



# IDEAL 2021

22nd International Conference on  
Intelligent Data Engineering and  
Automated Learning (IDEAL)

25-27 November 2021, Manchester, UK



UNIVERSIDAD  
DE GRANADA

## Federated Learning for Preserving Data Privacy



### Francisco Herrera

Andalusian Research Institute in Data Science and Computational Intelligence

University of Granada - Spain

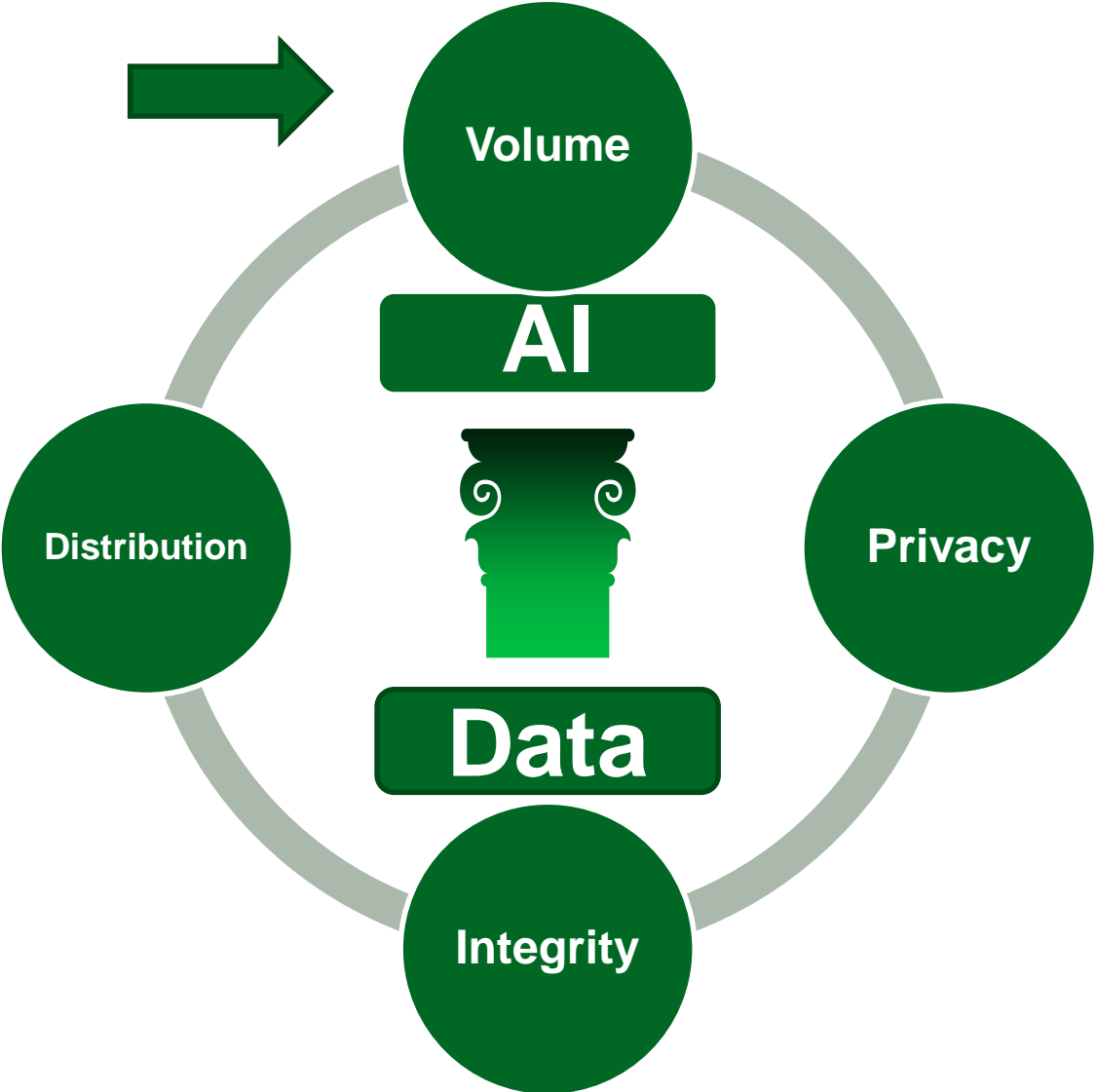
[herrera@decsai.ugr.es](mailto:herrera@decsai.ugr.es)



## Outline

- Artificial intelligence challenges
- Federated Learning
  - Definition
  - Key elements
  - Categories
  - Architecture (client-server; peer-to-peer)
- Federated Learning workflow
- Federated Learning libraries
- Case of study
- Adversarial attacks. Proposal and Case of study
- Concluding remarks. What's next?

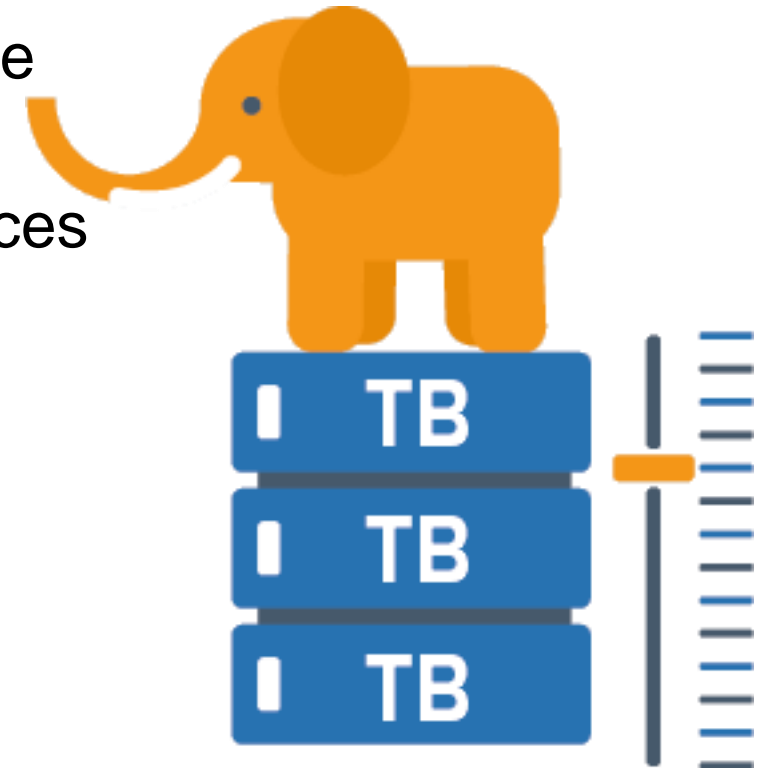
# Artificial Intelligence Challenges



# Artificial Intelligence Challenges: Volume of data

- Current AI models, and specially deep learning models, are fuelled by **vast amount of data**.
- Although fine tuning reduces the amount of training data, we still **need pre-trained models on big datasets**.
- For instance, the amount of data and computational resources of some current language models for NLP are:

Model	Data	Tokens	Parameters
BERT-base	13GB	250 billions	110M
GPT-3	570GB	300 billions	175M



# Artificial Intelligence Challenges: Volume of data, different sources

## Case of use

- Data with sensitive information, as: emails, personalised recommendations or health records. The data must be kept in their corresponding data owner silo.
- Data is stored in data silos, as the data stored by the healthcare industry.
- Data protected by legal regulations, as data from banks, telecom or hospitals that cannot be shared.

# Artificial Intelligence Challenges: Volume of data, different sources

## Case of use

- **Data with sensitive information**, as: emails, personalised recommendations or health records. The data must be kept in their corresponding data owner silo.
- **Data is stored in data silos**, as the data stored by the healthcare industry.
- **Data protected by legal regulations**, as data from banks, telecom or hospitals that cannot be shared.

**These use cases would benefit from learning models from data silos, sharing models instead of data**

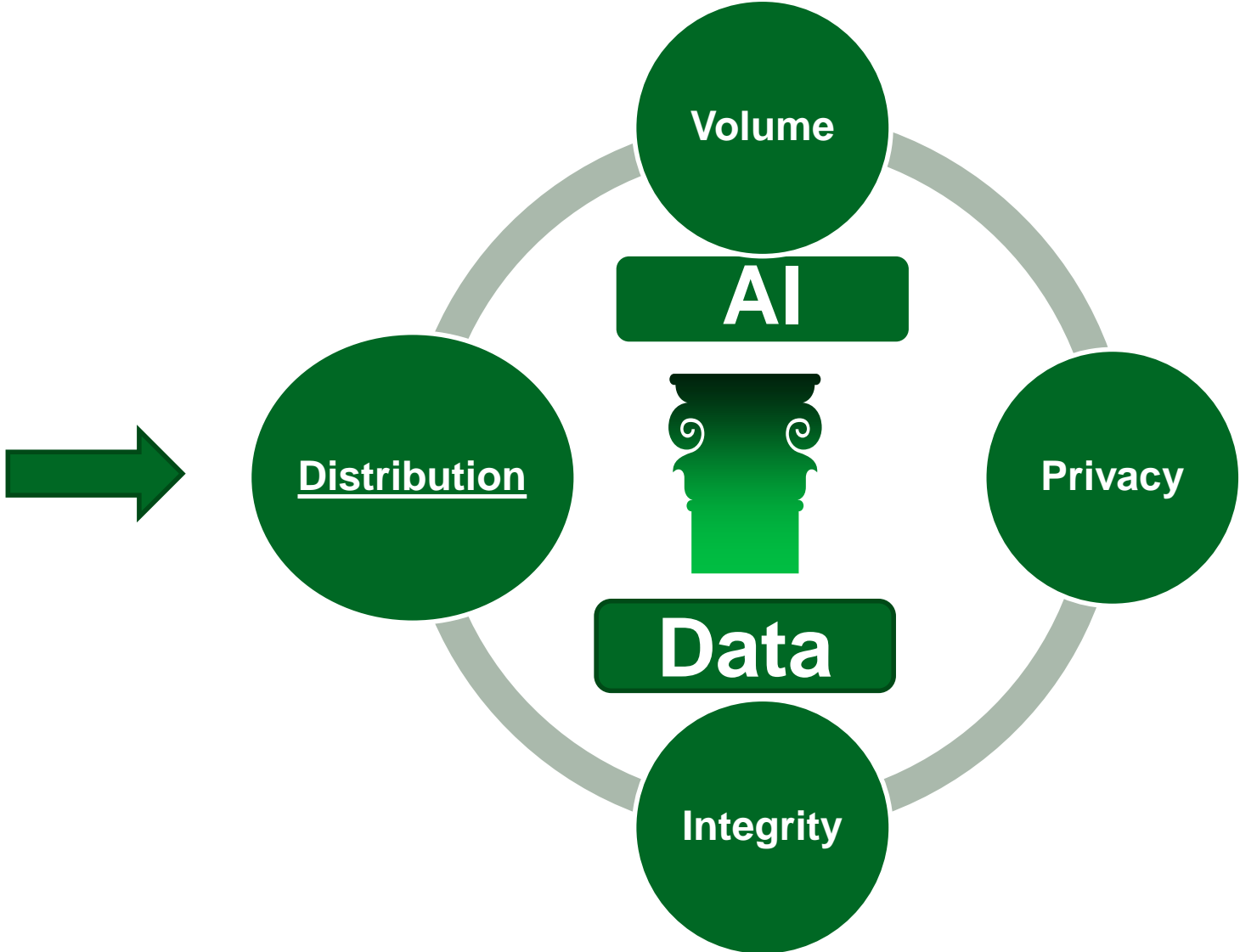
# Artificial Intelligence Challenges: Volume of data, different sources

- Possible solution: **Distributed Machine Learning**

- **Weaknesses:**

- **Communication costs.** That huge amount of data must be communicated among the nodes of the distributed machine learning setting.
- **Latency time.** The latency time proportionally increase to the amount of data.
- **Storage capacity.** The storage of so huge amount of data requires the use of so much storage nodes, which makes wider the communication costs and latency time.
- **Computation bottleneck.** In those situations where the communication costs may be afforded, the bottleneck of the setting is on the computation time.

# Artificial Intelligence Challenges





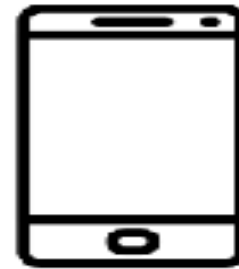
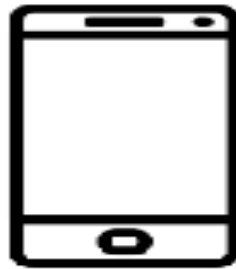
# Artificial Intelligence Challenges: Data Distribution

Independent and identically distributed random variables (IID)

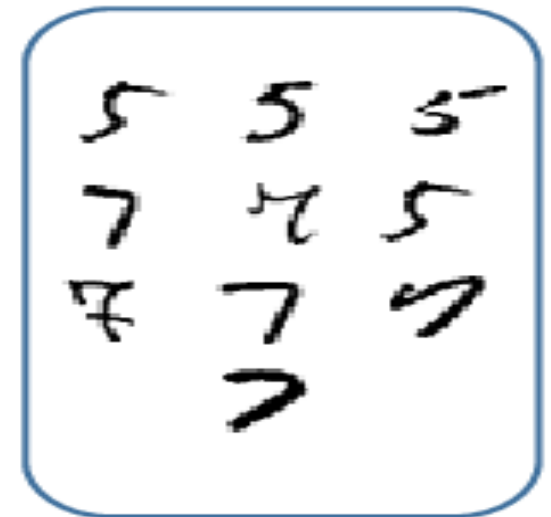
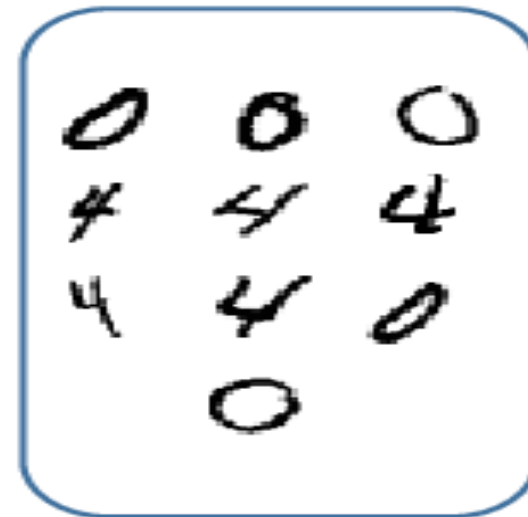
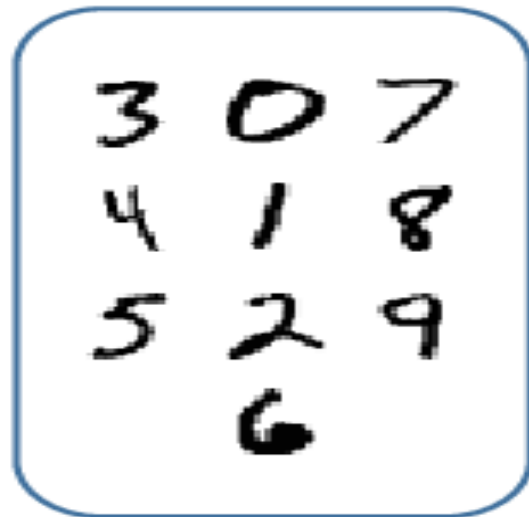
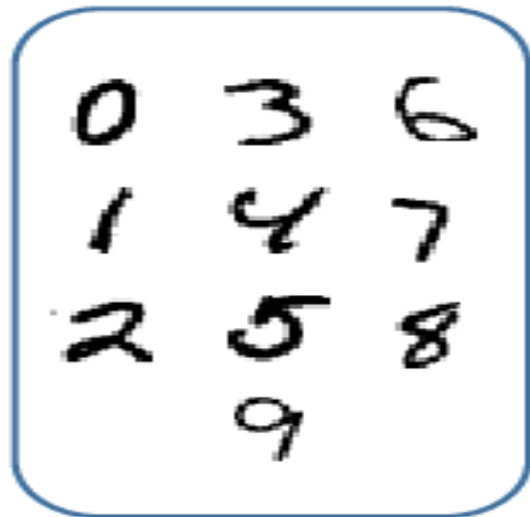
Non-Independent and identically distributed random variables (Non-IID)

IID dataset

Non-IID dataset



↕ same probability distribution



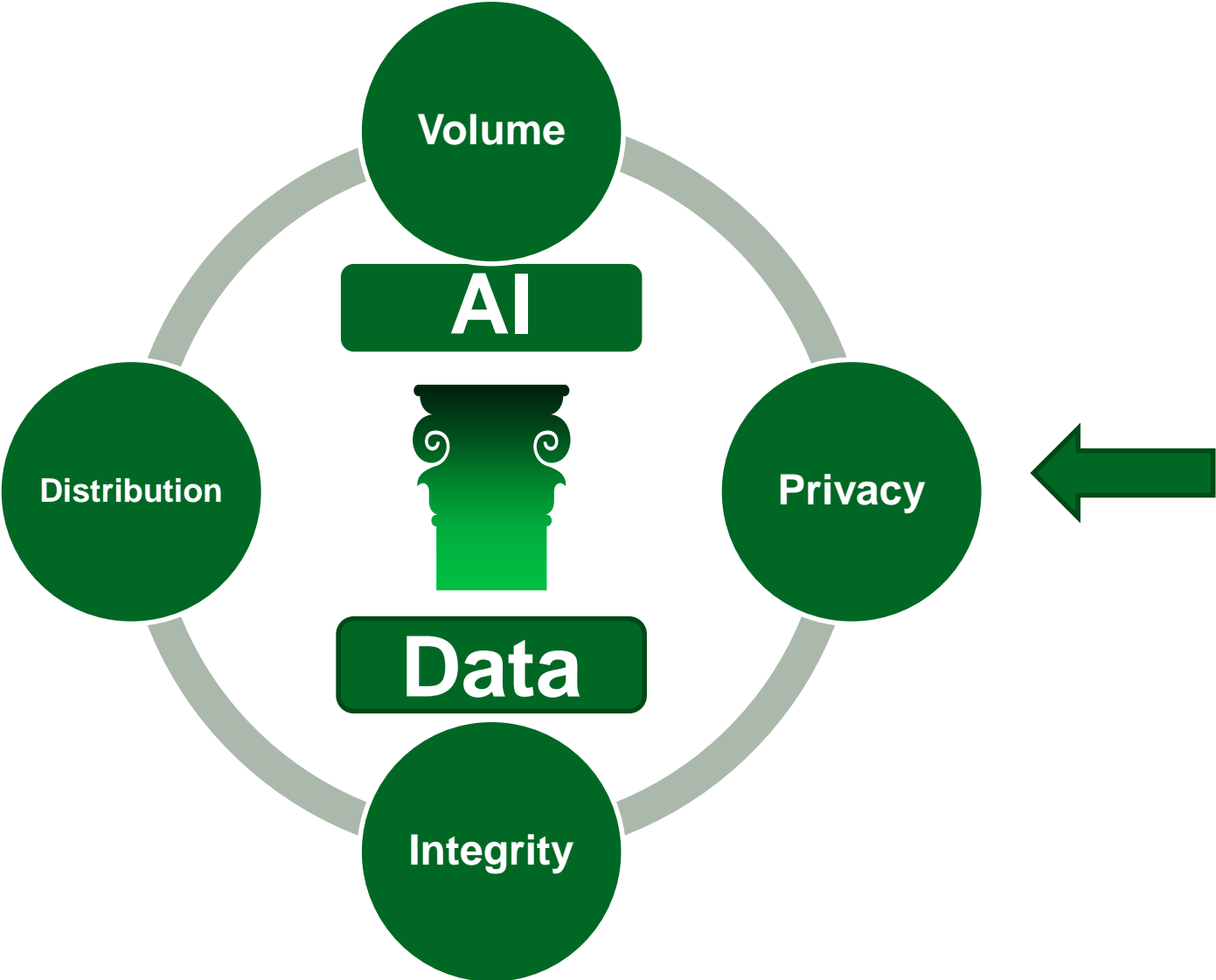
# Artificial Intelligence Challenges: Data Distribution

## Non-Independent and identically distributed random variables (Non-IID)

- In a distributed scenario, we can set an IID distribution if we have access to the data.
- If the data is distributed in several data silos or devices (clients), we cannot make the IID assumption [2,3].
- If we have a set of clients, let me consider:
  - $Q$ : Distribution over the set of clients that own a set of data points (dataset).
  - $P_i(x, y)$ : Local data distribution from the client  $i$ .
- If we consider each dataset (silo or device) as a random variable, we say that they follow a **Non-IID if they do not follow the same distribution**. Or in other words, if there are differences between  $P_i$  and  $P_j$  for different clients  $i$  and  $j$  ( $P_i \neq P_j$ ).

- [2] Sattler, F., Wiedemann, S., Müller, K. R., & Samek, W. (2019). Robust and communication-efficient federated learning from non-iid data. IEEE transactions on neural networks and learning systems, 31(9), 3400-3413.
- [3] Hsieh, K., Phanishayee, A., Mutlu, O. & Gibbons, P.. (2020). The Non-IID Data Quagmire of Decentralized Machine Learning. Proceedings of the 37th International Conference on Machine Learning, in Proceedings of Machine Learning Research 119:4387-4398

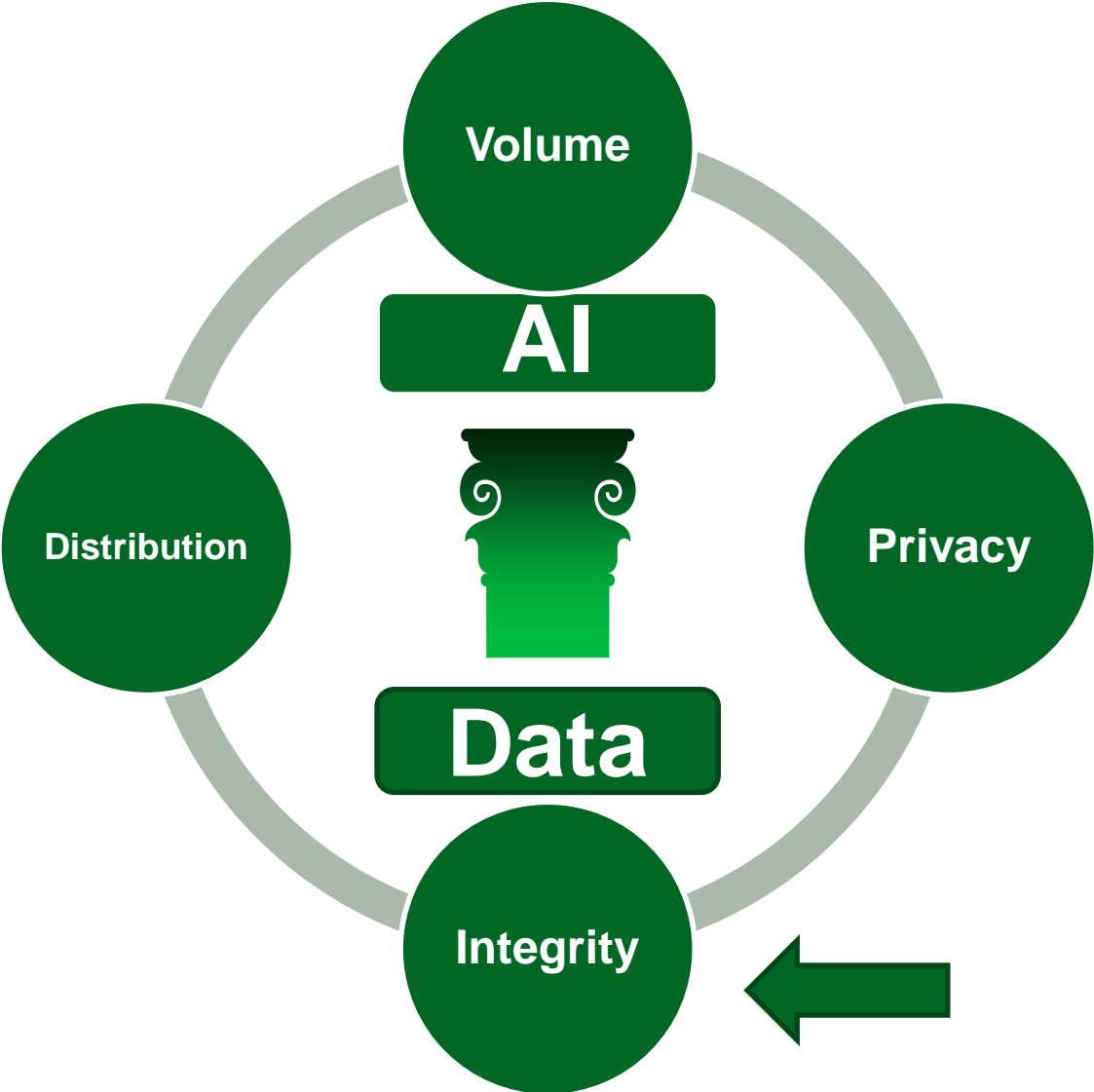
# Artificial Intelligence Challenges



# Artificial Intelligence Challenges: Privacy

- Currently, people are increasingly aware about the relevance of **preserving the privacy and ownership of their personal data**.
- People are demanding more AI-based services, but at the same time they are reluctant to share their personal data.
- Likewise, several legal frames are coming up to regulate how data should be managed, shared and used, for instance the GDPR regulation of the EU.
- These new legislative landscape and the incessant growing concern on **preserving data privacy make more difficult to develop distributed machine learning settings**, which may hinder the progress of data-based AI.

# Artificial Intelligence Challenges



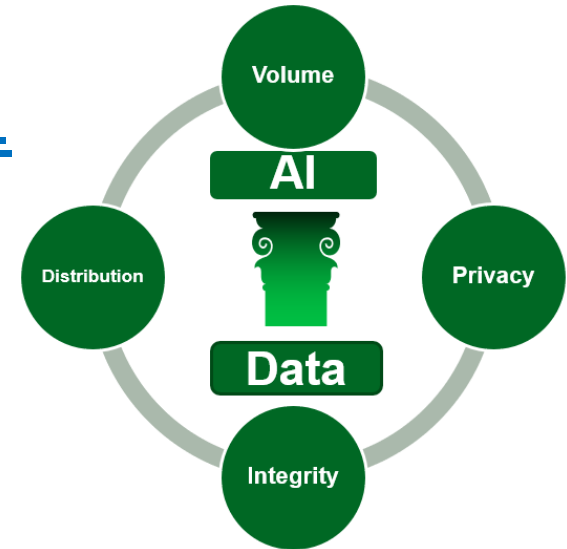
# Artificial Intelligence Challenges: Data Integrity

- **The standard distributed machine learning requires to transfer data among nodes.**
  - This exposes the data to be used by malicious agents to corrupt the learning model by corrupting them.
  - Likewise, the transferring of data may result in data leaks.
  - **Current AI recommendations as Trustworthy AI [4] asserts that AI systems have to preserve data from illegitimate access.**
- [4] European Commission. High-Level Expert Group on Artificial Intelligence: Ethics guidelines for trustworthy AI (2019).

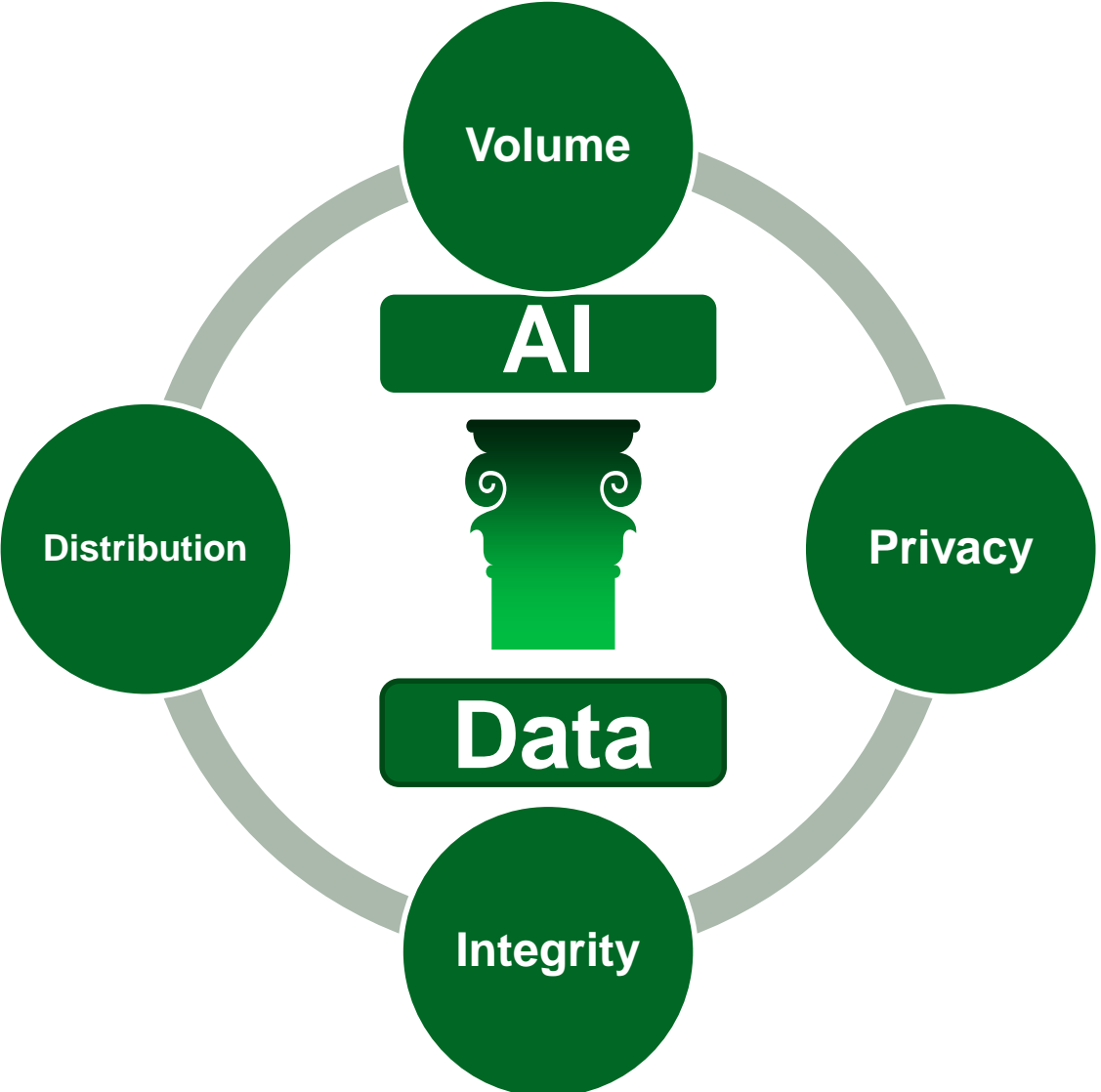
# Artificial Intelligence Challenges

The standard AI-based systems or machine learning settings needs to:

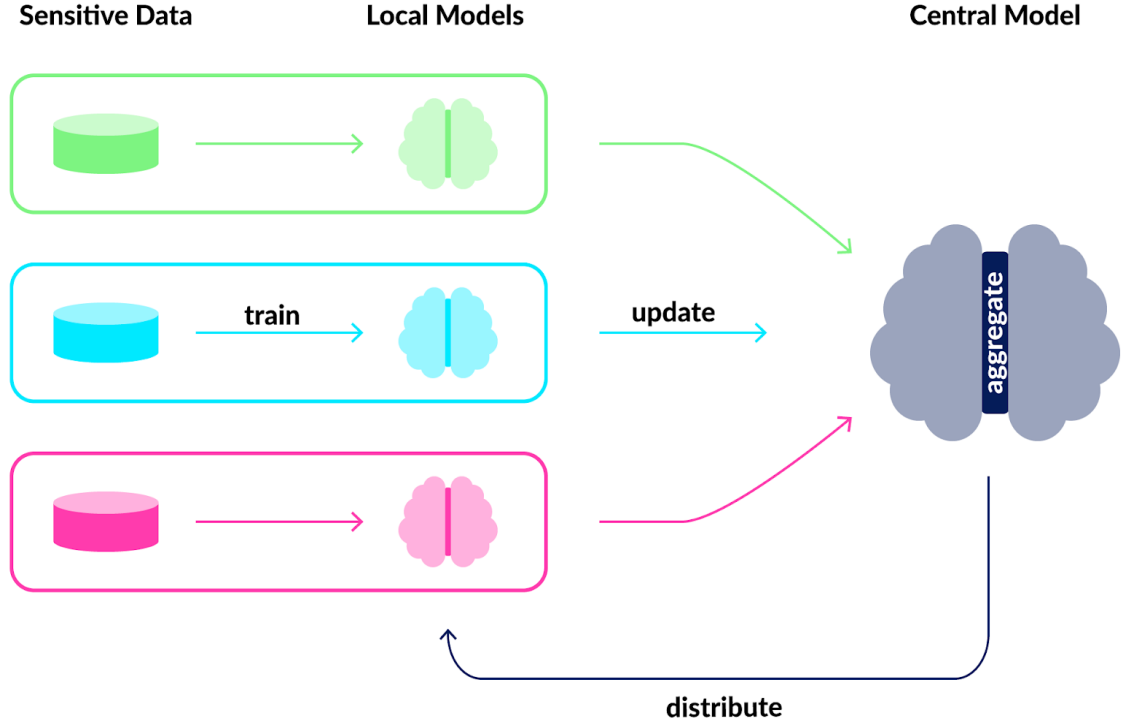
1. Work with a vast amount of distributed sensitive data stored.
2. Work with data that follows a Non-IID distribution.
3. Preserve data privacy.
4. Protect the integrity of data and the learning model from malicious attacks.



# Artificial Intelligence Challenges



# Federated Learning





## Outline

- Artificial intelligence challenges
- Federated Learning
  - Definition
  - Key elements
  - Categories
  - Architecture (client-server; peer-to-peer)
- Federated Learning workflow
- Federated Learning libraries
- Case of study
- Adversarial attacks. Proposal and Case of study
- Concluding remarks. What's next?

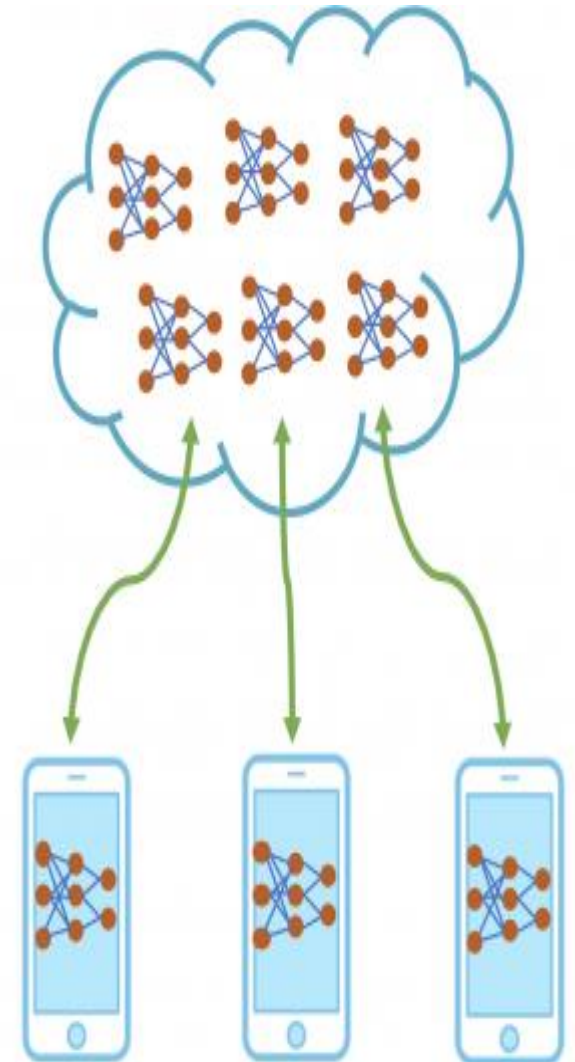
# Federated Learning. Definition

**Federated Learning (FL)** is a machine learning setting where **multiple entities (clients)** collaborate in solving a machine learning problem, under the **coordination of a central server** or service provider.

Each client's **raw data is stored locally** and not exchanged or transferred; instead **local learning** focused updates intended for **immediate aggregation** are used to achieve the learning objective [5].

- **Key points:**

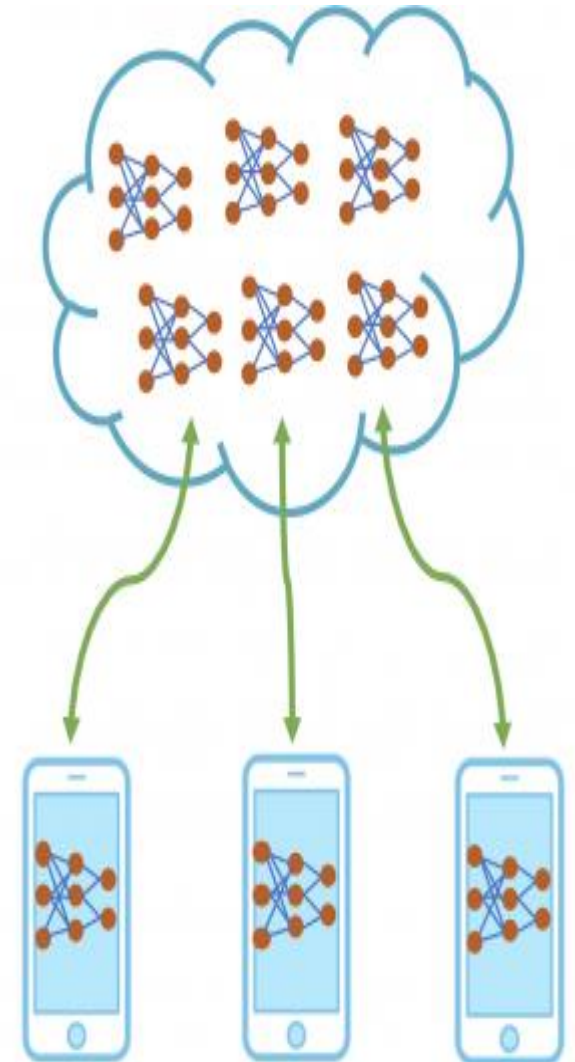
- **Raw data is stored locally** → Its privacy is protected.
- **Raw data is not transferred** → It prevents data leaks.
- **Learning local models, aggregation of local models to get a global model**



[5] Kairouz, P. and McMahan, H. B. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1), 2021.

# Federated Learning. Definition

- In FL, there are two kind of nodes or clients [6]:
  - Data owners nodes  $\{C_1, C_2 \dots, C_n\}$ : They own the raw data, *i.e.* the local datasets  $\{D_1, D_2 \dots, D_n\}$ .
  - Aggregation nodes  $\{G_1, G_2 \dots, G_n\}$ : They aggregate the **local learning models** to learn a global learning model from the data kept in the data owners node.
- **FL aim.** Learning a global learning model  $(\theta, GLM)$  based on the aggregation of the local learning models  $(\theta_i, LLM_i)$  through several rounds of learning.



[6] Rodríguez-Barroso, N., Stipcich, G., Jiménez-López, D., Ruiz-Millán, J. A., Martínez-Cámara, E., ... & Herrera, F. (2020). Federated Learning and Differential Privacy: Software tools analysis, the Sherpa.ai FL framework and methodological guidelines for preserving data privacy. *Information Fusion*, 64, 270-292.

# Federated Learning. Definition – Rounds of Learning

- It is an iterative learning process composed of the following steps:

1. Each client  $i$  trains its  $LLM_i$  on its local training data  $D_i^t$  and it updates the parameters of the  $LLM_i$ ,  $\theta_i^t$ .

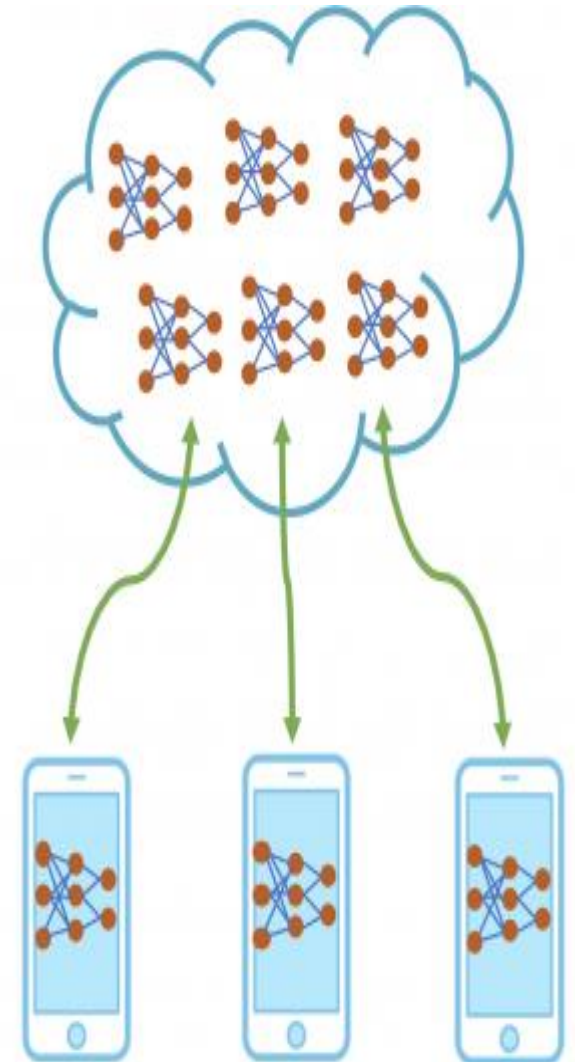
2. The clients send their updated parameters  $\theta_i^t$  to the server.

3. The server computes the global parameters  $\theta^t$  by aggregating the local parameters  $\{\theta_1^t, \theta_2^t, \dots, \theta_n^t\}$  of all the  $LLM_i$  using an specific federated aggregation operator  $\Delta$ , formally

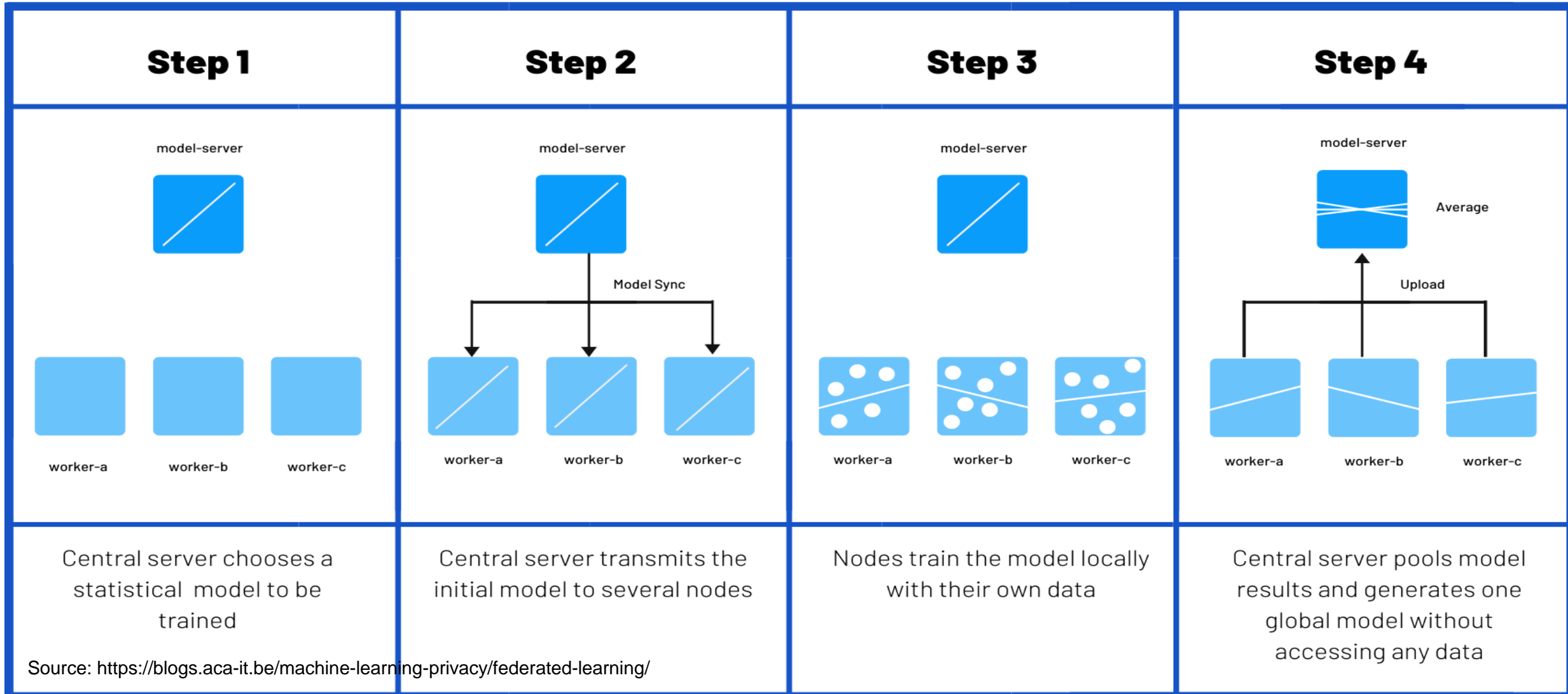
$$\theta^t = \Delta(\theta_1^t, \theta_2^t, \dots, \theta_n^t).$$

4. After the aggregation, the  $LLM$  are updated with the aggregated parameters:

$$\theta_i^{t+1} \leftarrow \theta_i^t, \forall i \in \{1, \dots, n\}$$



# Federated Learning. Definition – Rounds of Learning



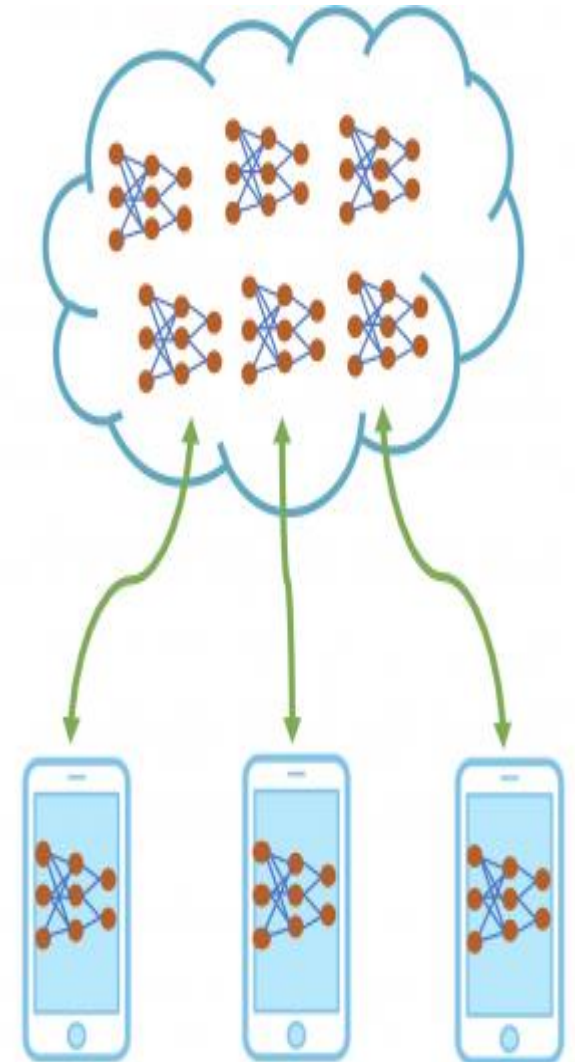
# Federated Learning. Definition – Learning goal

- The rounds of learning are repeated until reach the learning goal.
- The learning goal is to minimise the following objective function:

$$\min_{\theta} F(\theta), \text{ with } F(\theta) := \sum_{i=1}^n w_i F_i(\theta)$$

Where:

- $n$  is the number of nodes or clients.
- $F_i$  is the local objective function for the  $i$ -th client fitted to each client's data.
- $w_i \geq 0$ ;  $\sum_i w_i = 1$ .



# Federated Learning. Key Elements

- FL is a specific machine learning setting, which incorporates some elements to the standard workflow. In particular:
- **Data.** It plays a central role in FL as in machine learning. The **data stored in each client can follow a IID or a Non-IID** distribution. In real FL scenarios, the **Non-IID** distribution is the most likely.
- **Learning model.** It is composed of:
  - The **Local Learning Models** (LLM), which are locally trained in each client.
  - The **Global Learning Model** (GLM), which is obtained by aggregating the parameters of LLMs. Hence, the GLM is not trained.

# Federated Learning. Key Elements

- **Federated Aggregation Operator**. It is the responsible of aggregating the parameters of the LLM in the server. It has to match the following requirements:
  1. To assure a **proper fusion** of the LLM to optimise the objective function.
  2. To facilitate a **fast convergence** of the objective function for reducing the number of communication rounds among the clients and the server.
  3. To be **resilient and robust against clients with poor data quality and adversarial clients** for preserving data integrity.



# Federated Learning. Key Elements

## Federated Averaging (FedAVG) [7]

- So far, there are several aggregation operators to optimise the objective function of the GLM.
- The first one and the most used is FedAVG, which is designed to optimise non-convex objective functions commonly seen when training deep learning methods.

- [7] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication- Efficient Learning of Deep Networks from Decentralized Data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54, 2017, pp. 1273–1282.
- Image source [5].

## Server executes:

initialize  $x_0$

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

$S_t \leftarrow$  (random set of  $M$  clients)

**for** each client  $i \in S_t$  **in parallel do**

$x_{t+1}^i \leftarrow \text{ClientUpdate}(i, x_t)$

$x_{t+1} \leftarrow \sum_{k=1}^M \frac{1}{M} x_{t+1}^k$

## ClientUpdate( $i, x$ ):

**for** local step  $j = 1, \dots, K$  **do**

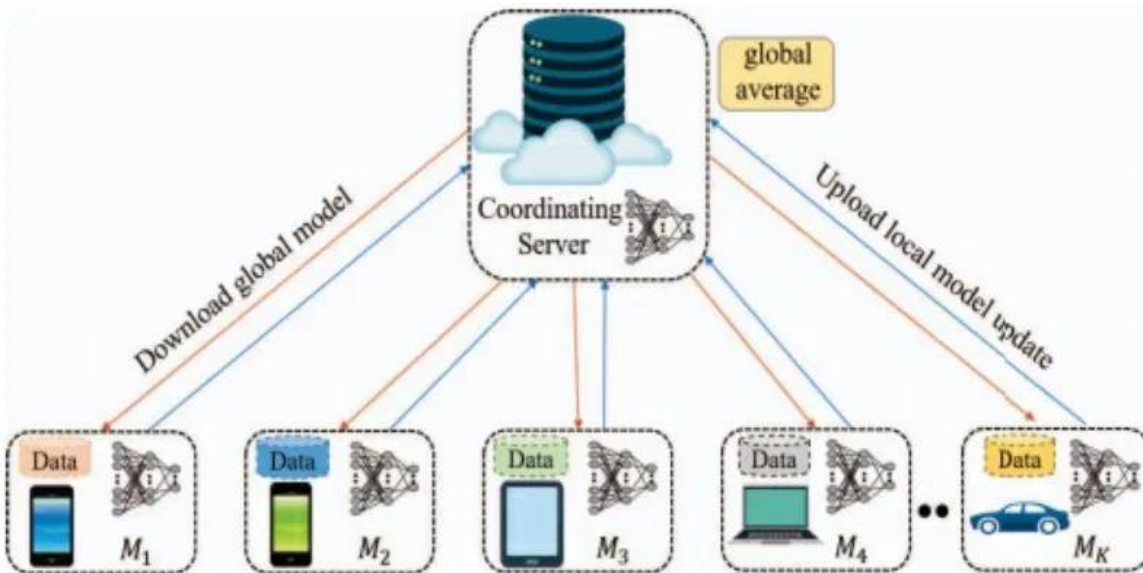
$x \leftarrow x - \eta \nabla f(x; z)$  for  $z \sim \mathcal{P}_i$

**return**  $x$  to server

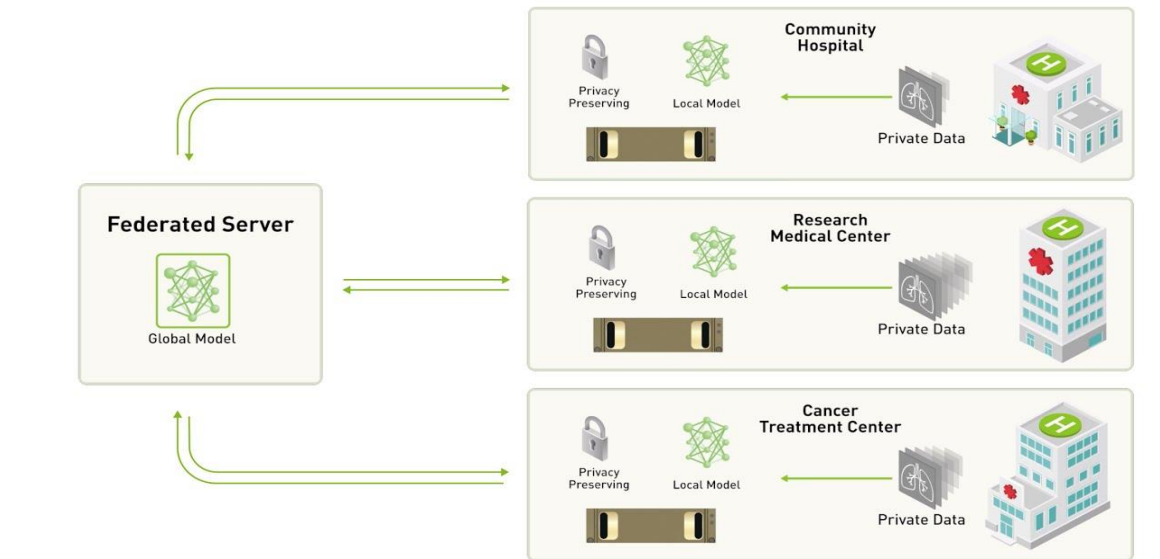
# Federated Learning. Key Elements

- **Clients.** They represent the nodes in a FL setting.
- The clients may be: IoT devices, mobile phones, self-driven cars, organisations. Their main characteristic is that they own the data, and the data is not shared with any other node of the FL setting.
- Depending on the nature of the clients, we have:

## Cross-device FL



## Cross-silo FL



# Federated Learning. Key Elements

- **Federated server**. It orchestrates the iterative learning process.

The server participates in:

1. Receiving the trained parameters of the LLM.
  2. Aggregating the trained parameters of each client model using the federated aggregation operator.
  3. Updating all the learning model with the aggregated parameters.
- The server stores the GLM, which represents the aggregation of all the LLM, and hence the final model after the learning process.
  - The GLM is used for predicting, but the prediction is done in each client.

# Federated Learning. Key Elements

- **Communication**

- It is crucial in FL, since FL is a distributed learning process.
- It is one of the weakest element of FL, since the model parameters are shared with the sever.
- **If the parameters are not protected**, an malicious agent can use them to reconstruct the data of the clients by means **model-inversion adversarial attacks** [8].
- **Strategies to mitigate the communication risks:**
  - Differential privacy techniques to obfuscate the model parameters.
  - Secure Multi-party Computation.
  - Homomorphic Encryption.

## Outline

- Artificial intelligence challenges
- Federated Learning
  - Definition
  - **Key elements**
  - **Categories**
  - Architecture (client-server; peer-to-peer)
- Federated Learning workflow
- Federated Learning libraries
- Case of study
- Adversarial attacks. Proposal and Case of study
- Concluding remarks. What's next?

**Data.**  
**Learning model**  
**Federated Aggregation Operator**  
**Clients**  
**Federated server**  
**Communication**

# Federated Learning. Categories

- The FL setting extremely depends on the data distribution.
- Let consider:
  - $D_i$ : the data of the client  $i$ .
  - $X$ : it represents the feature space.
  - $Y$ : it represents the label space.
  - $I$ : it represents the sample ID space.
  - $(I, X, Y)$ : it represents the dataset of the client  $i$ .
- Depending on how data is portioned among the clients, there are 3 categories of FL:
  - **Horizontal FL**
  - **Vertical FL**
  - **Transfer FL**

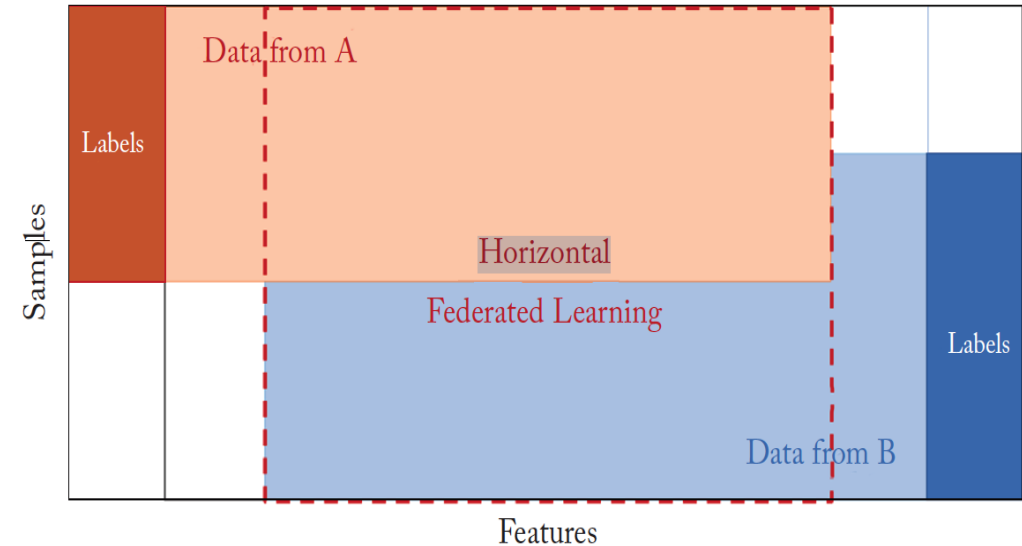
# Federated Learning. Categories – Horizontal FL

- The feature space ( $X$ ) is shared among the clients.
- The label space ( $Y$ ) is shared among the clients, although there may be some differences in the label distribution (prior probability shift or *unbalancedness*).
- The example or ID space ( $I$ ) is not shared, although it is possible to exist some overlap.

- Formally:

$$X_i = X_j, Y_i = Y_j, I_i \neq I_j, \forall D_i, D_j, i \neq j$$

- **Example:** Two banks that serve two different regional markets, **they may share some users but their data have the same or very similar features and labels.** Hence, they can collaboratively build a machine learning model through horizontal FL.



Source: [9]

[9] Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., & Yu, H. (2019). Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3), 1-207.

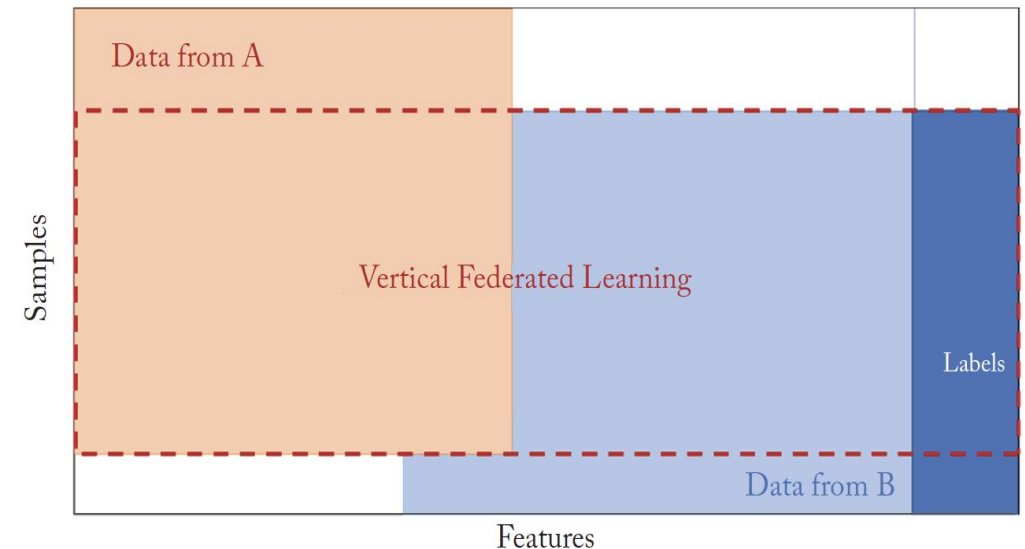
# Federated Learning. Categories – Vertical FL

- The feature space ( $X$ ) is not shared among the clients.
- The label space ( $Y$ ) is not shared among the clients.
- The example or ID space ( $I$ ) is shared.

- Formally:

$$X_i \neq X_j, Y_i \neq Y_j, I_i = I_j, \forall D_i, D_j, i \neq j$$

- **Example:** When two companies provide different services but **they share a large amount of users** (telco and insurance companies). They may collaborate in building a machine learning model through a Vertical FL setting (**different features and labels**).



Source: [9]

[9] Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., & Yu, H. (2019). Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3), 1-207.



# Federated Learning. Categories – Transfer FL

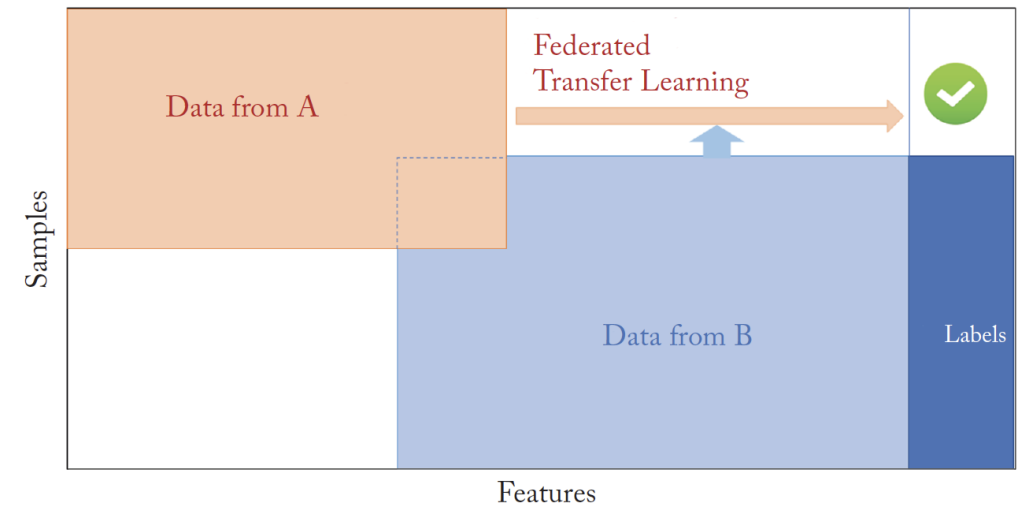
- The feature space ( $X$ ) is not shared among the clients.
- The label space ( $Y$ ) is not shared among the clients.
- The example or ID space ( $I$ ) is not shared.

- Formally:

$$X_i \neq X_j, Y_i \neq Y_j, I_i \neq I_j, \forall D_i, D_j, i \neq j$$

- Example:

- Those scenarios where the **mismatch among the feature, label and instance spaces are large.**
- Different organisations may collaboratively build machine learning models, where each of them may be benefited from the data of the other party.
- **It resembles the standard transfer learning scenario.**



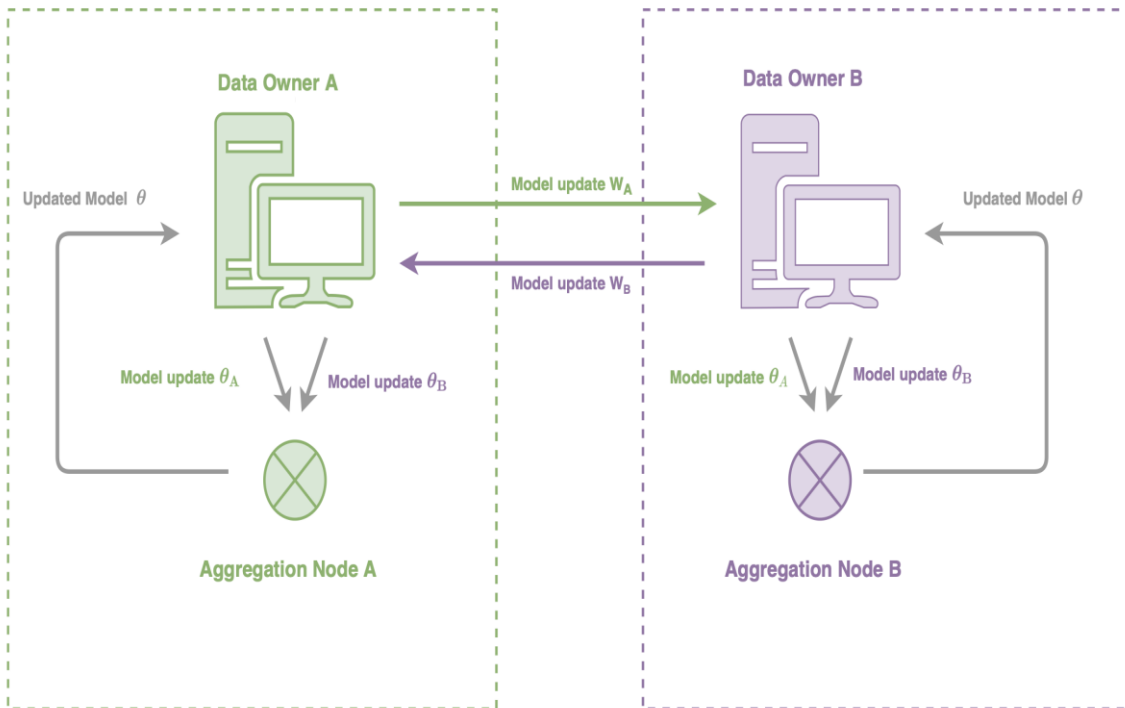
Source: [9]

[9] Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., & Yu, H. (2019). Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 13(3), 1-207.

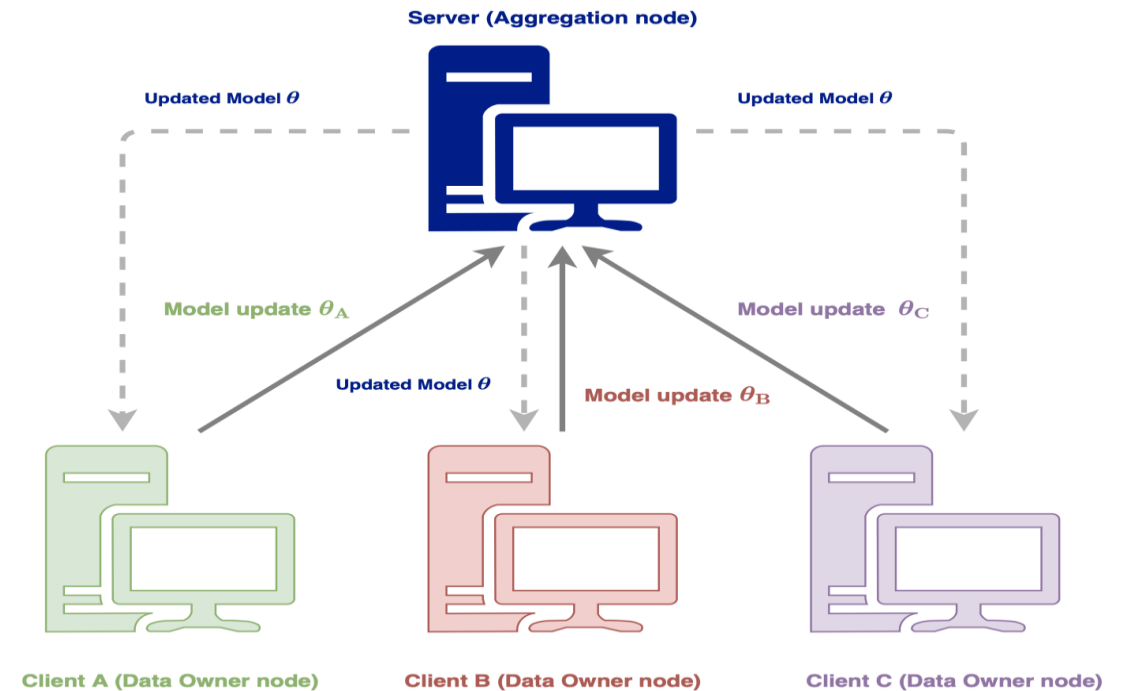
# Federated Learning. Architecture

- **FL is a distributed setting**, so it can be deployed as other distributed architectures:
  - Peer-to-peer: All the nodes can be data owners and aggregation nodes.
    - It is more common when there is not any agent with the enough reliability to be a server.
  - Client-server: There is a difference among data owners (clients) and aggregation nodes (server).

## Peer-to-Peer



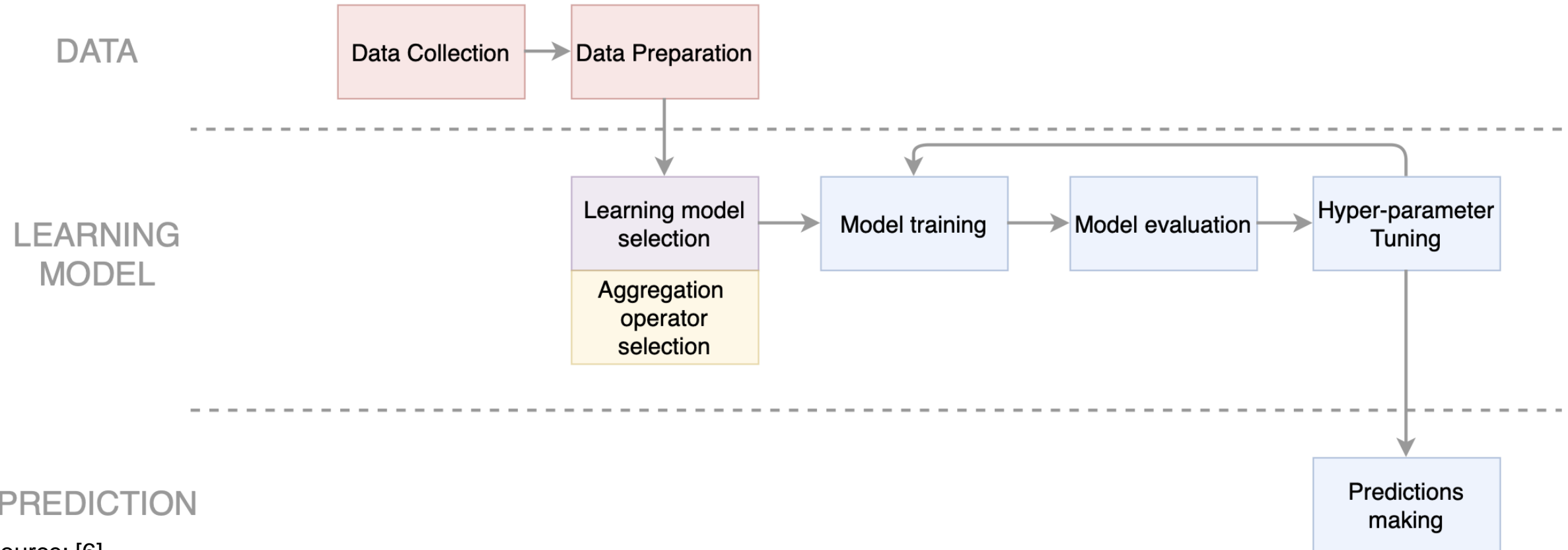
## Client-Server



## Outline

- Artificial intelligence challenges
- Federated Learning
  - Definition
  - Key elements
  - Categories
  - Architecture (client-server; peer-to-peer)
- **Federated Learning workflow**
- **Federated Learning libraries**
- Case of study
- Adversarial attacks
  - Taxonomy
  - FL-IOWA-DQ
  - On-going

# Federated Learning workflow



Source: [6]

Rodríguez-Barroso, N., Stipcich, G., Jiménez-López, D., Ruiz-Millán, J. A., Martínez-Cámara, E., ... & Herrera, F. (2020). Federated Learning and Differential Privacy: Software tools analysis, the Sherpa.ai FL framework and methodological guidelines for preserving data privacy. *Information Fusion*, 64, 270-292.

# Federated Learning Libraries

- **Sherpa.ai:** <https://github.com/sherpaai/Sherpa.ai-Federated-Learning-Framework>  
(The Sherpa.ai Federated Learning and Differential Privacy Framework is a project by [Sherpa.ai](https://sherpa.ai), in collaboration with the [Andalusian Research Institute in Data Science and Computational Intelligence \(DaSCI\)](https://www.dasci.es/) research group from the [University of Granada](https://www.unigra.es/).)
- **Tensorflow:** <https://www.tensorflow.org/federated?hl=es-419>
- **PySyft:** <https://github.com/OpenMined/PySyft>
- **Fate:** <https://fate.fedai.org>
- **Paddle Federated Learning:** <https://github.com/PaddlePaddle/PaddleFL>

**sherpa.ai**



**FATE**



**PFL**  
Paddle Federated Learning



UNIVERSIDAD  
DE GRANADA



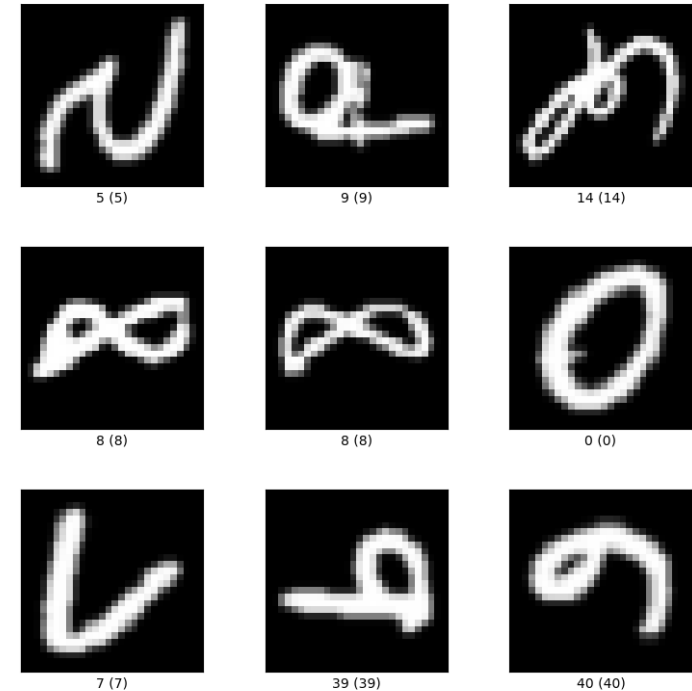
# Outline

- Artificial intelligence challenges
- Federated Learning
  - Definition
  - Key elements
  - Categories
  - Architecture (client-server; peer-to-peer)
- Federated Learning workflow
- Federated Learning libraries
- **Case of study: Image classification with Sherpa.ai (EMNIST data set)**
- Adversarial attacks. Proposal and Case of study
- Concluding remarks. What's next?

# Case of Study: Image classification with Sherpa.ai

- Dataset: EMNIST. The federated and extended version of the MNIST dataset.

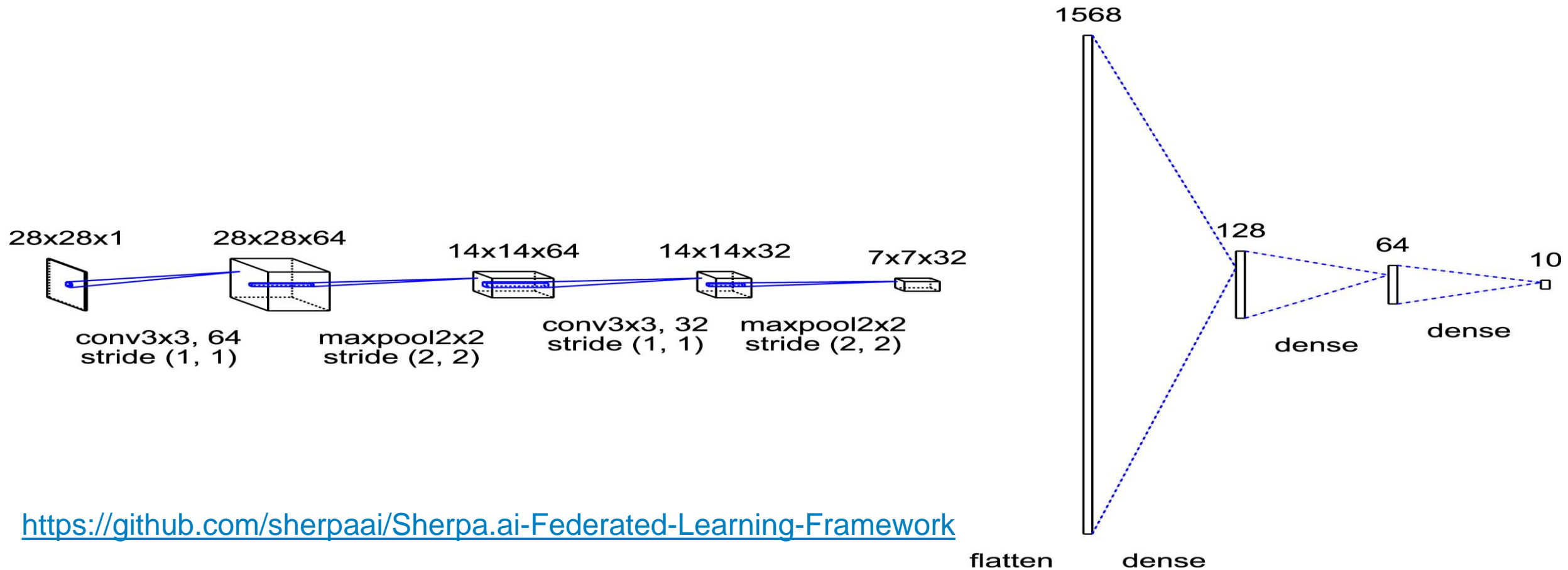
Train set	Test set	Total
240,000	40,000	280,000



<https://github.com/sherpaai/Sherpa.ai-Federated-Learning-Framework>

# Case of Study: Image classification with Sherpa.ai



- Classification model: feedforward network composed by two CNN layers with its corresponding maxpooling layers.



<https://github.com/sherpaai/Sherpa.ai-Federated-Learning-Framework>



# Case of Study: Image classification with Sherpa.ai

- The development of the model is not complicated.
- Data loading 
- Distribution of the data among 5 clients following an IID distribution. 

```
import matplotlib.pyplot as plt
import shfl
from shfl.private.reproducibility import Reproducibility

# Comment to turn off reproducibility:
Reproducibility(1234)

database = shfl.data_base.Emnist()
train_data, train_labels, test_data, test_labels = database.load_data()
```

```
iid_distribution = shfl.data_distribution.IidDataDistribution(database)
federated_data, test_data, test_labels = iid_distribution.
    ↪ get_federated_data(num_nodes=5, percent=50)
```

<https://github.com/sherpaai/Sherpa.ai-Federated-Learning-Framework>

# Case of Study: Image classification with Sherpa.ai

- Deep learning image classification model with Tensorflow.

• First CNN layer →

• Second CNN layer →

• Classification layer →

• Compilation of the model →

<https://github.com/sherpaai/Sherpa.ai-Federated-Learning-Framework>

```
[6]: import tensorflow as tf

def model_builder():
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Conv2D(32, kernel_size=(3, 3), padding='same', activation='relu', strides=1,
input_shape=(28, 28, 1)))
    model.add(tf.keras.layers.MaxPooling2D(pool_size=2, strides=2, padding='valid'))
    model.add(tf.keras.layers.Dropout(0.4))
    model.add(tf.keras.layers.Conv2D(32, kernel_size=(3, 3), padding='same', activation='relu', strides=1))
    model.add(tf.keras.layers.MaxPooling2D(pool_size=2, strides=2, padding='valid'))
    model.add(tf.keras.layers.Dropout(0.3))
    model.add(tf.keras.layers.Flatten())
    model.add(tf.keras.layers.Dense(128, activation='relu'))
    model.add(tf.keras.layers.Dropout(0.1))
    model.add(tf.keras.layers.Dense(64, activation='relu'))
    model.add(tf.keras.layers.Dense(10, activation='softmax'))

    model.compile(optimizer="rmsprop", loss="categorical_crossentropy", metrics=["accuracy"])

    return shfl.model.DeepLearningModel(model)
```

# Case of Study: Image classification with Sherpa.ai

- Setting of the aggregation operator. In this case, we use the FedAVG aggregation operator.
- The last step is the definition of the FA architecture with:
  - The classification model: `model_builder`.
  - The data to process: `federated_data`.
  - The aggregation operator: `aggregator`.

```
[7]: aggregator = shfl.federated_aggregator.FedAvgAggregator()  
federated_government = shfl.federated_government.FederatedGovernment(model_builder, federated_data, aggregator)
```

<https://github.com/sherpaai/Sherpa.ai-Federated-Learning-Framework>

# Case of Study: Image classification with Sherpa.ai

- The results of the federated model with an IID and a Non-IID data distribution is similar to the results reached by the same classification model in a centralised setting.

	IID (Accuracy)	Non-IID (Accuracy)
Centralised setting	0,9904	<b>0,9901</b>
Federated setting	<b>0,9921</b>	0,9855

- The configuration of the federated setting is:
  - Number of clients: 25.
  - Rounds of learning: 10.
  - Epochs: 5 per client.

## Outline

- Artificial intelligence challenges
- Federated Learning
  - Definition
  - Key elements
  - Categories
  - Architecture (client-server; peer-to-peer)
- Federated Learning workflow
- Federated Learning libraries
- Case of study: Image classification with Sherpa.ai (EMNIST data set)
- **Adversarial attacks. Proposal and Case of study**
- Concluding remarks. What's next?

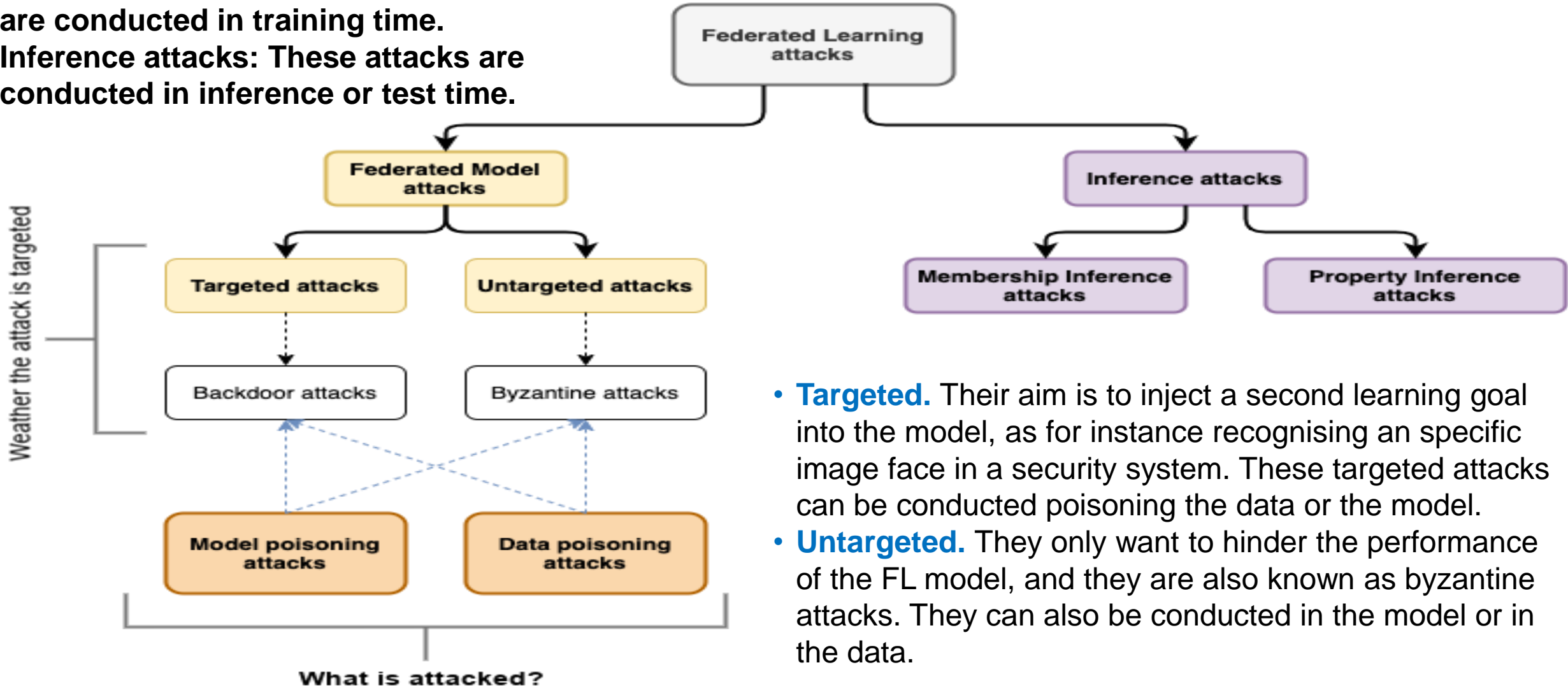
# Federated Learning: Adversarial Attacks

- FL as machine learning is vulnerable to adversarial attacks.
- As a reminder, FL is featured by:
  - Its distributed nature.
  - The inaccessibility of the training data, since it is sequestered in the clients.
- These two characteristics hinder the defence against adversarial attacks.
- The aim of the malicious agents (client or server) may be:
  - Misleading the behaviour of the FL model by poisoning the data or altering the parameters of the model.
  - Injecting a backdoor task without affecting the main learning task.
  - Breaking data privacy by discovering if an instance is in the training data or any property of the training data.

# Federated Learning: Adversarial Attacks

**Federated Model attacks:** These attacks are conducted in training time.

**Inference attacks:** These attacks are conducted in inference or test time.



- **Targeted.** Their aim is to inject a second learning goal into the model, as for instance recognising a specific image face in a security system. These targeted attacks can be conducted poisoning the data or the model.
- **Untargeted.** They only want to hinder the performance of the FL model, and they are also known as byzantine attacks. They can also be conducted in the model or in the data.

# Case of study: DDaBA against Byzantine attacks

- Byzantine attacks are a type of untargeted adversarial attacks grounded in the maliciously alteration of the data or the model.
- We propose DDaBA (Dynamic Defense Against Byzantine Attacks), which is a robust aggregation operator against Byzantine attacks regardless it is based on the manipulation of the data or the model.
- We evaluate DDaBA in three different Byzantine Attacks:
  - Label-flipping: It consists of randomly flipping the labels of the examples.
  - Out-of-distribution attack: It consists of introducing in the training dataset examples that do not follow the training data distribution.
  - Random weights: It consists of randomly generating the parameters of the local learning model.



# Case of study: DDaBA against Byzantine attacks

DDaBA is based on:

- The availability of a global validation subset in the FL server.
- The hypothesis is that a local learning model built upon altered dataset would underperform on the global validation set, and we consider them as outliers.
- We define an induced OWA (ordered weighted averaging) operator to average the contribution of each client according to its performance on the global validation set.
- Thanks to the induced OWA operator, **DDaBA dynamically selects the clients to be aggregated in the global learning model (FL-IOWA-DQ operator)**

# Case of study: DDaBA against Byzantine attacks

- We evaluate DDaBA on three federated datasets and we compare it with other defences from the state of the art.
- DDaBA shows in terms of Accuracy a strong performance independently the number of adversarial clients (1/30, 5/30, 10/50).
- DDaBA allows to avoid the effect of adversarial clients since the results are the same or higher when there are not any adversarial client.

	Federated EMNIST			Fashion MNIST			CIFAR-10		
	1-out-of-30	5-out-of-30	10-out-of-50	1-out-of-30	5-out-of-30	10-out-of-50	1-out-of-30	5-out-of-30	10-out-of-50
No attack	0,9657	0,9657	0,9629	0,8719	0,8719	0,8697	0,8357	0,8357	0,8231
FedAvg	0,1591	0,4212	0,4007	0,1917	0,3665	0,4322	0,1184	0,1436	0,2448
Trim.-mean	0,9428	0,8739	0,8370	0,8672	0,8325	0,861	0,8239	0,7346	0,8220
Median	0,9313	0,9161	0,9097	0,8671	0,8473	0,8585	0,8287	0,8090	0,8289
Krum	0,8917	0,8706	0,8634	0,7264	0,7197	0,7473	0,7479	0,7610	0,7698
MultiKrum (5)	0,9132	0,9270	0,9189	0,8403	0,8433	0,8255	0,8164	0,8232	0,8114
MultiKrum (20)	0,9563	0,9571	0,9504	0,8727	0,8724	0,8680	0,8439	0,8479	0,8518
Bulyan (f=1)	0,9523	0,7813	0,5809	0,8689	0,7830	0,7875	0,8265	0,6595	0,6454
Bulyan (f=5)	0,9365	0,9421	0,9516	0,8617	0,8652	0,8726	0,8492	0,8451	0,8540
<b>DDaBA</b>	<b>0,9657</b>	<b>0,9663</b>	<b>0,9643</b>	<b>0,8817</b>	<b>0,8783</b>	<b>0,8807</b>	<b>0,8633</b>	<b>0,8503</b>	<b>0,8557</b>

# Case of study: DDaBA against Byzantine attacks

- We also evaluate DDaBA when there are a high number of adversarial clients.
- We see that a high number of adversarial clients harms the performance of DDaBA.
- We propose the static version of DDaBA (SDaBA), which sets the proportion of clients to not consider in the federated aggregation.
- SDaBA filters out those clients whose performance on the validation set distances itself from the best client in  $\alpha$  times.
- The results show that SDaBA can work in scenarios with a high number of adversarial clients.

	Label-flipping	Out-of-dist.	Random weights
No attack	0,9657	0,9657	0,9657
FedAvg	0,3561	0,4394	0,0994
Trimmed-mean	0,6256	0,5778	0,1002
Median	0,8595	0,8347	0,9355
Krum	0,8801	0,8678	0,8633
MultiKrum (5)	0,9336	0,9366	0,9349
MultiKrum (20)	0,9623	0,9617	0,8595
MultiKrum (25)	0,9623	0,9617	0,8595
Bulyan (f=1)	0,4755	0,5005	0,1000
Bulyan (f=5)	0,9485	0,9475	0,9455
DDaBA	0,4235	0,4819	0,0997
SDaBA (1/4)	<b>0,9654</b>	<b>0,9653</b>	<b>0,9629</b>

Dataset: Federated EMNIST.

Adversarial clients: 10 out of 30 (33,33%).

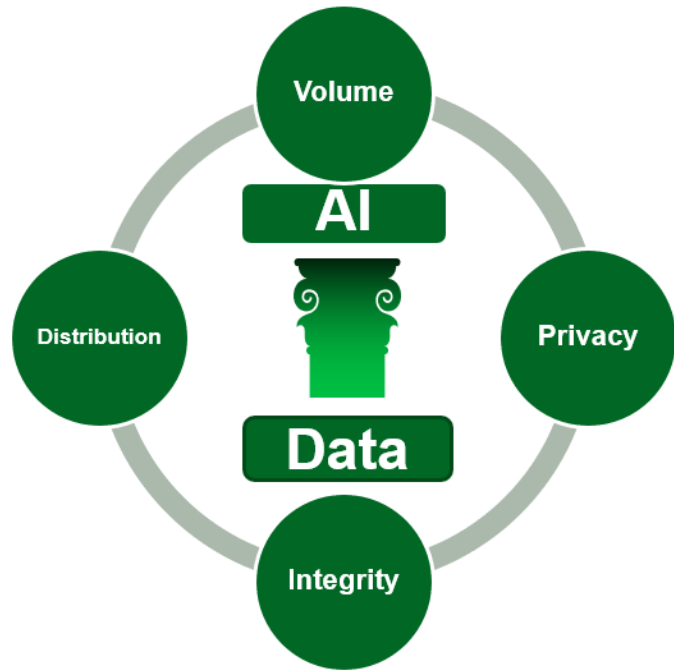
Evaluation measure: Accuracy

$\alpha$  value: 0.25.

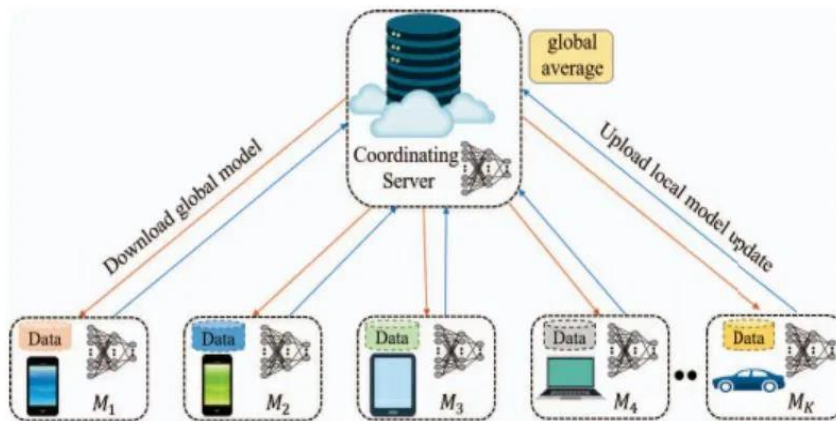
## Outline

- Artificial intelligence challenges
- Federated Learning
  - Definition
  - Key elements
  - Categories
  - Architecture (client-server; peer-to-peer)
- Federated Learning workflow
- Federated Learning libraries
- Case of study: Image classification with Sherpa.ai (EMNIST data set)
- Adversarial attacks. Proposal and Case of study
- **Concluding remarks. What's next?**

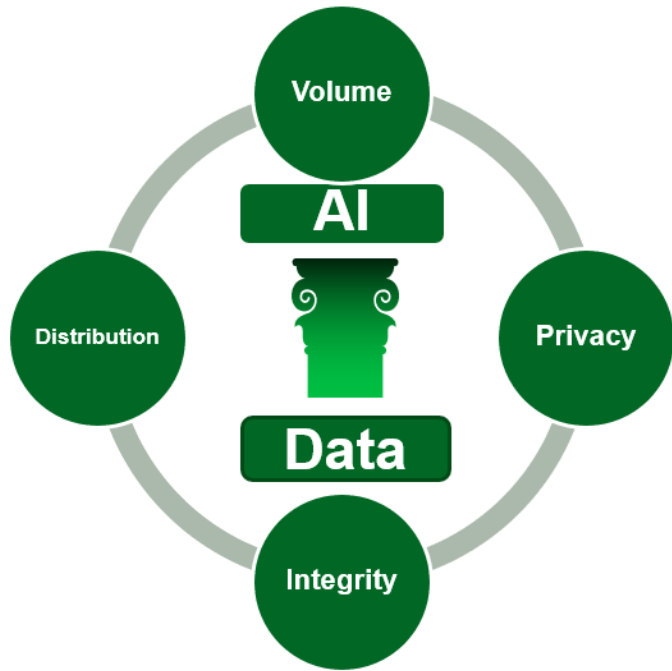
# Concluding remarks



- **Artificial Intelligence challenges:** Learning from data from different sources plus learning with privacy and integrity lead us to focus on new machine learning approaches. Federated Learning.
- **Key Elements:** Data. Learning model. Federated Aggregation Operator. Clients. Federated server. Communication.
- **Taking decisions:** Protection under adversarial tasks (low quality data, low quality models, attacks ...)
- **Mitigation of Communications risk:**
  - Differential privacy techniques to obfuscate the model parameters. Secure Multi-party Computation. Homomorphic Encryption.



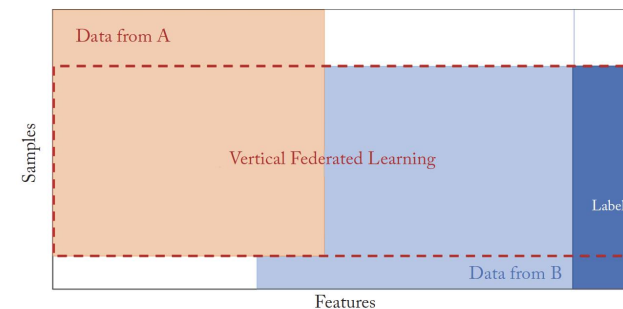
# Concluding remarks. Ongoing (What's next?)



- **eXplainable Artificial Intelligence and Federated Learning**
- **Personalized Federated Learning (per client)**
- **Unsupervised Federated Learning**
- **NLP under Federated Learning**
- **Key elements for Vertical Federated Learning**

## sherpa.ai

Go ahead with our framework:  
The Sherpa.ai Federated Learning and Differential Privacy  
Framework







**IDEAL 2021**

