Co-Funded by the Horizon 2020 programme of the European Union

Grant Agreement: 875358

Deliverable D4.1(a)

FAITH Federated AI Framework & Methodology

Work package:	WP4 – Data Analysis & Federated AI Services
Prepared By/Enquiries To:	Yahya Almardeny – WIT Diego Carvajal – UPM
Reviewers:	Maria Eugenia (Xenia) Beltran – UPM
Status:	For release.
Date:	30/04/2021
Version:	1.0
Classification:	Public

Authorised by:

Philip O'Brien

WIT

Authorised date: 17/05/21

Disclaimer:

This document reflects only authors' views. Every effort is made to ensure that all statements and information contained herein are accurate. However, the Partners accept no liability for any error or omission in the same. EC is not liable for any use that may be done of the information contained therein.

FAITH Project Profile

Contract No H2020-ICT- 875358

Acronym	**FAITH**
Title	**a Federated Artificial Intelligence solution for moniToring mental Health status after cancer treatment**
URL	**https://h2020-faith.eu/**
Twitter	**https://twitter.com/H2020_Faith**
LinkedIn	**linkedin.com/company/faith-project**
Facebook	**https://fb.me/H2020.FAITH**
Start Date	**01/01/2020**
Duration	**36 months**

FAITH Partners

List of Participants

Participant No	Participant organisation name	Short Name	Country
1 (Coordinator)	WATERFORD INSTITUTE OF TECHNOLOGY.	WIT	Ireland
2	UPMC Whitfield, Euro Care Healthcare Limited.	UPMC	Ireland
3	Universidad Politécnica de Madrid.	UPM	Spain
4	Servicio Madrileño de Salud.	SERMAS	Spain
5	UNINOVA, Instituto de Desenvolvimento de Novas Tecnologias.	UNINOVA	Portugal
6	Fundação D. Anna de Sommer Champalimaud e Dr. Carlos Montez Champalimaud.	CF	Portugal
7	Deep Blue.	DBL	Italy
8	Suite5 Data Intelligence Solutions Limited.	SUITE5	Cyprus
9	TFC Research and Innovation Limited.	TFC	Ireland

SC1-DTH-01-2019: Big data and Artificial Intelligence for monitoring health status and quality of life after the cancer treatment

H2020-SC1-DTH-2019

FAITH is co-funded by the European Commission - Agreement Number 875358 (H2020 Programme)

Document Control

This deliverable is the responsibility of the Work Package Leader. It is subject to internal review and formal authorisation procedures in line with ISO 9001 international quality management system procedures.

Version	Date	Author(s)	Change Details
0.1	05/02/21	Diego Carvajal (UPM)	Table of Contents defined & Executive Summary.
0.1.1	03/03/21	Yahya Almardeny (WIT)	Table of Contents Revised & Approve.
0.2	05/03/21	Yahya Almardeny (WIT)	Security and Privacy in the context of Federated Learning
0.3	15/03/21	Yahya Almardeny (WIT)	Federate Strategies.
0.4	20/03/21	Yahya Almardeny (WIT)	Distributed ML – 2nd part of the Introduction.
0.5	23/03/21	Diego Carvajal (UPM)	Federate Learning Frameworks + 1st part of the Introduction
0.6	24/03/21	Yahya Almardeny (WIT)	Implementation Strategy defined.
0.7	27/03/21	Yahya Almardeny (WIT)	Edge AI.
0.8	29/03/21	Yahya Almardeny (WIT)	Conclusion section.
0.8.1	30/03/21	Yahya Almardeny (WIT)	Abbreviations and Acronyms.
0.8.2	31/03/21	Yahya Almardeny (WIT)	Table of Contents, List of Figures and List of Tables.
0.9	27/04/21	Xenia Beltran (UPM)	Integration of content & Reviewed version.
0.9.1	30/04/21	Tom Flynn (TFC)	QA'd version.
1.0	13/0521	Philip O'Brien (WIT)	Final release for submission to European Commission Portal.

Executive Summary

Objectives:

This deliverable is the result from activities developed in T4.1 which comprised a comparative analysis mainly between the most important open-source Federated Learning libraries to define the Federated AI framework for the FAITH project, and an investigation on the most modern non-federated AI/ML deployment practices to discover the optimum deployment approach for Edge AI.

In the main section of the deliverable, a description of security and privacy in the context of Federated Learning is given, as well as federate strategies. Then, the most developed Federated Learning frameworks today are identified and an analysis of EDGE AI, model adaptation and devices for FAITH is done.

This document helps to better understand how security and privacy should be managed within FAITH, as well as the necessary strategies so that the data of the participants is protected and usable according to the requirements of the study. On the other hand, to choose the Federated Learning framework to be used in the project, it is necessary to have a clear vision of the most current developments that exist and to be able to ensure the selection of the framework that will allow to meet the specific objectives of FAITH.

This deliverable reflects the main ideas that will be used to distribute a globally trained artificial intelligence model to the different devices. Likewise, it will allow to understand what the most effective and efficient way is to train distributed models using local data and, it begins to lay the foundations to know how to combine the learning of these distributed models in order to produce an improved global model.

Results:

The primary results of this deliverable link with activities developed in T4.3 (explainable AI), T4.4 (Guarantee that generated models can be exploited by compression pipeline) and T6.3 (trial environment), and comprise the following outcomes:

- The definition of the security and privacy context within FAITH and the federate strategies.
- The Identification of Federated Learning frameworks and libraries to discover the optimum deployment approach for Edge AI.

TABLE OF CONTENT

TABLE OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The privacy of user data has been highly threatened in this time. Large leaks have occurred in companies such as Facebook or the US Customs and Border Protection, thus exposing the data they had stored on users. These and other cases of breaches have caused great concern and have led many governments to establish regulations to protect user data, such as GDPR in the European Union [1], PDPA in Singapore [2] and CCPA [3] in the United States. Failure to comply with these regulations can generate a high cost for companies like Google i.e. fined 50 million euros for breach of the GDPR [4].

It has been due to the events discussed above, that Federated Learning (FL) has gained a lot of attention in recent times, being a collaborative learning where there is no need to share users' personal data. While machine learning, especially deep learning, has attracted a lot of attention recently, the combination of federation and machine learning is emerging as a hot new research topic [5].

Federated Learning enables multiple parties to jointly train a machine learning model without exchanging local data, while meeting the requirements of various research areas, such as distributed system, machine learning, and privacy.

Federated learning has a couple of specific advantages:

- Ensuring privacy since the data remains on the user's device.
- Lower latency because the updated model can be used to make predictions on the user's device.
- Smarter models, given the collaborative training process.
- Less power consumption, as models are trained on a user's device.

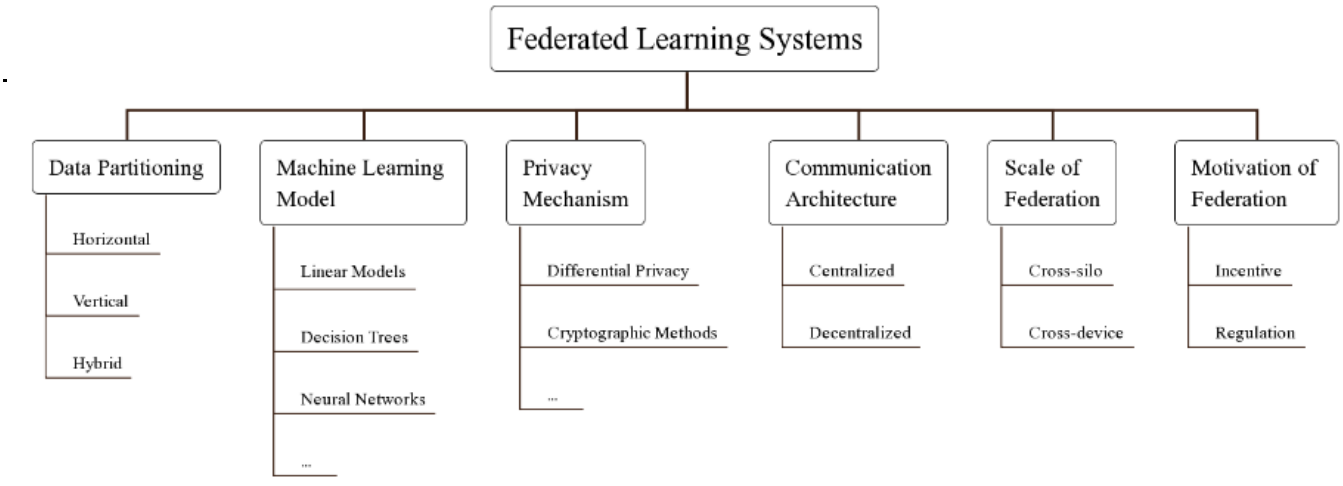

Figure 1 Taxonomy of federated learning systems [5]

Figure 1, shows the classification of the different Federated Learning Systems by six aspects: data partitioning, machine learning model, privacy mechanism, communication architecture, scale of federation, and motivation of federation. To better understand this classification, these six aspects can be applied to a use case in which several hospitals use Federated Learning to improve the prediction of lung cancer.

Data partitioning. It should be analysed how the medical records of patients are distributed among hospitals. Hospitals are likely to have different patients, however they may also have different information for a common patient. Therefore, both the non-overlapped instances and features in FL should be used.

Machine learning model. Investigate which machine learning model should be adopted to accomplish the task. For example, to perform a classification task on diagnostic images, a convolutional neural network must be trained in FL.

Privacy mechanism. A decision must be made on what techniques to use to protect privacy. Due to the high level of privacy of patient records, it is necessary to ensure that they cannot be inferred from the gradients and patterns exchanged. Differential privacy can be used to achieve privacy assurance.

Communication architecture. The communication architecture must be determined. If there is a reliable server, then it can be the administrator in FL. If not, a decentralized environment must be adopted.

Scale of federation. Unlike Federated Learning on mobile devices, in this scenario there is a relatively small scale and good stability of the federation. Furthermore, each part has a relatively large computing power, which means that more computing operations can be tolerated in the FL process.

Motivation of federation. It should be taken into account what incentives each party to participate in FL. In the present use case, a clear and direct motivation for hospitals is to increase the accuracy of lung cancer prediction. For this, FL should achieve a model with greater precision than local training for all hospitals.

Furthermore, Distributed Machine Learning (DML) covers the distributed training data storage and computations aspects, therefore DML is essentially designed to enhance the performance and scalability of the machine learning system in the presence of big data and large models. For instance, a typical DML system would have one server that splits the work over many computing nodes that act as workers. Each node works on a part of the dataset that is assigned to it, conducts a task such as the stochastic gradient descent (SGD), then sends back the weights of the local model to the server

which, in its turn, aggregates them [6]. There are two main categories of DML, namely: the DML that is motivated by the privacy, which ensures data security with decentralized sources of data; and the one that is motivated by the scalability, which aims to overcome memory and computation constraints via using different parallelism techniques. Not very different from DML, is the Federated Learning, which can be considered as a special type of DML, since both have many features in common, especially with regard to the data security and distributed datasets and training [7]. Beside enabling more privacy-preserving AI, Federated Learning solves some issues that exist in DML. For example, DML assumes that datasets are ideally distributed over the different sites, with approximately the same size and dimensions. On the other hand, Federated Learning does not have such underlying assumption, but rather it can work on heterogeneous datasets that differ in their sizes and dimensions [8], as will be shown in the next sections. Added to that, in Federated Learning, connected clients can involve light-powered smartphones or IoT devices that might also exhibit unreliable communication with the server, whereas in the DML systems, connected clients are typically nodes with high computational capabilities and fast network, such as the datacentres [9]. As a result, Federated Learning can be thought as an enhanced version of the distributed machine learning, that affords additional security and privacy of data and turns the one parameter server, in DML, into a coordinator party that assists connected clients to work together, hence Federated Learning disburdens the DML central server from many tasks that are typically performed by it.

2 ABBREVIATIONS AND ACRONYMS

Abbreviation	Description
CPU	Central Processing Unit.
DML	Distributed Machine Learning.
DF	Differential Privacy.
FL	Federate Learning.
FTL	Federated Transfer Learning.
FATE	Federated AI Technology Enabler.
GDPR	General Data Protection Regulation.
HE	Homomorphic Encryption.
HFL	Horizontal Federated Learning.
ISO9001-2015	International Quality Management Systems.
MPC	Multi-Party Computation.
ML	Machine Learning.
OT	Oblivious Transfer.
PPML	Privacy-Preserving Machine Learning.
PCA	Principal Component Analysis.
RAM	Random Access Memory.
ROM	Read-Only Memory.
SS	Secret Sharing.
SVD	Singular Value Decomposition.
TFF	TensorFlow Federated.
VFL	Vertical Federated Learning.
WP	Work Package.
WPL	Work Package Leader.

3 SECURITY AND PRIVACY IN THE CONTEXT OF FEDERATED LEARNING

Privacy-Preserving Machine Learning (PPML) is a term that broadly alludes to ML equipped with defence measures for protecting user privacy and data security. PPML is different from Secure ML, essentially, in the security infringement types that each deal with. Secure ML, on one hand, is confined to data integrity and availability, where an adversary manipulates the ML system to produce false negatives and/or false positives, hence the system becomes unusable. PPML, on the other hand, mainly deals with the privacy and confidentiality violations, where user's sensitive information is revealed to the attacker [10].

3.1 Models of Privacy Threat

There are different adversarial attacks that might target the ML privacy. For federated learning, however, the major concern is when an adversary uses reverse engineering techniques to reveal data used in training, in addition to any extra information about the model. Hence, in what follows, we overview the possible reconstitution attacks that can be conducted by an adversary.

3.1.1 Reconstruction Attacks

This attack can happen during the model training or model inference, in which the adversary aims to obtain training data or vectors of data features. The likelihood of such attack to happen increases in centralised learning, where different parties upload their raw data to a third party to perform computation, leaving raw data vulnerable to abuses. Contrary wise, in Federated Learning, each participant trains their model on local data, then model weights are only shared among different parties [11].

3.1.2 Model Inversion Attacks

In this type of attacks, the adversary conducts an equation solving attack, where they collect the responses of their queries sent to the model. The adversary might also be able to reconstruct the trained model by collecting enough query-response pairs, hence simulating the original model behaviour. In order to resist model inversion attacks, less information about the model should be visible, output should be limited and, if possible, rounded, and users of the model should always be granted a black-box access [12], [13].

3.1.3 Membership-Inference Attacks

In this attack, although the adversary has a black-box access to the model, they leverage the differences between the model predictions on samples provided by the attacker, and model predictions on the training dataset. As a result, the adversary might be able to gain more knowledge about the training data used. Differential Privacy is a major defence technique to resist Membership-Inference Attacks [14].

3.2 Privacy Preservation Techniques

3.2.1 Differential Privacy

The main idea of Differential Privacy (DF) is to confuse adversaries when they attempt to query the database for individual information [14]. In federated learning, Local Differential Privacy (LDP) is enabled via a randomised response technique, in which every party perturbs their data, then release their unintelligible version to the server. Hence, DP can be classified according to the perturbation applied into:

1. **Input perturbation**: The noise is added to the training data.
2. **Objective perturbation**: The noise is added to the objective function.
3. **Algorithm perturbation**: The noise is added to the intermediate values (e.g. gradients).
4. **Output perturbation**: The noise is added to the output parameters after training.

There are two main schemes of DP, one is by adding noise according to the function sensitivity [14], and another by adding noise according to an exponential distribution among discrete values [15].

For two datasets, D_1 and D_2, differing by only one record, and a function $M: D \rightarrow R^d$ over an arbitrary domain, the sensitivity of M is the maximum change in the output of M over all possible inputs:

$$\Delta\mathcal{M} = \max_{D_1, D_2} \|\mathcal{M}(D_1) - \mathcal{M}(D_2)\|$$

We denote the Laplace distribution with parameter b as Lap(b). Given a function M with sensitivity ΔM, the addition of noise drawn from a calibrated Laplace distribution Lap(ΔM / ϵ) maintains ϵ-differential privacy [14] and given by:

$$\mathcal{M}(X) + Lap\left(\frac{\Delta\mathcal{M}}{\epsilon}\right)^d$$

When an adversary queries the database, the ground truth of the sensitive data is returned with additional noise, hence the adversary is fooled. Nevertheless, since Laplace distribution is symmetric about the mean, the average of queries gets closer to the ground truth as the number of queries

increases. The latter is considered as the main limitation to DP. Therefore, a maximum number for queries should be imposed over a specific period of time, per participant.

It should be pointed out that other distributions can be used, such as adding noise from the Gaussian or Binomial distribution. Although the latter may yield better accuracy, DP would be weaker [16].

3.2.2 Secure Multi party Computation

In Secure Multi-Party Computation (MPC), multiple parties cooperatively compute a function from their private inputs, without disclosing such inputs among the parties. Given a secret value x that is split into n shares, all parties can jointly compute:

$$y_1, \ldots, y_n = f(x_1, \ldots, x_n)$$

hence, each party P_i only knows x_i and the corresponding y_i. MPC protocol can be proved, against adversaries that corrupt some parties, via simulation paradigm [17].

Three different frameworks can be used to implement MPC: Oblivious Transfer, Secret Sharing, and Threshold Homomorphic Encryption. Since the latter shares the idea of secret sharing with Oblivious Transfer, we will only overview the first two frameworks.

3.2.2.1 Oblivious Transfer

Oblivious Transfer (OT) was first proposed by Robin in 1981 [18]. In this protocol, there is a sender and a receiver. The former owns a database of a message-index pairs $(M_1, 1), \ldots, (M_N, N)$ where at every transfer, the receiver picks an index i in $[1, N]$, and receives M_i. As a result, the sender does not learn anything about the receiver's selection, and the receiver does not learn any extra information about the database.

3.2.2.2 Secret Sharing

In Secret Sharing (SS), the secret value is hidden by splitting it into different segments, then distributing those shares to the different parties. Hence each party only known one part of the secret value [19], [20]. It should be pointed out that the arithmetic secret sharing is the most adopted SS-based PPML approach, in which the addition operation is carried out at each party, locally.

3.2.3 Homomorphic Encryption

The main idea of Homomorphic Encryption (HE) is to conduct computation over a ciphertext without having to decrypt it [21]. Four functions build up the HE scheme H, as follows:

$$H = \{KeyGen, Enc\ Dec, Eval\}$$

Where:

- KeyGen: A cryptographic key generator as an input.
- Enc: Encryption function.
- Dec: Decryption function.
- Eval: Evaluation function that takes the ciphertext and public key and outputs the corresponding ciphertext to the plaintext.

HE schemes can involve an addition operation or a scalar multiplication, where that addition and multiplication are overloaded over ciphertexts, respectively [22].

HE schemes can be classified into three main categories:

1. Partially HE: Either the addition or the multiplication operation can be applied on ciphertexts for an unlimited number of times [21], [22].
2. Somewhat HE: Some of the operations can be applied on ciphertexts for only a limited number of times [23].
3. Fully HE: Both addition and multiplication operations cab be applied for unlimited number of times [24].

In-cloud computations on users' data by using HE, allows for higher level of protection as data will only be received encrypted and only the user can disclose the computation results.

3.2.4 Integration with GDPR

General Data Protection Regulation GDPR came into force from May 2018 to establish 7 main pillars regarding the personal data processing [25] that enforces sturdy PPML rules with regard to data processing. These principles are: Transparency & Lawfulness, Minimization of Data, Limitation on Purpose, Limitation on Storage, Accuracy & Precision, Confidentiality & Integrity, and Accountability. Nevertheless, it is profound the influence of GDPR on the AI industry, due to the limitation and rules that constrain the collection and processing of data distributed over different parties [26]. That being said, in compliance with the GDPR, explicit consents from users should be obtained even in federated AI, before initiating model training, with detailed explanation of what users' data will be used for.

Data Subject, Data Controller and Data Processor, are the three main participants roles that have associated obligations under the EU GDPR [27]. Data Subject ought to provide the end-users the right of access to get full insights about how their personal data is being processed. Data Controller is the main Service provider, who is responsible for ensuring the Data Subject and applying the appropriate

measures to comply with the GDPR. Finally, Data Processor is also the Service Provider, but not any third party, which under Federated Learning provides simple aggregation mechanisms of the local models to be used in updating the global model. The following table provides a summary comparison between the traditional centralised ML and the federated ML, with respect to the GDPR roles:

Table 1 Traditional Centralised ML vs Federated ML

GDPR Role	Traditional ML-based Service	Centralised FL-based Service
Personal Data	Original training data	Local model parameters
Data Subject	End-users	End-users
Data Controller	Service Provider	Service Provider
Data Processor	Service Provider, Third parties	Service Provider

One can see from the table how Federated Learning differs from traditional ML, mainly, but using the model parameters that have been trained locally, instead of the original data. Besides, FL does not involve any third parties for data processing.

4 FEDERATE STRATEGIES

4.1 Horizontal Federated Learning

Horizontal Federated Learning (HFL) is an example-partitioned Federated Learning that is applicable where datasets, over the different sites, have the same feature space, but differ in their sample space. Hence the term "horizontal", which is derived from the "horizontal partition" term in the traditional view of databases. HFL condition is given by:

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

where "X" is the feature space, "Y" is the sample space (assuming that they are the same), "I" is the user identifier and "D" is the dataset for the different i^{th} and j^{th} parties [8].

4.1.1 The Client-Server Architecture

Under this architecture, *K* clients collaborate to train the ML model with the assistance of a coordinator server. Here, we assume that both the clients and server are honest. However, we also consider the latter as curious. Therefore, we intend to stop any information leakage to the Server, from any Client [11]. The main four steps that are performed in such system are:

1. Clients conduct training gradients locally, mask results with one of the PPML methods, and then send them to the coordinator server.
2. The server conducts secure aggregation such as the weighted average.
3. Aggregated results are sent back to the clients.
4. Local models are updated on each client's side.

Those steps are repeated until the maximum number of iterations is reached or the cost function is converged.

There are two main types of what clients send to the server. The first is where clients send the gradients of their model, and this type is called "Gradient Averaging" (a.k.a Federated Averaging or FedAvg for short), and the second is where clients the weights of model themselves to the server, and this type is called "Model Averaging" [28]. The following table summarizes the key differences between the two types:

Table 2 Gradient Averaging vs Model Averaging

Method	Pros	Cons
Gradient Averaging	• Precise Information • Convergence is guaranteed	• Communication is heavy • Steadfast connection is required
Model Averaging	• Sporadic Synchronization (not bound to synchronous stochastic gradient descent)	• Convergence is not guaranteed • Loss in Performance

4.1.2 The Peer-to-Peer Architecture

Under this system, there is no coordinator server, rather it is a decentralized approach where each client is also a trainer or a worker. In this approach, each client trains their own model locally, then they securely transfer their model weights among each other [29], [30]. As there is no coordinator, trainers should agree on an order protocol for communication among them. Basically, there are two modes to maintain the transmission order among workers:

1) Cyclic Transfer: where clients are organized into a chain. The worker at the top of the chain sends weights to its downstream worker and so on.
2) Random Transfer (a.k.a Gossip Learning): where a trainer kth selects at random with equal probability a receiver with from a set $\{1,\ldots, L\}\backslash\{k\}$, then the latter updates its own model then sends at random to a receiver j in the set $\{1,\ldots, L\}\backslash\{j\}$. The process repeats until convergence [31].

Due to the nature of HFL, it is best suited for applications powered by a huge number of mobile devices. In this case, the clients/consumers of the application themselves are being federated under a business-to-consumer paradigm [32].

4.2 Vertical Federated Learning

Vertical Federated Learning (VFL) is an example of feature-partitioned federated learning, that follows a business-to-business paradigm, where the federated participants are organizations with different goals, but they are interested in cooperating with each other. VFL is applicable where datasets, over the different sites, have the same sample space, but differ in their feature space. VFL condition is given by:

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

where “X” is the feature space, “Y” is the label space, “I” is the sample identifier and “D” is the dataset for the different i^{th} and j^{th} parties [26].

4.2.1 VFL Architecture

In this system, there will be a number of organizations as participants and a third-party collaborator. We assume that the participants are honest but curious, whereas the collaborator is honest and legitimate (e.g. a government or secure nodes for computing such as SGX [33]). The training process in VFL is carried out under the following two steps:

- **Step 1**: The system identifies shared entities between the participants via an encryption-based user ID alignment technique, without disclosing to which company each user belongs to [34].
- **Step 2**: Common entities will be used to train a joint machine learning model as follows [26], [35]:
    1. The third-part collaborator sends a public key to the participants.
    2. Participants encrypts their intermediate results for the loss or gradients then share them between each other.
    3. Participants add a mask to the computed encrypted gradients and loss, then send them to the collaborator.
    4. The third-party collaborator decrypts loss and gradient, then it sends the results back to the participants, who update their models after unmasking the gradients.

Unlike HFL, participants under the VFL scheme need to interact with each other on a frequent basis due to the dependent computations required for exchanging the intermediate results. Hence, VFL requires a sturdy and reliable communication techniques as it is vulnerable to the failure in communications among participant due to the heavy exchanging of results.

4.3 Federated Transfer Learning

Federated Transfer Learning (FTL) is a learning framework that provides a federate solution when distributed datasets are heterogeneous. That is the participants’ local data do not share the feature space nor the sample space [36]. FTL enables different businesses and applications to collaborate to build sophisticated machine learning model, even if the participants have small data and few labels, while respecting the law of security and data privacy [26].

FTL can be used in both HFL and VFL under the following three main categories:

1. ***Based on Instance***: For HFL, participants can choose or reweight the samples of training data to reduce the difference among the distributions of each of the data drawn from the different participants' sites. For VFL, participating parties can selectively pick the samples and features to avoid the negative transfer resulted from the quite different objectives of those businesses [37].

2. ***Based on Feature***: For HFL, the representation or feature space that is common among participating parities can be learned by minimizing the maximum mean discrepancy [38]. For VFL, common feature space representation can be learned via minimizing the representations distance of the samples aligned among the different participating parties.

3. ***Based on Model***: Each of the federated party shares its model and learns from shared models. Otherwise, participating parties make use of models that are pre-trained as a part of the entire starting models for the task of Federated Learning.

FTL utilizes the traditional transfer learning into the paradigm of PPML, in which the goal is to predict the labels of new unseen data (or existing ones) as precisely as possible, by exploiting transferred knowledge of the source domain, in a secure manner [36].

5 FEDERATE LEARNING FRAMEWORKS

There are several frameworks that allow you to integrate the advantages of Federated Learning. Table 1, shows a comparison between the main Federated Learning systems, taking into account the models, data distribution strategy, security, among others.

Table 3 Federated Learning Frameworks

		Federated AI Technology Enabler	TensorFlow Federated (TFF)	OpenMined PySyft	PaddleFL
Models	Linear Models	X	x	x	x
	Decision Trees	X	_	_	_
	Neural Networks	X	x	x	x
	Operative System	Mac, Linux	Mac, Linux, TF Lite	Mac, Linux, Windows, PySyft for android	Mac, Linux, Windows, Paddle Lite 2.0
Data Distribution strategy	Horizontal	x	x	x	x
	Vertical	x	_	_	_
Security	Differential Privacy	_	_	x	x
	Cryptographic Methods	x	x*[1]	x	x
Supports GPUs		_	x	_	_
Compatible Frameworks		_	TensorFlow, Keras	Pytorch, TensorFlow, Keras	Paddle
Network Implementation		FATE Serving	TF serving, TensorFlow.js	PyGrid	
	Who is behind	Webank's AI Department	Google	Facebook	Baidu

5.1 Federated AI Technology Enabler (FATE)

FATE is an open-source project whose development was initiated by the Webank´s AI Department. It aims to provide a secure computing framework to support the federated AI ecosystem and

implements secure computing protocols based on homomorphic encryption and multi-party computing (MPC). It supports federated learning architectures and secure computation of various machine learning algorithms, including logistic regression, tree-based algorithms, deep learning, and transfer learning [39].

5.1.1 General Structure of FATE

Figure 2, shows the general structure of FATE that has six main modules: EggRoll, FederatedML, FATE-Flow, FATE Serving, FATEBoard and KubeFATE [40].

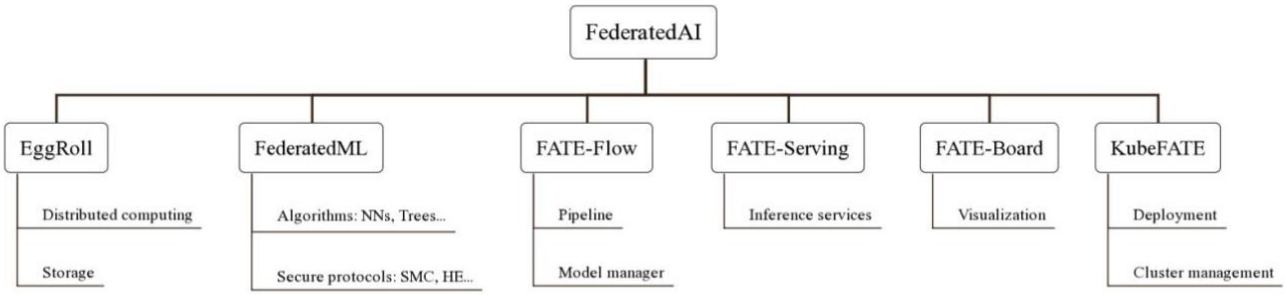

Figure 2 The FATE system structure [5]

Eggroll is a distributed infrastructure with a unity of computing, storage and communication targeted at large-scale machine learning and deep learning applications [41].

FederatedML includes federated algorithms and secure protocols. It currently supports training of many types of machine learning models in both horizontal and vertical federated environments, including NN, GBDT, and logistic regression. In addition, it integrates secure multi-party computing and homomorphic encryption to provide privacy guarantees [5].

FATE-Flow is a comprehensive pipeline platform for Federated Learning. Pipeline is a sequence of components specifically designed for highly flexible, high-performance federated learning tasks. That includes data processing, modelling, training, verification, publishing, and inference service [42].

FATE-Serving is a high-performance industrialized service system for Federated Learning models, designed for production environments [43]. This module provides the inference services for users. Supports loading FL models and making online inferences about them [5].

FATE-Board provides a visual way of probing models, from which you can efficiently reshape and enhance models; to facilitate understanding, tracking, debugging, and exploration of Federated Learning modelling, as well as examining, evaluating, and comparing multiple Federated Learning models [44].

KubeFATE manages federated learning workloads using cloud-native technologies such as Docker or Kubernetes. This module enables federated learning jobs to run in public, private, and hybrid cloud environments [45].

In conclusion, FATE is a powerful and easy-to-use Federated Learning system. Simply configure the parameters to run a Federated Learning algorithm. In addition, it provides detailed documents on its implementation and use. However, since FATE provides algorithm-level interfaces, professionals must modify the FATE source code to implement their own federated algorithms. This is not easy for non-expert users [5].

5.2 TensorFlow Federated

TensorFlow Federated (TFF) is an open-source framework for machine learning and other computations on decentralized data. TFF enables developers to simulate the federated learning algorithms that are included in their models and data, and to experiment with new algorithms. The building blocks provided by TFF can also be used to implement non-learning computations, such as aggregated statistics over decentralized data [46].

TFF provides two APIs of different layers [5]:

FL API offers high-level interfaces. It includes three key parts, which are models, federated compute constructors, and data sets. It also provides the mock federated data sets and the functions to access and list the local data sets for FL.

Federated Core (FC) API also includes lower-level interfaces as the basis of the FL process. Developers can implement their functions and interfaces within the federated core. Specifically, as a Python package, FC provides Python interfaces and developers can use them and write new Python functions. To be easy to use, especially for developers familiar with TensorFlow, it supports any type, such as tension types, sequence types, tuple types, and function types. Finally, FC supports several federated operators, such as federated sum, federated reduction, and federated broadcast. Developers can define their own operators to implement the FL algorithm.

In conclusion, TFF is a lightweight system for developers to design and implement new FL algorithms.

5.3 OpenMined PySyft

PySyft is a Python library for secure and private deep learning. PySyft decouples private data from model training, using Federated Learning, differential privacy, and encrypted computing (such as

multi-party computing (MPC) and homomorphic encryption (HE) within leading deep learning frameworks such as PyTorch and TensorFlow [47].

Although PySyft provides several tutorials for implementation, it has not been found detailed information on the architecture of the system and its interfaces.

5.4 PaddleFL

PaddleFL is an open-source federated learning framework based on PaddlePaddle [48]. Researchers can easily replicate and compare different federated learning algorithms with PaddleFL. Developers

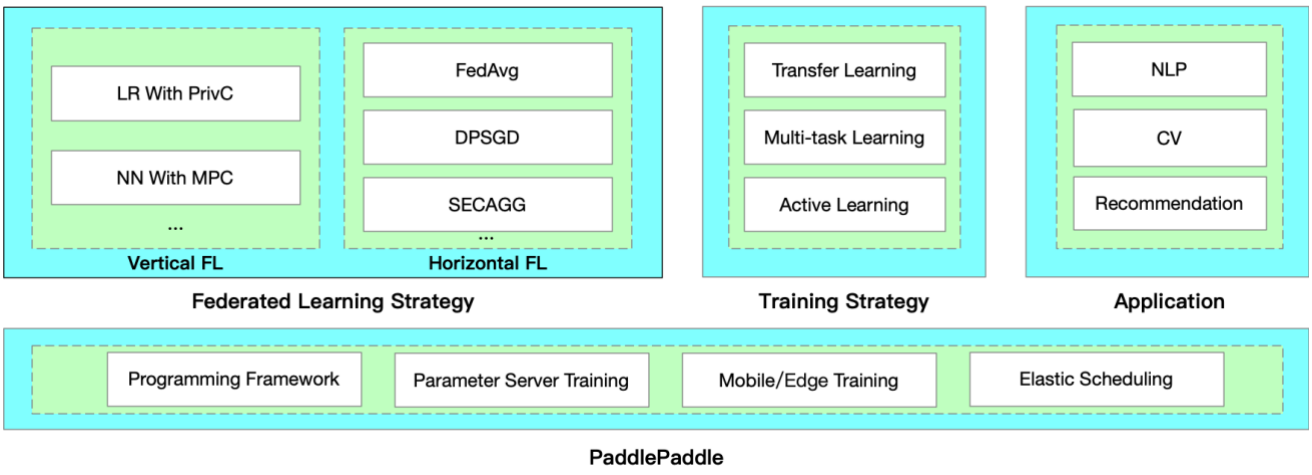

Figure 3 Paddle Strategies [49]

can also benefit from PaddleF, because it is easy to implement a Federated Learning system in large-scale distributed clusters. In PaddleFL, several Federated Learning strategies are provided with application in computer vision, natural language processing, recommendation, etc. It provides the application of traditional machine learning training strategies such as multitasking learning and transfer learning in federated learning environments. Based on PaddlePaddle's large-scale distributed training and elastic scheduling of training work in Kubernetes, PaddleFL can be easily deployed based on full-stack open-source software [49].

Figure 3 shows how the learning strategies that can be implemented with PaddleFL are categorized. Also, this Federated Learning system provides application demonstrations in natural language processing, computer vision, and recommendations.

6 EDGE AI

6.1 Model Optimization and Adaption Techniques

Due to the constrained memory and compute capabilities, edge AI is undoubtedly limited in comparison with could-based modelling. Hence, researchers have focused their efforts on improving the existing frameworks to be more suitable for edge devices. In what follows, we overview the most popular state-of-the-art methodologies and technologies of model adaption and optimization for Edge AI.

6.1.1 Model Comparison

This category includes a variety of different approaches that are mainly based on exploiting the sparsity inherited of the weights and gradients in order to minimize the memory requirement, up to the hilt. Those approaches include, but not limited to: Dimensionality Reduction, Quantization, Pruning Precision Downgrading, and others. To enable the latter, technologies such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and Huffman Coding, to name a few, are available. It should be pointed out that Model Compression is currently a very active direction in Edge AI as it is simple to implement, besides, it is suitable for both Model Training and Model Inference [50].

6.1.2 Algorithm Asynchronization

Algorithm Asynchronization aims to aggregate local models under the Federated Learning framework, in an asynchronization manner. Simply put, participating devices exchange gradients and weights in a peer-to-peer way in order to alleviate high concurrency on wireless channel, where connected devices have high chances of not completing the model download or upload, due to the wireless network unreliability and congestion [51].

6.1.3 Conditional Computation

Approaches that apply Conditional Computation aim to switch off some unimportant calculations, selectively, as a block-wise dropout [52]. Input Filtering, Components Shutoff, Components Sharing, Early Exit, Results Caching are popular examples of Conditional Computation approaches that are generally based on picking the most notable computation part or early stop once the threshold of threshold is reached.

6.1.4 Thorough Decentralization

Thorough Decentralization aims to remove the central aggregator in order to avoid any private data disclosure. By using blockchain-based federated learning architecture, a trustworthy edge learning environment can be built, where the central server for model aggregating is not required any further [53].

The following table maps the model adaption approaches to their desired enhancements:

Table 4 Enhancements of Model Adaption Methods

Method	Enhancement
Model Compression	Cost and Efficiency
Algorithm Asynchronization	Performance and Efficiency
Conditional Computation	Cost and Efficiency
Thorough Decentralization	Performance and Privacy

6.2 Edge Frameworks

In what follows, we review a set of frameworks that help developers run AI models on Edge, such as mobile, embedded, and IoT devices.

Table 5 Comparison of the Different Edge AI Frameworks

Framework	Overview	Pros	Cons
TensorFlow Lite	Lightweight solution for mobile and embedded devices	✓ Fast performance. ✓ Enables low-latency inference of on-device machine learning models with a small binary size.	▪ Some models are still relatively too big to store on devices. ▪ TensorFlow Lite models have lower accuracy than their counterparts.
TensorFlow.js	Machine Learning using in JavaScript from scratch	✓ NodeJS Powered. ✓ Deploy python ML model directly into JavaScript.	▪ Just available in JavaScript and TypeScript Languages
Keras	Deep Learning library for Theano and TensorFlow	✓ Quality Documentation. ✓ Easy and fast NN prototyping. ✓ Models can be saved and used in TensorFlow Lite	▪ Slower than its backend. ▪ Requires extra work to make it work on mobile devices.

PyTorch Mobile	A deep learning framework that puts Python first	✓ Available for iOS, Android, and Linux	▪ Relatively new. ▪ Less deployment options compared to TensorFlow.
ML Kit	Machine learning for mobile developers by Google	✓ It can be launched using Firebase. ✓ For Android and iOS apps.	▪ Custom models can be very large in size. ▪ It might be challenging for a new developer to implement ML Kit.

6.3 Edge Devices for FAITH

The rapidly increasing hardware capability of mobile devices along with the recent advancements in ultra-low-power machine learning hardware, the development of new algorithms for low-power devices, and the emerging technologies in wireless network communications, in addition to the rapidly decreasing prices of smartphones and digital appliances, are the main players in the paradigm shift of artificial intelligence to the "edge" of the network. The latter led to unlock an entirely new class of smart applications that enable mobile devices of building and training machine learning models.

Hereafter, we briefly overview the new computational capabilities and features, that have been added in new mobile devices, which clarify how FAITH is aligned with the trend and their feasibility for implementation.

6.3.1 Storage

Since 2012, there has been a noticeable increase in the storage capacity, where Random Access Memory (RAM) that is larger than 2GB and Read-Only Memory (ROM) that is larger than 16GB, have become dominant specs in more than 78% and 83% of manufactured smartphones, respectively [54]. Internal storage capacity is an important factor for Edge AI, as larger internal memory allows faster multi-task processing and loading of data, making small devices more capable of carrying out the machine learning training and inferences.

6.3.2 Chipset

The performance of Mobile devices can be boosted at more efficient energy consumption with multi-code Central Processing Units – CPUs. Added to that, high-performant CPUs are required to support the training and data modelling in Edge AI. Hence, vendors of smartphone tend to pick chipset with powerful multi-core CPUs. Unsurprisingly, from 2013 onwards, Quad-core CPUs have replaced

Dual-core ones, and recently, it was noticeable the emergence of Octa-core CPUs, making Dual-core ones almost obsolete [55].

6.3.3 Battery

Training AI models on Edge is considered a power-consuming task. Nonetheless, with the presence of lightweight AI and smart processors, and efficient network on Edge, battery drain has been reduced noticeably. The battery capacity in smartphones has been increased dramatically over the past few years. Since 2012, the battery capacity increased from less than 1750mAh to 2500mAh in 2015, and it is still surging to reach more than 5000mAh in many smartphones, nowadays [56]. LiPo batteries an Li-ion battery are the two main battery types on the market today, although they are relatively cheap, yet they efficiently boost the lifespan of the smartphone's energy, with an average capacity of more than 3500mAh and very fast recharging capability [57].

6.3.4 Network

With the presence of 4G network in late 2009, which offered much higher bandwidth, lower latency, and improved spectrum efficiency, the percentage of smartphone models that support 4G network increased from 20% to over 50% between 2012 and 2015 [56]. Moreover, the number of 4G LTE mobile phones in use worldwide, from 2014 to 2018, has increased from 257 to 921 million, according to Statista 2021. For instance, the results from the Mobile Consumer Experience Survey, which was conducted in 2019 by ComReg in Ireland, reveal that 62% of the smartphone holders use 4G network as their main service [58]. This dramatic change brought the computer closer to the user, making edge AI, certainly, more viable.

7 CONCLUSION

This document defines the available Federated AI frameworks for the FAITH project, in addition to how to build and utilize machine learning models in Artificial Intelligence (AI) applications where data is dispersed on different sites and run by different people or organizations. We first overviewed the taxonomy of federated learning systems, and the main differences between it and the distributed machine learning, where we showed how Federated Learning solves some issues that exist in distributed machine learning. As we are in the era of big data, privacy and security requirements make it increasingly infeasible to merge the data at different organizations in a simple way. Hence, we discussed and showed the different Privacy-Preserving Machine Learning techniques that enable federated learning to build high-performance models that are shared among multiple parties while still complying with requirements for data confidentiality, user privacy and GDRP rules. Added to that, we presented the different federate strategies where we illustrated how federated learning is viable in scenarios where local data that is distributed over multiple sites, differ in their dimensions. We further compared the several open-source frameworks that allow to integrate the advantages of Federated Learning, and their pros and cons. Besides the security and privacy concerns, another strong inspiration for federated learning is to utilize the computing power at the edge devices of a cloud system, maximally, where the communication is boosted efficiently only when the computed results, are exchanged between the servers and devices, rather than the whole training data. Therefore, we overviewed the most popular state-of-the-art methodologies and technologies of model adaption and optimization for Edge AI, in addition to the new computational capabilities and features, which have been added in new mobile devices, which clarify how FAITH is aligned with the trend and their feasibility for implementation. As a result, this deliverable provides a well-grounded reference that shall help in shaping the initial implementation strategy of FAITH project, in addition to make a clearer vision of the upcoming development phase. Finally, it should be pointed out that this deliverable would receive further updates in M30, as/if required.

8 BIBLIOGRAPHY

[1] P. Voigt και A. Von dem Bussche, The eu general data protection regulation (gdpr). A Practical Guide, First επιμ., Cham: Springer International Publishing, 2017.

[2] W. B.Chik, «The Singapore Personal Data Protection Act and an assessment of future trends in data privacy reform,» *Computer Law & Security Review,* τόμ. 29, αρ. 5, pp. 554-575, 2013.

[3] Californians for Consumer Privacy, «California Consumer Privacy Act Home Page,» [Ηλεκτρονικό]. Available: https://www.caprivacy.org/.

[4] Panda Security, «Google, sancionado con 50 millones de euros por incumplir con el GDPR,» [Ηλεκτρονικό]. Available: https://www.pandasecurity.com/es/mediacenter/seguridad/google-sancion-millonaria-por-incumplir-el-gdpr/. [Πρόσβαση March 2021].

[5] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, X. Liu και B. He, «A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection,» 2021.

[6] T.-Y. Liu, W. Chen και T. Wang, «Distributed Machine Learning: Foundations, Trends, and Practices,» σε *WWW '17 Companion Proceedings of the 26th International Conference on World Wide Web Companion*, 2017.

[7] J. Konečný, H. B. McMahan, D. Ramage και P. Richtarik, «Federated Optimization: Distributed Machine Learning for On-Device Intelligence,» *arXiv preprint arXiv:1610.02527,* 2016.

[8] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu και S. Zhao, «Advances and Open Problems in Federated Learning,» *Foundations and Trends® in Machine Learning,* τόμ. 14, αρ. 1, 2021.

[9] J. Konečný, H. B. McMahan και D. Ramage, «Federated Optimization: Distributed Optimization Beyond the Datacenter,» *arXiv preprint arXiv:1511.03575,* 2015.

[10] M. Barreno, B. Nelson, R. Sears, A. D. Joseph και J. D. Tygar, «Can machine learning be secure,» σε *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, 2006.

[11] L. T. Phong, Y. Aono, T. Hayashi, L. Wang και S. Moriai, «Privacy-Preserving Deep Learning via Additively Homomorphic Encryption,» *IEEE Transactions on Information Forensics and Security,* τόμ. 13, αρ. 5, pp. 1333-1345, 2018.

[12] M. Al-Rubaie και J. M. Chang, «Reconstruction Attacks Against Mobile-Based Continuous Authentication Systems in the Cloud,» *IEEE Transactions on Information Forensics and Security,* τόμ. 11, αρ. 12, pp. 2648-2663, 2016.

[13] M. Fredrikson, S. Jha και T. Ristenpart, «Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures,» σε *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[14] C. Dwork, F. Mcsherry, K. Nissim και A. Smith, «Calibrating noise to sensitivity in private data analysis,» *Lecture Notes in Computer Science,* pp. 265-284, 2006.

[15] F. McSherry και K. Talwar, «Mechanism Design via Differential Privacy,» σε *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, 2007.

[16] C. Dwork και K. Nissim, «Privacy-preserving datamining on vertically partitioned databases,» *Lecture Notes in Computer Science,* pp. 528-544, 2004.

[17] Y. Lindell, «How to Simulate It – A Tutorial on the Simulation Proof Technique,» *IACR Cryptology ePrint Archive,* τόμ. 2016, pp. 277-346, 2017.

[18] M. O. Rabin, «How to Exchange Secrets with Oblivious Transfer,» *IACR Cryptology ePrint Archive,* τόμ. 2005, p. 187, 1981.

[19] A. Shamir, «How to share a secret,» *Communications of The ACM,* τόμ. 22, αρ. 11, pp. 612-613, 1979.

[20] S. Tutdere και O. Uzunkol, «Construction of arithmetic secret sharing schemes by using torsion limits,» *Hacettepe Journal of Mathematics and Statistics,* τόμ. 49, αρ. 2, pp. 1-10, 2019.

[21] R. Rivest, L. Adleman και M. Dertouzos, «On data banks and privacy homomorphisms,» *Foundations of secure computation,* τόμ. 4, αρ. 11, pp. 169--180, 1978.

[22] P. Paillier, «Public-key cryptosystems based on composite degree residuosity classes,» σε *EUROCRYPT'99 Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, 1999.

[23] Z. Brakerski και V. Vaikuntanathan, «Efficient Fully Homomorphic Encryption from (Standard) LWE,» *Siam Journal on Control and Optimization,* 2014.

[24] A. Acar, H. Aksu, A. S. Uluagac και M. Conti, «A Survey on Homomorphic Encryption Schemes: Theory and Implementation,» *ACM Computing Surveys,* τόμ. 51, αρ. 4, p. 79, 2018.

[25] C. Addis και M. Kutar, «General Data Protection Regulation (GDPR), Artificial Intelligence (AI) and UK Organisations: A year of implementation of GDPR,» , 2020.

[26] Q. Yang, Y. Liu, T. Chen και Y. Tong, «Federated Machine Learning: Concept and Applications,» *ACM Transactions on Intelligent Systems and Technology,* τόμ. 10, αρ. 2, p. 12, 2019.

[27] N. B. Truong, K. Sun, S. Wang, F. Guitton και Y. Guo, «Privacy Preservation in Federated Learning: Insights from the GDPR Perspective.,» *arXiv preprint arXiv:2011.05411,* 2020.

[28] C. Yu, H. Tang, C. Renggli, S. A. Kassing, A. Singla, D. Alistarh, C. Zhang και J. Liu, «Distributed Learning over Unreliable Networks,» σε *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, 2019.

[29] K. Chang, N. Balachandar, C. K. Lam, D. Yi, J. M. Brown, A. Beers, B. R. Rosen, D. L. Rubin και J. Kalpathy-Cramer, «Institutionally Distributed Deep Learning Networks.,» *arXiv preprint arXiv:1709.05929,* 2017.

[30] L. T. Phong και T. T. Phuong, «Privacy-Preserving Deep Learning via Weight Transmission,» *IEEE Transactions on Information Forensics and Security,* τόμ. 14, αρ. 11, pp. 3003-3015, 2019.

[31] I. Hegedűs, G. Danner και M. Jelasity, «Gossip Learning as a Decentralized Alternative to Federated Learning,» σε *19th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS)*, 2019.

[32] H. B. McMahan, E. Moore, D. Ramage, S. Hampson και B. A. y. Arcas, «Communication-Efficient Learning of Deep Networks from Decentralized Data,» σε *Artificial Intelligence and Statistics*, 2017.

[33] R. Bahmani, M. Barbosa, F. Brasser, B. Portela, A.-R. Sadeghi, G. Scerri και B. Warinschi, «Secure Multiparty Computation from SGX,» σε *International Conference on Financial Cryptography and Data Security 2017 (FC'17)*, 2017.

[34] M. Scannapieco, I. Figotin, E. Bertino και A. K. Elmagarmid, «Privacy preserving schema and data matching,» σε *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007.

[35] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen και Q. Yang, «SecureBoost: A Lossless Federated Learning Framework,» *arXiv preprint arXiv:1901.08755,* 2019.

[36] Y. Liu, Y. Kang, C. Xing, T. Chen και Q. Yang, «A Secure Federated Transfer Learning Framework,» *The Missouri Review,* τόμ. 35, αρ. 4, pp. 70-82, 2020.

[37] S. J. Pan και Q. Yang, «A Survey on Transfer Learning,» *IEEE Transactions on Knowledge and Data Engineering,* τόμ. 22, αρ. 10, pp. 1345-1359, 2010.

[38] S. J. Pan, I. W. Tsang, J. T. Kwok και Q. Yang, «Domain Adaptation via Transfer Component Analysis,» *IEEE Transactions on Neural Networks,* τόμ. 22, αρ. 2, pp. 199-210, 2011.

[39] FATE Project, «FATE,» [Ηλεκτρονικό]. Available: https://github.com/FederatedAI/FATE. [Πρόσβαση March 2021].

[40] F. Project, «What is FATE,» 2021. [Ηλεκτρονικό]. Available: https://fate.fedai.org/overview/. [Πρόσβαση March 2021].

[41] FATE Project, «Federated AI Ecosystem,» [Ηλεκτρονικό]. Available: https://www.fedai.org/. [Πρόσβαση March 2021].

[42] FATE Project, «FATE-Flow,» [Ηλεκτρονικό]. Available: https://fate.fedai.org/fateflow/. [Πρόσβαση March 2021].

[43] FATE Project, «FATE-Serving,» [Ηλεκτρονικό]. Available: https://fate.fedai.org/fate-serving/. [Πρόσβαση March 2021].

[44] FATE Project, «FATEBoard,» [Ηλεκτρονικό]. Available: https://fate.fedai.org/fateboard/. [Πρόσβαση March 2021].

[45] FATE Project, «KubeFATE,» [Ηλεκτρονικό]. [Πρόσβαση March 2021].

[46] GOOGLE, «TensorFlow Federated: Machine Learning on Decentralized Data,» [Ηλεκτρονικό]. Available: https://www.tensorflow.org/federated. [Πρόσβαση March 2021].

[47] OpenMined, «Github-PySyft,» [Ηλεκτρονικό]. Available: https://github.com/OpenMined/PySyft. [Πρόσβαση March 2021].

[48] PaddlePaddle, «Github-Paddle,» [Ηλεκτρονικό]. Available: https://github.com/PaddlePaddle/Paddle. [Πρόσβαση March 2021].

[49] PaddlePaddle, «Github-PaddleFL,» [Ηλεκτρονικό]. Available: https://github.com/PaddlePaddle/PaddleFL. [Πρόσβαση March 2021].

[50] Y. Cheng, D. Wang, P. Zhou και T. Zhang, «A Survey of Model Compression and Acceleration for Deep Neural Networks.,» *arXiv preprint arXiv:1710.09282,* 2017.

[51] J. Daily, A. Vishnu, C. Siegel, T. Warfel και V. Amatya, «GossipGraD: Scalable Deep Learning using Gossip Communication based Asynchronous Gradient Descent.,» *arXiv preprint arXiv:1803.05880,* 2018.

[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever και R. Salakhutdinov, «Dropout: a simple way to prevent neural networks from overfitting,» *Journal of Machine Learning Research,* τόμ. 15, αρ. 1, pp. 1929-1958, 2014.

[53] H. Kim, J. Park, M. Bennis και S.-L. Kim, «On-Device Federated Learning via Blockchain and its Latency Analysis.,» *,* 2018.

[54] M. Rawat, «A Deep-dive into Memory Trends in Smartphones,» Survey, CMR Mobile Industry Consumer Insights (MICI), 2020. [Ηλεκτρονικό]. Available: http://cmrindia.com/a-deep-dive-into-memory-trends-in-smartphones/.

[55] M. P. Singh και M. K. Jain, «Evolution of Processor Architecture in Mobile Phones,» *International Journal of Computer Applications,* τόμ. 90, αρ. 4, pp. 34-39, 2014.

[56] Q. Han και D. Cho, «Characterizing the technological evolution of smartphones: insights from performance benchmarks,» σε *Proceedings of the 18th Annual International Conference on Electronic Commerce*, 2016.

[57] R. Triggs, «Fact check: Is smartphone battery capacity growing or staying the same?,» Android Authority, 2018. [Ηλεκτρονικό]. Available: https://www.androidauthority.com/smartphone-battery-capacity-887305/.

[58] ComReg, «Mobile Service Trends in Ireland, Results from the Mobile Consumer Experience Survey 2019,» https://www.comreg.ie/media/2019/09/ComReg-Mobile-Consumer-Experience-Survey-Infographic05092019.pdf, 2019.