biomedical data,' *Heliyon*, vol. 6, no. 6, E16925, 2023. [Online]. Available: `https://doi.org/10.1016/j.heliyon.2023.e16925` (cit. on pp. 22, 32, 39).

[39] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010 (cit. on p. 25).

[40] H. Maurer, 'Connectionist neuroarchitectures in cognition and consciousness theory based on integrative (synchronization) mechanisms. in: Pascal hitzler, md kamruzzaman sarker, aaron eberhart (eds.): Compendium of neurosymbolic artificial intelligence. chapter 10. pp. 210-234. ios press. amsterdam. 2023,' in. Aug. 2023, pp. 210–234, ISBN: ISBN: 978-1-64368-406-2 (print) — 978-1-64368-407-9 (online) (cit. on p. 26).

[41] J. L. McClelland and A. Cleeremans, 'Connectionist models,' in *The Oxford Companion to Consciousness*, B. Tim, C. Axel and W. Patrick, Eds., Oxford University Press, 2009 (cit. on p. 26).

[42] S. K. Card and A. Newell, '14 - cognitive architectures,' in *Human Performance Models for Computer-Aided Engineering*, J. I. Elkind, S. K. Card, J. Hochberg and B. M. Huey, Eds., Academic Press, 1990, pp. 173–179, ISBN: 978-0-12-236530-0. DOI: https://doi.org/10.1016/B978-0-12-236530-0.50019-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780122365300500194 (cit. on p. 26).

[43] E. I. Papageorgiou and A. Kannappan, 'Fuzzy cognitive map ensemble learning paradigm to solve classification problems: Application to autism identification,' *Applied Soft Computing*, vol. 12, no. 12, pp. 3798–3809, 2012, Theoretical issues and advanced applications on Fuzzy Cognitive Maps, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2012.03.064. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494612001779 (cit. on p. 26).

[44] R. Chrisley, 'Connectionism, cognitive maps and the development of objectivity,' *Artif. Intell. Rev.*, vol. 7, pp. 329–354, Oct. 1993. DOI: 10.1007/BF00849059 (cit. on p. 26).

[45] K. Kumar and S. Singh, 'Attack type active attack passive attack active attack masquerade alteration of message dos spoofing replay modification,' 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:30200318 (cit. on p. 27).

[46] M. Hemmat Esfe, S. A. Eftekhari, M. Hekmatifar and D. Toghraie, 'A well-trained artificial neural network for predicting the rheological behavior of mwcnt–al2o3 (30–70%)/oil sae40 hybrid nanofluid,' *Scientific Reports*, vol. 11, Aug. 2021. DOI: 10.1038/s41598-021-96808-4 (cit. on p. 28).

[47] L. Deng and D. Yu, 'Deep learning: Methods and applications.,' *Found. Trends Signal Process.*, vol. 7, no. 3-4, pp. 197–387, 2014. [Online]. Available: http://dblp.uni-trier.de/db/journals/ftsig/ftsig7.html#DengY14 (cit. on p. 27).

[48] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, 2016, Book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org (cit. on p. 27).

[49] F. Chollet, *Deep Learning with Python*. Manning, Nov. 2017, ISBN: 9781617294433 (cit. on p. 27).

[50] L. Rodriguez-Repiso, R. Setchi and J. L. Salmeron, 'Modelling it projects success with fuzzy cognitive maps,' *Expert Systems with Applications*, vol. 32, no. 2, pp. 543–559, 2007, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2006.01.032. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417406000510 (cit. on pp. 29, 33).

[51] G. Nápoles, A. Jastrzebska, C. Mosquera, K. Vanhoof and W. Homenda, 'Deterministic learning of hybrid fuzzy cognitive maps and network reduction approaches,' *Neural Networks*, vol. 124, pp. 258–268, 2020 (cit. on pp. 28–30).

[52] J. L. Salmeron and A. Ruiz-Celma, 'Synthetic emotions for empathic building,' *Mathematics*, vol. 9, no. 7, p. 701, 2021. [Online]. Available: https://www.mdpi.com/2227-7390/9/7/701/pdf (cit. on pp. 28, 30).

[53] S. Bueno and J. L. Salmeron, 'Benchmarking main activation functions in fuzzy cognitive maps,' *Expert Systems with Applications*, vol. 36, no. 3 Part 1, pp. 258–268, 2009 (cit. on p. 28).

[54] J. L. Salmeron and W. Froelich, 'Dynamic optimization of fuzzy cognitive maps for time series forecasting,' *Knowledge-Based Systems*, vol. 105, pp. 29–37, 2016 (cit. on pp. 29–31).

[55] C Lopez and J. L. Salmeron, 'Modeling maintenance projects risk effects on erp performance,' *Computer Standards & Interfaces*, vol. 36, no. 3, pp. 545–553, 2014 (cit. on pp. 29, 30).

[56] J. L. Salmeron and P. R. Palos-Sanchez, 'Uncertainty propagation in fuzzy grey cognitive maps with hebbian-like learning algorithms,' *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 211–220, 2019. DOI: 10.1109/TCYB.2017.2771387 (cit. on pp. 29, 31, 32).

[57] G Napoles, J. L. Salmeron and K Vanhoof, 'Construction and supervised learning of long-term grey cognitive networks,' *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 686–695, 2021 (cit. on pp. 30, 31).

[58] G. Napoles, J. L. Salmeron and K. Vanhoof, 'Construction and supervised learning of long-term grey cognitive networks,' *IEEE transactions on cybernetics*, vol. 51, no. 2, pp. 686–695, 2019 (cit. on p. 30).

[59] J. L. Salmeron, T. Mansouri, M. R. S. Moghadam and A. Mardani, 'Learning fuzzy cognitive maps with modified asexual reproduction optimisation algorithm,' *Knowledge-Based Systems*, vol. 163, pp. 723–735, 2019 (cit. on pp. 30, 32).

[60] F. Vanhoenshoven, G. Nápoles, W. Froelich, J. L. Salmeron and K. Vanhoof, 'Pseudoinverse learning of fuzzy cognitive maps for multivariate time series forecasting,' *Applied Soft Computing*, vol. 95, p. 106 461, 2020, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2020.106461. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620304014 (cit. on pp. 30, 32).

[61] G. A. Papakostas, Y. S. Boutalis, D. E. Koulouriotis and B. G. Mertzios, 'Fuzzy cognitive maps for pattern recognition applications,' *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 8, pp. 1461–1486, 2008 (cit. on p. 30).

[62] G. Papakostas and D. Koulouriotis, 'Fuzzy cognitive maps,' in, ser. Studies in Fuzziness and Soft Computing (volume 247). Springer, 2010, ch. Classifying Patterns Using Fuzzy Cognitive Maps, pp 291–306 (cit. on p. 30).

[63] P. Szwed, 'Classification and feature transformation with fuzzy cognitive maps,' *Applied Soft Computing*, vol. 105, p. 107 271, 1 2021 (cit. on p. 30).

[64] K. Wu, K. Yuan, Y. Teng, J. Liu and L. Jiao, 'Broad fuzzy cognitive map systems for time series classification,' *Applied Soft Computing*, vol. 128, p. 109 458, 2022 (cit. on p. 30).

[65] J. A. Ramirez-Bautista, J. A. Huerta-Ruelas, L. T. Kóczyb, M. F. Hatwágner, S. L.Chaparro-Cárdenasa and A. Hernández-Zavala, 'Classification of plantar foot alterations by fuzzy cognitive maps against multi-layer perceptron neural network,' *Biocybernetics and Biomedical Engineering*, vol. 40, pp. 404–414, 1 2020 (cit. on p. 30).

[66] A Baykasoğlu and I Gölcük, 'Alpha-cut based fuzzy cognitive maps with applications in decision-making,' *Computers & Industrial Engineering*, vol. 152, p. 107 007, 2021 (cit. on p. 30).

[67] G. A. Papakostas, D. E. Koulouriotis, A. S. Polydoros and V. D. Tourassis, 'Towards hebbian learning of fuzzy cognitive maps in pattern classification problems,' *Applied Soft Computing*, vol. 39, no. 12, pp. 10 620–10 629, 2012 (cit. on p. 30).

[68] J. L. Salmeron, A Ruiz-Celma and A Mena, 'Learning fcms with multilocal and balanced memetic algorithms for forecasting drying processes,' *Neurocomputing*, vol. 232, pp. 52–57, 2017 (cit. on pp. 31, 32).

[69] J. A. Dickerson and B. Kosko, 'Virtual worlds as fuzzy cognitive maps,' *Presence: Teleoperators & Virtual Environments*, vol. 3, no. 2, pp. 173–189, 1994 (cit. on p. 32).

[70] B. Kosko, *Fuzzy engineering*. Prentice-Hall, Inc., 1996 (cit. on p. 32).

[71] W. Stach, L. Kurgan and W. Pedrycz, 'Data-driven nonlinear hebbian learning method for fuzzy cognitive maps,' *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pp. 1975–1981, 2008 (cit. on p. 32).

[72] E. Papageorgiou, C. Stylios and P. Groumpos, 'Fuzzy cognitive map learning based on nonlinear hebbian rule,' *Australasian Joint Conference on Artificial Intelligence*, pp. 256–268, 2003 (cit. on p. 32).

[73] A. V. Huerga, 'A balanced differential learning algorithm in fuzzy cognitive maps,' *Proceedings of the 16th International Workshop on Qualitative Reasoning*, vol. 2002, 2002 (cit. on p. 32).

[74] A. Konar and U. K. Chakraborty, 'Reasoning and unsupervised learning in a fuzzy cognitive map,' *Information Sciences*, vol. 170, no. 2, pp. 419–441, 2005 (cit. on p. 32).

[75] E. Papageorgiou, C. D. Stylios and P. P. Groumpos, 'Active hebbian learning algorithm to train fuzzy cognitive maps,' *International journal of approximate reasoning*, vol. 37, no. 3, pp. 219–249, 2004 (cit. on p. 32).

[76] D. Koulouriotis, I. Diakoulakis and D. Emiris, 'Learning fuzzy cognitive maps using evolution strategies: A novel schema for modeling and simulating high-level behavior,' in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, IEEE, vol. 1, 2001, pp. 364–371 (cit. on p. 32).

[77] K. E. Parsopoulos, E. I. Papageorgiou, P. Groumpos and M. N. Vrahatis, 'A first study of fuzzy cognitive maps learning using particle swarm optimization,' in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, IEEE, vol. 2, 2003, pp. 1440–1447 (cit. on p. 32).

[78] N. H. Mateou, M. Moiseos and A. S. Andreou, 'Multi-objective evolutionary fuzzy cognitive maps for decision support,' in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, IEEE, vol. 1, 2005, pp. 824–830 (cit. on p. 32).

[79] W. Stach, L. Kurgan, W. Pedrycz and M. Reformat, 'Genetic learning of fuzzy cognitive maps,' *Fuzzy sets and systems*, vol. 153, no. 3, pp. 371–401, 2005 (cit. on p. 32).

[80] K. Poczeta, A. Yastrebov and E. I. Papageorgiou, 'Learning fuzzy cognitive maps using structure optimization genetic algorithm,' in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, IEEE, 2015, pp. 547–554 (cit. on p. 32).

[81] Y. Petalas, E. Papageorgiou, K. Parsopoulos, P. Groumpos and M. Vrahatis, 'Fuzzy cognitive maps learning using memetic algorithms,' in *Proceedings of the international conference of "Computational Methods in Sciences and Engineering"(ICCMSE 2005)*, 2005, pp. 1420–1423 (cit. on p. 32).

[82] M. Ghazanfari, S. Alizadeh, M. Fathian and D. E. Koulouriotis, 'Comparing simulated annealing and genetic algorithm in learning fcm,' *Applied Mathematics and Computation*, vol. 192, no. 1, pp. 56–68, 2007 (cit. on p. 32).

[83] S. Alizadeh and M. Ghazanfari, 'Learning fcm by chaotic simulated annealing,' *Chaos, Solitons & Fractals*, vol. 41, no. 3, pp. 1182–1190, 2009 (cit. on p. 32).

[84] S. Alizadeh, M. Ghazanfari, M. Jafari and S. Hooshmand, 'Learning fcm by tabu search,' *International Journal of Computer Science*, vol. 2, no. 2, pp. 142–149, 2007 (cit. on p. 32).

[85] X. Luo, X. Wei and J. Zhang, 'Game-based learning model using fuzzy cognitive map,' in *Proceedings of the first ACM international workshop on Multimedia technologies for distance learning*, ACM, 2009, pp. 67–76 (cit. on p. 32).

[86] P. Juszczuk and W. Froelich, 'Learning fuzzy cognitive maps using a differential evolution algorithm,' *Pol. J. Environ. Stud*, vol. 12, pp. 108–112, 2009 (cit. on p. 32).

[87] C. Lin, 'An immune algorithm for complex fuzzy cognitive map partitioning,' in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, ACM, 2009, pp. 315–320 (cit. on p. 32).

[88] E. Yesil and L. Urbas, 'Big bang-big crunch learning method for fuzzy cognitive maps,' *World Acad. Sci. Eng. Technol*, vol. 71, pp. 815–824, 2010 (cit. on p. 32).

[89] J Vaščák, 'Approaches in adaptation of fuzzy cognitive maps for navigation purposes,' in *Applied Machine Intelligence and Informatics (SAMI), 2010 IEEE 8th International Symposium on*, IEEE, 2010, pp. 31–36 (cit. on p. 32).

[90] Z. Ding, D. Li and J. Jia, 'First study of fuzzy cognitive map learning using ants colony optimization,' 2011 (cit. on p. 32).

[91] A. Baykasoglu, Z. D. Durmusoglu and V. Kaplanoglu, 'Training fuzzy cognitive maps via extended great deluge algorithm with applications,' *Computers in Industry*, vol. 62, no. 2, pp. 187–195, 2011 (cit. on p. 32).

[92] E. Yesil, C. Ozturk, M. F. Dodurka and A. Sakalli, 'Fuzzy cognitive maps learning using artificial bee colony optimization,' in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1–8 (cit. on p. 32).

[93] S. Ahmadi, N. Forouzideh, C.-H. Yeh, R. Martin and E. Papageorgiou, 'A first study of fuzzy cognitive maps learning using cultural algorithm,' in *Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on*, IEEE, 2014, pp. 2023–2028 (cit. on p. 32).

[94] S. Ahmadi, N. Forouzideh, S. Alizadeh and E. Papageorgiou, 'Learning fuzzy cognitive maps using imperialist competitive algorithm,' *Neural Computing and Applications*, vol. 26, no. 6, pp. 1333–1354, 2015 (cit. on p. 32).

[95] Y. Chi and J. Liu, 'Learning of fuzzy cognitive maps with varying densities using a multiobjective evolutionary algorithm,' *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 71–81, 2016 (cit. on p. 32).

[96] E. I. Papageorgiou and P. P. Groumpos, 'A new hybrid method using evolutionary algorithms to train fuzzy cognitive maps,' *Applied Soft Computing*, vol. 5, no. 4, pp. 409–431, 2005 (cit. on p. 32).

[97] Y. Zhu and W. Zhang, 'An integrated framework for learning fuzzy cognitive map using rcga and nhl algorithm,' in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, IEEE, 2008, pp. 1–5 (cit. on p. 32).

[98] Z. Ren, 'Learning fuzzy cognitive maps by a hybrid method using nonlinear hebbian learning and extended great deluge algorithm.,' in *MAICS*, 2012, pp. 159–163 (cit. on p. 32).

[99] E. I. Papageorgiou and J. L. Salmeron, 'A review of fuzzy cognitive maps research during the last decade,' *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 66–79, 2013. DOI: 10.1109/TFUZZ.2012.2201727 (cit. on p. 33).

[100] J. L. Salmeron, 'Fuzzy cognitive maps for artificial emotions forecasting,' *Applied Soft Computing*, vol. 12, no. 12, pp. 3704–3710, 2012, Theoretical issues and advanced applications on Fuzzy Cognitive Maps, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2012.01.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494612000415 (cit. on p. 33).

[101] S. Bueno and J. L. Salmeron, 'Fuzzy modeling enterprise resource planning tool selection,' *Computer Standards & Interfaces*, vol. 30, no. 3, pp. 137–147, 2008, ISSN: 0920-5489. DOI: https://doi.org/10.1016/j.csi.2007.08.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548907000682 (cit. on p. 33).

[102] J. L. Salmeron, 'Augmented fuzzy cognitive maps for modelling lms critical success factors,' *Knowledge-Based Systems*, vol. 4, no. 22, pp. 275–278, 2009 (cit. on p. 33).

[103] ——, 'Modelling grey uncertainty with fuzzy grey cognitive maps,' *Expert Systems with Applications*, vol. 12, no. 37, pp. 7581–7588, 2010 (cit. on p. 33).

[104] H. Zhu, J. Xu, S. Liu and Y. Jin, 'Federated learning on non-iid data: A survey,' *Neurocomputing*, vol. 465, pp. 371–390, 2021 (cit. on pp. 35, 40).

[105] J. Konecný, B. McMahan, D. Ramage and P. Richtárik, 'Federated optimization: Distributed machine learning for on-device intelligence,' *ArXiv*, vol. abs/1610.02527, 2016 (cit. on p. 35).

[106] J. Duan, J. Zhou, Y. Li and C. Huang, 'Privacy-preserving and verifiable deep learning inference based on secret sharing,' *Neurocomputing*, vol. 483, pp. 221–234, Apr. 2022. DOI: 10.1016/j.neucom.2022.01.061 (cit. on p. 35).

[107] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian and F. Wang, 'Federated learning for healthcare informatics,' *Journal of Healthcare Informatics Research*, vol. 5, no. http://dx.doi.org/ 10.1007/s41666-020-00082-4, 1—19, 2021 (cit. on p. 36).

[108] Z. Zhao, C. Feng, W. Hong, J. Jiang, C. Jia, T. Q. S. Quek and M. Peng, 'Federated learning with non-iid data in wireless networks,' *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1927–1942, 2022. DOI: https://doi.org/10.1109/TWC.2021.3108197 (cit. on p. 36).

[109] F. Sattler, S. Wiedemann, K.-R. Müller and W. Samek, 'Robust and communication-efficient federated learning from non-iid data,' *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, 2020. DOI: https://doi.org/10.1109/TNNLS.2019.2944481 (cit. on p. 36).

[110] L. Su and V. K. N. Lau, 'Hierarchical federated learning for hybrid data partitioning across multitype sensors,' *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 922–10 939, 2021 (cit. on p. 36).

[111] R. Hou, S. Ai, Q. Chen, H. Yan, T. Huang and K. Chen, 'Similarity-based integrity protection for deep learning systems,' *Information Sciences*, vol. 601, pp. 255–267, 2022, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.04.003 (cit. on p. 36).

[112] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen and H. Yu, *Federated Learning*. Morgan & Claypool, 2019, vol. 13, (3) , 2019, pp. 1–207 (cit. on pp. 37, 39).

[113] D. Gao, C. Ju, X. Wei, Y. Liu, T. Chen and Q. Yang, *Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography*, 2019. DOI: 10.48550/ARXIV.1909.05784. [Online]. Available: https://arxiv.org/abs/1909.05784 (cit. on p. 39).

[114] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos and Q. Yang, *Secureboost: A lossless federated learning framework*, 2019. DOI: 10.48550/ARXIV.1901.08755. [Online]. Available: https://arxiv.org/abs/1901.08755 (cit. on p. 39).

[115] S. Lee, M. E. Lacy, M. Jankowich, A. Correa and W.-C. Wu, 'Association between obesity phenotypes of insulin resistance and risk of type 2 diabetes in african americans: The jackson heart study,' *Journal of Clinical & Translational Endocrinology*, vol. 19, no. 3, p. 100 210, 2020 (cit. on p. 39).

[116] Y. Liu, Y. Kang, C. Xing, T. Chen and Q. Yang, 'A secure federated transfer learning framework,' *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020. DOI: 10.1109/mis.2020.2988525. [Online]. Available: https://doi.org/10.1109%2Fmis.2020.2988525 (cit. on p. 39).

[117] H. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, 'Communication-efficient learning of deep networks from decentralized data,' in *International Conference on Artificial Intelligence and Statistics*, 2016 (cit. on p. 40).

[118] P. Chan, Z.-M. He, H. Li and C.-C. Hsu, 'Data sanitization against adversarial label contamination based on data complexity,' *International Journal of Machine Learning and Cybernetics*, vol. 9, Jun. 2018. DOI: 10.1007/s13042-016-0629-5 (cit. on p. 43).

[119] Y. Yang, J. Cai, H. Yang, Y. Li and X. Zhao, 'Isbfk-means: A new clustering algorithm based on influence space,' *Expert Systems with Applications*, vol. 201, p. 117 018, 2022, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.117018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422004365 (cit. on p. 43).

[120] Y. Tian, W. Zhang, A. Simpson, Y. Liu and Z. L. Jiang, 'Defending Against Data Poisoning Attacks: From Distributed Learning to Federated Learning,' *The Computer Journal*, vol. 66, no. 3, pp. 711–726, Dec. 2021, ISSN: 0010-4620. DOI: 10.1093/comjnl/bxab192. eprint: https://academic.oup.com/comjnl/article-pdf/66/3/711/49530973/bxab192.pdf. [Online]. Available: https://doi.org/10.1093/comjnl/bxab192 (cit. on p. 43).

[121] Y. Qi, M. S. Hossain, J. Nie and X. Li, 'Privacy-preserving blockchain-based federated learning for traffic flow prediction,' *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2020.12.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X2033065X (cit. on p. 43).

[122] Z. Cui, F. Xue, S. Zhang, X. Cai, Y. Cao, W. Zhang and J. Chen, 'A hybrid blockchain-based identity authentication scheme for multi-wsn,' *IEEE Transactions on Services Computing*, vol. PP, pp. 1–1, Jan. 2020. DOI: 10.1109/TSC.2020.2964537 (cit. on p. 43).

[123] Y. Zhao, J. Chen, J. Zhang, D. Wu, M. Blumenstein and S. Yu, 'Detecting and mitigating poisoning attacks in federated learning using generative adversarial networks,' *Concurrency and Computation: Practice and Experience*, vol. 34, Jun. 2020. DOI: 10.1002/cpe.5906 (cit. on p. 44).

[124] X. Li, S. Cao, L. Gao and L. Wen, 'A threshold-control generative adversarial network method for intelligent fault diagnosis,' *Complex System Modeling and Simulation*, vol. 1, pp. 55–64, Mar. 2021. DOI: 10.23919/CSMS.2021.0006 (cit. on p. 44).

[125] S. Shi, C. Hu, D. Wang, Y. Zhu and Z. Han, 'Federated anomaly analytics for local model poisoning attack,' *IEEE Journal on Selected Areas in Communications*, vol. PP, pp. 1–1, Oct. 2021. DOI: 10.1109/JSAC.2021.3118347 (cit. on p. 44).

[126] X. Ma, Q. Jiang, M. Shojafar, M. Alazab, S. Kumar and S. Kumari, 'Disbezant: Secure and robust federated learning against byzantine attack in iot-enabled mts,' *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–12, Feb. 2022. DOI: 10.1109/TITS.2022.3152156 (cit. on p. 44).

[127] K. Zhai, Q. Ren, J. Wang and C. Yan, 'Byzantine-robust federated learning via credibility assessment on non-iid data,' *Mathematical Biosciences and Engineering*, vol. 19, pp. 1659–1676, Jan. 2021. DOI: 10.3934/mbe.2022078 (cit. on p. 44).

[128] M. Zhang and L. Mo, 'Mgwhd-svm: Maximum weighted heteroscedastic migration learning algorithm,' *Int. J. Comput. Sci. Math.*, vol. 14, no. 1, 89–106, 2021, ISSN: 1752-5055. DOI: 10.1504/ijcsm.2021.118078. [Online]. Available: https://doi.org/10.1504/ijcsm.2021.118078 (cit. on p. 44).

[129] W. Li and S. Wang, 'Federated meta-learning for spatial-temporal prediction,' *Neural Comput. Appl.*, vol. 34, no. 13, 10355–10374, 2022, ISSN: 0941-0643. DOI: 10.1007/s00521-021-06861-3. [Online]. Available: https://doi.org/10.1007/s00521-021-06861-3 (cit. on p. 44).

[130] M. Abadi, A. Chu, I. Goodfellow, B. McMahan, I. Mironov, K. Talwar and L. Zhang, 'Deep learning with differential privacy,' in *23rd ACM Conference on Computer and Communications Security (ACM CCS)*, 2016, pp. 308–318. [Online]. Available: https://arxiv.org/abs/1607.00133 (cit. on pp. 44, 45).

[131] A. Acar, H. Aksu, S. Uluagac and M. Conti, 'A survey on homomorphic encryption schemes: Theory and implementation,' *ACM Computing Surveys*, vol. 51, Apr. 2017. DOI: 10.1145/3214303 (cit. on p. 44).

[132] G. Kaissis, M. Makowski, D. Rückert and R. Braren, 'Secure, privacy-preserving and federated machine learning in medical imaging,' *Nature Machine Intelligence*, vol. 2, pp. 305–311, 2020 (cit. on p. 44).

[133] R. Hu, Y. Guo, H. Li, Q. Pei and Y. Gong, 'Privacy-preserving personalized federated learning,' in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9149207 (cit. on p. 44).

[134] Y. Cheng, Y. Liu, T. Chen and Q. Yang, 'Federated learning for privacy-preserving ai,' *Commun. ACM*, vol. 63, no. 12, 33–36, Nov. 2020, ISSN: 0001-0782. DOI: 10.1145/3387107. [Online]. Available: https://doi.org/10.1145/3387107 (cit. on p. 44).

[135] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu and J. Li, 'A training-integrity privacy-preserving federated learning scheme with trusted execution environment,' *Information Sciences*, vol. 522, Feb. 2020. DOI: 10.1016/j.ins.2020.02.037 (cit. on p. 44).

[136] F. Mo and H. Haddadi, 'Efficient and private federated learning using tee,' 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:211627925 (cit. on p. 44).

[137] G. Ateniese, G. Felici, L. Mancini, A. Spognardi, A. Villani and D. Vitali, 'Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers,' *International Journal of Security and Networks*, vol. 10, Jun. 2013. DOI: 10.1504/IJSN.2015.071829 (cit. on p. 44).

[138] B. Hitaj, G. Ateniese and F. Perez-Cruz, 'Deep models under the gan: Information leakage from collaborative deep learning,' in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17, Association for Computing Machinery, 2017, 603–618, ISBN: 9781450349468. DOI: 10.1145/3133956.3134012. [Online]. Available: https://doi.org/10.1145/3133956.3134012 (cit. on p. 44).

[139] L. Melis, C. Song, E. D. Cristofaro and V. Shmatikov, 'Exploiting unintended feature leakage in collaborative learning,' *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 691–706, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53099247 (cit. on p. 44).

[140] R. Kanagavelu, Q. Wei, Z. Li, H. Zhang, J. Samsudin, Y. Yang, R. S. M. Goh and S. Wang, 'Ce-fed: Communication efficient multi-party computation enabled federated learning,' *Array*, vol. 15, p. 100 207, 2022, ISSN: 2590-0056. DOI: https://doi.org/10.1016/j.array.2022.100207. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590005622000546 (cit. on p. 45).

[141] Y. Zhao, N. Suda, M. Li, D. Civin, L. Lai and V. Chandra, 'Federated learning with non-iid data: A metric learning approach,' *arXiv*, vol. 1806.00582, 2018 (cit. on p. 45).

[142] O. Ibitoye, R. Abou-Khamis, M. el Shehaby, A. Matrawy and M. O. Shafiq, *The threat of adversarial attacks on machine learning in network security – a survey*, 2023. DOI: 10.48550/ARXIV.1901.08755. [Online]. Available: https://arxiv.org/abs/1901.08755 (cit. on p. 46).

[143] U. Zia, M. McCartney, B. Scotney, J. Martinez, M. AbuTair, J. Memon and A. Sajjad, 'Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains,' *International Journal of Information Security*, vol. 21, no. 4, pp. 917–935, 2022. DOI: 10.1007/s10207-022-00588-5. [Online]. Available: https://doi.org/10.1007/s10207-022-00588-5 (cit. on p. 47).