Department of Excellence

Federated Learning and Explainable Artificial Intelligence: a Favorable Synergy

Francesco Marcelloni

Coordinator
Artificial Intelligence Group
Department of Information Engineering
Largo Lucio Lazzarino 1
56123 PISA

Coordinator
IT2PAO lab
Joint laboratory with LogObject AG (Switzerland)
Ponsacco

E-mail: francesco.marcelloni@unipi.it

1

Francesco Marcelloni

Syllabus

2

DIPARTIMENTO DI INGEGNERIA

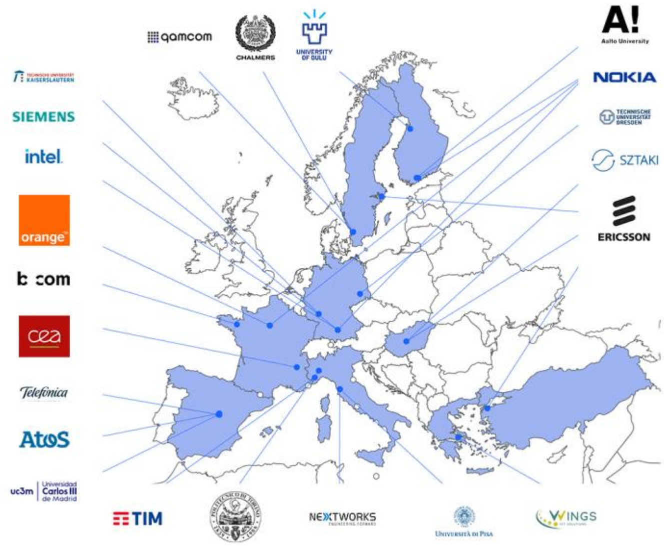
Francesco Marcelloni

HEXA-X: The European 6G flagship project

Hexa-X

*A **flagship for B5G/6G vision** and intelligent fabric of technology enablers connecting human, physical, and digital worlds*

The use of AI will be fundamental



EU project: HEXA-X - **Programme**: Horizon 2020 - **Grant Agreement ID**: 101015956

3

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

The EU view of AI

Seven requirements toward Trustworthy AI

- Human agency and oversight
- Accountability
- Technical robustness and safety
- Societal and environmental wellbeing
- Diversity non-discrimination and fairness
- Privacy and data governance
- Transparency

Need to collect data to train accurate AI models clashes with need to preserve privacy of data owners.

"AI systems and their decisions should be explained in a manner adapted to the stakeholder concerned."

Fed-XAI
Federated Learning of eXplainable AI models

4

Why do we need Federated Learning?

- Machine learning algorithms, especially **deep learning algorithms**, are **data hungry.**
- **Data are generally spread** over different devices with different owners and under the protection of **privacy restrictions**.
- In practice, we cope with isolated **data islands** and we cannot transfer data

5

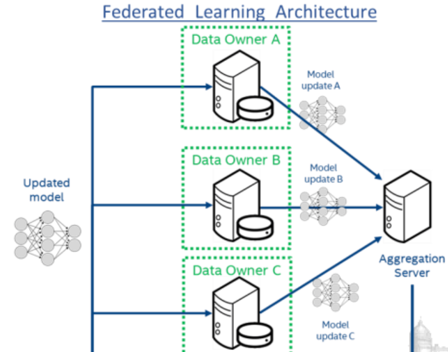
Why do we need Federated Learning?

«Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a Machine learning problem… Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective.»

«Advances and Open Problems in Federated Learning» Foundations and Trends® in Machine Learning, Vol 14

- Federated Learning:
  - **to aggregate updates**, learned at the edge devices, of the distributed individual versions of the global ML model and to exploit the aggregation for updating the model
  - **to broadcast the model** to allow edge devices to continuously refine their individual versions

Federated Learning Architecture



6

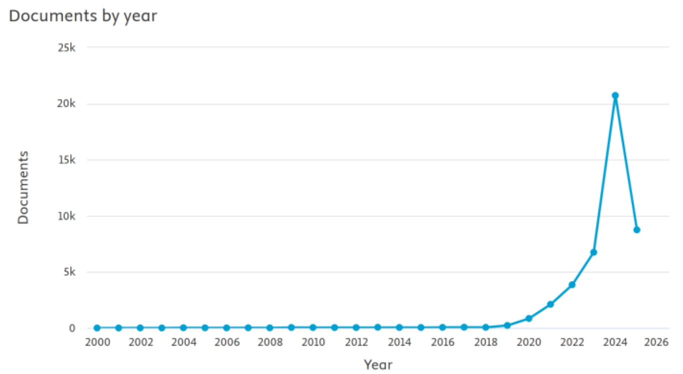
DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Federated learning: hot topic

- The interest in Federated Learning is rapidly increasing (Scopus)

Year ↓	Documents ↑
2025	8763
2024	20764
2023	6761
2022	3838
2021	2111
2020	850
2019	230
2018	63
2017	72



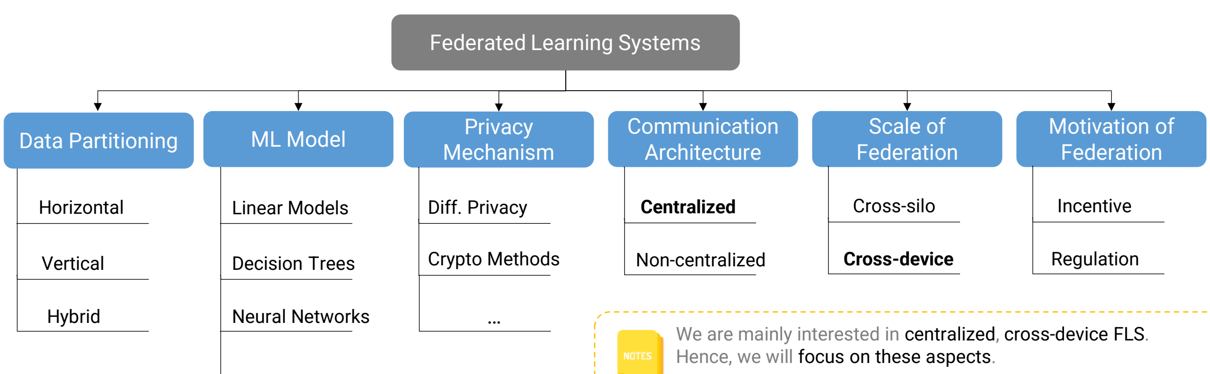
7

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Federated learning: a taxonomy

A Federating Learning System taxonomy according to six main characterizing aspects



NOTES We are mainly interested in centralized, cross-device FLS. Hence, we will focus on these aspects.

Li Q., Wen Z., Wu Z., Hu S., Wang N., Li Y., Liu X., He B.A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection(2021) IEEE Transactions on Knowledge and Data Engineering

8

Data Partitioning

Based on how data are distributed among the parties making up the federation over the sample and feature spaces, FLSs can be typically categorized in horizontal, vertical and hybrid FLSs:

In Horizontal FL the datasets of different parties share the **same feature space** but have **little or no intersection on the sample space** (CT scans reported in different hospitals).

This is a natural data partitioning especially for the **cross-device setting**, where different users try to improve their model performance on the same task using FL.

Device #1	Feature 1	...	Feature N
Sample 1			
Sample 2			
Sample 3			

Device #2	Feature 1	...	Feature N
Sample 4			
Sample 5			

Device #3	Feature 1	...	Feature N
Sample 5			
Sample 6			
Sample 7			

9

DIPARTIMENTO DI INGEGNERIA
Francesco Marcelloni

Data Partitioning

In Vertical FL the datasets of different parties have the **same or similar sample space** but **differ in** the **feature space** (for instance, municipality registry and hospital data).

Device #1	Feature 1	Feature 2
Sample 1		
Sample 2		
Sample 3		
Sample 4		
Sample 5		
Sample 6		

Device #2	Feature 3	Feature 4	Feature 5
Sample 1			
Sample 2			
Sample 3			
Sample 4			
Sample 5			
Sample 6			

Device #3	Feature 6	Feature 7
Sample 1		
Sample 2		
Sample 3		
Sample 4		
Sample 5		
Sample 6		

In Hybrid FL partition of data among the parties may be **a hybrid of horizontal partition and vertical partition**.

10

Francesco Marcelloni

ML Models

There have been many efforts in developing new models or reinventing current models to the federated setting. For the sake of brevity **we briefly cite the widely-used models nowadays**:

- Neural Networks: there are many studies on **federated stochastic gradient descent** which can be used to train NNs.
- Decision tree is another widely used model as it is highly efficient to train compared with NNs. (FLSs studies for Gradient Boosting decision trees - GBDTs have been proposed recently).
- SVM: there exist a number of examples in which SVM is successfully trained exploiting a federated stochastic gradient descent algorithm.

11

Francesco Marcelloni

Privacy Mechanisms

Model parameters exchanged during FL rounds may leak sensitive information about the data. Beyond attacks targeting user privacy, there are also other classes of attacks on federated learning (e.g. an adversary might attempt to bias the model to produce inferences that are preferable to the adversary and much else).

Technology	Main characteristics
Differential Privacy	Add properly tuned random noise to mask the influence of an individual instance on the output.
Secure Multi-Party Computation	Enables two or more parties to compute an agreed-upon function of their private inputs in a way that only reveals the intended output to each of the parties, while keeping those inputs private.
Homomorphic Encryption	Enables parties to perform mathematical operations directly on encrypted data without decrypting them.
Trusted Execution Environments	TEEs provide the ability to trustably run code on a remote machine, even if you do not trust the machine's owner. TEEs may provide confidentiality, integrity and remote attestation.

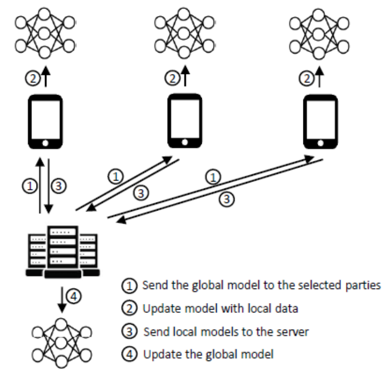
12

Francesco Marcelloni

Communication Architecture

Centralized versus non-centralized

In the **centralized architecture** the **data flow** is **asymmetric**: the **server aggregates** the information (e.g. gradients or model parameters) from the clients and **sends them back** the **updated global model**. The **process is executed iteratively until** a **convergence** criterion is met.



Li Q., Wen Z., Wu Z., Hu S., Wang N., Li Y., Liu X., He B.A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection(2021) IEEE Transactions on Knowledge and Data Engineering

13

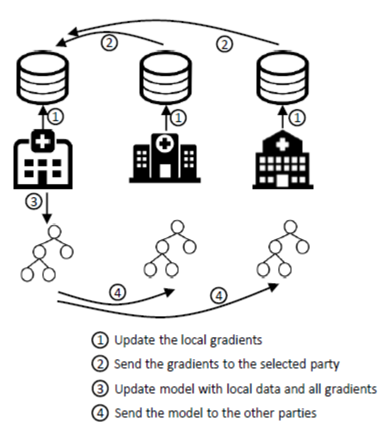
Francesco Marcelloni

Communication Architecture

Centralized versus non-centralized

In the **non-centralized architecture** the **communications** are **performed among the parties** and every party is able to **update** the **global parameters directly**. There is **no need for a trusted central aggregating server**.

In the **non-centralized architecture** the **major challenge** is that it is hard to design a protocol that treats every member almost fairly with reasonable communication overhead.



Li Q., Wen Z., Wu Z., Hu S., Wang N., Li Y., Liu X., He B.A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection(2021) IEEE Transactions on Knowledge and Data Engineering

14

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Scale of Federation

FLSs can be categorized into **two typical types** by the scale of federation: cross-silo FLSs and cross-device FLSs. The **main differences** between them lie on the **number of parties** and the **amount of data stored in each party**.

	Setting	Data Distrubution	Data Availability	Distribution Scale	Primary Bottleneck	Client reliability	Data partition axis
Cross-silo	Training a model on siloed data. **Clients are different organizations** or geo-ditributed datacenters.	**Data is generated locally and remains decentralized.** Each client stores its own data and cannot read the data of other clients.	**All clients are almost always available.**	**Typically, 2 – 100 clients.**	Might be **computation** or **communication.**	**Relatively few failures.**	Partition is fixed. Could be example-partitioned (horizontal) or feature-partitioned (vertical).
Cross-device	The clients are a **very large number of mobile or IoT devices.**		**Only a fraction of clients are available** at any one time, often with diurnal or other variations.	**Massively parallel, up to 10^{10} clients.**	**Communication** is often the **primary bottleneck.** Generally, cross-device computations use **wi-fi or slower connections.**	**Highly unreliable – 5% or more of the clients participating in a round of computation are expected to fail** or drop out.	Fixed partitioning by example (horizontal)

15

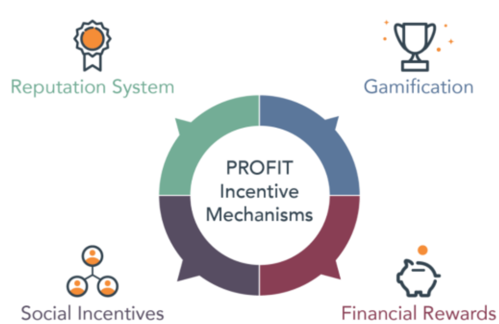
DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Motivation of Federation

In **real-world applications** of FL, **individual parties need the motivation to get involved** in the FLS. The motivation can be **regulations or incentives**. The parties inside the system can be collaborators as well as competitors.

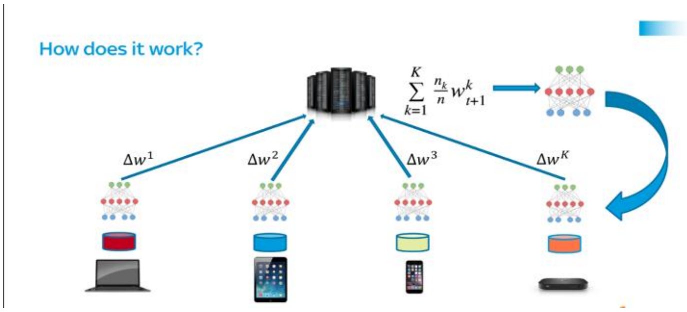
Take **Google Keyboard as an example**: Google cannot prevent users who do not provide data from Using Gboard. But those who agree to upload input data may enjoy a higher accuracy of word prediction.



16

Francesco Marcelloni

Popular approaches (Federated Averaging)



Federated Learning (Source: https://proandroiddev.com/federated-learning-e79e054c33ef)

- C = fraction of clients that participates in each federated round
- K = total number of clients (indexed by k)
- E = number of training passes each client makes over its local dataset on each round
- B = local minibatch size used for the client updates (B = ∞ indicates that the full local dataset is treated as a single minibatch
- P_k = set of indexes of data points on client k, with $n_k = |P_k|$

Algorithm 1 FederatedAveraging. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \ldots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow$ ClientUpdate(k, w_t)
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w)**:** // *Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

Zhu H., Xu J., Liu S., Jin Y.Federated learning on non-IID data: A survey (2021) Neurocomputing, 465, pp. 371 - 390

17

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Popular approaches

Federated Stochastic Gradient Descent (FedSGD) vs Federated averaging (FedAVG):

In **FedSGD** each client k computes the gradient on its local data at the current model w_t and the central server aggregates these gradients and updates the global model.
Note that FedSGD coincides to FedAvg with
$C = 1, B = \infty, E = 1$

In **FedAVG** each client locally takes one or multiple steps of gradient descent on the current model w_t using its local data, and the server then takes a weigthed average of the resulting models.

- C = fraction of clients that participates in each federated round
- K = total number of clients (indexed by k)
- E = number of training passes each client makes over its local dataset on each round
- B = local minibatch size used for the client updates (B = ∞ indicates that the full local dataset is treated as a single minibatch
- P_k = set of indexes of data points on client k, with $n_k = |P_k|$

For the current global model w^t, the average gradient on its global model is calculated for each client k.

$$F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w)$$

$$g_k = \nabla F_k(w_t)$$

The central server then aggregates these gradients and applies the update.

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k$$

Zhu H., Xu J., Liu S., Jin Y.Federated learning on non-IID data: A survey(2021) Neurocomputing, 465, pp. 371 - 390

18

XAI models: why?

INTEPRETABLE MODELS VS **MODEL INTERPRETABILITY TECHNIQUES**

Transparent Models. Models which are interpretable by design. Within trasparency **three levels** are contemplated:

Simulatability. The ability of a model of being simulated by a human.

Decomposability. The ability to explain each of the parts of a model.

Algorithmic transparency. The ability of the user to understand the process followed by the model to produce output from input.

Models that can be explained by means of **external XAI tecniques**: **post-hoc explainability**. This category comprises tecniques corresponding to the most common ways humans explain things by themselves:

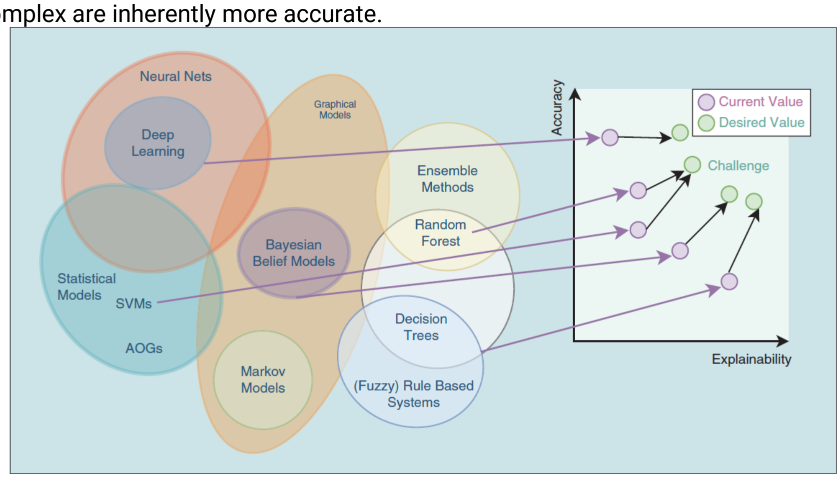
- **Text explanations**
- **Visual explanations**
- **Local explanations**
- **By example**
- **By simplification**
- **Feature relevance**

19

XAI: interpretability versus performance

Generally, models that are more complex are inherently more accurate.

The **two extremes** are represented on the **accuracy side** by **Neural Networks** which, indeed, are often called black-box models,
on **the explainability side** by all **rule-based systems** such as **decision trees**.



Fernandez A., Herrera F., Cordon O., Jose Del Jesus M., Marcelloni F., Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to?(2019) IEEE Computational Intelligence Magazine, 14 (1), pp. 69 - 81

20

Francesco Marcelloni

Federated Decision trees

Algorithm 1 The ICDTA4FL process

```
Client's side
    for C_i; i=1, ..., n do
        C_i ← train a decision tree, LocalDT_i, with its local data D_i.
        Send LocalDT_i to the Server.
Server's side
    Send the received trees to the clients.
Client's side
    for C_i; i=1 to n do
        C_i ← evaluate the local DTs, C_k LocalDT_i, k = 1, ..., n; i ≠ k
        Send the evaluation metrics to the server.
Server's side
    Delete the trees that do not surpass a filter selected for the metrics.
    Extract the rules for selected decision trees.
    Aggregate the rules applying the Cartesian product.
    Build a global decision tree, GlobalDT with the aggregated rules.
    Send GlobalDT and the aggregated rules to the clients.
Client's side
    for C_i; i=1 to n do
        Evaluate the GlobalDT with its local data D_i
```

A. Argente-Garrido, C. Zuheros, M.V. Luzón, F. Herrera, An interpretable client decision tree aggregation process for federated learning, Information Sciences, Volume 694, 2025

Francesco Marcelloni

Federated Decision trees

Client 1 extracted rules:

x0 <= 32.5 → Class 1; 2 instances; **1A**
x0 > 32.5 → Class 2; 4 instances; **2A**

Client 2 extracted rules:

x3>49; x54<=197085; x64<=0.5; x71<=14.5 → Class 1; 105 instances; **1B**
x3>49; x54>197085; x64<=0.5; x71<=14.5 → Class 1; 1 instances; **2B**
x0>39; x74>0.5;x3>22;x71<=10.5 → Class 2; 31 instances; **3B**
x0>35; x74>0.5;x3>22;x71>10.5; x51<=0.5 → Class 2; 29 instances; **4B**
x0>35; x74>0.5;x3>22;x71>10.5; x51>0.5 → Class 1; 3 instances; **5B**

Some aggregated rules

x0>32.5; x74>0.5;x3>22;x71<=10.5 → Class 2; **2A-3B**
x0>32.5; x74>0.5;x3>22;x71>10.5; x51<=0.5 → Class 2; **2A-4B**
x0>32.5; x74>0.5;x3>22;x71>10.5; x51>0.5 → Class 2; **2A-5B**

if two rules share a condition, i.e., rule 2A, and rule 4B, the less restrictive condition is selected

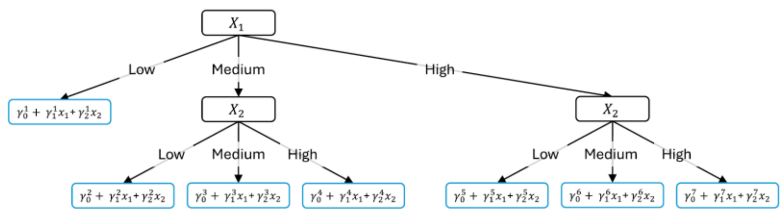
The rule 1A has the condition x0 <= 32.5, while the rule 3B has the condition x0 > 39, making them incompatible.

A. Argente-Garrido, C. Zuheros, M.V. Luzón, F. Herrera, An interpretable client decision tree aggregation process for federated learning, Information Sciences, Volume 694, 2025

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Federated Decision trees



The metrics used for determining the variable to be used in the decision node is the variance.

$$\mathrm{Var}(y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

where:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\mathrm{VarReduction} = \mathrm{Var}(S) - \left(\frac{n_L}{n}\mathrm{Var}(S_L) + \frac{n_R}{n}\mathrm{Var}(S_R)\right)$$

José Luis Corcuera Bárcena, Pietro Ducange, Francesco Marcelloni, Alessandro Renda, Increasing trust in AI through privacy preservation and model explainability: Federated Learning of Fuzzy Regression Trees, Information Fusion, Volume 113, 2025,

23

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Federated Decision trees

Observation: the variance can be computed as follows

$$\mathrm{Var}(y) = \frac{1}{n}\sum y_i^2 - \left(\frac{1}{n}\sum y_i\right)^2$$

For each variable, we compute the first and second sum in each node.

Then, each node transmits the two sums to the server which compute the Var(y) and the VarReduction

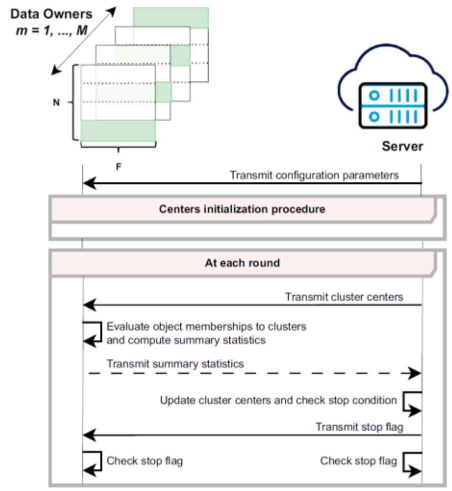
José Luis Corcuera Bárcena, Pietro Ducange, Francesco Marcelloni, Alessandro Renda, Increasing trust in AI through privacy preservation and model explainability: Federated Learning of Fuzzy Regression Trees, Information Fusion, Volume 113, 2025,

24

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Horizontal federated c-means (LLF-CM)



J. L. C. Bárcena, F. Marcelloni, A. Renda, A. Bechini and P. Ducange, "Federated c -Means and Fuzzy c -Means Clustering Algorithms for Horizontally and Vertically Partitioned Data," in IEEE Transactions on Artificial Intelligence, vol. 5, no. 12, pp. 6426-6441, Dec. 2024, doi: 10.1109/TAI.2024.3426408.

25

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Horizontal federated c-means (LLF-CM)

- Let $\mathbf{q}^{(t),m} = \left[q_1^{(t),m}, q_2^{(t),m}, \ldots, q_{N_m}^{(t),m}\right], \; q_j^{(t),m} \in \{1, \ldots, C\}$ be

  the vector that indicates the cluster for each object in dataset P^m

- Each data owner P^m assigns each object in the local dataset to the cluster with the nearest center.

26

Horizontal federated c-means (LLF-CM)

Algorithm 1: Horizontal Federated c-means (LLF-CM).
C: number of clusters, $\varepsilon > 0$: tolerance value for the stop condition. T: maximum number of rounds

Initialization stage

Server:

stop_flag = FALSE

Initialization procedure for C cluster centers $\mathbf{V}^{(0)} = \left\{ \mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_C^{(0)} \right\}$

Execution stage

At each round t, with t starting from 0:

Cluster assignment

Server:

Transmit $\mathbf{V}^{(t)}$ to each data owner

Each data owner P^m:

Evaluate $q_j^{(t),m} \in \{1, \ldots, C\}$ for each object j, as its own cluster assignment according to the nearest center $\mathbf{v}_c^{(t)}$

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Horizontal federated c-means (LLF-CM)

Centers update

Each data owner P^m:

for each cluster Γ_c **do**

$n_c^{(t),m} \leftarrow$ count of the number of objects in cluster Γ_c

if $n_c^{(t),m} > 1$ **then**

compute $\mathbf{Ls}_c^{(t),m}$ as per Eq. (1)

else

$(\mathbf{Ls}_c^{(t),m}, n_c^{(t),m}) \leftarrow (\mathbf{0}, 0)$

end if

end for

Transmit to the server all the pairs $(\mathbf{Ls}_c^{(t),m}, n_c^{(t),m})$ calculated above

Server:

Update cluster centers evaluating $\mathbf{V}^{(t+1)}$ as per Eq. (2)

$$\mathbf{Ls}_c^{(t),m} = \sum_{\mathbf{x}_j^m \in \Gamma_c} \mathbf{x}_j^m \quad (1)$$

$$\mathbf{v}_c^{(t+1)} = \frac{\sum_{m=1}^{M} \mathbf{Ls}_c^{(t),m}}{\sum_{m=1}^{M} n_c^{(t),m}}, \quad \forall\, c \in \{1, \ldots, C\} \quad (2)$$

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Horizontal federated c-means (LLF-CM)

Termination

Server:

16: **if** NOT $\left(\|\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}\|_F < \varepsilon \quad \text{OR} \quad t > T\right)$ **then**

17:     *stop_flag* = TRUE

18: **end if**

19: Transmit *stop_flag* to each data owner

Each data owner P^m & Server:

20: **if** NOT *stop_flag* **then**

21:     Proceed with the next round (line 4)

22: **end if**

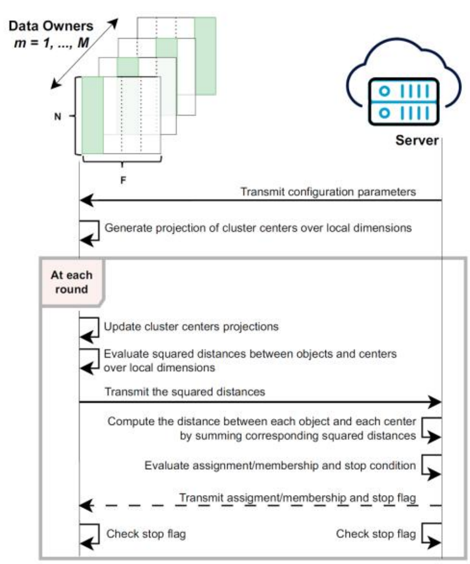
23: ⟨Termination⟩

29

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Vertical federated c-means (LLF-CM)



30

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Vertical federated c-means (LLF-CM)

Initialization stage

Server:

1: Transmit the number of clusters C to each data owner

2: *stop_flag* = FALSE

Each data owner P^m*:*

3: Randomly generate the projections of the cluster centers over the features defined on P^m: $\mathbf{V}_c^{(0),m} = \left\{\mathbf{v}_1^{(0),m}, \mathbf{v}_2^{(0),m}, \ldots, \mathbf{v}_C^{(0),m}\right\}$

31

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Vertical federated c-means (LLF-CM)

Execution stage

4: At each round t, with t starting from 0:

Centers update

Each data owner P^m*:*

5: **if** NOT *stop_flag* AND $t > 0$ **then** 5

6: Evaluate $\mathbf{v}_c^{(t),m}$ for each cluster Γ_c

7: **end if**

$$\mathbf{v}_c^{(t),m} = \frac{\sum_{\mathbf{x}_j^m \in \Gamma_c} \mathbf{x}_j^m}{n_c^{(t-1)}} \quad \forall\, c \in \{1, \ldots, C\}$$

32

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Vertical federated c-means (LLF-CM)

Cluster assignment

Each data owner P^m:

8: Evaluate $d_{j,c}^{(t),m} = \sum_{f=1}^{F^m} \left(x_{j,f}^m - v_{c,f}^{(t),m} \right)^2$ for each object j and cluster Γ_c

9: Transmit the $N \times C$ matrix $\mathbf{D}^{(t),m}$ to the server

Server:

10: Evaluate $d_{j,c}^{(t)} = \sqrt{\sum_{m=1}^{M} d_{j,c}^{(t),m}}$ for each object j and cluster Γ_c

11: Evaluate assignment/membership for each object j and cluster Γ_c i.e., $q_j^{(t)}$ and $n_c^{(t)}$

12: **if** $t \geq 1$ AND $\left(||\mathbf{D}^{(t)} - \mathbf{D}^{(t-1)}||_F < \varepsilon \text{ OR } t > T \right)$ **then**

13: *stop_flag* = TRUE

14: **end if**

15: Transmit object assignment/membership to clusters and *stop_flag* to each data owner

33

DIPARTIMENTO DI INGEGNERIA

Francesco Marcelloni

Vertical federated c-means (LLF-CM)

Termination

Each data owner P^m & Server:

16: **if** NOT *stop_flag* **then**

17: Proceed with the next round (line 5)

18: **end if**

19: ⟨Termination⟩

34