UNIVERSITA' DEGLI STUDI DI VERONA

DEPARTMENT OF

Computer Science

GRADUATE SCHOOL OF

University of Verona

DOCTORAL PROGRAM IN

Computer Science

Cycle / year 37°

TITLE OF THE DOCTORAL THESIS

Advancing Federated Learning in Biomedical Applications:
Tools, Best Practices, and Implementations

S.S.D. IINF-05/A

Coordinator: Prof./ssa ~~Ferdinando~~ Cicalese

Signature _____

Tutor: Prof./ssa Gloria Menegaz

Signature _____

Doctoral Student: Dott. Walter Riviera

Signature _____

* For the list of S.S.D. please refer to the Ministerial Decree of 4th October 2000, Attachment A "*Elenco dei Settori Scientifico – Disciplinari"* available at: http://www.miur.it/atti/2000/alladm001004_01.htm

"You already have a job, why doing a Ph.D.?". For myself.

Abstract

The impact of Deep Learning (DL) and Artificial Intelligence (AI) in biomedicine has been transformative and far-reaching. These technologies have revolutionized a variety of biomedical fields by improving diagnostic accuracy, predicting patient outcomes, and personalizing treatments. With the increasing need for large-scale data to train robust machine learning models, data privacy and security concerns have become paramount. A clear demonstration of this growing importance is represented by the introduction of dedicated laws around the world. Federated learning (FL) can help mitigating these concerns by allowing multiple institutions to collaboratively train models without sharing sensitive data. This approach ensures that data remain local, thus preserving privacy, while using larger and more diverse datasets from different sources. Although the potential impact of FL can be easily appreciated, it remains a relatively new concept with growing research interest. In addition, the complexity introduced by all the degrees of heterogeneity that might occur in a FL pipeline (related to data, models, and system diversity), has produced many fragmented and parallel lines of investigation in the research community. Because of this, most of the problems remain open and best practice on how to implement the FL pipeline for real-life applications in the biomedical field remains weak.

With this thesis, our aim is to accelerate the adoption of FL in real-life scenarios applied to the biomedical field by advancing on two levels: (i) enable the researcher and developer community with suitable tools and (ii) address the convergence and scalability issues of the FL settings by validating them in different scenarios including multimodal data. The first step was achieved through three fundamental contributions: i.1 by actively developing modules of an open source software tool named OpenFL [187]; i.2 providing a ranking of all open source tools based on a proposed taxonomy of the key features required to implement FL pipelines; i.3, evaluating complex settings, such as Vertical FL setup, on multimodal data and releasing the code for further exploitation. The second step was well achieved through extensive validation of different FL settings (Horizontal and Vertical) using multi-modal real-

life dataset, with different deep learning models distributed on a cluster of multiple machines to draw more realistic conclusions.

With this thesis, we contributed to the advancement of the research applied to FL in the biomedical field, increasing awareness of the concept and open challenges, stimulating adoption by simplifying entry points and contributing to implementation in real-life environments to foster advancements in healthcare.

Contents

1 Introduction **1**

1.1 What is *Federated Learning* . . . 1
1.2 Motivation . . . 3
1.2.1 The Need . . . 3
1.2.2 Data regulations . . . 5
1.2.3 The opportunity . . . 7
1.3 FL: make it real . . . 10
1.3.1 Types of FL . . . 10
1.3.2 Types of Orchestration . . . 11
1.3.3 FL Main challenges and Research Areas . . . 13
1.3.4 Seeding work . . . 14
1.3.5 Objectives . . . 14
1.3.6 Research questions . . . 15
1.3.7 Execution plan . . . 15
1.3.8 Contribution summary . . . 16
1.3.9 Thesis organization . . . 18

2 State of the Art **21**

2.1 The roots . . . 21
2.2 Heterogeneity concept in FL . . . 23
2.2.1 Data Heterogeneity . . . 23
2.2.2 Model Heterogeneity . . . 25
2.2.3 System Heterogeneity . . . 26
2.3 FL in Biomedical environment today . . . 26
2.4 FL effectiveness on model convergence . . . 30
2.4.1 HFL aggregation functions . . . 31
2.5 Security and privacy . . . 35
2.5.1 Threats and Risks . . . 36
2.5.2 Protection approaches and strategies . . . 37

3 FeLebrities: a user-centric assessment of Federated Learning frameworks 41
3.1 Chapter organization . . . 45
3.2 Related Works . . . 45
3.2.1 FL settings . . . 46
3.2.2 FL challenges . . . 47
3.2.3 Study relevance . . . 48
3.3 Federated Learning Tools . . . 49
3.3.1 Methods and premises . . . 49
3.3.2 Exploring tools . . . 51
3.4 Tools popularity and level of adoption . . . 53
3.5 Priorities in FL tools for Research . . . 55
3.5.1 Evaluation metrics . . . 58
3.6 Discussion . . . 61
3.7 Conclusion . . . 64

4 MERGE: a model for multi-input biomedical federated learning 69
4.1 Introduction . . . 70
4.1.1 Federated learning in the bio-medical field . . . 72
4.1.2 Multi-input classification . . . 72
4.1.3 Federated Multi-input . . . 73
4.2 Results . . . 74
4.3 Discussion . . . 77
4.3.1 Conclusions . . . 79
4.3.2 Limitations and future work . . . 80
4.4 Experimental procedures . . . 81
4.4.1 Data and code availability . . . 81
4.4.2 Resource availability . . . 81
4.4.3 Architecture . . . 81
4.4.4 Datasets . . . 82
4.4.5 CoViD-19 dataset . . . 83
4.4.6 ADNI dataset . . . 84

5 FLavourite: Horizontal and Vertical Federated Learning setting comparison 89
5.1 Introduction . . . 89
5.1.1 Chapter organization . . . 91
5.2 related works . . . 91
5.2.1 Types of FL orchestration . . . 91
5.2.2 HFL and VFL in medical imaging . . . 92

5.2.3 Study relevance . . . 94
5.3 Method . . . 95
5.3.1 Data partitioning . . . 95
5.3.2 FL description . . . 97
5.4 Experiments . . . 99
5.4.1 Experimental set-up . . . 99
5.4.2 ADNI dataset . . . 101
5.4.3 Resources . . . 102
5.5 Results . . . 103
5.5.1 Q1: Which FL setting should be chosen when the context would allow both? . . . 103
5.5.2 Q2) In what way does the previous answer differ, given that HFL permits the exploration of various aggregation functions compared to VFL? . . . 106
5.6 Conclusions . . . 108
5.6.1 Future Directions . . . 108

6 Other contributions **109**
6.1 OpenFL: the open federated learning library . . . 109
6.1.1 Abstract [72] . . . 109
6.2 Model-Agnostic Federated Learning . . . 110
6.2.1 Abstract [161] . . . 110
6.3 A Performance Analysis for Confidential Federated Learning . 110
6.3.1 Abstract - PREPRINT version . . . 110
6.4 Submitted Patent . . . 111

7 Conclusions and future work **113**
7.1 Thesis status: what's been done . . . 113
7.2 Adherence to RQs and Objectives . . . 116
7.3 Key Learnings . . . 118
7.3.1 Federated Dataset . . . 119
7.3.2 Experimental environment . . . 120
7.3.3 Scientific aspects . . . 121
7.4 Future directions . . . 122
7.4.1 Stress multiclass . . . 123
7.4.2 Formalize model convergence problem . . . 123
7.4.3 Deep dive on VFL . . . 124

Ringraziamenti **161**

List of Figures

1.1 "if you want to go fast go alone, if you want to go far go together 2

1.2 Photo by Anastasiia Chepinska on Unsplash 5

1.3 We're better when we're united: photo by Clay Banks on Unsplash 8

1.4 Example of Horizontal FL pipeline. Different institutions owning local dataset with same features but belonging to different instances (patients, IDs, etc.). 11

1.5 Example of Vertical FL pipeline. Different institutions owning local dataset with different features but belonging to same instance across the participants (patients, IDs, etc.). 12

2.1 Picture taken from [253], showing the four different skew patterns in data heterogeneity 23

2.2 Picture taken from [56], showing articles published in the last ten years that were retrieved from the SCOPUS database using the keywords "federated learning", "federated machine learning", "medical image", and "medical imaging" 27

2.3 Table extracted from [221], showing the comparison of contributions and references retrieved by the survey. Last column mentioning "our study" refers to [221]. 28

2.4 Picture taken from [221], showing the percentage of FL applied in different diseases diagnosis research. 30

3.1 Data-to-model: example of legacy approach where data would move to a centralized training facility. Here the AI model is represented as a graph or neural network. 43

3.2 Model-to-data: example of a federated approach. A central unit called aggregator would clone and distribute copies of the same model to each collaborating institution. Each of them would then train its copy of the model using local datasets before sharing it back to the aggregator. As the name suggests, the aggregator would ultimately merge the models from different institutions and restart the process by sending out the latest aggregated version. 44

3.3 Collection pipeline implemented for the harvests. On each arrow is outlined the number of articles retrieved and filtered. Numbers between squared brackets refer to the outcome of H_2, while the curly brackets summarize the numbers from H_3. The output numbers of the second review are obtained by summing the numbers of a given category from the first review with the respective class of the second review. Please note that despite the September harvest returning 1292 articles, only 270 were new findings. The same consideration applies for H_3, with 1298 to 248 articles. 52

3.4 A subset of examples of selected articles for each category: "relevant" green, "uncertain" blue and "non-relevant" red. . . 54

3.5 Popularity growth rate: this graph illustrates which tools have been gaining more popularity in the community between March and September '22 GR_{12}, between September '22 and April 23 GR_{23}, and over an observation window of 374 days (roughly one year), GR_{13}. Tools that did not have a repository available in H_1 (March '22) were excluded from this chart as it would not be possible to track any evolution. 56

4.1 FL settings: **(a)** HFL, with collaborating institutions sharing the same features space but different samples. In this setting, only one model is being trained locally by each institution that is also shared and aggregated. **(b)** VFL, with collaborating institutions having different features for the same samples. In this case, each C_i has its model that flows into a unified 2nd-level model shared across the federation. 86

4.2 Accuracy values obtained by testing the *isolated* models on data belonging to other institutions. 87

4.3 A general FL infrastructure with Multi-Input Neural Network models 88

5.1 Example of our HFL pipeline for the Multi-Input model. Each i holds its own version of the ADNI dataset which is composed by tabular (TAB) and images (IMG) data. Each local model has the same architecture of the global model hosted by the aggregator. 92
5.2 Diagram representing our pipeline for the VFL with Multi-Input model experiments. Each i holds its own data, in this case respectively Tabular (TAB) and Images (IMG). For the Tabular-only and Image-only models, we would just remove one of the two i. 93
5.3 Detailed architecture of the Multi-Input Model. The Image-only model is obtained by using only the layers with cyan background. The Tabular-only MLP is presented with the yellow background. Finally, the connection layer is highlighted with the magenta background. 100
5.4 Aggregation functions compared in terms of Training Loss, Validation accuracy and validation F1 score. Each row represent the metrics obtained by running the training with the multi-input setting over the 5-folds cross-validation. As depicted, in most of cases the FedProx [128] algorithm seems to converge earlier and to higher values than FedAVG [154] . . . 107

List of Tables

1.1 Overview of Ph.D. Work Packages (WPs) organized by Years (Y) and semesters (H) 15

1.2 Main contribution of this Ph.D. path where the candidate appear as first author or equal contributor. 19

1.3 Contribution of this Ph.D. path achieved through external collaborations. 20

2.1 Threats and Vulnerabilities in ML Frameworks as summarized by [164] 38

3.1 Tool popularity table, March harvest H_1. Legend: This table shows the list of 36 tools retrieved in March with their respective Git indicators and the cumulative scores. The results are sorted from the most popular tools on the top to the less popular tools on the bottom. Indicated with NA are the tools for which the Git repository was unavailable or not publicly accessible. If not specified otherwise, all the repositories are Github projects. 65

3.2 Tool evaluation table: in documentation ("Docs.") and distribution channels ("Dist. channel") columns, P = Paper, Gh = Github, Gl = GitLab, W = Website. In the "Supported AI Type" column, DL = Deep Learning and ML = Machine Learning. Column "H/V" differentiates between "Horizontal" and "Vertical" FL, while the "Sync/Async" column indicates whether the tool supports synchronous S or asynchronous (A) workflows. The NM label refers to "Not Mentioned", meaning that the information did not appear as mentioned in the available documentation. 66

3.3 Tool scoring table 67

4.1 **Accuracy in centralized and federated setting.** Results (mean ± standard deviation) obtained with 5-fold cross-validation (centralized) and 5 averaged runs (federated). For each experiment setting (column) we highlighted the best performing model. 75

4.2 **F1 score in centralized and federated setting.** Results (mean ± standard deviation) obtained with 5-fold cross-validation (centralized) and 5 averaged runs (federated). For each experiment setting (column) we highlighted the best performing model. 75

4.3 **Accuracy in the isolated setting.** Results (mean ± standard deviation) obtained with 5-fold cross-validation (centralized) and 5 averaged runs (federated). For each experiment setting (column) we highlighted the best performing model. . 77

4.4 **F1 score in the isolated setting.** Results (mean ± standard deviation) obtained with 5-fold cross-validation (centralized) and 5 averaged runs (federated). For each experiment setting (column) we highlighted the best performing model. . 78

4.5 Statistics of the CoViD-19 CXR dataset with class-balance percentages 83

4.6 Main demographic and clinical data for the three ADNI study cohorts. Age is reported as mean ± standard deviation values, gender as the number of males/females, while APOE4 refers to the number of $\varepsilon 4$ alleles (0, 1, or 2, respectively). 85

5.1 Main demographic and clinical data for the three ADNI study cohorts. Age is reported as mean ± standard deviation values, gender as the number of males/females, while APOE4 refers to the number of $\varepsilon 4$ alleles (0, 1, or 2, respectively). 102

5.2 **Accuracy values for all models in each setting:** "Exp" describes the type of experiment which is also associated to the model used: "IMG" is the Image-Only model, "TAB" the Tabular-only model and "MI" the Multi-Input model. The table shows how accuracy changes depending on the type of experiment, whether is training or validation and on which setting: VFL, HFL or Centralized (CL). 103

5.3 **F1 values for all models in each setting,** "Exp" describes the type of experiment which is also associated to the model used: "IMG" is the Image-Only model, "TAB" the Tabular-only model and "MI" the Multi-Input model. The table shows how F1-scores change depending on the type of experiment, whether is training or validation and on which setting: VFL, HFL or Centralized (CL). 104
5.4 **Execution time among settings** This table shows the total Epoch time (Ep.), the compute time (CT) as the time spent by all components to compute their part of the pipeline, and communication time (NT) of each setting. All timings are expressed in seconds. The best values are highlighted. 105
5.5 **Execution time among settings** This table shows the Accuracy (left) and f1-score (right) obtained in each experiment setting by the two aggregation functions FedAVG [154] and FedProx [128]. The best values are highlighted. 106

Chapter 1

Introduction

1.1 What is *Federated Learning*

> "If you want to go fast go alone, if you want to go far go together"

When I think of Federated Learning (FL), I think of it as a natural direction of an evolutionary process for technology. It happened for the hardware and the evolution of computing units (CPUs), going from one core to multiple ones on the same chip set. It was applied to software, leaving monolitic approaches for more modular products based on microservices. When the resources of a system are not enough, more systems are pulled together to unlock new possibilities. As technology advances, science follows, and vice versa. A clear demonstration of this phenomenon are high performance computing systems (HPCs), where a collection of many separate servers (computers), called nodes, are connected via a fast interconnect. The idea of building a powerful distributed system, composed of many subsystems, tuned for maximum performance is at the base of many engineering and scientific research laboratories and businesses or modern cloud infrastructures. Famous examples of HPC systems [7] that have considerably enabled major state-of-the-art advancements in scientific environments are:

- The HPC system of the Frontier Development Lab at Nasa
- The Openlab cluster at the European Council for Nuclear Research (CERN), in Geneva
- The COSMOS supercomputer, launched by Prof. Stephen Hawking, hosted at University of Cambridge

Figure 1.1: "if you want to go fast go alone, if you want to go far go together

This is a short, representative, and non-exhaustive list of HPC centers around the world where some of the most advanced research is carried out by various laboratories.

I always liked to challenge myself by asking what could be the next innovation step, what obstacle we should see removed. I guess some time, but in this age of Deep Learning (DL) everywhere now called "Artificial Intelligence" (AI), I could only stand in front of the unanswered question:

> *How are we going to unlock new capabilities to push the boundaries forward? What comes next?*

These types of system are the pinnacle of technological computing; however, there is a hidden aspect that requires them to evolve: Data. In a standard HPC system, the data need to be available in the system itself. Being extremely optimized for maximum performance, every bottleneck has to be removed by design. This means that having access to the latest most powerful HPC system but having to wait to load data from external sources that could introduce delays or slow down the computation, might not be contemplated.

This aspect represents a huge limitation considering that not all data are always available in the same premise or easily accessible when hosted

across external infrastructures. When data is distributed among different organizations or storage systems, the challenge is to find the optimal path to retrieve it in the fastest and most effective way. In addition, it is worth noting that accessing and using data leads to two separate types of challenge. Finding the fastest and most cost-effective way to access certain information does not take into account whether we have permission to use those data for our purposes.

Here is where FL can help. FL is a machine learning approach in which multiple entities, often referred to as clients or collaborators, work together to train or validate a model while keeping their data decentralized. This method aligns with the spirit of the initial quote, emphasizing the power and need for collaboration to reach for farther goals. This definition applies regardless of the type of system we are considering. HPCs are great examples of datacenter facilities, but conscious of how the approach can unlock access to new data, it is reasonable to think that it can of course be applied to smaller computing premises, like smartphones or smart domestic environments. In general, a device or appliance that works as a data source by generating or acquiring new information, which is also equipped with a computing unit, can be turned into a member of a FL pipeline.

In FL, each participant contributes to a common goal without sharing their individual data, thus going "together" towards building a robust, shared model. This collaborative effort can lead to far-reaching advancements in AI, as it combines diverse datasets that are not limited by the constraints of centralization. It is a journey where the collective intelligence and contributions of many can lead to greater achievements than any single entity could accomplish alone.

The quote encapsulates the essence of FL, where the journey to develop powerful AI models is not a sprint, but a marathon, requiring the collective effort and persistence of many to reach new heights of innovation and utility.

Following along with the marathon analogy, as happens in real life with marathons run by humans, there is more to a pair of comfortable shoes (hardware) and a path with rules to follow (software) to complete the task: motivation is key. The same applies to FL.

1.2 Motivation

1.2.1 The Need

> "Inclusion is not a matter of political correctness. It is the key to growth." — Jesse Jackson.

Not long ago, the entire world pressed the "pause" button due to coronavirus restrictions (Fig.1.2). A sad time where all intensive care units (ICUs) in all hospitals around the world were constantly running at full capacity. Tough and delicate times where everyone could appreciate the need for advanced research laboratories. In addition to the various versions of vaccines that help mitigate the effect of the virus, the ability to effectively prevent transmission in each community and globally has remained an urgent task since its outbreak. In those days, a good way to obstacle the rapid spread of infection consisted of identifying the infected individuals and isolating them. That is when screening computed tomography (CT scan), X-ray and other acquisitions became more and more instrumental in separating patients infected with COVID-19 from others [83, 200, 214].

Several studies in the literature have exploited Deep Learning (DL) approaches to accurately identify infection from a medical image. However, most of them relied on locally available or publicly downloadable data. This means that for as good as these results could be, the effectiveness of the proposed methods was only evaluated on restricted datasets. The question I asked myself is:

> *What would happen if we could combine the dataset, meaning all the acquisitions, from all the hospitals in the world, to train and validate models for COVID-19 detection?*

I think everyone can easily appreciate the impact that such technology would have on the good of everyone. Regardless of the specific application, that model would have access to a library of knowledge never experienced before. And, as happens with physicians and nurses, the more exposure and experience they gather in critical situations, the stronger their learning will be. The original question led me to a key consideration: the elements required to build such scenario would be mainly 3:

1. Data: for training and validation
2. Models: for addressing specific tasks (i.e. segmentation, detection, classification, etc.)
3. Infrastructure: powerful hardware coupled with software to scale

If we extend these ingredients at world level including all the hospitals on the planet, we already have data, we also have mature state-of-the-art models to tackle specific tasks, and as mentioned above, we can rely on HPCs systems to fulfill the compute required to train and validate models at scale. This is where another interrogative comes up:

Figure 1.2: Photo by Anastasiia Chepinska on Unsplash

We have all the ingredients required to make it happens: but why it is not happening?

Mainly because of two aspects: data regulations and practical feasibility. Where present, data regulations can limit access to sensitive data belonging to patients. On the other hand, in cases where data regulations are not in place or data can be accessed freely, the problem of how to collect distributed data remains open. In the COVID-19 example discussed above, gathering the distributed data might not be easily achievable even if it is done under the assumption of dealing with publicly available information. For example, given that all hospitals on the planet would agree in sharing their data, how would one practically gather those data? Medical records can be pretty cumbersome and transferring data over a cabled network might congestion the communication channel.

1.2.2 Data regulations

"Once you've lost your privacy, you realize you've lost an extremely valuable thing." — Billy Graham.

As a native Italian, I am proud to live in the European Union (EU), which introduced the General Data Privacy Regulation (GDPR) in 2016. The definition of Privacy provided by the EU council has its roots in what is recognized as an "absolute human fundamental right: Human Dignity". It is a matter of personal dignity and autonomy [1].

It is good to know that Europe is not the only region with such high consideration of privacy. Infact, the majority of countries across the five continents can demonstrate dedicated efforts to recognize privacy among their values, either in their constitution or in other provisions.

However, despite the fact that most countries share the same good general sentiment about the social value that privacy represents for civil societies, the same sensitivity is not reflected for data protection. At least not yet.

This means that data regulations are rising almost everywhere, at different pace, following different priorities and with varying levels of success. EU is a great example of data regulation with inclusion as the design principle. It is inclusive because it applies to all digital records of an individual. Other examples of data regulations with a limited range of action were explicitly created to govern the access and use of Health Data. Is the case of the *Health Insurance Portability and Accountability Act* (HIPAA) [194] in the USA, the *Protection of Personal Information Act* (POPIA) [4] or the *Data Protection ACT* (DPA), which was created in 2018 following the GDPR [97]. These regulations aim to give individuals greater control over their personal data and to hold organizations accountable for how they handle and protect these data.

The fact that these regulations were created with a specific application area is a critical aspect, as it depicts a clear image of where we are at the moment and what to expect next. Europe has prioritized data regulations [97] over technological advancement, while most of the other advanced countries have followed the opposite path. Despite this phenomenon, it is interesting to see how countries such as China that have invested a great deal of effort to build strong AI-based expertise and products have recently felt the need to redirect government energies from innovating to regulating [31]. It is the case of the *Personal Information Protection Law of the People's Republic of China*, simply referred to as PIPL, introduced in 2021.

I truly believe that people have the right to decide who can access information, especially when it comes to their physical and mental health status. This, of course, goes beyond the application we mentioned above about Coronavirus detection. It includes and is not limited to any application in the healthcare landscape. Medical privacy ensures that these sensitive details are not disclosed to unauthorized individuals, which could result in discrimination, stigma, or personal embarrassment. Moreover, keeping medical records

confidential is essential to maintain trust between patients and healthcare providers [57, 92]. This trust is the foundation of the therapeutic relationship and is crucial for accurate diagnosis and effective treatment. On a larger scale, protecting medical information is important in preventing financial fraud, identity theft, and data misuse. Thus, privacy in healthcare is not just a preference, but a vital safeguard to protect the integrity of healthcare systems and the personal rights of individuals [85].

However, I also believe that data protection should be a must in all fields and not limited to healthcare or delicate aspects of human life. With the increasing ability of DL models to *see, listen, read* and capture multimodal context more and more effectively, it is crucial to enable people with the ability to fulfill their right to protect their privacy and personal information. With the increasing amount of data generated and shared online, the risks of data breaches and unauthorized access have grown significantly. Effective data protection measures help prevent identity theft, financial loss, and confidentiality breaches that can have serious personal and business consequences.

However, driven by our motivating factor of building an impactful AI application for the good of humanity, like the Coronavirus detection example shared before, there is a key consideration -which acts as the elephant in the room- that needs to be addressed: although protecting sensitive information is a critical mission from a governance perspective, from an AI perspective, introducing all these regulations limits access to data. This is why FL represents a viable way to access sensitive data in a privacy-compliant way.

1.2.3 The opportunity

> "Strength lies in differences, not in similarities." — Stephen Covey.

Among the benefits of the FL approach, in addition to being privacy-compliant, there are scientific advantages. When it comes to Machine Learning (ML) algorithms, diversity is an added value. The more diverse and complete the training dataset, the more robust and reliable the model will be [212].

Going back to the Coronavirus example mentioned above, unlocking access to a multitude of distributed set of data can enrich the training or validation experience, resulting in more robust models for the good of all the institutions taking part to the federation. This aspect is known as the ability of the model to generalize [76, 110], that is, to preserve performance even with unseen samples [153].

Figure 1.3: We're better when we're united: photo by Clay Banks on Unsplash

As mentioned above, FL is a distributed machine learning approach that trains models across multiple decentralized devices that hold local data samples, without exchanging them. So far we have only referred to examples involving big datacenters or HPCs computing facilities, but is worth nothing to mention that the paradigm can be applied to smaller or edge devices.

An example of FL applied to portable devices is the one exploited by Google [251] in 2017. The team trained a recurrent neural network language model using on-device learning, for the purpose of next-word prediction on a virtual keyboard for smartphones. Given the example, finding ways to leverage the FL paradigm for edge devices is a creativity effort. Using the Coronavirus example, [182] et al. built an automatic COVID-19 diagnosis based on multimodal information coming from edge devices. The intriguing aspect is that, thanks to FL, new horizons and opportunities are unlocked. Is FL example used to facilitate in-home health monitoring, by training a global model from distributed homes, while preventing data leakage by keeping user data locally [242]. Moreover, since FL attracts large computation and dataset resources from a number of health data clients to train or validate AI models, the quality of the health data training, for example, accuracy, would be significantly improved, which might not be achieved by using centralized AI approaches with fewer data and limited computational capabilities [171,256]. By extending FL beyond the healthcare scenario, one can easily imagine how many opportunities this approach can unlock due to its benefits.

In summary, the key benefits of FL include:

- **Privacy Preservation**: Since data do not need to be transferred to a central server, sensitive information remains local, reducing the risk of data breaches and enhancing user privacy.

- **Reduced Latency**: By performing computations closer to the data source, FL can reduce the latency associated with data transfer and centralized processing.

- **Scalability**: FL enables the use of a large number of devices for parallel model training, thus improving the scalability of the learning process.

- **Cost Efficiency**: By leveraging local computation resources, FL can potentially lower the computational and storage costs compared to traditional centralized approaches.

- **Personalization**: Models can be tailored to the specific data characteristics of each client, leading to improved performance and user personalization.

FL presents significant opportunities for both data centers and edge devices by enabling decentralized data processing while preserving data privacy. For data centers, FL reduces the need to centralize voluminous datasets, thereby decreasing storage costs and enhancing computational efficiency. This approach allows data centers to focus on model aggregation and refinement, rather than raw data handling. On the other hand, edge devices benefit from improved personalized user experiences, as models are trained directly on local data, which can lead to reduced latency and improved responsiveness in applications. Furthermore, the privacy-preserving nature of FL encourages users to participate more willingly, knowing that their sensitive data remain on the device, fostering a collaborative and secure AI ecosystem.

1.3 FL: make it real

As introduced in Section1.1, FL is a machine learning technique that enables decentralized devices to collaboratively learn a shared model while maintaining the training data localized. There are several types of FL, each suited for different scenarios that depend on how data are originally partitioned or distributed among collaborators. We are going to share an overview of the main characteristics that can describe different federated settings. More related work and background will be provided in the state-of-the-art chapter.

1.3.1 Types of FL

Depending how data are organized among the various institutions taking part to the federation, different FL architecture can be implemented.

- **Horizontal Federated Learning (HFL)**, also known as sample-based FL, is applied when datasets share the same feature space but differ in samples. In this scenario, different entities that hold similar types of data participate in a collaborative learning process [72, 156, 212, 217]. A typical HFL pipeline is shown in Fig. 1.4

- **Vertical Federated Learning (VFL)**, also known as feature-based federated learning, occurs when datasets share the same sample space but differ in features. This type is useful when different entities hold complementary information about the same users, such as banks and insurance companies [46, 53, 126]. A typical VFL pipeline is shown in Fig. 1.5

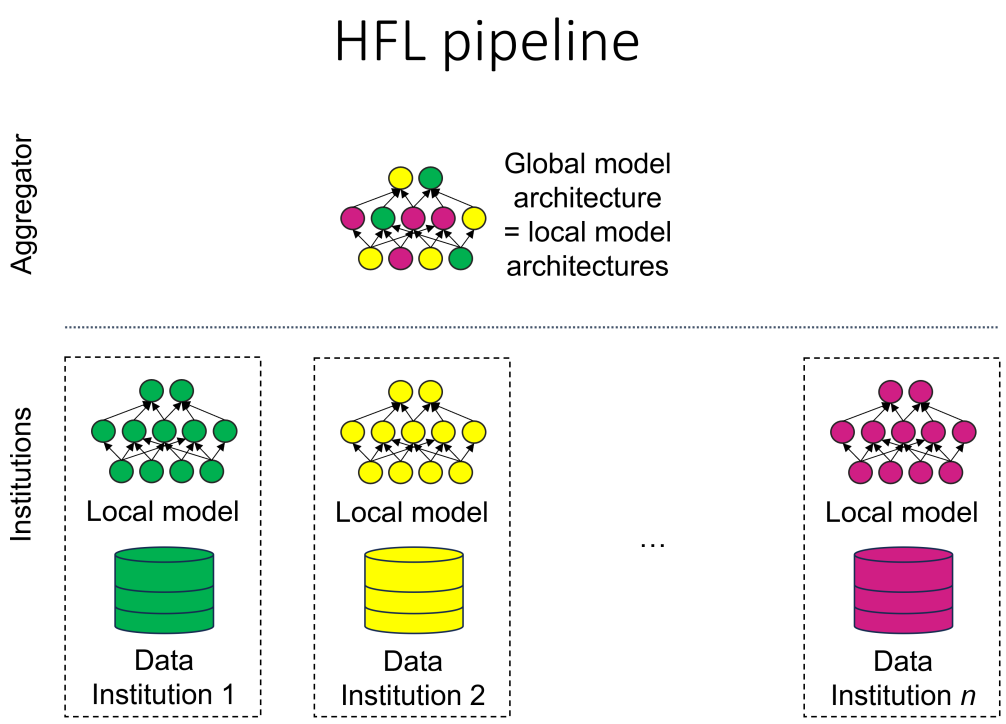


Figure 1.4: Example of Horizontal FL pipeline. Different institutions owning local dataset with same features but belonging to different instances (patients, IDs, etc.).

- **Federated Transfer Learning (FTL)** is useful when both the sample and the feature space do not fully overlap but have a small intersection. This method is particularly beneficial in scenarios where data distributions vary significantly across different entities but there is still a small common subset that can be leveraged to improve learning [111,141,232].

1.3.2 Types of Orchestration

As we have previously discovered, a federation can involve different types of client institutions. Depending on whether the process is taking parts among data centers or across "smart" Internet-of-things (IOT) devices at the edge, we might observe different strategies for orchestrating the processes.

- **Synchronous Orchestration** requires all participating devices or nodes to sync up at regular intervals [212] [60]. This means that data updates from all the nodes need to be received before proceeding to the next

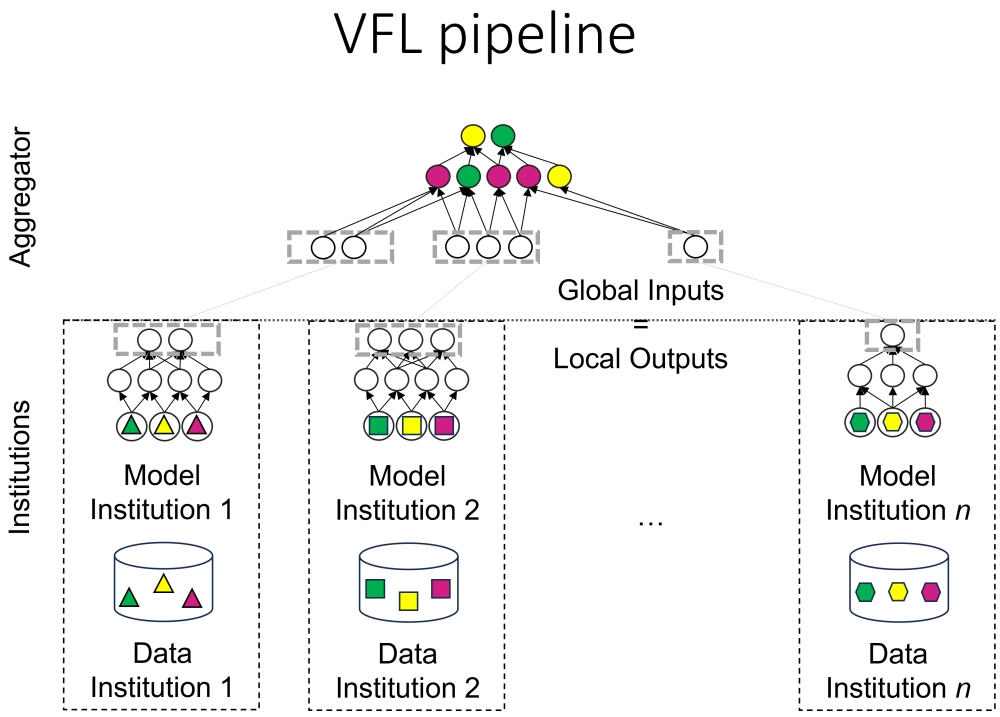


Figure 1.5: Example of Vertical FL pipeline. Different institutions owning local dataset with different features but belonging to same instance across the participants (patients, IDs, etc.).

iteration of training (or validation) process, ensuring a unified pace and consistency across the network. Synchronous orchestration can be subject to inefficiency and delays when certain devices are slower or experience connectivity issues. This is why is more adopted among datacenters with more stable connectivity, powerful equipment to ensure low computation times and limited amount of collaborators.

- **Asynchronous Orchestration** allows nodes to operate independently and send updates as soon as they are available, without waiting for others [50,170,205]. This can significantly improve the efficiency and speed of the training (or validation) process as it eliminates waiting times. Asynchronous orchestration may come at the cost of consistency and may require additional mechanisms to handle the variability of updates from different devices. This is why is more adopted for federations of smart-edge devices, with a varying (potentially big) number of con-

nected devices that might be subject to failures (sensors) [50,104], weak connectivity or different latency and working frequency.

1.3.3 FL Main challenges and Research Areas

Regardless to what FL setting is adopted by a given federation, the research community is currently focused on three main areas:

1. **Aggregation Functions**: whether is a HFL, VFL or FTL setting, these functions play a crucial role in ensuring the effective combination of locally trained models into a robust global model [184]. The choice of aggregation function affects the overall reliability, effectiveness, and security of the FL pipeline. Through the study and optimization of these functions, researchers can improve their capacity to address challenges associated with data heterogeneity, communication efficiency, and adversarial threats [132].

2. **Privacy and Security**: since FL can help accessing data in a privacy-compliant way, it is inherently introducing risks related to data leakage, adversarial attacks, and potential breaches of sensitive information [14,158]. By focusing on privacy-preserving techniques, such as differential privacy (DP) [63], secure multi-party computation (SMP) [66], and homomorphic encryption (HE) [165], researchers can ensure that participants' data remains confidential and secure.

3. **Communication Efficiency**: communication efficiency is an essential aspect for any FL setting because is related to many of the key aspects of this decentralized paradigm, such as:

   - Number of clients (or collaborators) [158, 245, 268].
   - Network bandwidth: can impact the size of models and gradient updates [158, 245].
   - System heterogeneity: can impact execution times within and across members [8, 132, 148].
   - Statistical (data) heterogeneity: can impact model convergence, hence computation time, hence update exchange [132, 148, 157].

   Regardless of what setting or which orchestration strategy have been chosen, training or validating models across multiple nodes or edge devices requires frequent communication updates. Efficient communication protocols ensure that these updates are transmitted quickly

and with minimal bandwidth usage, reducing latency and the overall burden on the network. The main factors influencing communication efficiency [210].

1.3.4 Seeding work

"OpenFL: The Open Federated Learning Library"

All the experimental work performed during this Ph.D. path was done using an FL tool called OpenFL [72] implemented by Intel Labs. It is worth mentioning that we played a role in the development of the tool itself, not only through an extensive testing and review cycle to provide feedbacks, but also at the software development level. More in detail, the active and direct contributions to the tools concerned: the design and development of the containerized version of the tool through Docker containers. This would simplify portability and the adoption of the tool across various infrastructures by using the same common low-level software layer called Docker. The code of this contribution can be found on the official official OpenFL repository.

The tool is now part of the Linux Foundation and is available to the developers community as free software. Although not being the first author limits the extent to which we can elaborate on this achievement (no dedicated chapter, but abstract reported in Chapter 6), its global recognition and usage underscore its importance for this work and for the broader research community.

More details on the additional contributions and results achieved due to this seeding work are reported in Section 1.3.8.

1.3.5 Objectives

This project was born with extremely ambitious goals for bringing clear progress to the State-of-the-Art through three main contributions:

1. Devising and assessing novel and revised approaches to help addressing and understanding how hyper-parameter tuning, data distribution, collaborators settings and aggregation functions can affect convergence and performance

2. Investigating ways to improve security and privacy issues within a Federated Learning pipeline to increase usability and reliability.

3. Proposing and investigating novel use-cases and areas of adoption - from an application perspective - for Federated learning and others distributed intelligent systems set-up.

Table 1.1: Overview of Ph.D. Work Packages (WPs) organized by Years (Y) and semesters (H)

	Y1 - **1H**	Y1 - **2H**	Y2 - **1H**	Y2 - **2H**	Y3 - **1H**	Y3 - **2H**
WP1						
WP2						
WP3						
WP4						
WP5						

1.3.6 Research questions

To ensure adherence to these ambitious objectives, we have identified and built a plan around two driving research questions, reported in the following:

- **RQ 1)** How can the development and analysis of federated learning tools and methodologies improve the efficiency and efficacy of multi-modal data integration in biomedical applications?
- **RQ 2)** How to exploit the advantages and address the challenges of Horizontal and Vertical federated learning approaches in the same context?

It is worth mentioning that the contribution of this Ph.D. work is focused not only on scientific aspects but also on technological contributions.

1.3.7 Execution plan

The project was developed over 3 years and consisted of 5 Work Packages (WP). WP1 was devoted to further appreciate the State of The Art; WP2 focused on implementing an FL set-up that would be used as ground tool to run experiments for the next WPx. WP3 was devoted to addressing the data challenges. WP4 concerned the security and privacy aspects of the federation. WP5 included use-cases, applications, and dissemination activities. Depending on the type of experiments performed, the WP3, WP4, and WP5 activities could overlap in some cases. Table 1.1 captures the initial proposal with the time allocation estimate for each activity.

The actual work plan is slightly different from that indicated in Table 1.1, however, all activities carried out adhere to the provided structure. Activities such as data set preparation, maturity status of technology and tools,

and testing, in some cases, took more than estimated. An additional factor that is worth mentioning is the maturity of the State-Of-the-Art regarding Federated Learning that was overestimated at the beginning, but required more investigation at the foundation level. More details on this aspect are discussed in the dedicated chapter 2.

The specific activities involved the investigation of Vertical and Horizontal FL settings, distributed across multiple nodes and simulated (on single node) environments, for disease classification tasks based on bidimensional and 3D images and tabular data.

1.3.8 Contribution summary

The three-year journey has led to various contributions to help address the two research questions highlighted in Section1.3.6. A summary of all the contributions, including external collaborations in terms of peer-reviewed publications and main dissemination activities, is provided below.

External collaborations refer to activities conducted with industrial or research partners for which the candidate does not appear as the first or second author despite providing relevant contribution to the context. The abstract of these contributions are provided in Chapter 6.

1. Contributed to the creation of an FL tool for the research community. The tool was published in Physics in Medicin & Biology: OpenFL: the open federated learning library [72]

2. "FeLebrities: a user-centric assessment of Federated Learning frameworks", Accepted Journal Paper (IEEE Access) [189]

3. "MERGE: a model for multi-input biomedical federated learning", Accepted Journal Paper (Cell-Patterns) [39]

4. "FLavourite: Horizontal and Vertical Federated Learning setting comparison", to be submitted

5. "Model-Agnostic Federated Learning", Published Conference Paper (Euro-Par 2023: Parallel Processing) [161]

6. "A Performance Analysis for Confidential Federated Learning", Submitted at AAAI Conference - Under review.

On a different perspective, the Ph.D. path can be described identifying the following main contributions:

1. Technical development and improvement of a free FL tool to enable the research community in the domain

2. Exploitation and investigation of techniques concerning data fusion, multimodal training and model convergence

3. a Federated Multi-Input Architecture for multimodal biomedical data, validated with COVID-19 and Alzheimer's datasets.

4. - the comparative experimental Study of Horizontal FL (HFL) vs. Vertical FL (VFL) offering reproducibility and implementation guidelines, demonstrating clear advantages of HFL with respect to VFL for multimodal data in terms of accuracy, computational and synchronization needs with respect to a clearly optimal but less practical centralized solution.

5. The Security and Confidentiality Analysis: Benchmarks using LibOS and TEEs demonstrating feasibility of secure, end-to-end FL with measured performance loss.

As highlighted in Section 1.3.4, among the external collaborations is the publication with Intel Labs [72]. Although this thesis remains focused on the achievement as the first author or equal contributor, it is worth mentioning the specific contribution provided. The efforts invested did not only help advance the development of the tool in terms of feature availability and maturity, but also included dissemination activities through hands-on workshops for other Ph.D. students, in an international Summer School in Neuro-Imaging, Neuro-Oncology, Neuro-Science 2022, 1^{st} workshop on FL at webconf2023 at the University of Austin, TX in 2023, the creation and delivery of a course for Master students (of 20h of front lecture) on Federated Learning at the University of Verona in 2023 and the delivery of a hands-on tutorial on FL at the 21^{st} International Symposium on Biomedical Imaging ISBI2024 in Athen, Greece.

Additionally, given the lack of suitable demo templates for NLP and Medical imaging, we implemented hands-on tutorial to allow beginners to get closer to the topic and facilitate the code implementations for their own research activity. The NLP tutorial can be found at https://github.com/intel/openfl/tree/develop/openfl-workspace/keras_nlp. The Medical imaging examples are based on the MedMnist dataset which has a 2D and a 3D version; therefore, two codes were produced: the Federated MedMnist 2d version that can be found at https://github.com/intel/openfl/tree/develop/

openfl-tutorials/interactive_api/PyTorch_MedMNIST_2D and the 3D example that can be found at this link. Both MedMnist-based examples have also been made available through Google Colab to facilitate broader adoption.

A more comprehensive review of all the main activities can be found in Table 1.2, while external contributions are detailed in Table 1.3.

1.3.9 Thesis organization

The rest of this thesis is organized as follows: in Chapter2 a comprehensive overview of the state-of-the-art is provided for the key aspects of this work. Chapters 3, 4, and 5, the main contributions are reported as original publications respectively focussed on understading the FL tool landscape, investigating multimodality model convergence on clinical use-cases and understanding convergence behavior when comparing HFL and VFL settings. Chapter 6, shares the abstracts of the external contributions. Finally, conclusions and future perspectives are provided in Chapter 7

Table 1.2: Main contribution of this Ph.D. path where the candidate appear as first author or equal contributor.

Contribution	Status	Venue	Title (and Link)
Paper	Published	IEEE Acces: Journal	FeLebrities: A User-Centric Assessment of Federated Learning frameworks [189]
Paper	Published	Cell-Patterns: Journal	MERGE: a model for multi-input biomedical federated learning [39]
Paper	Published	STAR-Protocol	Protocol for training MERGE: A federated multi-input neural network for COVID-19 prognosis [40]
Patent	Rejected	Intel Corp.	Dynamic bootstrapping of new devices for learning federations in smart connected environments
Paper	Under Review	IEEE Acces: Journal	FLavourite: Horizontal and Vertical Federated Learning setting comparison

Table 1.3: Contribution of this Ph.D. path achieved through external collaborations.

Contribution	Status	Venue	Title (and Link)
Paper	Published	Physics in Medicin & Biology	OpenFL: the open federated learning library [72]
Paper	Published	European Conference on Parallel Processing, Europar Conference	Model-Agnostic Federated Learning [161]
Worskhop	Organized	7^{th} IEEE, Deep Learning Security and Privacy Workshop	A Performance Analysis for Confidential Federated Learning [36]

Chapter 2

State of the Art

This chapter provides a summary of background information on the origin and updates on latest advances in FL for key areas relevant to this research work. The key areas focuses on synchronous large-scale institutions (computing clusters of servers over edge devices), applied in biomedical environments for both settings HFL and VFL.

2.1 The roots

The key initial challenge of the "distributed approaches" was centered on communication efficiency [211]. In this work, the focus was to minimize some stochastic (population) objective (e.g. expected loss or generalization error)on stochastic optimization (learning) settings, for optimization functions defined as $\phi_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \ldots, m$ distributed across m machines:

$$\phi(w) = \frac{1}{m} \sum_{i=1}^{m} \phi_i(w). \tag{2.1}$$

The goal was to reduce the number of steps required to converge to an optimal solution. The proposed method, called "DANE" from Distributed Approximate NEwton, paved the way for one of the first papers in the domain introducing the "Federated Optimization" concept [121]. In this work the authors extended the previous study to heterogeneous data distributed over an extremely large number of nodes, whilst preserving the main goal of training a high-quality centralized model. The function to optimize followed the shape of the one proposed in 2.1 [211], but included linear logistic regressions, support vector machines, and more complicated algorithms such as conditional random fields or neural networks. Thanks to this advancement,

the work was further evolved into the contribution that gave the name "Federated Learning", starting a new research branch [154]. The study proposed the federated averaging algorithm (FedAvg) to consolidate the contribution of each collaborating institution by averaging (weighted average) their gradient updates in a horizontally partitioned environment (HFL). The work was targeting mobile devices for two reference use-cases: image classification, for predicting which photos are most likely to be viewed multiple times in the future, or shared; and language models, which can be used to improve voice recognition and text entry on touch-screen keyboards by improving decoding, next-word prediction, and even predicting whole replies. In addition to the proposed FedAvg algorithm, the authors shared some practical insights on how the number of local training rounds performed by each individual collaborator impacted the model convergence. The more local rounds, the faster the convergence. However, for a very large number of local epochs, FedAvg can plateau or diverge. This milestone also focused on edge devices, such as mobile devices and smart phones. When it comes to federation between data centers, the cost of communication is minimal, while computational costs are predominant.

Then, different approaches have been proposed to leverage the contribution of multiple institutions. Some studies examined collaborative learning techniques that preserve privacy for medical models, in which institutions train one after another in a serial chain instead of simultaneously in parallel [44]. Sheller et al. [212], named this implementation institutional incremental learning (IIL) and extended it to cyclic institutional incremental learning (CIIL). In IIL, each institution trains the model and then passes it to the next institution for training, until all have trained once. CIIL repeats this process in a loop, fxing the number of training epochs at each institution and cycling repeatedly through the institutions. The study highlighted how serial training methods of IIL and CIIL can lead to what is known in the literature as 'catastrophic forgetting', where the trained model strongly favors the most recently seen data [73]. The repetitive cycles and limited epochs per institution performed during CIIL enable it to make gradual progress, despite the forgetting, resulting in better models than IIL produces [44]. Hence, the study concluded that FL was the best approach to maximize training performance compared to centralized settings (that is, scenarios in which all data are gathered together).

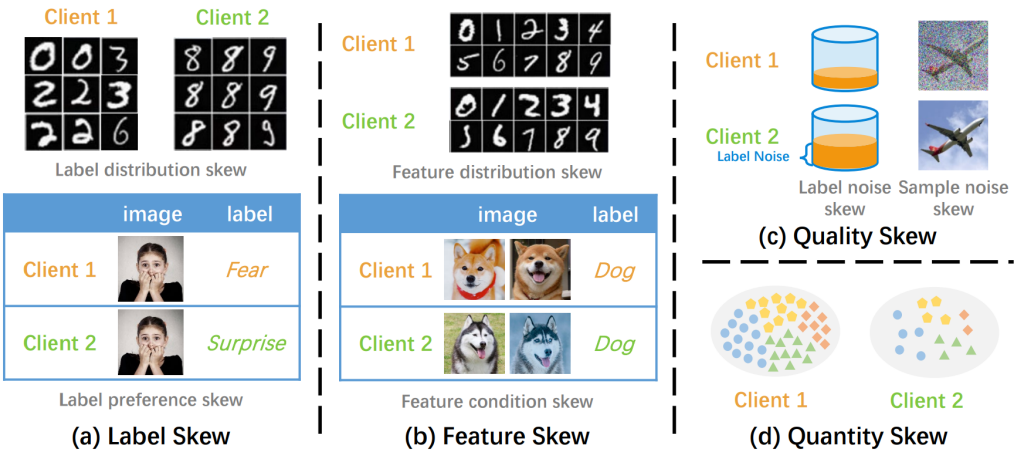


Figure 2.1: Picture taken from [253], showing the four different skew patterns in data heterogeneity

2.2 Heterogeneity concept in FL

As we have learned from Chapter 1, from a philosophical perspective, FL is all about bringing together diverse institutions to achieve a common objective. If we had to represent FL using one term, it would be *heterogeneity*. In general, *heterogeneity* refers to the quality or state of being diverse in character or content. Indicates a mix of different elements, often leading to complexity and varied characteristics within a system. The diversity of data, model architectures, physical infrastructure, and knowledge is what makes FL fascinating but also extremely challenging at the same time. This diversity is present in any single aspect of an FL pipeline and is often captured in several works such as *heterogeneity*.

This section will help setting the scene on the different types of *heterogeneity* that needs to be considered when dealing with FL pipeline by providing a definition and research example for the following aspects: Data, Models and Infrastructures.

2.2.1 Data Heterogeneity

The most obvious and known type of heterogeneity in FL is related to the data. Data across different clients can be nonidentically distributed (non-i.i.d.) and may have different distributions, sample sizes, and feature spaces [39, 154, 187]. In the literature, data heterogeneity is also often referred to as *statistical heterogeneity*. In fact, this diversity is somehow encouraged and expected, as AI can benefit from having a training dataset where more

representations or views of the same item are provided. However, on the other hand, this makes it challenging to aggregate models, as each model update could reflect different data patterns. Further context on the various aggregation functions is provided in section 2.4.

In [27] and [253] the authors propose a categorization of the variables that concur to make data non-I.I.D. By summarizing both contributions, we can draw a more complete categorization. The proposed categories are built on the assumption that the clients do not have access to data drawn from the same underlying distribution $\mathcal{P}$; therefore: $-\mathcal{P}_i \neq \mathcal{P}_j$ for all pairs of clients i and j. Examples of the types of heterogeneity described in the following are provided in Fig. 2.1.

- **Feature distribution skew** [27]: The $\mathcal{P}_i(x)$ marginal distributions vary between clients. This is the case of HFL, where the input features are not evenly distributed between clients despite belonging to the same feature space. In the formula, x represent a data sample used as input to the AI (federated) model.

- **Label distribution skew** [27]: The $\mathcal{P}_i(y)$ marginal distributions vary between clients. In the formula, y refers to an instance label used. This is the case where the data labels are not evenly distributed between clients. For example some clients only have access to data from a subset of all possible labels for a given task. This phenomenon is commonly encountered in practical applications in which data collection or annotation is inconsistent.

- **Concept shift (same features, different label)** [27]: The $\mathcal{P}_i(y|x)$ conditional distributions vary between clients. In this scenario, different labels are assigned for the same features across clients. For example, client i labels all cat images as *cat*, but client j labels all cat images as *dog*. In [253], the authors propose to enrich this category with two specifications: *label distribution skew* and *label preference skew* to distinguish the cases where: label distribution might be different for different clients even if feature distribution is shared, i.e. $\mathcal{P}_i(y) \neq \mathcal{P}_j(y)$ but $\mathcal{P}_i(x|y) = \mathcal{P}_j(x|y)$ or label distribution is different across clients despite having the same feature distribution, i.e. $\mathcal{P}_i(x) = \mathcal{P}_j(x)$ but $\mathcal{P}_i(y|x) \neq \mathcal{P}_j(y|x)$.

- **Concept shift (same labels, different features)** [27]: The $\mathcal{P}_i(x|y)$ conditional distributions vary between clients. In this case the different features across clients are labelled with the same label. For example, client i labels all *cat* images as *cat*, but client j labels all dog images

as *cat*. In [253], the authors propose to enrich this category with two specifications: *feature distribution skew* which implies consistent label distribution but different feature distribution, i.e. $\mathcal{P}_i(x|y) \neq \mathcal{P}_j(x|y)$ even if $\mathcal{P}_i(y) \neq \mathcal{P}_j(x)$, and the *feature condition skew*, which captures the case when feature distribution varies among clients even if label distribution is the same: $\mathcal{P}_i(x|y) \neq \mathcal{P}_j(x|y)$ but $\mathcal{P}_i(y) = \mathcal{P}_j(y)$.

- **Quality skew** [253]: captures the scenario where labels or data collection qualities are inconsistent across the various clients, for example, due to different machinery used for data acquisition or different processing techniques. Again a twfold subcategorization is proposed to discriminate by the proportion of noisy labels contained differs across clients or the variation in sample noise at feature level.

- **Quantity skey** [253]: Describes the scenario where the amount of data stored and owned by each institutions is different from one client to another. On one hand, this unbalanced situation can justify the need for FL to compensate dataset that would not be enough to train a model if taken alone, but on the other, introduce additional challenges for model aggregation strategies.

It should be noted that the categories described above include both the HFL and the VFL settings. An additional aspect to include is related to the multi-modal dataset, which can provide another degree of complexity when dealing with data heterogeneity.

In this work, we explored various scenarios to capture most of the data heterogeneity types described in the provide categorization, including multi-modal examples [39, 40].

2.2.2 Model Heterogeneity

In FL model heterogeneity refers to the presence of diverse models in different client devices that participate in the learning process. This is particularly emphasized in VFL scenarios [184], where each client device can utilize a different architecture or hyperparameter tailored to its specific needs or capabilities. In traditional HFL, where the local model is a copy of the global architecture as shown in Fig. 1.4, model heterogeneity still applies and is more related to ways of sharing or aggregating hyperparameters or local model weights. Issues might arise when the global model is aggregated, as the combined gradient updates may not accurately reflect the data from any single client. A detailed analysis on these aspects is provided in Section 2.4.

Solutions such as model aggregation techniques (discussed in 2.4), knowledge distillation, or custom training protocols are often used to address these challenges, ensuring that the collaborative learning process is effective and benefits all participating devices.

2.2.3 System Heterogeneity

As mentioned in Chapter 1 and at the beginning of this chapter, FL can also be applied to IoT environments with a multitude of edge devices potentially different from one to another. Given the scope of this thesis, the heterogeneity of the system will be addressed for datacenter environments. According to [8], there are two types of system heterogeneity:

- **Device Heterogeneity** (DH): which results in population bias in the training process due to unequal contribution by and/or failures of collaborating institution. This includes variations in adopted hardware architectures that might use newer computing units with higher frequencies or core-counts, access to more and faster memory, as well as to software tools. When it comes to software we must consider that for delicate environments like the biomedical one, with lot of sensitive patients data, Software must be genuine and up-to-date to ensure maximum protection from attackers. A more extensive list of types of security threats and privacy risks, with attacks and possible solutions can be found in Section 2.5. All these aspects can add various level of complexity to the overall FL pipeline heterogeneity.

- **Behavioral Heterogeneity** (BH): results from the turnover of client devices. In datacenters, this scenario might occur when institutions are dislocated in different areas of the world, which might suffer from unreliable network connections or power supply failures that could cause intermittent activity [179]. BH results in sampling bias in the training process due to uneven participation of devices and unpredictable dropouts. Despite being realistic, this scenario is more related to asynchronous environments which are not the focus of this work.

2.3 FL in Biomedical environment today

Now that we have learned the roots of the FL paradigm, we are going to provide a review of the literature on the application of FL in medical imaging. FL has quickly become popular in the healthcare environment because

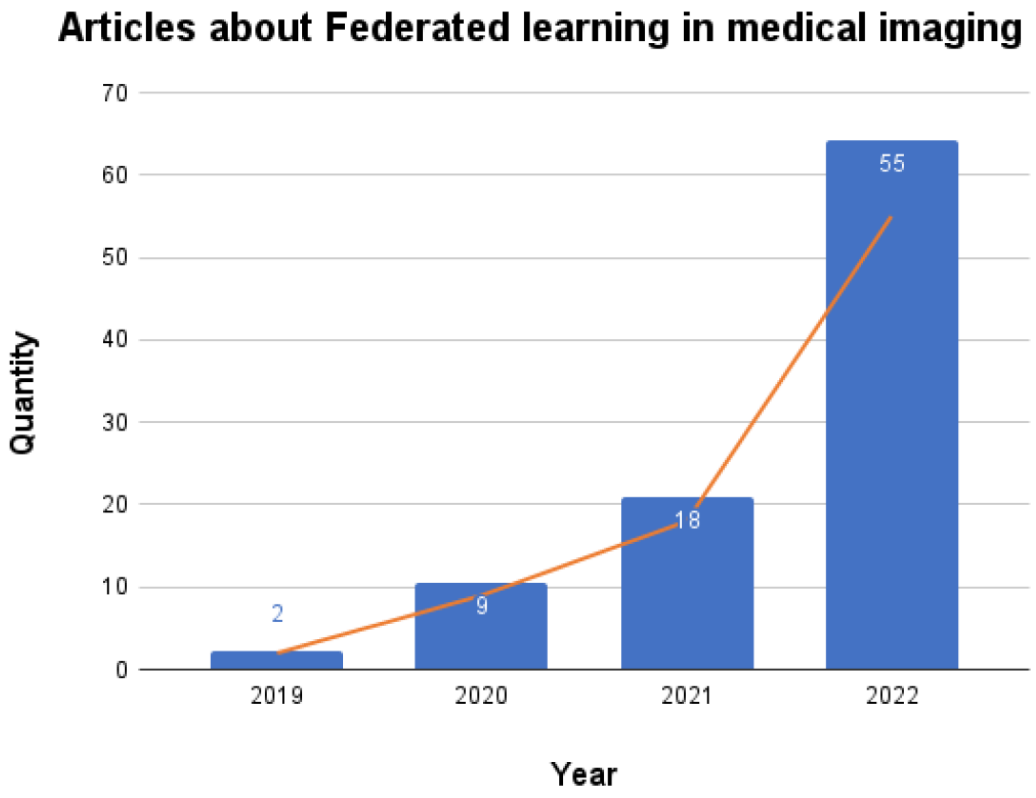


Figure 2.2: Picture taken from [56], showing articles published in the last ten years that were retrieved from the SCOPUS database using the keywords "federated learning", "federated machine learning", "medical image", and "medical imaging"

privacy and data security are paramount and can help unlock collaborations beyond a locally owned dataset by a single clinical institution.

Although healthcare care was a fertile ground for FL exploitation, recent surveys demonstrate how FL is still in a blooming phase of research [56,221]. In the last ten years of activity, less than 100 articles have been retrieved for the survey [56]. More details are shown in Fig. 2.2. Similar numbers have also been reported by another survey with an exclusive focus on FL in the biomedical environment [221]. A summary of these results is provided in Fig. 2.3.

Most of these works involve the use of convolutional neural networks (CNNs) [174] to deal with medical images. Researchers have used a variety of models, datasets, and structures to develop their FL frameworks. Types of acquisitions include: Magnetic Resonance Imaging (MRI or MR) [102,135,137], Computed Tomography (CT) [71,247,263], X-Ray [69,112,144]

Contributions\Reference	[5]	[6]	[7]	[8]	**Our Study**
Area of investigation	All medical data	Healthcare area	Biomedical data	Oncology and cancer research	**All medical image data**
Number of articles	80	24	13	63	**17**
Demographic data presented	yes	yes	yes	no	**yes**
FL process disscused	Yes	yes	no	yes	**yes**
Applications of FL	no	yes	no	no	**yes**
Characteristics of federated datasets	yes	yes	no	no	**yes**
Additional privacy methods disscused	yes	yes	yes	yes	**yes**
Performance discussion	no	no	no	no	**yes**
Performance comparison with traditional ML	no	no	no	no	**yes**
Highlights	Elaborately discussed the FL process and privacy concern; but no discussion over performance.	All of the materials discussed broadly without performance comparison.	Keen about presenting survey data but very short discussion.	A comparative discussion of the architectures and learning approach between FL and general methods on cancer data.	**A SLR, covers all of the discussed issues in the FL oriented medical image analysis.**

Figure 2.3: Table extracted from [221], showing the comparison of contributions and references retrieved by the survey. Last column mentioning "our study" refers to [221].

and Histology [88]. MR images generally have shown superior accuracy compared to CT and X-ray images, which probably explains their increased focus in the survey [56]. The survey also emphasizes how MR scans are more effective than X-ray or CT images in visualizing soft tissues. Additionally, while many datasets of MR and X-ray images are publicly available, there are only a few that include CT images, indicating a gap that needs to be addressed in future research. Histology images represent a critical medical imaging technique that aids in clinical decision-making; however, there is an absence of FL solutions tailored for this imaging modality. In general, the various studies discuss the applications of FL in several research fields, such as: Diabetic retinopathy (DR), MRI classification, cancer segmentation, pneumonia, COVID-19 detection, and few others. These subjects are well-known in the field of medical image analysis using traditional Machine Learning (ML). Consequently, FL opens up new avenues of research thanks to its efficient privacy-preserving capabilities, which are crucial for this type of imaging research.

An estimated percentage of FL applied in the diagnosis of different diseases is presented in Fig.2.4. Examples of "others" include (but are not limited to) neural disease prediction [102] and classification [39], diagnose cardiovascular disease [135], and segment and classify vertebral body images [137].

Most of the reviewed articles refer to HFL for data centers. When it comes to VFL, the research field is even newer; however, few considerations can be drawn. Recent studies on VFL have identified different architectures for implementing VFL pipelines. In [126] the authors propose a taxonomy with three branches (Split VFL, Non-Split VFL and Customized VFL), while [141] partitions the alternatives into four main types (SplitVFL, AggVFL, SpliVFL$_c$ and AggVFL$_c$). In summarizing the contributions, we can identify two main categories of VFL. They both start from the VFL assumption of having heterogeneous data distributed across multiple institutions providing a different perspective of the same data instance. What changes is the way the model are designed:

- Split VFL: takes the name from Split Neural Networks (SNN) [41, 119] and consists of models partitioned across various institutions where each hosts a part of the trainable model. It is literally like taking a model, which accepts several inputs, and splitting it across different computing nodes.

- Custom VFL: where only the baseline assumption of data heterogeneity is preserved and the model partitions are custom and depends on the

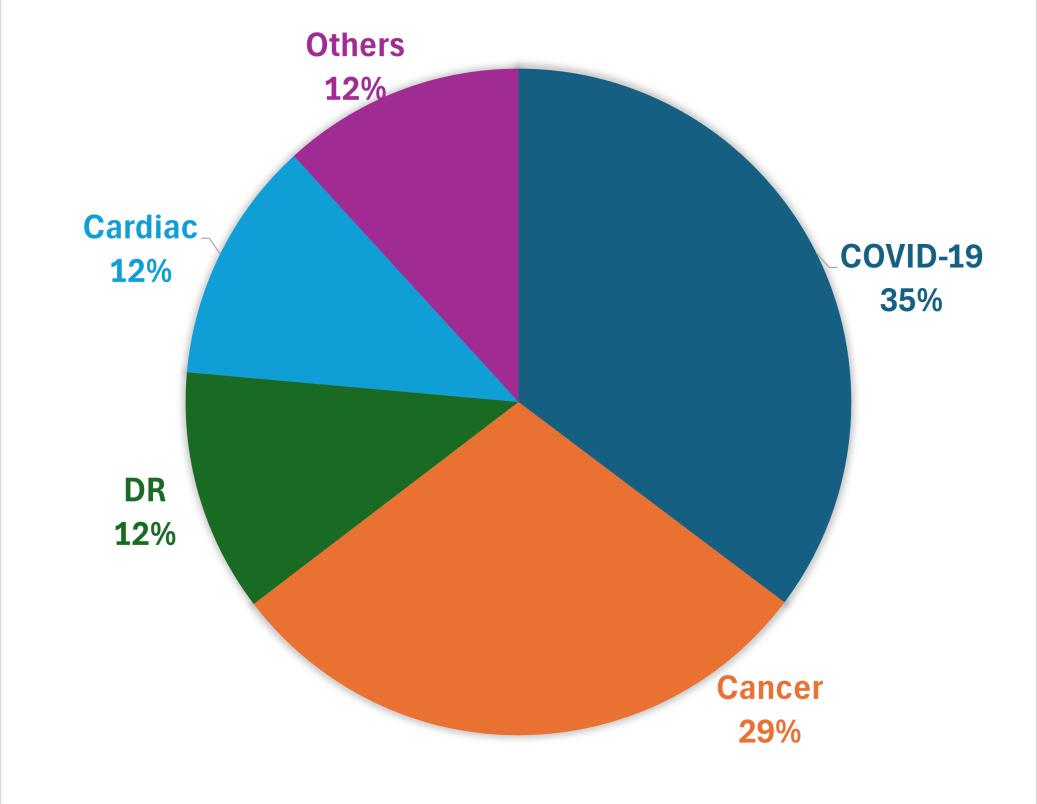


Figure 2.4: Picture taken from [221], showing the percentage of FL applied in different diseases diagnosis research.

model architecture. Types of custom VFL includes decision trees [52], linear and logistic regression [86, 142], support vector machines (SVM) [99] and DL models.

The scope of this work includes HFL and VFL settings, the latter intended as split VFL, both applied to data centers to simulate collaborations between hospitals or biomedical institutions. Therefore, in this thesis, examples of FL applied to IOT or edge devices or alternative VFL approaches will not be discussed.

2.4 FL effectiveness on model convergence

As mentioned in Chapter 1.3.3, aggregation functions are not only responsible for improving model convergence, but they can also be adopted to speed-up communication exchange between parties and improve the system security.

In this work, we focus on identifying the elements to improve model convergence and to produce models that are more able to generalize on unseen samples. An intuitive explanation can be found if we consider how FedAvg [154] works. Averaging the various contributions weighted by the amount of data owned by a single institution, in a typical HFL environment, would not consider any qualitative significant difference in the heterogeneous dataset. This means that if two institutions I_1 and I_2 would provide respectively n_1 and n_2 data samples, with $n_1 == n_2$, but with $\sigma^2(D_1) >> \sigma^2(D_2)$, where σ^2 is the variance of each respective dataset D_i, their support for the ultimate convergence of the model would be considered the same based on the FedAvg [154] algorithm.

This might not be convenient, and the research field has worked to propose alternative and more effective ways to collate the various local outputs. A comprehensive summary of the most impactful aggregation methods is provided below. Due to the different design principles that differentiate HFL and VFL, the aggregation functions are more customizable for the HFL pipeline, while in VFL settings, the observations discussed in the previous section apply 1.3.3. More in detail, we can say that the HFL setting performs a parameter-based aggregation, while VFL leverages an output-based aggregation. In the first case, the parameters, such as gradients or model weights, of each local model are fused to produce the global model (as highlighted in Fig. 1.4. This is not possible in typical VFL settings due to the architectural differences between the models. Hence, the focus is on identifying ways to consolidate the output of local models as input to the aggregated module. Further discussion on the VFL aggregation techniques are provided in Chapter 5.

2.4.1 HFL aggregation functions

As introduced in 1.3.2, the methods for aggregating local models depend on whether the HFL pipeline is synchronous or asynchronous. Considering the scope of this thesis, the methods reported in the following are related to synchronous pipelines and dedicated to improving the effectiveness of the model on the specific tasks. Aspects such as communication speed-up or security improvement will not be included.

Let an FL setting be composed of I collaborating institutions with $i \in [1, I], and I \geq 2$. Each institution i is expected to access a local dataset D_i, composed of $|D_i| = n_i$ data instance.

- FedAvg [154]: the first and most adopted method. Consists in averag-

ing the model weights coming from each institution.

$$\mathcal{L}(\Omega, D) = \frac{1}{I}\sum_{i=1}^{I} \ell(\omega_i, x_i, y_i) \tag{2.2}$$

- FedProx [128]: an improvement in the FedAvg algorithm designed to address the problem of optimization variance inherent to SGD-based methods. The researchers argue that conducting numerous local iterative training steps in FedAvg could lead each client to focus on optimizing its local objective instead of the global objective, potentially causing divergence of the model. In FedProx, a proximal term is added to the objective function to control the impact of local models and provide convergence assurances.

$$\mathcal{L}(\Omega, D)^{HFL} = \frac{1}{I}\sum_{i=1}^{I} \ell_i(\omega_i, x_i, y_i) + \frac{\mu}{2}||\omega_i^s - \omega_i^{s-1}||^2 \tag{2.3}$$

  where $||\omega_i^s - \omega_i^{s-1}||$ captures the differences between the local model hosted by i at step s and $s-1$.

  When mu = 0, tutto e' uguale a FedAVG

- FedMA [233]: The name stands for "Federated Matched Averaging" and proposes to build the aggregated global model in a layer-wise manner by comparing and fusing together similar hidden elements of the neural network, such as channels of convolution layers, hidden states for Recurrent Neural Networks (RNNs) or neurons for fully connected layers. The work proposes a new layer-wise FL algorithm to adapt to heterogeneity in the data based on the Bayesian non-parametric method called "Probabilistic Federated Neural Matching" (PFNM) [257]. This method is limited to CNN and Long-short term memory networks (LSTM). Details of Layer-wise conversion function for the two categories are available in the original work [233].

- FedNova [235]: The name stands for "Federated Normalized Averaging" and proposes a method to normalize local model updates when averaging, based on rounds of local iterations. FedNova demonstrates how previous aggregation functions, such as FedAvg [154] or FedProx [128] are subject to objective inconsistency caused by different data variants, different number of optimization rounds and different types of hardware (with different working frequency) hosted by each single institution. FedNova introduces a normalization and scaling factor in the

global model update. The proposed update rule for the global model is provided below:

$$w^{t+1} \leftarrow w^t_{\text{glob}} - \tau_{\text{eff}} \sum_{k \in S_t} \frac{n_k}{n} \cdot \eta d^t_k \tag{2.4}$$

where $d^t_k = G^t_k a^t_k / \|a^t_k\|_1$

- FL-HC [27]: The name stands for "Federated Learning with Hierarchical Clustering" and the idea consists of separating clusters of clients by the similarity of their local updates to the global joint model. Once separated, the clusters are trained independently and in parallel on specialised models. The study shows how the HC component can speed-up model convergence and allow for greater number of clients to join the federation. The specific procedure is explained in the original paper [27].

- Scaffold [116]: The name stands for "Stochastic Controlled Averaging algorithm". It tries to compensate for the "client-drift" issue by predicting the update direction of the global gradient and adjusting the local model updates. Introduces a client control variable c_i (for each collaborator i) and a server control variable c (for the aggregator). The disparity between local and global control variables is utilized to adjust/influence gradient updates during local training. The update procedure for the control variables in Scaffold is defined as:

$$c^+_k \leftarrow c_k - c + \frac{1}{e\eta l} \left(w^t_{glob} - w^t_k \right) \tag{2.5}$$

$$c^+ \leftarrow c + \frac{1}{|N|} \sum_{k \in S_t} \left(c^+_k - c_k \right) \tag{2.6}$$

- MOON [125]: The name stands for "Model-contrastive learning" and consists in correcting the local updates by maximizing the convergence of contributions learned by a local model and the representation learned by the global model. The technique is based on the intuitive idea for which the model trained on the whole dataset is able to extract a better feature representation than the model trained on a partitioned subset. For example, given a federation of collaborating institutions that own datasets of different animals, the local model learning about dogs will not necessarily generalize or learn features helpful for identifying birds. By transposing the same concept to medical imaging, we

can see where the MOON idea comes from. This is why this function is particularly relevant for computer vision tasks. It builds on top of FedAvg [154] adding a loss function to minimize the distance between the global model and the local model representations. The definition of the contrastive loss function is defined below:

$$l_{\text{con}} \leftarrow -\log \frac{\exp(\text{sim}(w_k^t, w_{\text{glob}}^t)/\tau)}{\exp(\text{sim}(w_k^t, w_{\text{glob}}^t)/\tau) + \exp(\text{sim}(w_k^t, w_{\text{glob}}^{\text{prev}})/\tau)} \tag{2.7}$$

- FedDist [64]: The name is inspired by the Euclidean Distance computed between layers. The algorithm integrates elements of FedAvg [154] and FedMA [233], emphasizing euclidean distance dissimilarity to match neurons from client and server models. Recognizing the divergence in client models due to non-IID data, FedDist identifies neurons that cannot be matched and incorporates them into the aggregated model, improving its ability to handle specific client data. This approach, particularly effective for sparse data, involves adding new neurons and performing layer-wise training to ensure models generalize better, resulting in an enhanced federated learning process. Details of Layer-wise matching and neuron creation policies are available in the original work [64].

- FedSim [176]: The name is inspired by the "Similarity" guided model aggregation. Based on the same line of thinking of FedMA [233] and FedDist [64], the idea of FedSim consists in leveraging inter-client similarities based on model gradients to preserve model convergence. FedSim breaks down the standard aggregation process into two stages: local and global aggregation. Following the global model's initialization and broadcasting, the aggregator gathers gradients from participants' local models. It then applies Principal Component Analysis (PCA) to reduce the matrix dimensionality of the collected gradients. The aggregator forms clusters of participants using the K-means algorithm, performs local aggregation within each cluster, averages the results during the global aggregation phase, and finally disseminates the global model across the network.

- FedBN [131]: The name is inspired by the application of "Batch Normalization" (BN) to the FL pipeline. The study consider the deviation in features space of each collaborating institutions and identify this scenario as feature shift. Similar to FedAvg [154], FedBN performs local updates and averages local models. However, FedBN assumes local models have BN layers and excludes their parameters from the

averaging step. The detailed mathematical background on how BN is applied to the layers of the local models are explained in the original paper [131].

- FedFM [254]: The name stands for "Federated Feature Matching", which tries to mitigate the adverse impacts of data variability in federated learning by harmonizing each client's feature representation. The work proposes a definition of landmarks as the average of features for the same class and names them as anchors. In the process of updating anchors, initially, each client computes its own local anchors; subsequently, through communication with the server, these local anchors are aggregated to form global anchors, which are then distributed back to each client. Hence, each client's feature is encouraged to reduce the distance with the global anchor of the corresponding category during local model training. The updating formulas and detailed definitions for the anchors are provided in the original paper [254].

Despite all these proposed methods, to date, there is no rigorous understanding of objective inconsistency and the speed of convergence for FL settings with heterogeneous data, systems, and local updates metrics. The picture becomes even more complicated if we consider the VFL approach, which changes the mathematical formulation of the problem.

2.5 Security and privacy

As we mentioned in Section 1.3.3, the ability of FL to go beyond the limits of a local institution and allow distributed collaboration could also increase the risk of external attacks. Risks related to data leakage, adversarial attacks, and potential breaches of sensitive information [14, 158] can discourage the construction of FL pipelines. These risks are not only associated with the Privacy concept but are also extremely entangled with the Security aspects. Before diving into the details of the latest research activities on this topic, intuitive definitions are provided for both Security and Privacy.

Security and privacy are two fundamental concepts often mentioned together, but refer to different aspects of information protection. Security involves the measures and protocols put in place to protect data from unauthorized access, breaches, and other malicious activities. It encompasses a broad range of practices, including encryption, firewalls, and secure coding. Privacy, on the other hand, is centered on the rights of individuals to control their personal information and how it is collected, used, and shared. Security is primarily concerned with safeguarding data against external threats,

whereas privacy is focused on ethical and legal handling of data within an organization. Together, they work to ensure that information is both shielded from risks and managed in a way that honors individuals' rights.

2.5.1 Threats and Risks

In [188], the authors anticipate what the possible privacy challenges would be when applied to FL environments. The paper highlights how privacy is linked to performance as an additional cost that needs to be paid to ensure higher security standards. The authors identify two types of FL collaborations:

- Trusted: a federation where all parties are considered trustworthy and are bound by an enforceable collaboration agreement. This scenario simplifies the work flow as the assumption is that many of the threats, such as deliberate attempts to extract sensitive information or to intentionally corrupt the model, do not need to be addressed. This reduces the need for sophisticated countermeasures, falling back to the principles of standard collaborative research.

- Untrusted: a federation with so many collaborating institutions for which an enforceable collaboration agreement might not be doable. In this case we consider that certain clients can not be trusted as they could intentionally attempt to reduce performance, disrupt the system, or gather data from other participants. In this scenario, security measures will be necessary to address these threats. Examples of actions that can help mitigating these risks includes (but are not limited to): robust encryption of model submissions [165], secure authentication of all participants, activity traceability, DP [63], verification mechanisms, execution integrity, maintaining model confidentiality, and safeguards against adversarial attacks.

More in general, the privacy and security risks that should be considered when building FL pipelines, regardless to whether is a *trusted* or *Untrusted* setting, are proposed below [87]:

- **Data Poisoning**: Data poisoning is an attack strategy in which an adversary intentionally inserts false or misleading information into a training data set. The purpose is to disrupt the learning procedure and undermine the model's precision, frequently by increasing its susceptibility to mistakes or by inserting particular biases [173]. In FL this can also happen when a participant inadvertently transmits incorrect data to the aggregator. This kind of attack can be especially harmful,

as it can reduce the efficacy of machine learning models and can be difficult to identify and counteract [244].

- **Data Leakage/Extraction Attack**: In general terms a data extraction attack strategy consists of an attacker that aims to extract confidential, sensitive, or proprietary information from a targeted system [82]. Considering all communication rounds to/from the collaborators and the aggregator of an FL pipeline, these types of attacks have additional entry points in federated settings. In fact, an attacker could intercept the model weights during update exchanges and reconstruct the original data from there. Studies [109, 238], highlight the vulnerability of FL to gradient data leakage, which compromises the privacy of participant training data. These types of attacks exploit the fact that ML models can unintentionally memorize and reveal details about their training data.

- **Model Inversion Attack**: Similar to the data extraction attack, a model inversion attack is a type of privacy attack where an adversary aims to replicate the trained model instead of stealing the training data [93, 101]. In FL it might occur when an adversary leverages the updates provided to the aggregator to reconstruct an approximation of the original model.

- **Membership inference Attack**: A Membership Inference Attack is a type of attack in which an adversary aims to infer whether a particular data record was part of the training dataset of an ML model [168]. This can reveal sensitive information about individuals whose data might have been used in the training process. In FL this could reveal information regarding the particular institution hosting the version of the model or, even worse, expose details about the specific user to which the training data refers to [98, 260].

The list provided highlights only the main types of attack. For a more inclusive overview, see Table 2.1 as proposed by [164].

2.5.2 Protection approaches and strategies

Now that we have understood the most common types of attack and their impact on FL pipelines, we can provide a quick overview of the main defense mechanisms, some of which were anticipated in Section 1.3.3.

There are two categories of defence strategies when it comes to cybersecurity [84, 164]: proactive and reactive.

Table 2.1: Threats and Vulnerabilities in ML Frameworks as summarized by [164]

Threats	Severity	ML Framework	Source of Vulnerability
Poisoning	High	DML/FL	Client Data Manipulations, Compromised Central Server
Inference	High	FL	Client Data Manipulations, Compromised Central Server
Backdoor Attacks	High	DML/FL	Client Data Manipulations
GANs	High	FL	Client Data Manipulations, Compromised Central Server
Malicious Server	High	DML/FL	Compromised Central Server
Communication bottlenecks	High	FL	Weaker Communication bandwidth
Free-riding	Medium	FL	Clients in FL
Unavailability	Medium	FL	Clients in FL
Eavesdropping	Medium	FL	Weaker Communication Protocol
Interplay with data protection laws	Low	FL	Implementers of FL Environment
System disruption IT downtime	Low	FL	Clients and Centralized Server in FL

- Proactive: Proactive defense strategies involve anticipating potential threats and vulnerabilities before they can be exploited. This includes activities such as regular system updates, patch management, threat hunting, penetration testing, user education, and implementing cutting-edge technologies like AI for threat detection.

- Reactive: Reactive defense strategies, on the other hand, are measures taken in response to a security incident after it has occurred. This includes activities such as incident response, forensic analysis, patching identified vulnerabilities, and learning from incidents to improve future defenses.

In addition to these, the following techniques were adopted to improve the overall security of the FL pipeline.

- **Secure Multi-Party Computation (SMP or MPC)**: SMP is a cryptographic protocol that enables multiple parties to jointly compute a function over their inputs while keeping those inputs private [87]. The main goal is to ensure that no party learns anything more than the output of the computation and their own input. When applied to FL, SMC only needs to encrypt the parameters instead of the full training set. Although this approach prevents leaks from an infected aggregator, the encryption is expensive to use and may have an impact on a larger scale. Thus, the main cost of this solution is efficiency loss due to encryption [84, 87].

- **Homomorphic Encryption (HE)**: HE is a form of encryption that allows computations to be carried out on cipher-data, generating an encrypted result which, when decrypted, matches the result of operations performed on the plain-data [165]. This type of encryption enables

performing operations on encrypted data without needing to decrypt it first, thus providing data privacy and security, even while data is being processed. In FL, HE plays a critical role by aggregating model updates across various devices while safeguarding user data privacy [91]. A detailed example showing all the communication step for an HE procedure applied to FL is provided in [87]. The ownership of the secret key is a crucial point in public-key crypto-systems used for FL aggregation. Various techniques based for HE in FL are further described in [184].

- **Differential Privacy (DP)**: DP is used to ensure that the privacy of individuals in a dataset is maintained when performing data analysis. This is typically achieved by adding a controlled amount of random noise to the data, which makes it difficult to identify or retrieve initial information within the dataset while still allowing useful aggregate information to be extracted [63]. Despite the perturbation, the quality loss caused by the added noise of each user is relatively low compared to the increased privacy protection [84]. In FL, to avoid inverse data retrieval, DP is introduced to add obfuscate the uploaded parameters of the various models. A more estensive list of examples of HE applied to FL is provided in [87].

- **BlockChain**: Blockchain refers to the use of distributed ledger technology to improve the security, transparency and integrity of data exchanges between multiple decentralized nodes without the need for a central authority [166,227]. In FL, blockchain can provide a secure and immutable record of transactions and interactions, ensuring trustworthiness and reliability among participating entities [87]. More in detail, Blockchain brings two main advantages to FL: first, it would incentivize user participation in the global model's creation by compensating contributors for their input; second, it would record the progression of parameters on the blockchain. This approach could bolster user trust and mitigate the threat of poisoning attacks [84]. A clear example is provided by *BlockFL* detailed in [118].

Chapter 3

FeLebrities: a user-centric assessment of Federated Learning frameworks

Federated Learning (FL) is a new paradigm aimed at solving data access problems.

It provides a solution by moving the focus from sharing data to sharing models. The FL paradigm involves different entities (institutions) holding proprietary datasets that, contributing with each other to train a global Artificial Intelligence (AI) model using their own locally available data. Although several studies have proposed methods to distribute the computation or aggregate results, few efforts have been made to cover on how to implement FL pipelines. With the aim of accelerating the exploitation of FL frameworks, this chapter proposes a survey of public tools that are currently available for building FL pipelines, an objective ranking based on the current state of user preferences, and an assessment of the growing trend of the tool's popularity over a one year time window, with measurements performed every six months. These measurements include objective metrics, like the number of "Watch," "Star" and "Follow" available from software repositories as well as thirteen custom metrics grouped into three main categories: Usability, Portability, and Flexibility. Finally, a ranking of the maturity of the tools is derived based on the key aspects to consider when building a FL pipeline.

Federated Learning (FL) is a paradigm that aims to solve the data access problem. In the Artificial Intelligence (AI) domain, data represents the starting point for many research and development activities [248] [127] [212]. With increasing attention given to the field, data have also grown in demand and appreciation, redefining priorities in designing and building solutions for real-world applications. A clear demonstration of this growing impor-

tance is the creation of dedicated laws, such as the General Data Privacy Regulations (GDPR) [97] in place in the European Union, the Protection of Personal Information Act (POPIA) [4], and the Health Insurance Portability and Accountability Act (HIPAA) [194] in the USA, which is specific for accessing clinical data and medical records. From the AI perspective, this reflects the need to access data to advance the State of the Art (SOA) in a given environment while fully complying with regulations. FL is an effective way to satisfy all these requirements. In a federation of collaborating institutions, what is shared is a common global model that is partially trained by every collaborator using local data. Historically, the approach of training AI models assumes that data would be collected and centralized in a unique infrastructure appropriately equipped with dedicated hardware and software to sustain the computation. High performance computing (HPC) centers are great examples of this approach, as illustrated in Figure 3.1. In constrast, in an FL setting, data are expected to stay in the exact location where they were collected, while a copy of the AI global model is shared across all institutions participating in a federation. A generic example is shown in Figure 3.2.

The research community has already started investigating this emerging topic either for its privacy-compliant aspects [248] [149] or as a viable tool for addressing AI challenges in critical domains such as the biomedical context [145] [195] [216]. Although the domain is still relatively new, the literature can already provide helpful surveys on how the concept works and complies with privacy aspects [248], how it can be transferred to the *Internet of Things (IoT)* world [186], and what are the steps to implement it from a protocol, software, and hardware standpoint [186]. The rising interest in the research community and industries *R&D* departments has enriched the literature, which has, in turn, influenced the development and evolution of many new tools for implementing FL pipelines. If, from one perspective, this aspect is encouraging, it reflects the need to obtain clear indications about what tools are currently available, which are the most popular, and what is their level of maturity (in terms of features).

This chapter aims at providing four main contributions:

1. Provide an updated list of tools publicly available for implementing FL pipelines.

2. Share the current state of adoption of each tool, including growth trends calculated over a one year time window where measurements from three "harvests" (H_i) were performed.

3. Identify and describe the key aspects required by the research commu-

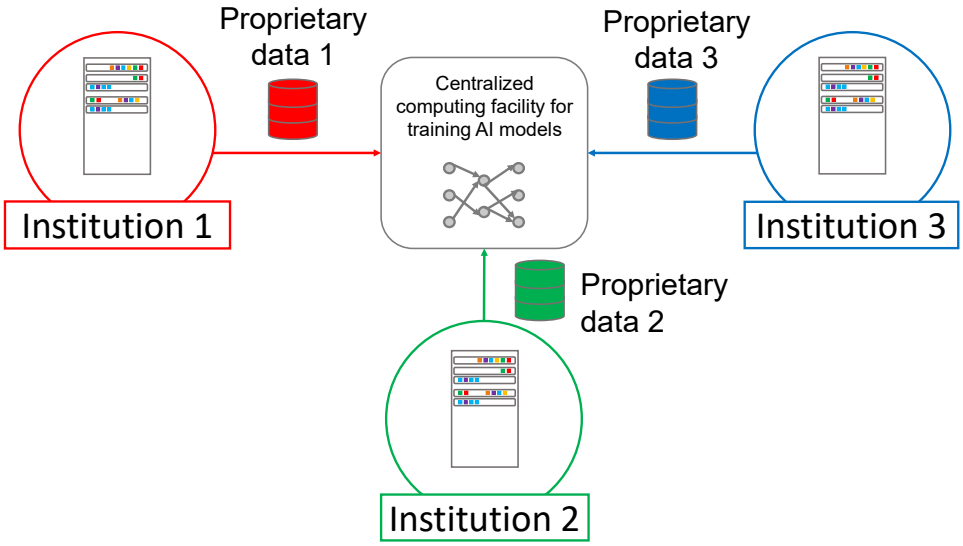


Figure 3.1: Data-to-model: example of legacy approach where data would move to a centralized training facility. Here the AI model is represented as a graph or neural network.

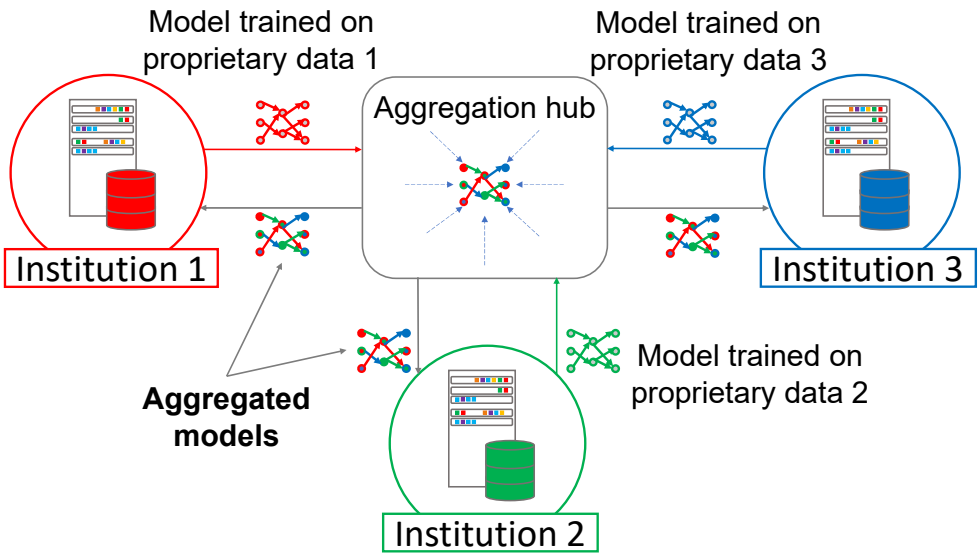


Figure 3.2: Model-to-data: example of a federated approach. A central unit called aggregator would clone and distribute copies of the same model to each collaborating institution. Each of them would then train its copy of the model using local datasets before sharing it back to the aggregator. As the name suggests, the aggregator would ultimately merge the models from different institutions and restart the process by sending out the latest aggregated version.

nity and map them into a list of features that tools should include.

4. Propose a ranking based on objective metrics, including common indicators and the ability to match the needs highlighted in the previous point.

We genuinely believe that by providing a quantitative and qualitative survey of the FL tools, the research community will be able to: accelerate its activities, promote fairness by proposing an inclusive method to collect comparable studies, and help tool providers identify ways to improve their products. The availability of a ranking of FL tools will also boost their exploitation in the production environment, where such tools still need to be explored.

3.1 Chapter organization

This chapter is composed of six Sections. In section II, we discuss FL implementation related works. Section III focuses on the list of tools currently available to the community, sharing a high-level overview of their popularity and adoption. Section IV augments the retrieved list of tools with the current state of adoption, including the growth trend observed over one year, and Section V discusses the key aspects that should be considered when implementing federated environments for research purposes. These factors are then translated into requirements that FL tools need to satisfy for successful exploitation and consolidated in a raking table. The results are discussed in Section VI and future directions and conclusions are finally addressed in VII.

3.2 Related Works

FL is a distributed machine learning (ML) approach that enables organizations to collaborate on projects without sharing sensitive data [213], such as patient records [188], [226] or financial data [143], or data that is not easily accessible, such those stored in remote locations such as satellites or space stations from high-resolution sensors [219]. The basic premise of FL [248] [127] is that the model `moves` to meet the data rather than the data `moving` to meet the model. Therefore, the only minimum data movement required across the federation is the model parameters and their updates.

3.2.1 FL settings

There are two essential components of an FL pipeline: one or multiple institutions owing data and a mechanism to orchestrate the process. Each institution must have local data and be accountable for hosting the training process on proprietary data. The orchestration mechanism may vary, but is mainly of two types: Synchronous or Asynchronous.

In a synchronous scenario, the idea is to have a central unit, often identified as an aggregator [213] [188], acting as a central pivot and determining when to start a new iteration. The aggregator is responsible for cloning the initial model to each collaborating institution, waiting to receive the locally trained copies, and finally merging them, as the name suggests. This type of FL pipeline is usually implemented in big data centers (cross-silo), such as those involved in medical environments [212] [60]. Data centers can store vast amounts of data and provide the computational power required to process them. In addition, big computing infrastructures, such as HPC centers, can rely on fast and stable connections to the network, simplifying the creation of a more reliable communication channel to interact with a hypothetical aggregator unit.

However, as soon as we move away from data centers towards edge devices, new challenges arise owing to the high variance in products and manufacturers. Devices with different latencies, working frequencies, and hardware features can have different computation times [50] [104]. These are the reasons for the need for an asynchronous FL pipeline. In this scenario, each collaborating institution can share its update at any time, either to a unique aggregator [243], [50], [251] or to other participants in an "all-to-all" setup [205], [170].

Another critical point to address is the difference between the horizontal (HFL) and vertical (VFL) federated learning. To understand this difference, we need to consider the space of the features and the model type. In the examples shared thus far, we implicitly refer to the Horizontal FL, where the different collaborators have different data but contribute to the federation by sharing the feature space and training the same model. This is the case for institutions with offices distributed across different locations that would like to train a common model by leveraging the local data stored in each facility in a privacy-compliant manner. In Vertical FL, each collaborator is expected to contribute by providing different bits of information from the same sample. This leads to a scenario in which the feature space accessed by every collaborator may be different from the others. Therefore, each collaborator might train a different model in the vertical configuration. Aggregation, in this case, is represented by the interoperability between collaborators, where

to update a model, information coming from the model of another collaborator might be required [237], [49]. For example, in a typical VFL setting, we can see a life insurance agency collaborating with hospitals to build a decision model to obtain more precise estimations of their affiliates. In this case, it is expected that the entities involved in the federation can provide different information about the same user. These two ways of articulating the data for a federation impact the choice of model and how the federation is orchestrated. While in the HFL, there is only one model, and all the collaborators are responsible for ensuring that data are normalized to feed it, the VFL brings some more complexity. In this case, to handle different data types from several institutions, each collaborator should have a local model that can accept the data from that specific institution as input. In addition, there must be a federated model that takes all outputs of the various local models as inputs. As illustrated by Chen et al. [49], the procedure for training Deep Learning (DL) models based on back-propagation [17, 240], needs to deal with the two-level training procedure represented by the different models that need to be managed: one at the collaborator level and the other at the aggregation point. This complexity is also reflected in the challenges that might arise in finding a satisfactory convergence point for the adopted DL model.

3.2.2 FL challenges

Regardless of which FL setting (Synchronous or Asynchronous) or configuration (Horizontal or Vertical) is adopted by a given federation, the research community currently adresses three main areas:

1. Aggregation functions and model convergence starting from different data distributions.

2. Privacy aspects and ways to build a secure FL pipeline for protecting the IP during experiments.

3. Communication efficiency and protocols to improve the FL base infrastructure.

Protecting dataset ownership implies that in most cases, the assumption of dealing with independent and identically distributed (i.i.d.) samples across local nodes does not hold for FL setups [130] [204]. Data distribution can severely impact the training performance by affecting the total accuracy [28], convergence capability, authentication processes (especially in the case of different devices), and speed of the process intended as total time-to-train

[26]. In summary, in this setting, the performance of the training process may vary significantly according to the imbalance of the local data samples and the particular statistical distribution of the training examples (i.e., features and labels) stored at the local nodes [127].

In the past few years, institutions have introduced FL deployments to address the need to train AI models. Sectors such as healthcare and finance would benefit from having a setting with greater access to more extensive and diverse datasets without violating privacy laws [215] [25], such as HIPAA, GDPR [97], and POPIA [4]. On the one hand, FL has been designed with security in mind [28], and the set up is just the beginning. Securing execution environments introduces many open challenges to the research field [16]. Key questions include finding a consolidated method to guarantee secure execution (encryption, key exchange, and hardware features) and validating the reliability of intermediate results and collaborators within the federation.

Massive amounts of data are usually stored in "Data-Lake" infrastructures. The more machines/institutions that participate in a federation, the more critical is the ability to scale. As mentioned in the previous section, a consolidated method for detecting scarse training contributions (coming from institutions with corrupted or redundant data) is still lacking, to the best of our knowledge. Aggregation functions are currently being evaluated by the research community [130] [25] [250]. Another implication when discussing big scales is the infrastructure and the connectivity chosen by the institutions for communication [127].

3.2.3 Study relevance

Several studies have proposed surveys to illustrate the advancement in the field [248] [186]; however, to the best of our knowledge, no one has provided a ranked list based on the ad hoc quality assessment criteria of all the (possible) tools available to the community to implement FL experiments. A comparison of five tools is provided in [117], which are accessible through a licensed service, without clarifying why or how these tools were precisely selected. Another study [47] provides an attractive comparison table. However, the main focus of this work is to promote an alternative tool specifically for FL benchmarks instead of providing a complete list of the available options to boost the exploitation of FL across the community. Even in this related work, it is unclear why and how the tools discussed were selected. Similarly, [139] proposed a complete benchmarking suite with a helpful decision tree to help users choose a tool based on their requirements. Their recommended ranking also includes some of the evaluation metrics proposed in this study with an even deeper level of detail. However, while we believe in the value of such an

approach, the breadth of the offer in terms of tools that can be chosen might represent a constraint for end users. In fact, [139] centers its evaluation on nine tools, but the criteria for which those tools were identified and selected need to be clarified. As we discovered in this work, the list of open-source FL tools can exceed 30, and it is interesting to note how the most popular tool to date was not considered in their decision tree.

3.3 Federated Learning Tools

3.3.1 Methods and premises

This article aims to provide an inclusive and informative list of the current FL tools available to the community for implementing research pipelines in any environment in which accessing distributed data is challenging. To better understand the present scenario, we performed three literature searches, H_i, where $i \in 1, 2, 3$. H_1 was conducted on March 28, 2022, H_2 on September 28, 2022, and H_3 on April 10, 2023.

This activity was inspired by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines. More specifically, we followed the *Preferred Reporting Items for Systematic Review and Meta-Analysis of Diagnostic Test Accuracy Studies (PRISMA-DTA): explanation, elaboration, and checklist* [201]. In particular, the guidelines we followed were a selection of the guidelines described in the *PRISMA 2020 checklist*, accessible on the official PRISMA website: http://www.prisma-statement.org/. Below is a detailed description of the items extracted from the PRISMA guidelines that were identified as applicable to this collection. Items not included in the list below were either not sharing best practices on the "Method" (i.e., best practices for "Title," "Introduction," and "Abstract" for systematic reviews) or not directly relatable to this contribution as it does not fully match a "Systematic Review." Some examples of discarded items are as follows:

- 12) *Specify for each outcome the effect measure(s) (e.g., risk ratio, mean difference) used in the synthesis or presentation of results.*
- 14) *Describe any methods used to assess risk of bias due to missing results in a synthesis (arising from reporting biases).*
- 19) *For all outcomes, present, for each study: (a) summary statistics for each group (where appropriate) and (b) an effect estimate and its precision (e.g., confidence/credible interval), ideally using structured tables or plots.*

On the contrary, here is the list of items considered for this work. The numbers reported below are a direct references to the PRISMA document.

- 5) *Specify the inclusion and exclusion criteria for the review and how studies were grouped for the syntheses.*
- 6) *Specify all databases, registers, websites, organisations, reference lists and other sources searched or consulted to identify studies. Specify the date when each source was last searched or consulted.*
- 7) *Present the full search strategies for all databases, registers and websites, including any filters and limits used.*
- 8) *Specify the methods used to decide whether a study met the inclusion criteria of the review, including how many reviewers screened each record and each report retrieved, whether they worked independently, and if applicable, details of automation tools used in the process.*
- 13.b) *Describe any methods required to prepare the data for presentation or synthesis, such as handling of missing summary statistics, or data conversions.*
- 13.d) *Describe any methods used to synthesize results and provide a rationale for the choice(s). If meta-analysis was performed, describe the model(s), method(s) to identify the presence and extent of statistical heterogeneity, and software package(s) used.*
- 16.a) *Describe the results of the search and selection process, from the number of records identified in the search to the number of studies included in the review, ideally using a flow diagram.*
- 16.b) *Cite studies that might appear to meet the inclusion criteria, but which were excluded, and explain why they were excluded.*
- 23c) *Discuss any limitations of the review processes used.*

Each of these items was used to frame the study. The following map illustrates how the single guidelines contributed to shaping the sections:

- Items 5, 6, 7 and 8 were considered for building this Section;
- Items 13a and 13d, were used to build the comparison table in Section IV;
- Items 16a, 16b and 23c, were used to structure the discussion of the results provided in Section V.

3.3.2 Exploring tools

To objectively build the list of tools, we performed three harvests, H_1, H_2, and H_3, with roughly six months of cadence (184 and 194 days respectively). The collection method was the same as that described below. We used three different search engines: Google Scholar [3], Semantic Scholar [5], and the standard Google website.

For the first two, we developed a script to automatically query search engines using a collection of keywords on the topic. We built such a collection by combining each item p of a list of prefixes P with each element s in a list of suffixes S. The set of prefixes was populated with the "federated learning" keyword, and other synonyms or more related terms used in the literature to express similar concepts: P='federated learning', 'privacy-preserving machine learning', 'collaborative learning', 'collaborative machine learning'.

The set of suffixes was built around adjectives and secondary aspects, like 'tools, library' and 'open-source': S='framework', 'tool and framework open source', 'tool and framework open-source', 'open source framework', 'open source tool', 'open source library'.

This led to a prosperous and inclusive search of all the relevant articles and works in the domain.

Google Scholar helped capture all related works where a given keyword (or part of it) was mentioned in the paper and not only in the title. In H_1, we identified a cumulative list of 420 related articles, of which 217 were unique. Despite the contribution, due to a service limitation, the reported numbers refer to H_1: the article harvest completed in March '22 only.

To build a more robust and consistent set of related works, we leveraged the Semantic Scholar service [74]. The website allows users to perform queries and sort the outcome according to four metrics: "Relevance", "Citations-count", "Most Influential Paper", and "Recency".

We repeated searches using all the keywords for all the four sorting types mentioned above, obtaining 1121 (out of 8320) unique articles during H_1, 1291/8284 unique articles with H_2 and 1298/8360 unique articles in H_3. The number of new articles retrieved in H_2 that were unavailable in H_1 was 270, while H_3 enriched the search with 248 new findings.

Finally, we used the standard Google search engine to ensure we could capture all the relevant FL tools yet to be described in a published paper. To do so, we evaluated the first ten results obtained by querying the search engine with the same list of keywords used previously. This step allowed us to enrich the list with additional FL frameworks, such as Nvidia Flare [197], Tensorflow Federated [6], and IBM Federated [146].

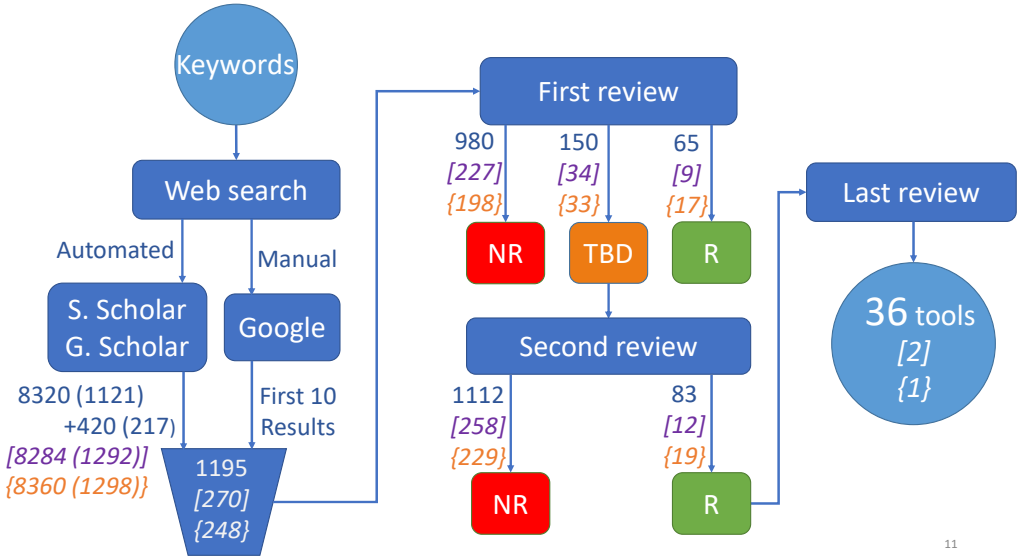


Figure 3.3: Collection pipeline implemented for the harvests. On each arrow is outlined the number of articles retrieved and filtered. Numbers between squared brackets refer to the outcome of H_2, while the curly brackets summarize the numbers from H_3. The output numbers of the second review are obtained by summing the numbers of a given category from the first review with the respective class of the second review. Please note that despite the September harvest returning 1292 articles, only 270 were new findings. The same consideration applies for H_3, with 1298 to 248 articles.

Once we obtained the three lists of unique titles described above, we finally merged them, resulting in 1195 unique articles discovered in H_1, 1292 retrieved in H_2, and 1298 in H_3. We then started pruning results by manually reviewing and labeling the list in three different buckets: "relevant"(R), "non-relevant" (NR), and "uncertain" (TBD).

Articles considerably unrelated to the topic (e.g., work mentioning ML methods or collaborative learning platforms for schools) were discarded from the collection. After the first labeling cycle, we had 65 R, 980 NR, and 150 TBD for H_1, 9 R, 227 NR, and 34 TBD for H_2 and 17 R, 198 NR, and 33 TBD for H_3

Figure 3.4 shows an example of articles captured by the three categories.

The "uncertain" category required us to conduct a deeper review of the work. All articles in this list underwent a second round of labeling. The objective was to review 150 papers on the TBD list and to allocate them to the R or NR. As a result, we obtained 83 R and 1112 NR for H_1, 12 R and 258 NR for H_2 and 19 R with 229 NR for H_3.

A summary of the adopted research pipeline and the results collected during each harvest is shown in Figure 3.3.

Ultimately, a deeper understanding of the relevant papers was performed to draw the final list of FL tools. This final review allowed us to identify 36 suitable tools during H_1, two additional tools during H_2, and one last tool added to H_3. The complete list of tools retrieved in H_1 with the indicators from the Github and Gitlab repositories is presented in Table 3.1.

3.4 Tools popularity and level of adoption

After retrieving the list of tools, our goal was to understand each item's popularity and adoption level from a community perspective. Each Git repository has public indicators, such as the number of watches (W), forks (F), and stars(S). The Watch indicator captures the number of users who actively watch the repository. These users receive updates when new actions are taken from a repository. The number of forks indicates the number of times a repository has been forked. It is a good indicator of how many interested users might develop code to extend the tool. Finally, the number of stars indicates the number of likes that the repository has received. This final indicator might need to be more accurate in capturing actual users, but it can provide a reasonable estimation of reach in terms of how many people have seen the tool at least once. For practicality, we aimed to combine these three aspects into one consolidated score to provide a *popularity* driven ranking of the tools. Since popularity would also depend on the time a given

fl_pytorch: optimization research simulator for federated learning
fedlab: a flexible federated learning framework
sunday-fl – developing open source platform for federated learning
gfl: a decentralized federated learning framework based on blockchain
hyfed: a hybrid federated framework for privacy-preserving machine learning
graphfl: a federated learning framework for semi-supervised node classification
openfl: an open-source framework for federated learning

federated learning for vehicular networks
securefl: privacy preserving federated learning with sgx and trustzone
secureml: a system for scalable privacy-preserving machine learning
fedgraphnn: a federated learning system
flaas: federated learning as a service
a crowdsourcing framework for on-device federated learning
fl-ntk: a neural tangent kernel-based framework for federated learning analysis

effects of mobile gaming patterns on learning outcomes: a literature review
collaborative logical framework: an e-learning assesment tool in .lrn platform
sagittarius: a tool to enhance the collaboratibe work in virtual learning environments
predicting machine learning pipeline runtimes in the context of automated ML
model-sharing games: analyzing federated learning under voluntary participation
tensorflow lite micro: embedded machine learning on tinyml systems
collaborative creation and training of social bots in learning communities

Figure 3.4: A subset of examples of selected articles for each category: ”relevant” green, ”uncertain” blue and ”non-relevant” red.

repository was made available to the community, we wanted to normalize all the values with a timing factor. This step ensured that newer repositories with less exposure to the community would not be affected by a low score. The scores are calculated as follows:

$$\text{Score}_{\text{H}_\text{i}} = \frac{\frac{1}{3}(W + F + S)^{H_i}}{\text{ET}_{\text{H}_\text{i}}} \tag{3.1}$$

Where $\text{ET}_{\text{H}_\text{i}}$ is the time elapsed between the day of the tool's first commit and the harvest date H_i. Table 3.1 shows the scores associated with the tools retrieved in H_1, the March harvest.

An initial understanding of which tools were accessible to the community was helpful but could provide a limited view of the bigger picture. Indeed, while Git indicators can share important insights about user preferences in a given time frame, they do not necessarily capture community trends from a popularity growth rate perspective.

To access this information, we observed the list of tools over a time window to determine which tools were being considered by the community at a higher pace. Thanks to the second harvest H_2 held on September 22) and the third H_3 done on April 23, we discovered new tools to add to the list and updated the values of W, F, and S for each of the tools found in March. Knowing the differences between the indicator's value recorded in the various harvests, given two harvests, H_i and H_j where $j > i$, we computed the growth rate for each tool as follows:

$$\text{GR}_{\text{ij}} = \frac{\frac{1}{3}((W + F + S)^{H_j} - (W + F + S)^{H_i})}{\Delta_{ji}} \tag{3.2}$$

Where Δ_{ji}, is the difference in days between $\text{Date}(\text{H}_\text{j})$ and $\text{Date}(\text{H}_\text{i})$. More precisely, $\Delta_{21} = 184$ days, $\Delta_{32} = 194$ days and $\Delta_{31} = 378$ days. We calculated the growth rate GR for all possible combinations: GR_{12} captures the growth of the repositories between H_1 and H_2, GR_{23} for the growth between H_2 and H_3 and GR_{13} captures the yearly evolution between H_1 and H_3. The results of these computations are shown in Figure 3.5.

Interestingly, the order of the tools based on the popularity level observed in H_1 and captured in Table 3.1 does not match the growth rate highlighted in Figure 3.5.

3.5 Priorities in FL tools for Research

The previous Sections illustrated how we could retrieve a list of relevant tools for building FL pipelines and which are preferred by the community. In this

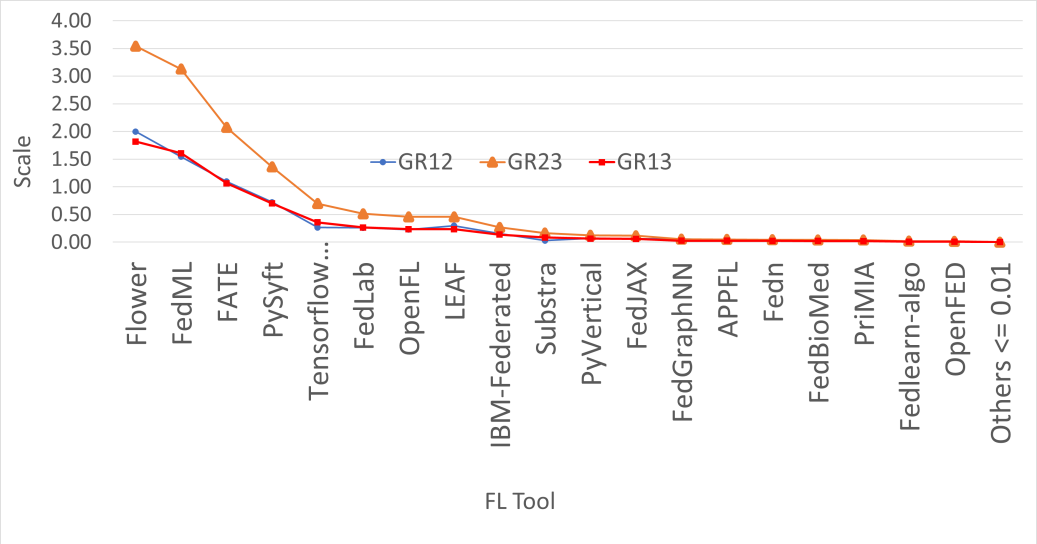


Figure 3.5: Popularity growth rate: this graph illustrates which tools have been gaining more popularity in the community between March and September '22 GR_{12}, between September '22 and April 23 GR_{23}, and over an observation window of 374 days (roughly one year), GR_{13}. Tools that did not have a repository available in H_1 (March '22) were excluded from this chart as it would not be possible to track any evolution.

Section, our goal is to provide a new ranking of tools to identify the most mature. We focused on the specific features of each tool, regardless of the popularity aspects, as outlined in the previous Sections.

The task consists of retrieving and evaluating FL tools that can be adopted to boost the exploitability. To perform this classification, we defined a set of measures based on the different needs and expectations that a tool should satisfy according to the application field and final objectives.

As described in the SOA Section, there are several ways of implementing FL. From bridging different data-center institutions together at the production level to leverage the agile nature of IoT devices, FL pipelines must be shaped according to the needs, goals, and constraints to consider. However, in line with our purpose of identifying the most mature FL tools for research activities, there is no need to filter results based on data centers or edge devices as long as the tools will provide the possibility to simulate multiple decentralized abstracted computing hubs.

However, other considerations should be drawn around what has been highlighted in the FL challenges regarding data distribution, confidential computing, and communication efficiency.

Indeed, those aspects are essential because they can directly reflect requirements that FL tools need to fulfill to be considered. For example, they all highlight the need for a flexible and modular architecture to allow maximum research customization for aggregation functions, communication protocols, or privacy-preserving and security features. Another practical insight derived from Items 2 and 3 concerns the ability of a tool to scale out on multiple computing machines. As demonstrated in [219], the applicability of FL is not only related to the need to access data complying with regulations. It can also refer to data that are not readily retrievable, such as those from a satellite or space station. Furthermore, the Medical [212], or Geo-spatial [219] environments are usually sources of high-resolution data acquired by machines manufactured by different companies, which could map the need for dedicated pre-processing routines to be used to feed an AI model. FL approaches can be tested on multiple machines hosting different datasets (generated by different equipment) or by simulating multiple parallel instances running on the same computing node. The first setting is preferred because it enhances the reliability of the conclusions when investigating the privacy and communication efficiency aspects.

In addition to what has been outlined, other practical considerations related to the research environment may also apply. The easier the path to results in a research environment, the faster the deployment of this technology in real institutions. For example, being able to set up a federated environment quickly by leveraging friendly API, re-using common and well-

established language (such as Python), and AI platforms (such as PyTorch or Tensorflow to mention two) having access to direct support channels or useful documentation can represent critical aspects for simplifying research activities in different domains.

3.5.1 Evaluation metrics

Based on these observations, we consolidated a list of evaluation parameters and guidelines organized as follows:

1. Usability
    - Documentation (Docs.)
    - Developer Experience (DX)
    - Language (Lang.): Python, R, C++, etc.
    - Supported AI frameworks: PyTorch, Tensorflow (TF), etc.
    - Type of AI: Machine Learning (ML), Deep Learning (DL)
2. Portability
    - Templates/Examples availability (T/E)
    - Distribution channels: Anaconda, Pipy, etc.
    - Multi-node mode
    - Open-source
3. Flexibility
    - Containers/Virtualization (C/V)
    - Modular architecture
    - Horizontal or Vertical (H/V) FL
    - Synchronous or Asynchronous (S/A) FL
    - Privacy and Security independent module? (PVC/SEC)
    - Easy integration with other tools

We used these parameters to build an "Evaluation Table" [**??**] for the tools identified in the previous Section. The table was populated with information retrieved from publicly available resources for each tool (see "Docs." column in Table **??**). As can be seen, there is a mismatch between the tools listed in Table 3.1 and those listed in Table **??**. This is mainly due to the following four reasons:

1. Tools not open-source, like Sherpa-ai [18].

2. Missing repositories: is the case of tools that have not yet released their codes after the paper publication: Chiron [103], FedHealth [51], FAE [255], GENO [35], FedTGan [267] and IPLS [178].

3. Coherent but not suitable: is the case for LEAF [30], FL-Bench [134], and PyFed [70], which are positioned for benchmarking purposes and, therefore, might lack essential features for conducting more extensive research activities. FedGraphNN [94] is a sub-project of the more significant initiative known as FedML [95] already included in this survey.

4. New tools or new openings: is the case for Nvidia-Flare [197] (a sub-project of Nvidia-Clara) and FL-Pytorch [29] which opened their repositories at some point after H_1 harvest, as well as FLUTE [61], and PLATO [108], which were retrieved (and added) during H_2 and XFL [234] retrieved in H_3.

After pruning the 12 tools that did not qualify, adding 2 (despite the substitution with Nvidia-Flare [197], Nvidia-Clara was already captured by Table 3.1) at the end of April '23 harvest H_3, we ended up with a list of 28 total tools.

In a second instance, a score was associated with each cell based on a quantitative assessment. Aiming at objective classification of the tools, we captured qualitative aspects in a very inclusive manner. These rewarding tools demonstrated additional development efforts for the community through the available material without penalizing new promising tools that might still be under development.

More precisely, the proposed scoring method rewards completeness rather than excellence. This means that a tool that supports 20 programming languages but needs other relevant features would not outperform a tool that supports only one programming language but has more features that simplify the user experience. This is achieved using scores with a reduced range of values to allow newer but promising tools to compete with more mature tools. More in detail, we adopted a simple approach to assign a score to each cell and designed the "Score Table" 3.3:

- Documentation: We considered having Paper P and/or a public repository for the tool Gh (Github) or Gl (Gitlab) as a minimum requirement. Therefore, we assigned zero to all the tools that did not match this expectation; rewarded with 1 point the tools with at least one additional source of information (such as a dedicated web page or richer documentation that would go beyond Readme files on repositories or Slack

support). Finally, 1.5 points were given to all those that provided two or more sources.

- Developer Experience (DX): we assigned 0 points to all the tools that did not seem to mention nor provide a user interface of some sort (e.g., Jupyter notebook [120] or Google Colab [34] to mention two). We assigned 1 point to all the tools with at least 1 form of user interface abstracting from programming on the command line. Finally, 1.5 points were given to all the tools with two or more user interfaces.

- Language: We assigned 0 points where information about the supported version was not clearly outlined in the documentation. One point was given to the tools supporting at least one language (or one version), and 1.5 points were given to all tools supporting two languages (or two versions of a language). Finally, 2 points were given to all the tools where the engineering team made an the extra effort to support more than two languages (or more than two versions of the same language).

- Supported AI frameworks: We assigned 0 points where the information about the supported AI frameworks was not clearly outlined in the documentation; 1 point was given to each supported framework. When the number of different supported frameworks exceeded 2, we assigned a maximum score of 2.5 points.

- Type of AI: 0 points if not mentioned in the documentation, and 1 for each type of AI supported (ML or DL).

- Templates/Examples availability.

- Distribution channels: we set the minimum requirement for the ability to download a repository and install the tool from there. Therefore we assigned zero points to all the tools respecting this minimum requirement, 1 point to all the tools that had at least one additional way to access the software package (e.g., Pipy or Anaconda); finally, 1.5 points for all the tools that could be installed in two or more ways.

- Multi-node mode: zero when only the simulated environment on a single computing machine was mentioned. One point to all the tools that allow implementing real federation on multiple nodes and 0.5 if this capability has limitations or constraints.

- Open-source: all the tools presented in the table are open-source. This column was not included in the table.

- Containers/Virtualization: zero was given where the documentation did not provide any of the two options. One point was given where either a containerized or virtualized environment was supplied. 1.5 points when two or more options were listed and 0.5 when containers were available but also presented as the only way to access the tool.

- Modular architecture: based on the analysis of the repositories of the tools, we trust that all of them respect this parameter. More specifically, they all have proven to have separate entities (such as client-server and orchestration processes) that can be launched independently.

- Horizontal or Vertical: a tool must implement at least one. Therefore we rewarded 1 point only to the tools that would allow executions in both settings.

- Synchronous or Asynchronous: Asynchronous tools can simulate synchronous orchestration with fewer efforts (i.e., active waiting from the aggregation point) than the one required in the opposite scenario (i.e., different communication protocol orchestration). Therefore, we rewarded 1 point only to the tools that would allow execution in Asynchronous. Where "Not mentioned", given that a tool must implement at least one, we assumed the default to be synchronous and assigned 0 points.

- Privacy and Security independent module: zero point to the tools that focused on the ability to implement an FL pipeline but did not seem to mention nor highlight the possibility of tweaking or injecting any privacy or security module (i.e., Homomorphic encryption, secured communication protocols, blockchain). One point to all the tools that included at least one.

- Easy integration with other tools: same process applied for evaluating the containers and virtualization mechanism.

3.6 Discussion

This chapter proposes a survey of the public tools currently available for building FL pipelines. After retrieving the list of tools, we evaluated them using three different metrics: the tool's popularity (based on community adoption), growth rate, and maturity (based on our proposed review). These evaluations led to various rankings. An in-depth discussion of the results is provided below.

Based on the resulting ranking in Table 3.3, the most mature tools are Flower [22], OpenFL [187], and IBM-Federated [146]. Although the firsts two are fairly close to each other, the third appears to have a solid distance. PySyft [269], Nvidia-flare [197] and FedML [95] followed the same rate. Fedn [65] comes 7th.

This is just an initial observation, but things change when we integrate the popularity results highlighted in Table 3.1 and the growth rate outlined in Figure 3.5 into the equation. In Table 3.1, PySyft [269] and FATE [2] are the two most popular tools according to the developer's community, while Flower [22], OpenFL [187], and IBM-federated [146], cover the 5th, 6th, and 7th placement, respectively, with a considerable distance from the first two. An interesting aspect is the clear gap between what the community awarded as the most popular tools and what this work outlined as the most matures. Similarly, another essential element is the result highlighted by the growth rates reported in Figure 3.5. Leading that ranking is Flower [22], followed by FedML [95], FATE [2]and PySyft [269]. OpenFL [187] is in the 7th placement, with a growth rate of 0.23, approximately nine times smaller than the Table leader, which has a value of 1.82. With a deeper look at the various growth rates, we can see how generally $GR_{23} > GR_{12}$. This means that the tools grew more between H_2 and H_3, than between H_1 and H_2 despite having a time gap of approximately six months in each window. While this is true for almost all tools, the first four seem to have taken a much more significant leap. This could confirm the effectiveness of the work done by the development team and community contributions. However, it is crucial to note that these leaps did not have a significant impact over an observation window of 374 days (nearly one year). With some exceptions, growth rate GR_{13} seems to confirm the growth rates highlighted by GR_{12}. By comparing these two curves, we can see how Flower [22] had a stronger start in the first six months but then decreased over one year (i.e., $GR_{12} > GR_{13}$) compared to FedML [95], which gained more points in GR_{23} than it had in GR_{12}. The same considerations apply to Tensorflow-Federated (tff) [6], and Substra [78].

One interesting aspect is that regardless of the scoring we decide to consider, the top five placements seem to be occupied by the same names, reshuffled a bit. Of the 15 possible names, we can only count up to eight different tools. Of these eight different names, four are more dominant as they appear in at least two rankings: Flower [22], FedML [95], FATE [2], and PySyft [269]. The remaining four were Tensorflow-Federated (tff) [6], OpenFL [187], Nvidia-flare [197] and IBM-federated [146]. Among these eight, only OpenFL [187] supports Vertical FL.

Despite our efforts to adopt objective scoring when building Table 3.3 as

described in Section V, we are aware that other valid scoring alternatives might exist. For example, a more in-depth analysis of all functional features provided by each tool (such as communication protocols or level of modularity in the architecture) and a more granular differentiation of external tools that can be integrated into the FL pipeline could lead to different results. However, although we appreciate that such a finer approach might eventually change the distances of the current points between the elements in the lists, we would expect the main order to remain the same. This consideration arises when examining the definition of the current ranking. The success of the top two tools is mainly justified by the high scores obtained in the "Usability" and "Portability" factors outlined in Section V. This suggests that when tools have similar features with an equivalent level of maturity, the preference goes to the one with a lower entry barrier for users. Providing different documentation sources, tutorials, and access to multiple standard languages and tools may be critical for the community. As confirmed in the lower part of Table 3.3, low scoring for the worse-ranked tools might not necessarily be related to a lack of critical features, but rather to insufficient documentation that might have compromised exploitation. However, we noticed a discrepancy in Table 3.1 that led us to the following question: Why are tools with features comparable to the most popular ones, but with better documentation and more accessible entry points, not currently being considered at the same (or higher) level by the community?

Among the possible causes, we identified three main factors: participation in more significant international projects involving multiple institutions, tool adoption in various application fields, and dissemination and marketing activities by the respective engineering teams.

Although summarizing the results of the three tables might be difficult, we can say that if someone does not know where to start with FL, tools such as Flower [22] or PySyft [269] represent a good compromise between maturity and popularity for horizontal pipelines (HFL) either in data-centers or IoT devices. When Vertical FL (VFL) is required, OpenFL [187] can be the tool of choice. These recommendations are valid regardless of the application fields, as all the tools can support different models and data types. At the same time, we recognize that as future directions, more in-depth benchmarking with dedicated tools such as [139], LEAF [30], or FL-bench [134] may be needed to further understand the different peculiarities of each tool.

Another future goal is to revise the proposed criteria to account for these arguments and other factors to get closer to a comprehensive measure harmonizing the overall results.

3.7 Conclusion

Several tools for implementing FL pipelines can accelerate research activities in this field. In this study, we provided a survey of all open-source solutions and two rankings based on the tools' popularity and readiness with the aim of guiding users (including non-experts) in adopting FL solutions, boosting their exploitation, and accelerating their research and development. One key aspect of this study is that tools primarily adopted by the community are not necessarily the most mature tools available. Owing to the three harvests (searches) performed over nearly one year (374 days), we could understand the growth rate of the majority of the tools. With all the data collected, we were able to provide clear recommendations to end-users on what tool to choose when starting a new journey in FL research.

	TOOL	Watch	Fork	Star	ETA (days)	SCORES
1	PySyft [269]	211	1800	8000	1714	1.95
2	FATE [2]	134	1200	4100	1134	1.60
3	FedML [95]	37	331	1100	614	0.80
4	Tensorflow Federated [6]	66	447	1800	1301	0.59
5	Flower [22]	20	15	825	770	0.37
6	OpenFL [187]	10	76	286	437	0.28
7	IBM-Federated [146]	20	85	292	647	0.20
8	FedLab [258]	6	30	180	383	0.19
9	LEAF [30]	18	187	470	1249	0.18
10	FedGraphNN [94]	9	33	147	383	0.16
11	Fedlearn-algo [136]	8	34	80	255	0.16
12	FedJAX [191]	11	28	172	461	0.15
13	PyVertical [15] [192]	12	35	101	661	0.07
14	PriMIA [113]	8	18	102	707	0.06
15	Substra [78]	8	24	143	1252	0.05
16	Fedn [65]	7	20	59	622	0.05
17	FedBioMed (GitLab) [185]	NA	23	3	332	0.04
18	OpenFED [47]	2	1	17	300	0.02
19	APPFL [198]	2	1	7	172	0.02
20	HyFed [167]	4	3	9	361	0.01
21	PyFed [70]	3	2	5	544	0.01
22	dsMTL [33]	2	2	0	266	0.01
23	Sunday FL [172]	1	4	2	524	0.00
24	DecFL [163]	2	2	3	717	0.00
25	MTC-ETH [207]	1	1	2	881	0.00
26	Vantage6 [162]	5	2	0	1835	0.00
27	Sherpa ai [18]	3	0	0	0	-
28	FL-Pytorch (Pytorch Federated) [29]	NA	NA	NA	NA	NA
29	Chiron [103]	NA	NA	NA	NA	NA
30	FedHealth [51]	NA	NA	NA	NA	NA
31	FAE [255]	NA	NA	NA	NA	NA
32	GENO [35]	NA	NA	NA	NA	NA
33	FedTGan [267]	NA	NA	NA	NA	NA
34	FL-Bench [134]	NA	NA	NA	NA	NA
35	IPLS [178]	NA	NA	NA	NA	NA
36	Nvidia-Flare [197]	NA	NA	NA	NA	NA

Table 3.1: Tool popularity table, March harvest H_1. Legend: This table shows the list of 36 tools retrieved in March with their respective Git indicators and the cumulative scores. The results are sorted from the most popular tools on the top to the less popular tools on the bottom. Indicated with NA are the tools for which the Git repository was unavailable or not publicly accessible. If not specified otherwise, all the repositories are Github projects.

TOOL	Docs.	DX	Lang.	AI frameworks	AI type	Examples	Dist. channels	Multinode?	C/V	H/V	Sync/Asyn	PVC/SEC	Tools integration
APPFL [198]	P, Gh, Pipy, W	N.M.	Python 3.6+	Pythorc	DL	Yes	Gh, Pipy	Yes	Docker	H	S,A	N.M.	MPI orchestration
DecFL [163]	P, Gh	N.M.	N.M.	TF	DL	Yes	Gh	Yes	Required	H	S	Yes	N.M.
dsMTL [33]	P, Gh,	N.M.	R only	R-based	ML	Yes	Gh	Yes	N.M.	H	N.M.	Yes	N.M.
FATE [2]	P, Gh, W	N.M.	3.8, 3.9	N.M.	ML, DL	Yes	Gh	Yes	Docker	H,V	S,A	Yes	KubeFATE, flake, Spark
FedBioMed [185] (Gl)	P, Gh	Jupyter	python	Pythorc, TF	DL	Yes	Gh	Yes	Required	H	N.M.	Yes	Flake, Sklearn
FedJAX [191]	P, Gh, W	G. Colab, Jupyter	Python	Pythorc, TF	DL	Yes	Gh, Pipy	Simulated	N.M.	H	S,A	N.M.	N.M.
FedLab [258]	P, Gh, W	N.M.	python 3.6+	Pytorch	DL	Yes	Gh, Pipy	Yes	Docker	H	S,A	Yes	N.M.
Fedlearn-algo [136]	P, Gh	N.M.	Python 3.6, 3.7	Pythorc, TF	ML, DL	Yes	Gh	Yes	Docker	H,V	N.M	Yes	Hugging Face, Sklearn
FedML [95]	P, Gh, W, dedicated links, Slack	N.M.	Python	Pytorch, TF, JAX, mxnet	ML, DL	Yes	Gh, Pipy	Yes	Docker	H	S,A	N.M.	MPI, NCCL, MQTT, gRPC, Pytorch RPC
Fedn [65]	P, Gh, dedicated docs	Yes	Python 3.8, 3.9, 3.10	Pytorch, TF	DL	Yes	Gh	Yes	Required	H,V	N.M.	Yes	C++, scikit-learn
Flower [22]	P, Gh, slack, W, +	Jupyter, colab	Python 3.6+	Pytorch, TF, MxNet, Jax	ML, DL	Yes	Gh, Pipy	Yes	Yes	H	S,A	Yes	Android, flake, hugging Face
FLUTE [61]	P, Gh, W	N.M.	Python 3.8	Pytorch	DL	Yes	Gh	Azure Cloud	N.M.	H	S,A	N.M.	HuggingFace
HyFed [167]	P, Gh	WebAPP	N.M.	N.M.	N.M.	Yes	Gh	Yes	N.M.	H	N.M.	Yes	N.M.
IBM-Federated [146]	P, Gh, W, video tutorials	Jupyter	Python 3.6	Pytorch, TF	ML, DL	Yes	Gh, wheel	Yes	Docker	H	N.M.	Yes	Ray, Openshift
IPLS [178]													
MTC-ETH [207]	P, Gh	N.M.	N.M.	N.M.	N.M.	N.M.	Gh	Yes, but low number	Required	H	N.M.	Yes	N.M.
Nvidia Flare [197]	P, Gh, W	Jupyter	Python 3.8, 3.10	Pytorch, TF	DL, ML	Yes	Gh, pip	Yes	Docker	H	S	Yes	MONAI
OpenFED [47]	P, Gh, W	Jupyter	Python 3.6	Pytorch	DL	Yes	Gh, Pipy	Simulated	N.M.	H	N.M.	Yes	N.M.
OpenFL [187]	P, Gh, slack, W, videos	Jupyter, Colab	Python 3.6, 3.7, 3.8	Pytorch, TF	DL	Yes	Gh, Pipy	Yes	Docker	H,V	S,A	Yes	MonAI, Hugging Face
PLATO [108]	P, Gh, Website	N.M.	Python 3.9	Pytorch, TF, Mindspore	DL	Yes	Gh, pip	Yes	Docker	H	S,A	N.M.	Catalyst, Mindspore
PriMIA [113]	P, Gh, W	N.M.	N.M.	Pytorch	DL	Yes	Gh	Simulated	N.M.	H	N.M.	Yes	PySyft, K8s
PySyft [269]	P, Gh, Slack, Video, Courses	Jupyter	Python 3.8	Pythorc, TF	DL	Yes	Gh, Pipy	Yes	Docker, VMs. +	H	N.M.	Yes	PySyft, k8s
FL-Pytorch [29]	P, Gh, W, slack, videos	Custom GUI	Python 3.9	Pytorch	DL	Yes	Gh, pipy	Simulated	N.M.	H	N.M.	N.M.	N.M.
PyVertical [15, 192]	P, Gh	Jupyter	Python 3.6,3.7,3.8	N.M.	DL	Yes	Gh	Yes	Docker	V	N.M.	Yes	Syft
Substra [78]	P, Gh, W, Slack	N.M.	N.M.	N.M.	N.M.	Yes	Gh, Pipy	Yes	docker	H	S,A	N.M.	N.M.
Sunday FL [172]	P, Gh, youtube	N.M.	Python, java	N.M.	N.M.	Yes	Gh	Yes	Docker	H	N.M.	N.M.	Azure
Tensorflow Federated [6]	P, Gh, W	Colab	Python 3	TF	DL	Yes	Gh, Pipy	Simulated	N.M.	H	S,A	N.M.	N.M.
Vantage6 [162]	P, Gh, W, youtube	N.M.	Python 3.7	N.M.	N.M.	N.M.	Gh, Pipy	Yes	Docker	H	N.M.	N.M.	N.M.
XFL [234]	P, Gh, W	N.M.	Python 3.9	Pytorch, TF	DL, ML	Yes	Gh	Yes	Docker	H,V	N.M.	Yes	Crypto Algorithms

Table 3.2: Tool evaluation table: in documentation ("Docs.") and distribution channels ("Dist. channel") columns, P = Paper, Gh = Github, Gl = GitLab, W = Website. In the "Supported AI Type" column, DL = Deep Learning and ML = Machine Learning. Column "H/V" differentiates between "Horizontal" and "Vertical" FL, while the "Sync/Async" column indicates whether the tool supports synchronous S or asynchronous (A) workflows. The NM label refers to "Not Mentioned", meaning that the information did not appear as mentioned in the available documentation.

#	TOOL	Docs.	DX	Lang.	AI frameworks	AI type	Examples	Dist. channels	Multinode?	C/V	H/V	Sync/Async	PVC/SEC	Tools integration	TOTAL
1	Flower [22]	1.5	1.5	2	2.5	2	1	1	1	1	0	1	1	1.5	17
2	OpenFL [187]	1.5	1.5	2	2	1	1	1	1	1	1	1	1	1.5	16.5
3	IBM-Federated [146]	1.5	1	1	2	2	1	1	1	1	0	0	1	1.5	14
4	PySyft [269]	1.5	1	1	2	1	1	1	1	1.5	0	0	1	1.5	13.5
5	Nvidia Flare [197]	1	1	1.5	2	2	1	1	1	1	0	0	1	1	13.5
6	FedML [95]	1.5	0	1	2.5	2	1	1	1	1	0	1	0	1.5	13.5
7	Fedn [65]	1	1	2	2	1	1	0	1	0.5	1	0	1	1.5	13
8	Fedlearn-algo [136]	0	0	1.5	2	2	1	0	1	1	1	0	1	1.5	12
9	XFL [234]	1	0	1	2	2	1	0	1	1	1	0	1	1	12
10	PLATO [108]	1	0	1	2.5	1	1	1	1	1	0	1	0	1.5	12
11	FATE [2]	1	0	1.5	0	2	1	0	1	1	1	1	1	1.5	12
12	APPFL [198]	1	0	2	1	1	1	1	1	1	0	1	0	1	11
13	FedLab [258]	1	0	2	1	1	1	1	1	1	0	1	1	0	11
14	FedBioMed [185] (GitLab)	0	1	1	2	1	1	0	1	0.5	0	0	1	1.5	10
15	FedJAX [191]	1	1.5	1	2	1	1	1	0.5	0	0	1	0	0	10
16	OpenFED [47]	1	1	2	1	1	1	1	0.5	0	0	0	1	0	9.5
17	Tensorflow Federated [6]	1	1	2	1	1	1	1	0.5	0	0	1	0	0	9.5
18	PyVertical [15] [192]	0	1	2	0	1	1	0	1	1	0	0	1	1	9
19	FL-Pytorch [29]	1.5	1	1	1	1	1	1	0.5	0	0	0	0	0	8
20	FLUTE [61]	0	0	1	1	1	1	0	0.5	0	0	1	0	1	7.5
21	PriMIA [113]	1	0	0	1	1	1	0	0.5	0	0	0	1	1.5	7
22	Sunday FL [172]	1	0	1.5	0	0	1	0	1	1	0	0	0	1	6.5
23	dsMTL [33]	0.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	0	1.0	0.0	6
24	Substra [78]	1	0	0	0	0	1	1	1	1	0	1	0	0	6
25	DecFL [163]	0	0	0	1	1	1	0	1	0.5	0	0	1	0	5.5
26	Vantage6 [162]	1.5	0	1	0	0	0	1	1	1	0	0	0	0	5.5
27	HyFed [167]	0	1	0	0	0	1	0	1	0	0	0	1	0	4
28	MTC-ETH [207]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.5	0	0	1.0	0.0	2.5

Table 3.3: Tool scoring table

Chapter 4

MERGE: a model for multi-input biomedical federated learning

Driven by the Deep Learning (DL) revolution, Artificial Intelligence (AI) has become a fundamental tool for many bio-medical tasks, including AI-assisted diagnosis. These include analysing and classifying images (2D and 3D), whilst, for some tasks, DL exhibits superior performance. Diagnostic imaging, however, is not the only source of information. Tabular data, such as personal data, vital signs, and genomic/blood tests, are commonly collected for every patient entering a clinical institution. However, it is rarely considered in DL pipelines, although it carries diagnostic information.

The training of DL models requires large datasets, so large that every institution might need more data that should be pooled from different sites. Data pooling generates newfound concerns about data access and movement across other institutions spawning multiple dimensions, such as performance, energy efficiency, privacy, criticality, and security. Federated Learning (FL) is a cooperative learning paradigm aiming at addressing these concerns by moving models instead of data across different institutions.

This chapter proposes a federated multi-input model that leverages images and tabular data, providing a proof of concept of the feasibility of multi-input FL architectures. The proposed model was evaluated on two showcases: the prognosis of CoViD-19 disease and the patients' stratification in Alzheimer's disease.

Results show that enabling multi-input architectures in the FL framework allows improving the performance regarding accuracy, F1-score and generalizability with respect to non-federated models while allowing to preserve the security and data protection properties peculiar to FL.

4.1 Introduction

Artificial intelligence techniques, such as Machine Learning (ML) and Deep Learning (DL), are increasingly being exploited as tools for tackling challenges in various research fields, including the bio-medical one. One of the strengths of ML models is their capability to capture complex hidden relationships among multi-dimensional data. They have been explored for several tasks, including inter-alia, disease classification [10, 123, 151], human body segmentation [23, 140, 222, 241], the definition of diagnostic scores [59], drug discovery [81,261] and data augmentation through the generation of synthetic samples [55, 75, 208, 265]. Despite most of these examples leveraging medical images as the preferred data format, a variety of data types are collected by hospitals, clinical laboratories, and other healthcare institutions. Beyond magnetic resonance imaging (MRI), x-rays, or other types of body scans, there are also time series like the ones coming from the electrocardiograms or devices for monitoring vital signs, video from cameras recording patients' movements and positions overnight or during a rehabilitation therapy and, last but not least, text or tabular data coming from surveys, administrative and clinical records. Very often, these data are used singularly to solve specific problems.

Although these data are different, they have one common feature: they are critical data for privacy and security and must be treated appropriately.

Examples of data regulations created explicitly for managing the access and use of Health Data are the Health Insurance Portability and Accountability Act (HIPAA) [209] in the USA and the Protection of Personal Information Act (POPIA) [194]. While protecting sensitive information is a critical mission from a governance perspective, from an AI perspective the introduction of all of these regulations limits access to data.

Computing paradigms like Federated Learning (FL) [155] can help addressing this challenge [164].

In an FL scenario, multiple institutions C_i holding proprietary and critical data collaborate to train a global AI model M. The crucial aspect of the FL is that data belonging to a specific C_i never exits from the IT facility of its owner. Instead of sharing or exchanging data, different institutions iteratively aggregate local models in a single global model M that approximates the one that could be achieved by gathering all data in a single data lake.

FL is generally categorized along two main approaches: Horizontal (HFL) and Vertical (VFL) Federated Learning [249].

In both cases, multiple institutions C_i holding proprietary data are willing to train a shared model while never sharing their sensitive information.

The main difference between HFL and VFL resides in the assumption of

how data are split. In the HFL, the basic assumption is that each C_i taking part in the federation has the same feature space (i.e., data format, like MRIs) but different sample instances (i.e., different patients). Conversely, in the VFL environment, each C_i contributes to the federation by leveraging different data with different feature spaces, while the sample instances are the same. An example of HFL and VFL settings can be found in Figure 4.1.

In this chapter, we propose a multi-input model that can handle different types of data, namely images and tabular text, trained by following the orchestration paradigm of the HFL schema. Multi-input FL tasks are usually implemented by a VFL schema by design. In this case, multiple used models are specifically tailored to the different data sources as illustrated in Figure 4.1.

In this context, the proposed solution solves the trade-off between using diverse data to train different input-specific models (VFL) and gathering homogeneous data from different populations (HFL), better reflecting the needs encountered in multi-centric studies.

In addition, we demonstrate the suitability of our approach in solving medical image classification tasks and provide evidence on how our federated multi-input model can outperform the typical single-input models. To this end, we focused on two case studies addressing core issues in the field, that is COVID-19 prognosis and classification of Alzheimer's disease patients, relying on the CoViD-19 CXR (2D data) and the ADNI study (3D data), respectively.

The contributions of this work can thus be summarized as follows:

- We propose an approach to translate FL multi-input classification tasks from a complex VFL to a more straightforward HFL orchestration.
- We demonstrate how the proposed approach can effectively capture knowledge from multiple data sources, even when exploiting standard (i.e., not custom) models.

This work proposes an example of a federated multi-input model for classification orchestrated following an HFL schema. The proposed framework is open and flexible, being able to incorporate any neural network architecture and allowing to shape the parameters of the orchestration on the specific problem.

Hence, outperforming the state-of-the-art in the classification tasks was not within the scope of this contribution. We are aware that choosing the best model for the problem at hand would lead to better performance in terms of the classification scores. We leave this issue for further investigation.

Inline with open science initiatives, to help reproduce our results and facilitate the customization of the model, we provide the code and instructions on how to use it in Section 4.4.2EXPERIMENTAL PROCEDURES.

4.1.1 Federated learning in the bio-medical field

In recent years, the evolution of machine learning techniques has been dominated by the need to improve the performance of models up to levels that can be used for practical purposes. The introduction of DL has been a real revolution for many tasks, especially for image and text analysis. Unfortunately, DL is data-hungry: large-size DL models require more and more data to be trained, and pooling data is often needed to build an appropriate dataset. Pooling increases the model generalizability [76, 110], i.e. the ability of a model to deal with unseen data reliably [153]. Various efforts have been made in the research community in this respect [54, 122, 133, 169, 206]. The studies show how higher exposure to data can increase generalizability, hence the need for real-life federations [54, 122, 133, 169, 206]. FL allows the virtual pooling of different datasets while maintaining the data private and allowing the model to learn features appearing in examples from different data owners.

An example of an HFL setting could be represented by a set of institutions, like hospitals, aiming to train a shared model (i.e., for brain tumour segmentation) by leveraging MRI scans. In this scenario, the assumption is that each C_i would be able to provide MRI scans of their patients. However, in a real-world scenario, patients differ from hospital to hospital, leading to a non-i.i.d. data distribution known for degrading FL performance [37].

Despite being relatively new, FL has already demonstrated its value in addressing the generalizability challenge [203, 231]. Due to the high sensitivity of the data, many other works have been performed to explore FL in the bio-medical field [110, 212, 216]. While the effectiveness of FL in increasing the model generalizability has already been proven, finding the best way to aggregate the contributions from all the data owners remains an open challenge.

4.1.2 Multi-input classification

The idea of using multiple input sources to perform a classification task is well-known in the literature [181]. It can be found with different names like multi-modal [218], multi-view [246], multi-channel [177], or mixed data ML [224]. For the sake of clarity, we are going to refer to this topic using

the multi-input label. One of the driving factors beyond our adoption of multi-input AI approaches is data availability.

Several multi-modal datasets can be found publicly [159, 202, 229, 230]. In the bio-medical field, it is common to see different types of images coupled together. Classification of Breast tumour [20] and Alzheimer's disease (AD) [138], or the fusion of learning strategies based on different views of the same image [32], are some examples.

In other cases, the fusion involves different data types, like time series [160], or features extracted from different sources [129, 228]. Multi-input models have also been tested in contexts other than the bio-medical field. For example, a recent work [48] proposes MI-DCNN, a complete end-to-end multi-input convolutional neural network that can take full advantage of multi-modal physiological signals and automatically complete the process from feature extraction to emotion classification simultaneously.

Regardless of the data type being considered, according to Sleeman et al. [218], co-training and co-regularization are the two main possible settings of a multi-input-based approach. The co-training refers to those problems in which the information from one source can help to estimate missing details (like labels) on the other, as it happens in semi-supervised learning [62, 225]. Concerning FL, the co-training setting can be easily mapped into the VFL scenario. In the co-regularization case, the various input sources are considered contributors to build a common descriptor for representing the instances of the problem being addressed. In this last case, the fusion occurs within the classification model itself after each input has been encoded independently [114]. For this work, we relied on the co-training setting to train models in an HFL architecture.

4.1.3 Federated Multi-input

Despite a few works have addressed the challenge of having a multi-input model employed in a FL setting, the field remains less well-explored in the literature. Most of the articles propose multi-input-based approaches that are directly associated with the VFL setting. Huang et al. [100] describe a scenario where M views are distributed across M devices. The same considerations apply to other recent works. [115, 183, 252].

In the bio-medical environment, Che et al. [45] propose a detailed approach that can work in both HFL and VFL settings. However, the main focus is pipeline orchestration and data leakage prevention. Furthermore, the examples provided refer to preserving the privacy of sequential data, like real-world keyboard data, collected from the BiAffect study [11]. Qayyum et al. [183] propose a clustered FL approach where the clinical institutions are

organized in two clusters depending on what type of data they own: X-ray data cluster and Ultrasound data cluster.Mahbub Ul and Rahim [12] present a FL multi-input approach working on Internet-of-Medical-Things (IoMT). However, as claimed by the authors, the main focus is to study the heterogeneity of the hardware equipment used to simulate the different clients (medical institutions), performing evaluations on time-series models. Regarding the tasks, a recent work [266] focuses on the classification of signals from Internet-of-Things (IoT) devices, which use autoencoders to extract common representations from the different data sources. However, this contribution is not directly linked to the bio-medical field. Bernecker et al. [21] tackle the liver segmentation problem by proposing a multi-input normalization technique, which tries to encode CT and MRI scans into a common representation.

This work proposes a multi-input model that works in an HFL setting to solve classification tasks. The ultimate ambition is not to provide a specific model architecture but to share an approach that consists in combining multiple input sources to improve classification performance for an HFL pipeline. The model is meant to be a proof of concept, that is open and flexible such that it could be easily customized for other tasks also in different domains. We demonstrate the feasibility of our approach by evaluating the model on two classification tasks: prognosis of COVID-19 (2D data) disease and patients classification in Alzheimer's disease, relying on the ADNI initiative (3D data). While our final goal is not to improve the state-of-the-art performance for the two specific problems, it is essential to highlight how the two considered challenges are open and might benefit from the proposed alternative approach [9]. Alternative routes to solve the classification task have been explored by considering different feature extraction techniques [19,175]. Similar considerations apply to the classification in Alzheimer's disease. Many works demonstrate multiple efforts in addressing the challenge by using transfer-learning [199], multiple-modal [264] and multi-input techniques [80,236,259].

In summary, the few works available in the literature that implement multi-input models in federated contexts either refer directly to the VFL setting or are unrelated to the bio-medical environment, or do not address the classification task. This chapter addresses the problem of having a multi-input classification model in an HFL setting in the bio-medical field.

4.2 Results

This chapter proposes to leverage data of different types, i.e. images and tabular data, to improve testing accuracy and f1-score of federated mod-

els. Results show how the federated version of the multi-input model can outperform baseline single-input models and increase overall generalizability compared to equivalent centralized (non-federated) settings.

To provide evidence, we evaluated our approach on two different datasets: the COVID-CXR dataset [220] (2D) and ADNI dataset (3D). The first is publicly available while the second can be accessed under mild conditions.

Table 4.1: **Accuracy in centralized and federated setting.** Results (mean ± standard deviation) obtained with 5-fold cross-validation (centralized) and 5 averaged runs (federated). For each experiment setting (column) we highlighted the best performing model.

	CoViD-19 CXR		**ADNI**	
Input	**Centralized**	**Federated**	**Centralized**	**Federated**
Only images	0.731 ± 0.06	0.558 ± 0.02	0.777 ± 0.01	0.855 ± 0.04
Only tabular	$\mathbf{0.740 \pm 0.03}$	0.696 ± 0.02	0.638 ± 0.02	0.715 ± 0.03
Multi-input	0.733 ± 0.01	$\mathbf{0.734 \pm 0.01}$	$\mathbf{0.811 \pm 0.03}$	$\mathbf{0.866 \pm 0.02}$

Table 4.2: **F1 score in centralized and federated setting.** Results (mean ± standard deviation) obtained with 5-fold cross-validation (centralized) and 5 averaged runs (federated). For each experiment setting (column) we highlighted the best performing model.

	CoViD-19 CXR		**ADNI**	
Input	**Centralized**	**Federated**	**Centralized**	**Federated**
Only images	0.515 ± 0.30	0.197 ± 0.14	0.662 ± 0.03	0.735 ± 0.05
Only tabular	$\mathbf{0.562 \pm 0.28}$	0.623 ± 0.01	0.379 ± 0.02	0.573 ± 0.04
Multi-input	0.520 ± 0.30	$\mathbf{0.636 \pm 0.05}$	$\mathbf{0.742 \pm 0.03}$	$\mathbf{0.745 \pm 0.04}$

For both case studies, we ran several experiments involving training multiple models in multiple settings. More precisely, we evaluated the performance by comparing the classification results obtained by three model architectures, depending on three types of input:

- Image-only input, using a ResNet-18 [96] model;
- Tabular-only input, leveraging a Multi-Layer Perceptron (MLP) [262] model;

- Multi-input, using a concatenation layer to fuse together the outputs of the previous two models (Image-only and Tabular-only) as shown in Figure **4.3**.

In the image-only and multi-input cases, we used the 2D version of ResNet-18 for the experiments based on the CoViD-19 CXR dataset and the 3D version for experiments on the ADNI dataset.

Experiments were conducted to analyze the DL model behaviours in the following settings:

- **Isolated**: To simulate no collaboration among institutions. Models are trained independently and tested using data belonging to every single organization. In this scenario, we have three models (one for each input type) for each of the six hospitals of the CoViD-19 dataset and each ADNI (ADNI1, ADNI2, and ADNI3) study. Results are shown in Table 4.3. Moreover, isolated models have also been tested on the data of other organizations in order to discuss generalizability properties. Results are shown in Fig. 4.2.

- **Centralized**: To simulate collaboration among institutions without privacy constraints. Models are trained on a unified dataset by forcing all the data to be hosted in the same computing facility. Results are shown in Table 4.1.

- **Federated**: To simulate collaboration among institutions subject to privacy constraints. Models are trained following an HFL orchestration schema. Results are shown in Table 4.1.

For the isolated and centralized experiments, the datasets were split into three subsets: train, validation, and test using 80%, 10%, and 10% quotas, respectively. We kept the test set fixed and used the train and validation sets to train the models on five stratified folds in each centralized experiment. The reported accuracy values were obtained by evaluating the best model with the best validation accuracy on the test set.

For the federated experiments, datasets were partitioned only in train and test sets. This is mainly due to two reasons. Firstly, since each C_i would perform only one iteration before sending the model back to the aggregator, running a cross-validation step would not bring any benefits as there would not be any iterative process to optimize. Then, the tool used for running the federated experiments would not allow a five-fold cross-validation phase without massive intervention to the low-level code. To have comparable results, we preserved the percentage of training data by splitting the dataset into train (80% of the entire dataset) and test (the remaining 20%) sets,

Table 4.3: **Accuracy in the isolated setting.** Results (mean ± standard deviation) obtained with 5-fold cross-validation (centralized) and 5 averaged runs (federated). For each experiment setting (column) we highlighted the best performing model.

	CoViD-19 Hospital					
Input	**A**	**B**	**C**	**D**	**E**	**F**
Only images	0.683 ± 0.06	0.524 ± 0.06	0.619 ± 0.09	0.550 ± 0.13	0.552 ± 0.06	0.524 ± 0.04
Only tabular	0.817 ± 0.03	$\mathbf{0.838 \pm 0.04}$	0.677 ± 0.05	$\mathbf{0.900 \pm 0.03}$	0.867 ± 0.06	$\mathbf{0.786 \pm 0.02}$
Multi-input	$\mathbf{0.883 \pm 0.03}$	0.733 ± 0.04	$\mathbf{0.768 \pm 0.06}$	0.743 ± 0.09	$\mathbf{0.876 \pm 0.07}$	0.750 ± 0.02

Input	**ADNI1**	**ADNI2**	**ADNI3**
Only images	0.521 ± 0.02	$\mathbf{0.717 \pm 0.06}$	0.756 ± 0.02
Only tabular	0.725 ± 0.00	0.615 ± 0.03	$\mathbf{0.881 \pm 0.03}$
Multi-input	$\mathbf{0.871 \pm 0.07}$	0.605 ± 0.05	0.808 ± 0.08

respectively. Further considerations are shared in the Discussion subsection. We ran the federated experiments five times to ensure that the performance would not depend on a specific data split, and we averaged the results.

4.3 Discussion

This paper proposes to leverage multiple inputs, i.e. images and tabular datasets, to improve testing accuracy and f1-score of (horizontally) federated models. To determine the goodness of this approach, we structured the investigation in three steps: we first assessed the generalizability of each model in the isolated setting, we then evaluated how the performance would improve when pooling the data together in a centralized setting, and finally, we repeated the evaluation in a federated setting.

For the first step, we evaluated each model trained on the data of a specific CoViD-19 hospital (or ADNI partition) on a testing set from each of the other CoViD-19 hospitals (or ADNI partitions). Results are reported in Figure 4.2. The diagonals show the accuracy obtained when testing a model using data with the same training set distribution. Figure 4.2 shows how isolated models do not generalize well on new data. Indeed, when tested on data from other institutions, the accuracy drops significantly compared to when tested on local proprietary data. One exception is represented by the ADNI evaluation on tabular data, where the highest performance is obtained when testing the model using ADNI3. This behaviour is due to the high-class

Table 4.4: **F1 score in the isolated setting.** Results (mean ± standard deviation) obtained with 5-fold cross-validation (centralized) and 5 averaged runs (federated). For each experiment setting (column) we highlighted the best performing model.

	CoViD-19 Hospital					
Input	**A**	**B**	**C**	**D**	**E**	**F**
Only images	0.808 ± 0.03	0.364 ± 0.31	0.616 ± 0.13	0.147 ± 0.23	0.314 ± 0.32	0.260 ± 0.33
Only tabular	0.893 ± 0.01	$\mathbf{0.800 \pm 0.40}$	0.686 ± 0.08	$\mathbf{0.943 \pm 0.08}$	0.848 ± 0.11	$\mathbf{0.793 \pm 0.05}$
Multi-input	$\mathbf{0.978 \pm 0.04}$	0.200 ± 0.40	$\mathbf{0.826 \pm 0.11}$	0.600 ± 0.49	$\mathbf{1.00 \pm 0.00}$	0.687 ± 0.11

Input	**ADNI1**	**ADNI2**	**ADNI3**
Only images	0.289 ± 0.29	$\mathbf{0.667 \pm 0.08}$	0.081 ± 0.13
Only tabular	0.684 ± 0.02	0.542 ± 0.05	$\mathbf{0.173 \pm 0.12}$
Multi-input	$\mathbf{0.703 \pm 0.03}$	0.463 ± 0.13	0.058 ± 0.07

imbalance of ADNI3, as reported in Table 4.6.

For the second step, we evaluated models in a centralized setting, gathering all samples in a single data lake to simulate collaboration among institutions without privacy constraints. Pooling data from several institutions can improve the generalizability of the models.

Table 4.1 show how the centralized models generally perform better than the isolated models as in Fig. (4.2). This aspect can be appreciated by comparing the centralized accuracy with all the off-diagonal values of the two tables.

As previously noted, ADNI3 class imbalance represents an exception to this consideration, as also confirmed by looking at the f1-score obtained by the three models trained on ADNI3 in Table Creftab:f1isolated.

Overall improvements in model generalizability are also confirmed by looking at the f1-score values in the isolated and centralized setting, respectively reported in Table 4.4 and Table 4.2. Similarly to the accuracy values, the same conclusions can be drawn for the f1-scores.

In real-life scenarios, however, a centralized setting is unrealistic due to data regulations preventing medical institutions from sharing data. For this reason, it is fundamental that FL performance remains aligned with centralized performance. Sheller et al. [212] demonstrated that FL approaches are a great option when we want to increase the ability of a model to generalize by leveraging multiple data sets, but when compared to a centralized setting leveraging the same data sets is just slower in converging and

slightly less performing. For the CoViD-19 dataset, the single-input results presented in Table 4.1 confirm this behavior. However, our proposed FL approach using multi-input models outperforms the federated single-input versions and remains aligned with the centralized counterpart. This means that the approach of translating a VFL to HFL orchestration was successful. For the ADNI dataset, all the federated models outperform the centralized counterparts. While counterintuitive, this can be explained if we look at the different data partitions between the two experimental settings. Indeed, the centralized values are obtained by querying the best validation model on the test set, using an 80/10/10% train/validation/test split as opposed to the 80/20% train/test split used for the federated setting without cross-validation. Validation sets are usually helpful in identifying and selecting the most performing model among various epochs of the training process. For our federated experiments, since each collaborator C_i would perform only one training epoch before sending the model back to the aggregator, running a cross-validation step would not bring any benefits as there would not be any iterative process to optimize or best model to select. Nonetheless, running multiple folds cross-validation in a federated setting would have required a massive refactoring of the selected tool, as this feature is not available. Finally, we choose to have one training epoch because, as stated in [38], there is no scientific evidence of advantages brought by having more iterations.

Furthermore, in real-life scenarios, hospitals willing to train a federated learning model would rather use common (ideally public) external sources to test their model instead of sharing a subset of their proprietary datasets. By choosing the 80/20% training/test split, we were able to simulate a more realistic scenario by keeping the testing set external to the training process.

4.3.1 Conclusions

In this paper, we propose a new approach for solving classification tasks in the bio-medical environment in a privacy-compliant way. The method introduces an HFL setting with the advantage of leveraging multiple input sources. The basic assumption for the proposed approach is that each C_i taking part in the federation has both data types, images and tabular data, locally available and accessible. We demonstrated the goodness of our approach by running several tests based on 2D and 3D images, respectively combined with tabular data, and by comparing the results obtained by the multi-input model with the only-images and only-tabular models. Results show that enabling multi-input architectures in the FL framework allows improving the performance regarding both accuracy and f1-score with respect to non-federated models while complying with data protection practices. In particular, we tested our

method on two classification tasks: prognosis of COVID-19 disease from chest X-rays data (CoViD-CXR dataset) and detection of Alzheimer's disease from neuroimaging data (ADNI dataset).

4.3.2 Limitations and future work

As mentioned above, the main goal of this paper was to demonstrate the feasibility of a horizontal federated multi-input architecture suitable for the biomedical field. Consequently, optimizing the performance in the non-federated conditions was not targeted, and improvements concerning state-of-the-art in this respect could not be demonstrated. However, making a federated architecture available enables the exploitation of multiple sources of unshared data that allows building on top of current cutting-edge single-institution solutions overcoming the low data numerosity issue while improving the generalization ability of the overall system and naturally enabling multicentric studies.

The proposed approach does not consider the problem of missing views, which also affects clinical data processing. However, we are confident that the openness and flexibility of the proposed approach will foster research in the field, marking a step in data sharing and distributed processing.

An additional limitation of the current study is the lack of an intermediate "validated federation" setting. This scenario would re-use the same 5-folds data split used to run the centralized experiments. Despite not being as realistic as the federated scenario presented here, it would add more comparable results between the centralized and federated settings and provide additional indicators to the current study.

The future directions are:

- Propose a VFL setting where each client has a different type of input source and DL model. In particular, in a real federation, each participant can have various types of datasets, such as images, tabular features, or text reports. In this scenario, there is a need to explore new alternative ways of concatenating the different input sources. Preliminary results show that aggregating only the identical architectural layers of different networks (particularly the classifier of two different CNNs) leads to performance comparable to the typical case in which all model parameters are aggregated. Despite this new aggregation technique presenting the same typical limitations of FL (i.e., layers must be identical to be aggregated), further investigation is required to analyze the importance of the input features of the classifier.

- Explore the "validated federation" scenario described above for an additional perspective and to try to answer the question: does aggregating the best-validated models lead to a better global model?

- Expand the hyper-parameter tuning phase, starting with parameters specifically related to training a federated model compared to a centralized one. In particular, we will test our proposed architecture with more than one epoch per round, enabling us to select the best model to simulate cross-validation and, ideally, achieve performance gains.

4.4 Experimental procedures

4.4.1 Data and code availability

The code used for experimental evaluation is publicly available [39].

4.4.2 Resource availability

All the experiments for COVID-19 prognosis were performed on the HPC4AI [13] facility located in Turin (node: 8 cores per CPU, AMD EPYC-IPBP, 1 NVIDIA A40 GPU). For FL experiments, we adopted Open Federated Learning (OpenFL) [187], the new framework for FL developed by the Intel Internet of Things Group (IOTG) and Intel Labs. FL experiments were executed on a distributed environment encompassing six Collaborators (clients in the federation that train a global model on a local dataset) and one Aggregator (that aggregates the model updates received from Collaborators), each of which ran on the previously described node.

For the ADNI case study, all the experiments were performed on a 4 nodes cluster of dual socket machines equipped with Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz, with 40 physical cores per socket.

4.4.3 Architecture

Figure 4.3 displays the architecture of a multi-input NN. The general idea is to aggregate two different NNs trained on the same dataset but using different data types. In particular, the ultimate goal is to aggregate a Convolutional Neural Network (CNN) [174] and a Multi Layer Perceptron (MLP) [262], respectively trained using as input features a set of images and a tabular data frame. CNN and MLP are used as feature extractors. We used a ResNet-18 [96] as reference model for the CNN and defined a custom MLP consisting of 3 hidden layers.

Both models architectures are available on the project repository [39].

Evaluation metric: for each experiment, we returned the test accuracy value defined as the ratio between correct guesses among all guesses, more precisely:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4.1}$$

Where T and F stays for $True/False$ and P or N refers to $Positive/Negative$.

However, for applications in the bio-medical, accuracy only might not be enough to understand the goodness of a classifier. For a more in-depth analysis, we also calculated the $F1score$ for each experiment, which takes into account data unbalance. This metric is defined as the harmonic mean of $Precision$ and $Recall$ and can give us a hint about the ability of the classifier to correctly identify TP cases. More precisely, the metrics are defined as follows:

$$Precision = TP/(TP + FP) \tag{4.2}$$

$$Recall = TP/(TP + FN) \tag{4.3}$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

Model: models were trained by minimizing the binary cross-entropy loss with mini-batch gradient descent using the Adam optimizer with learning rate 1e−4 and OneCycleLR as scheduler. The local batch size was 8. The number of training epochs and FL rounds on the COVID-19 classification task was set to 100, while on the Alzheimer's detection task was set to 200.

4.4.4 Datasets

We tested the multi-input NN on two tasks:

1. Prognosis of COVID-19 disease from chest X-rays data (CXR), using the CoViD-CXR dataset

2. Detection of Alzheimer's disease from neuroimaging data, using the ADNI dataset.

The federated setting emulates a realistic medical non-i.i.d. scenario, where each C_i is hosted on an independent computing node using its dataset, contrasting with standard procedures where non-i.i.d. distributions are often simulated by splitting a single source dataset hosted on a single machine.

4.4.5 CoViD-19 dataset

This task relied on real-world data consisting of CXR and clinical parameters, divided into train and test sets. Data were collected from six hospitals in emergency conditions during the first outbreak in Northern Italy in collaboration with Centro Diagnostico Italiano and Bracco Imaging. Due to the different data collection procedures, the distribution of image features varies between each hospital, leading to the well-known problem of non-iidness [37,124]. The CoViD-19 CXR dataset consists of 1589 patients. Each of them is provided with a CXR and some clinical parameters (namely age, sex, positivity at admission, temperature, days of fever, cough, difficulty in breathing, WBC, RBC, CRP, glucose, LDH, INR, PaO2, PaCO2, pH, high blood pressure, diabetes, dementia, BPCO, cancer, CKD and respiratory failure). The dataset details are summarized in Table 4.5. Additional information about this dataset can be found at https://aiforcovid.radiomica.it/.

Table 4.5: Statistics of the CoViD-19 CXR dataset with class-balance percentages

Hospital	Samples	Positives (%)	Negatives (%)
A	120	85 (70.83%)	35 (29.17%)
B	104	59 (56.73%)	45 (43.27%)
C	151	81 (53.64%)	70 (46.36%)
D	139	76 (54.68%)	63 (45.32%)
E	101	55 (54.46%)	46 (45.54%)
F	974	546(56.06%)	428 (43.94%)

This dataset also exhibits a clear quantity skew distribution because, as shown in Table 4.5, more than 60% of the data is contained in hospital F. However, recent works [37,124] in FL literature show that quantity skew does not degrade the model's performance because most FL algorithms, such as FedAvg [155], adopt a weighted averaging of the parameters. As a result, the distribution of samples (except for the quantity) is uniform among parties, which is the easiest setting. All the images, provided in JPEG format, were rescaled to 256x256. As for data augmentation, we performed random horizontal flips and random crops with a probability of 50%.

4.4.6 ADNI dataset

Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (https://adni.loni.usc.edu). The ADNI dataset represents an ongoing, longitudinal, and multicenter study, the main landmark repository currently available for AD. Beginning in October 2004, ADNI has the primary goal of defining outcome measures to be used in clinical trials for assessing the treatment effectiveness in AD patients. However, its scope has been further widened over the years, pointing to identifying early-diagnosis biomarkers in the pre-dementia stage. A comprehensive set of clinical, neuropsychological, neuroimaging (MRI and positron emission tomography), genetic, and biochemical data are currently collected in large cohorts of healthy elderly subjects, mild cognitive impairment (MCI), and AD patients. In particular, this study has been organized into different subsequent phases, the main ones being ADNI1 (2004-2011), ADNI2 (2011-2016), and ADNI3 (2016-ongoing), each of which has witnessed the enrollment of a significant number of new subjects over time and the progressive expansion of the adopted technologies and collected data [79,239]. Up-to-date information is available at https://adni.loni.usc.edu.
For this study, we downloaded and used the 3D T1-weighted MRI scans acquired at baseline as imaging data. We leveraged healthy controls (CN) subjects and AD patients for each of the ADNI sets, as two distinct classes.Details about the acquisition protocols in terms of scanners, sequences and corresponding parameters can be found at https://adni.loni.usc.edu/methods/documents/mri-protocols/. The coded information in the updated "ADNIMERGE.csv" file were retained to build the list of tabular features used to feed the models. More precisely, the following indicators were considered: age, gender, and APOE4 ($\varepsilon 4$ allele of Apolipoprotein E). The latter, in particular, represents the most decisive known genetic risk factor for AD and assumes either 0, 1, or 2 according to the number of $\varepsilon 4$ alleles of the APOE gene. Subjects with incomplete data were removed, leading to the final samples reported in Table 4.6.

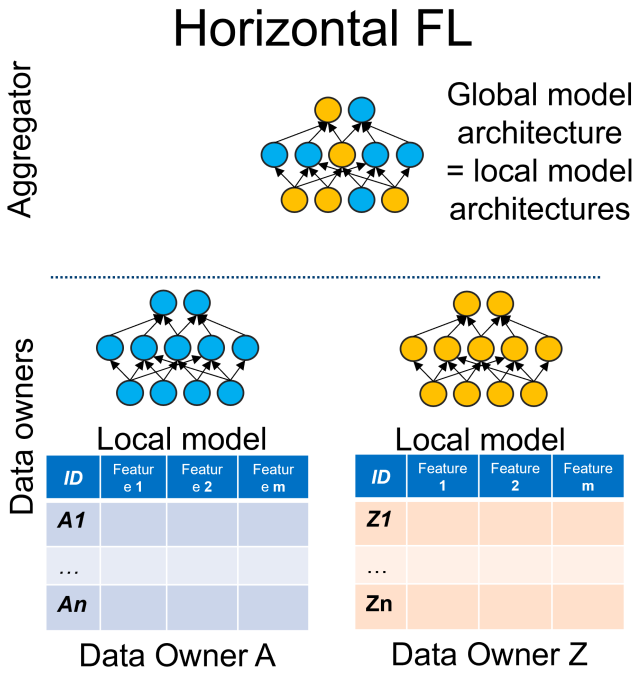
ADNI Pre-processing: The individual 3D T1-weighted volumes were minimally pre-processed, including reorientation, bias-field correction and non-linear registration to the MNI152-2mm standard space with dimensions of 91x109x91 (*fsl_anat* tool [106]). Data used to feed the models have also been normalized using the "min-max" scaling formula reported below.

For each input image x:

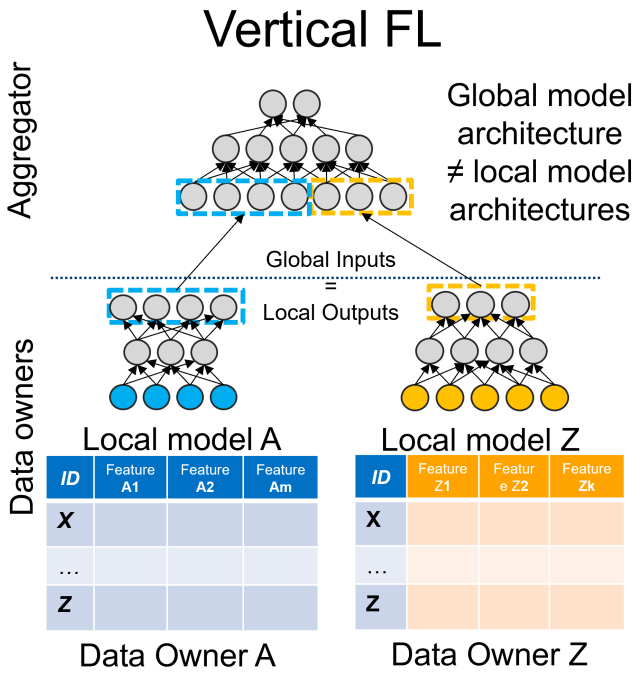
$$Scaled_x = \frac{x - min(x)}{max(x) - min(x)} \tag{4.5}$$

Table 4.6: Main demographic and clinical data for the three ADNI study cohorts. Age is reported as mean ± standard deviation values, gender as the number of males/females, while APOE4 refers to the number of $\varepsilon 4$ alleles (0, 1, or 2, respectively).

					Gender		**APOE4**		
ADNI	**Samples**	**AD (%)**	**CN (%)**	**Age (avg ± std)**	**F**	**M**	**type0**	**type1**	**type2**
1	411	184 (44.77%)	227 (55.23%)	75.58±6.21	198	213	229	143	39
2	288	143 (49.65%)	145 (50.35%)	73.69±7.35	130	158	149	107	32
3	262	51 (19.47%)	211 (80.53%)	72.01±6.44	136	126	169	75	18



(a) HFL



(b) VFL

Figure 4.1: FL settings: **(a)** HFL, with collaborating institutions sharing the same features space but different samples. In this setting, only one model is being trained locally by each institution that is also shared and aggregated. **(b)** VFL, with collaborating institutions having different features for the same samples. In this case, each C_i has its model that flows into a unified 2nd-level model shared across the federation.

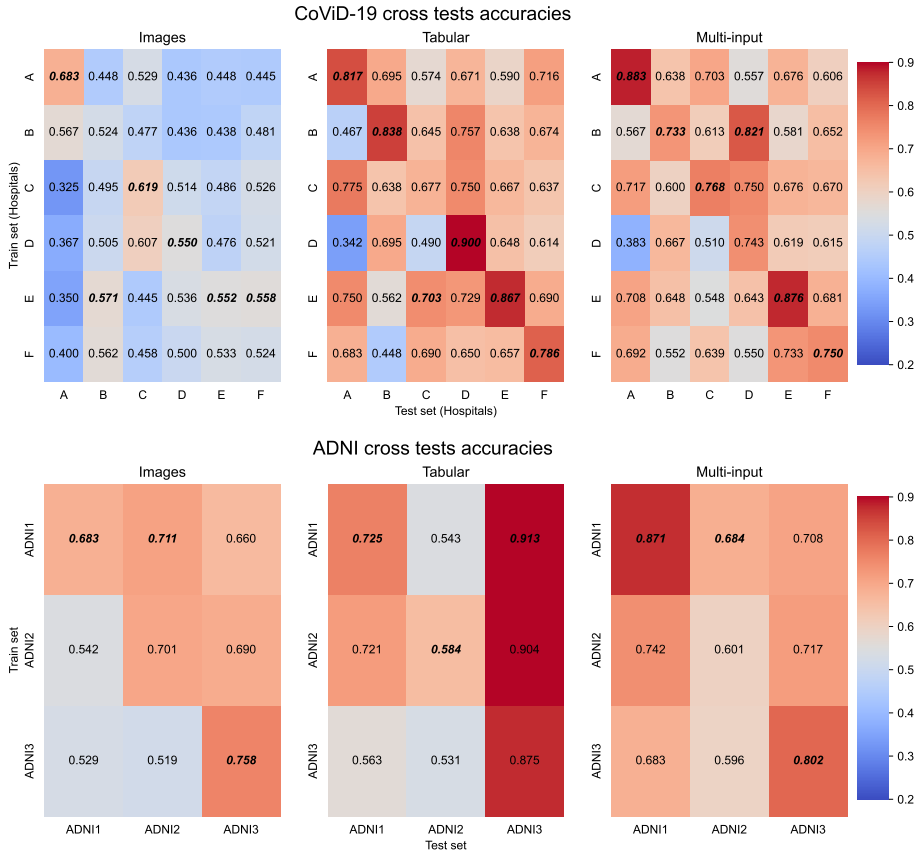


Figure 4.2: Accuracy values obtained by testing the *isolated* models on data belonging to other institutions.

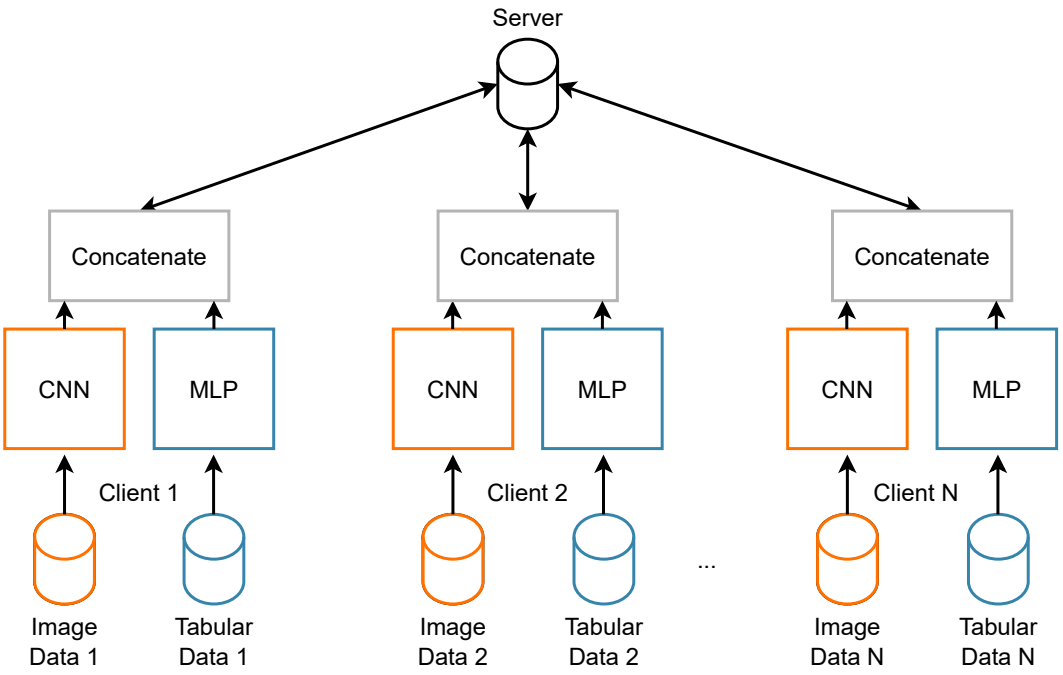


Figure 4.3: A general FL infrastructure with Multi-Input Neural Network models

Chapter 5

FLavourite: Horizontal and Vertical Federated Learning setting comparison

Federated learning (FL) enables multiple organizations to collaboratively train AI models while keeping data separate. Depending on data distribution, FL can be set up horizontally (HFL) or vertically (VFL). When multi-modal data are already available in the same infrastructure, the choice of FL setting depends on the workload needs, such as level of privacy, evaluation metrics, and time-to-train. In this paper, we perform a qualitative and extensive evaluation of the HFL and VFL settings to provide guidance on how to choose a specific setting based on workload needs. We assess the classification of Alzheimer's disease using non-independent and identically distributed (non-i.i.d.) 3D images and tabular data. The results demonstrate how, with the same aggregation function, HFL converges faster to higher accuracy and f1-score than VFL and demonstrate how HFL can be further improved due to a more flexible architecture. Finally, we provide practical considerations on when to choose which setting and share the implementations of the pipelines for reproducibility.

5.1 Introduction

In the realm of Bio-Medical research and healthcare, the exponential growth of sensitive data presents both a colossal opportunity and a significant challenge. Federated learning (FL), a cutting-edge technique that enables the training of machine learning models across decentralized data sources without compromising privacy, holds the key to unlocking unprecedented insights

while maintaining stringent data security [212, 248]. FL can help address the current needs to access data to train AI models in a privacy-compliant way [4, 97, 194].By harnessing the power of federated learning, Bio-Medical researchers and practitioners can collaborate across institutions and borders to make their models able to generalize better and preserve performance behavior when queried with new unseen samples [133, 153, 206].

There are two main types of FL settings, respectively, Horizontal FL (HFL) and Vertical FL (VFL). The discriminating factor is the way data is organized among the various collaborating institutions. In the HFL settings, each client holds the same type of data represented with the same descriptors (or set of features), but it differs in the data instances. An example of this are hospitals willing to train a joint segmentation model using brain tumor acquisitions: each hospital will align on the type of data, but patients will be different between them [72, 212]. The example of the HFL pipeline is reported in Figure 5.1.

On the other hand, in a typical VHL setting, each institution is expected to provide different types of data (diverse feature space) belonging to the same instance. Examples of VHL include life insurance agencies that collaborate with hospitals to build a decision model to obtain more precise estimates for their affiliates [42, 141]. Figure 5.2 shows our VFL pipeline.

Despite the choice of which FL setting to apply depends on how the data are organized, there are some scenarios where the context would allow for both routes. Is the case of institutions willing to train AI models to address a specific task, for which they already have all the required data in-house, organized in distributed databases. In these cases, FL represents a suitable solution to implement a distributed training to deal with large data stored in multiple clusters [43, 77, 105]. It is important to note that the suggested scenario does not match what is described as the "centralized" approach (CL) in the literature [212], which assumes that the data are stored in one location and accessible for centralized training.

For cases where both approaches would work, the choice needs to be done based on the workload needs, for example, level of privacy, evaluation metrics, and time-to-train. Despite the FL literature is constantly growing, it remains focused on studying one or the other approach, without addressing the comparison between the two scenarios.

In this chapter, we perform a qualitative evaluation of the HFL and VFL settings to share guidance on how to choose one over the other and provide a better understanding of the performance expectations of the two settings. We investigate the performance of binary classification tasks applied to the Bio-Medical scenario for both FL settings, observing the convergence behavior of three different Deep Learning (DL) models respectively trained on tabular

data, 3D images, and a multi-modal set obtained by the combination of the two previous datasets. Each of the six settings (HFL and VFL measured in the three DL models) is evaluated in terms of model accuracy, f1-score, and time-to-train.

This chapter aims at providing three main contributions.

1. Provide a qualitative evaluation of the HFL and VFL settings in terms of model accuracy, f1-score, time-to-train and degree of customization for model convergence strategies.

2. Share guidelines on how to select the most appropriate FL setting where context would allow for both options

3. Add an additional proof-point to confirm the validity of our previous work [39]

To the best of our knowledge, this is the first paper addressing the comparison of the two settings. We believe that by providing a qualitative assessment of the two FL types, the research community will be able to better understand the implications of one implementation over another and take more informed decisions in context where both can apply. We also plan to release the code if the article is accepted.

5.1.1 Chapter organization

This chapter is composed of six Sections. In section II, discusses HFL and VFL related works and applications in the Bio-Medical environment. Section III focuses on the method, including formal definitions of the two paradigms. Section IV describes the experimental phase. Results are shared and discussed in Section V. Finally, future directions and conclusions are addressed in Section VI.

5.2 related works

5.2.1 Types of FL orchestration

Depending on how the data are distributed across the different participants of a federation, we can have two main types of FL settings: HFL or VFL. In HFL scenario, each participants holds same type of data belonging to different instances. In a tabular representation this would be referred as horizontal partitioning [217]. A precursor example of HFL was introduced by Google, to train a federated model for text recommendation, using datasets from

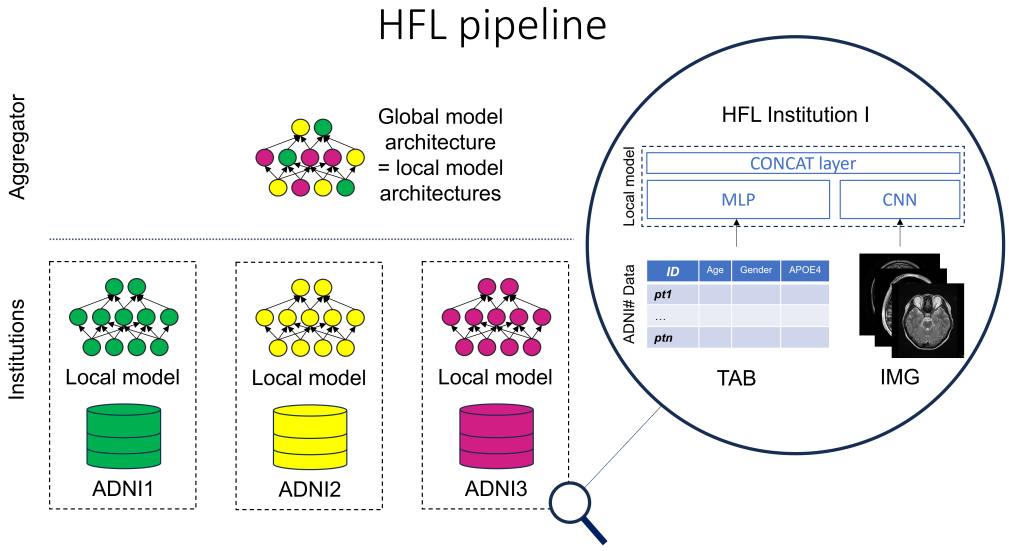


Figure 5.1: Example of our HFL pipeline for the Multi-Input model. Each i holds its own version of the ADNI dataset which is composed by tabular (TAB) and images (IMG) data. Each local model has the same architecture of the global model hosted by the aggregator.

various mobile phones [156]. In VFL, each participants provide different type of data, views, for the same instance. In a tabular representation this would referred as vertically partitioned [217]. VFL is currently explored for cross-collaborations among different institutions. An example is represented by the finance entities using VFL to estimate risk for small and micro enterprise (SME) loans [46, 53, 126].

Some studies also report a third type of FL, called Federated-Transfer-Learning (FTL) [141], described as the intersection of settings where datasets differ in both feature and sample spaces with limited overlaps [111]. With FL, collaborations can include different type of devices: from datacenters to Internet-of-Things (IOT). Some study proposes to combine the two, like VHFL [232] that leverages the aggregator computing resources to perform a parallel VFL training while waiting for the local HFL updates.

5.2.2 HFL and VFL in medical imaging

HFL was the first approach being explored, hence more extensive studies have been published on medical imaging than on VFL.

In [58] the authors provide an overview of common HFL algorithms and evaluate their performance on two machine learning prediction tasks: in-

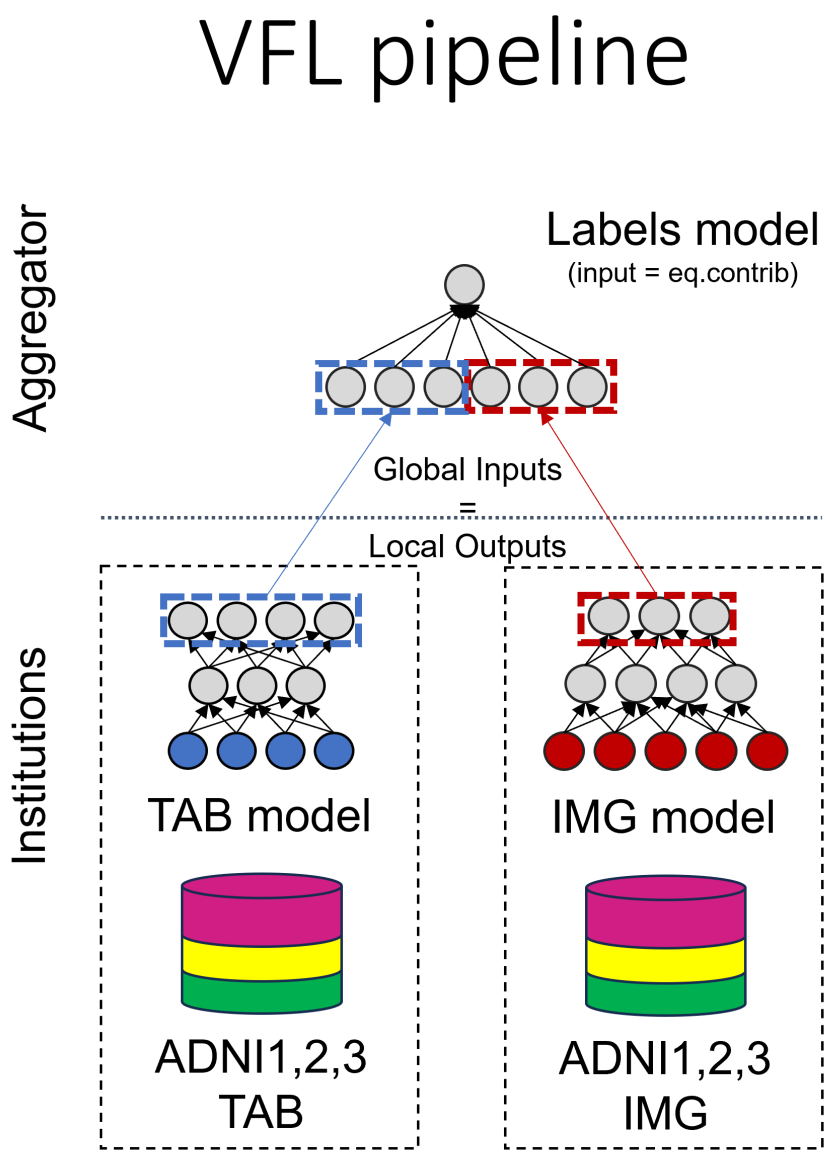


Figure 5.2: Diagram representing our pipeline for the VFL with Multi-Input model experiments. Each i holds its own data, in this case respectively Tabular (TAB) and Images (IMG). For the Tabular-only and Image-only models, we would just remove one of the two i.

hospital mortality and acute kidney injury (AKI). In-hospital mortality and prolonged length of stay prediction have also been investigated by [180], which not only compares the centralized and HFL settings, but also studies the privacy implications of the two scenarios. In [89], an HFL method to enable multi-institutional collaborations for Magnetic-Resonance-Imaging (MRI) reconstruction in a privacy-preserving manner is provided. Additional studies leverage HFL on various type of acquisitions such as gigapixels and large scale data [144,179], 3D data [39,67,196]. [39] addresses the performance improvement obtained by leveraging a multi-modal model in a HFL scenario. Similarly, [150] share a multi-modal VFL classification of Alzheimer disease using OASIS dataset [152]. Being newer VFL is still being explored, but few studies have already shared a perspective on the level of maturity and current open challenges [126, 141]. In [193] the authors propose to apply VFL to predict the likelihood of patients surviving time, after being diagnosed with cancer. [42] leverages a VFL approach to use autoencoders for hearing impairment prediction after surgery. From an edge perspective, [223] shows how different institutions such as mobile network operators and healthcare providers can collaborate through VFL.

When it comes to leveraging multi-modal data, few works explore the usage of hand-written health records [90, 147]or speech data [107] to extract and interpret medical information.

5.2.3 Study relevance

Based on the studies mentioned so far we believe this work enriches the literature by filling a gap regarding the comparison of the two methods, HFL and VFL, when the context lends itself to both implementations. Is the case of institutions owning multi-modal representation of the same data instance, like hospitals having access to tabular records (such as vital parameters or other clinical information) and visual acquisitions of their patients. Few papers addresses the comparison between HFL and centralized setting [68, 180, 212], centralized and VFL [150] or different ways to implement collaborative learning among various institutions [212, 248]. Additional studies investigate the challenges of both approaches, from the security, aggregation methods and application perspectives [126, 141, 184, 248], however nobody seems to have addressed how to select which setting, when the context would allow for both implementation. To the best of our knowledge, this is the first paper drawing some considerations around this choice.

5.3 Method

This section provides an exhaustive and formal definition of both HFL and VFL settings and the two aggregation functions explored for the HFL experiments. Before we dive into the mathematical definitions, a more detailed description of the two approaches is provided.

In a general FL pipeline, a global model Ω is trained by merging contributions from a distributed set of collaborating institutions i. However, as highlighted in Section 5.1, the way the data are organized dictates the choice of HFL or VFL setting. This choice impacts the model architecture and how the federation is orchestrated. While in the HFL, there is only one model architecture, and all the i are responsible for ensuring that data get normalized to feed it, the VFL brings some more complexity. In this case, to handle different types of data from several institutions, each i should have a local model ω_i that can accept data from a specific institution i as input. In addition, there must be a federated model Ω, which takes all the outputs of the various ω_i models as input.

A direct consequence of this difference is the level of freedom that the two approaches provide for the policy to aggregate the models. It is well known that data distributions have an impact on model convergence [39, 126]. Since HFL works with different data distributions that train the same model architecture, the setting allows more customization in terms of the ways and policies to follow when consolidating the various ω_i models into Ω [184]. Algorithms such as FedAvg [154, 156] which averages weights, and FedProx [128] which mitigates model variance when computing the average, are examples of aggregation functions [184] used in HFL. Unfortunately, in VFL, the same level of control is not guaranteed. Although it is true that the contributions from the various ω_i can be weighted differently, the fact that all the ω_i differ from each other makes the fusion less controllable.

Having highlighted these differences, the mathematical definition of the FL processes is provided after a formal representation of the data partitioning.

5.3.1 Data partitioning

As mentioned in Section 5.1, HFL and VFL differ from the way data are distributed among the various institutions I, so the formal definition should be able to address these differences. Given the multi-modal nature of our dataset used also for HFL, we define multi-modal FL as federated systems containing at least two data modalities among all the local datasets.

HFL data partitions

Let an FL setting be composed of I collaborating institutions that own, respectively, M types of data (modalities), where $I \geq 2$ and $M \geq 2$. Each institution i is expected to access a local dataset D_i defined as:

$$D_i = \{(x_i^{m_1}, x_i^{m_2}, ..., x_i^{m_M}, y_i)\}^{N_i} \tag{5.1}$$

where $i \in [1, I]$, and $N_i = |D_i|, \forall m \in [1, M]$.

In the HFL setting, the $n-$th data sample of the i-th local dataset is:

$$D_{i,n}^{HFL} = (x_{i,n}^{m_1}, x_{i,n}^{m_2}, ..., x_{i,n}^{m_M}, y_{i,n}) \tag{5.2}$$

and D_I can be defined as the total size of the distributed dataset:

$$D_I = \sum_{i=1}^{I} |D_i| = \sum_{i=1}^{I} N_i \tag{5.3}$$

VFL data partitions

As described in Section 5.3, the advantage of VFL is to allow collaborations among institutions working with different modalities belonging to the same instances (in this case, patient). Therefore, in the VFL setting, the labels are provided by one of the collaborators i.

The $n-$th data sample of the i-th institution local dataset in a vertically partitioned scenario corresponds to:

$$D_{i,n}^{VFL} = x_{i,n}^{m} \tag{5.4}$$

where $m \in [1, M]$ is a specific modality owned by institution i. The aggregation point in the VFL setting corresponds to the institution i that provides the labels. We can identify this institution as i^*, since it can either provide actual labels y_{i^*} or data of a specific modality $x_{i^*}^m$ that would serve as target to train the model on. Formally:

$$D_{i^*}^{VFL} = \tilde{y} \tag{5.5}$$

where $\tilde{y} \in \{y_{i^*}, x_{i^*}^m\}$

In our HFL scenario, we leverage a dataset composed of two modalities $images(img)$ and $tabular(tab)$ data distributed across three institutions $i \in [1, 2, 3]$:

$$D_i^{HFL} = \{(x_i^{img}, x_i^{tab}, y_i)\}_{i=1}^{N_i} \tag{5.6}$$

More details on each N_i can be found in table 5.1.

For the VFL scenario, we simulate that each i contributes with a different modality m. Depending on the type of experiment, whether it is Tabular-only, Image-only or Multi-Input, we can have, respectively, one or two i and an additional one i acting as i^*. More formally, (5.4) is defined as:

$$D_{i,n}^{VFL} = x_{i,n}^{m} \tag{5.7}$$

where $m \in \{img, tab\}$, and i^* as:

$$D_{i^*,n}^{VFL} = \tilde{y}_i^* = y_{i^*,n} \tag{5.8}$$

Since labels are provided separately.

5.3.2 FL description

FL is described [248] as an empirical risk minimization problem following the form:

$$\min_{\Omega} \mathcal{L}(\Omega, D) \tag{5.9}$$

given:

$$\mathcal{L}(\Omega, D) = \frac{1}{I} \sum_{i=1}^{I} \ell_i(\omega_i, x_i, y_i) \tag{5.10}$$

where $\mathcal{L}$ is the loss function of the global model Ω, which we need to minimize given the federated dataset D_I. The problem can be solved by minimizing the sum of all ℓ_i losses of local models ω_i at each i, given the respective x_i and y_i.

HFL description

In the standard HFL setting, the following condition is verified:

$$\tilde{\omega_i} = \tilde{\omega_j} = \tilde{\Omega}, \forall \{(i,j) \mid i,j \in [1,I]\} \tag{5.11}$$

where $\tilde{\omega}$ refers to the specific model architecture and does not consider the individual model weights ω which varies from one institution to another. This aspect helps to aggregate the various local models ω_i. For standard HFL [154], (2.2) becomes:

$$\mathcal{L}(\Omega, D)^{HFL} = \frac{1}{I} \sum_{i=1}^{I} \mathcal{W} \ell_i(\omega_i, x_i, y_i) \tag{5.12}$$

where $\mathcal{W} \geq 0$ is a normalization term that weights the contribution of the single i. Based on [154], $\mathcal{W} = N_i / D_I$, which means that each contribution

is weighted by the number of samples N_i used to train the local model ω_i, considering the overall size of the training data D_I defined in (5.3). Based on this, (5.12) becomes:

$$\mathcal{L}(\Omega, D)^{HFL} = \frac{1}{I}\sum_{i=1}^{I}\frac{N_i}{D_I}\ell_i(\omega_i, x_i, y_i) \tag{5.13}$$

That corresponds to HFL when Federated Averaging (FedAVG) [154] is in place. By adding a regularization terms to (5.10), we obtain FedProx [128]:

$$\mathcal{L}(\Omega, D)^{HFL} = \frac{1}{I}\sum_{i=1}^{I}\frac{N_i}{D_I}\ell_i(\omega_i, x_i, y_i) + \frac{\mu}{2}||\omega_i^s - \omega_i^{s-1}||^2 \tag{5.14}$$

where $||\omega_i^s - \omega_i^{s-1}||$ captures the differences between the local model hosted by i at step s and $s-1$. This explains why FedProx can help mitigate the effect of dealing with non-i.i.d. data: by reducing the impact that high-variance update can have on the global model, it facilitates faster convergence. As highlighted in [128], it is important to note that when $\mu = 0$, we get the FedAVG setting described by (5.10).

VFL description

In VFL settings, the condition (5.11) does not apply. Instead we have:

$$\tilde{\omega_i} \neq \tilde{\omega_j} \neq \tilde{\Omega}, \forall\{(i,j) \mid i,j \in [1,I]\} \tag{5.15}$$

while (5.10) becomes:

$$\mathcal{L}(\Omega, D)^{VFL} = \mathcal{L}(\Omega(\ell_1(\omega_1, x_1^1), \ell_2(\omega_2, x_2^2), ..., \ell_I(\omega_I, x_I^M), \tilde{y})) \tag{5.16}$$

where $\tilde{y}$ derives from (5.8).

Thanks to (5.16), the Multi-Input setting of our study can be now expressed as:

$$\mathcal{L}(\Omega, D) = \mathcal{L}(\Omega(\ell_i(\mathcal{I}_i, x_i^{img}), \ell_i(\mathcal{T}_i, x_i^{tab}), y)) \tag{5.17}$$

where $\mathcal{I}$ is the Image-only model, $\mathcal{T}$ is the Tabular-only model and Ω refers to the *concat* trainable layer hosted at the i^* institution.

To have comparable performance, we ensured:

$$\ell_i = \ell_j, \forall\{(i,j) \mid i,j \in [1,I]\} \tag{5.18}$$

More precisely, we used the the binary cross entropy with mini-batch gradient descent using the Adam optimizer with learning rate 1e−4 and OneCycleLR as a scheduler for all the experiments and all the models.

5.4 Experiments

We demonstrate the effectiveness of our research by measuring accuracy, f1-scores, and timing needed to perform the experiments to address the following two research questions:

- Q1) Which FL setting should be chosen when the context allows for both?
- Q2) In what way does the previous answer differ, given that HFL permits the exploration of various aggregation functions compared to VFL?

5.4.1 Experimental set-up

We evaluated the HFL and VFL settings by performing an extensive test on the The Alzheimer's Disease Neuroimaging Initiative (ADNI), extracting images and tabular data for each patient. We used both data types to train three different DL models:

- Tabular-only: a multi-layer perceptron (MLP) model with 3271 trainable parameters [262]
- Image-only: a Convolutional Neural Network (CNN) with 81733 trainable parameters already validated on MRI-based Alzheimer's disease classification [24], that we customized to work on 3D data. This is a key differentiator of this work compared to the previous one [39].
- Multi-Input: a fusion of the previous two models with a total of 85041 trainable parameters.

The detailed architecture for the Multi-Input model containing the Tabular-only and the Image-only models is presented in Fig. 5.3

Initially, we performed Centralized (CL) tests by merging all the data available in one unique institution and running the training procedure without distributing the computation. In normal conditions, the CL scenario could not occur due to the impossibility of merging data from different institutions in one unique hub. However, the results obtained with this test would serve as a reference to help set the expectations on the best obtainable validation results, since FL cannot exceed the CL performance

Similarly to what done in Chapter 4.2, we first partitioned the data set into three stratified subsets using 80% data for the training set, and the

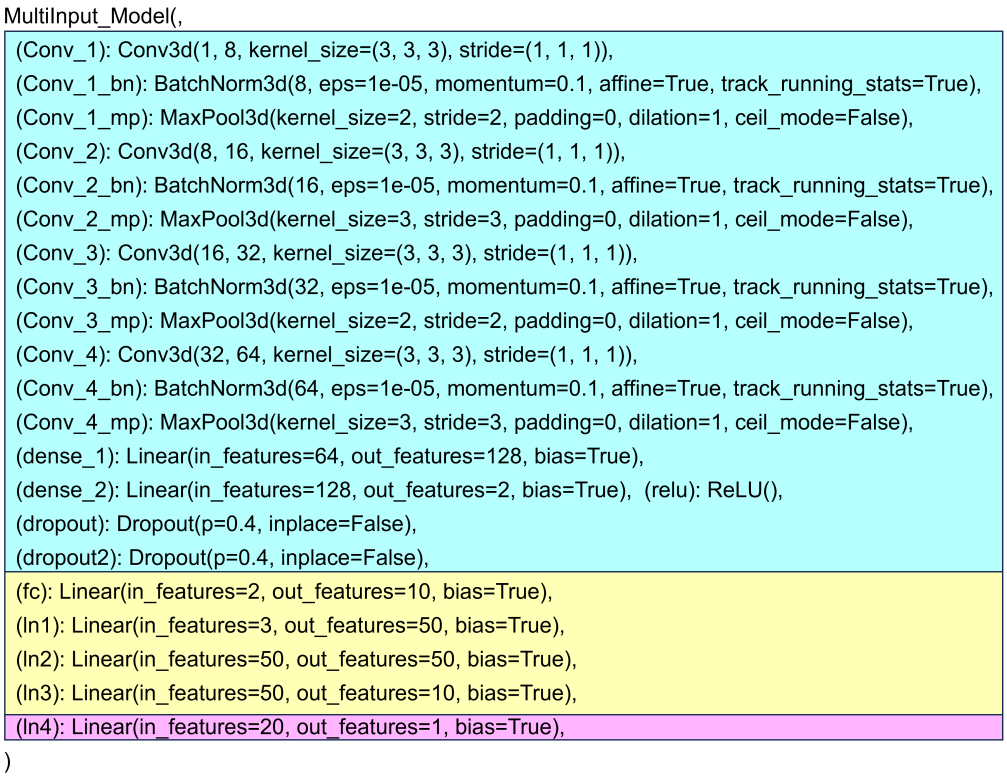


Figure 5.3: Detailed architecture of the Multi-Input Model. The Image-only model is obtained by using only the layers with cyan background. The Tabular-only MLP is presented with the yellow background. Finally, the connection layer is highlighted with the magenta background.

remaining 20% for the testing and validation sets (10% each). We then produced 5 folds ensuring that the data distributions were preserved among all the shards. We trained each model, in both settings, for 200 epochs capturing accuracy, f1-scores and execution time. The main difference between these tests and those performed in Chapter 4 is on the adopted folds: in this case the very same folds where used for all the experiments, while in Chapter 4, the data splits of the various folds might differ from one setting to another.

For the HFL scenario, we used the provided ADNI categorization based on collection dates to simulate the proprietary dataset of three different insitutions i: ADNI1 (2004-2009), ADNI2 (2010-2016) details and ADNI3 (2016-to-date) details. Each ADNI set includes different patients and since data were collected in different timeframes, it is fair to expect that machinery and clinical equipment used to acquire the data were different. This means that the data we used to train our models were non-i.i.d. For the VFL scenario, we preserved the same fold partitions, however, we organized the data in two vertical silos containing all images and all tabular data, respectively, to simulate a two entities willing to collaborate on the same task. For all experiments in the VFL setting, the aggregation hub was the only one that held the ground-truth labels for the classification task.

To better understand the behavior of the convergence of each model, we finally explored the impacts of aggregation functions on HFL by implementing a version that would use the FedProx algorithm [128] to consolidate all local models ω_i into the global Ω version.

This section illustrates the adopted architecture and describes the data pre-processing phases for both datasets.

5.4.2 ADNI dataset

The dataset adopted including the data preprocessing procedures and the definition of the evaluation metrics are the same as those adopted in Chapter 4, described in Section 4.4.6.

The main demographic and clinical data for the three cohorts of the ADNI study described in Table 4.6, are also reported below for readability in Table 5.1.

ADNI Pre-processing: The individual 3D T1-weighted volumes were minimally pre-processed, including reorientation, bias-field correction and non-linear registration to the MNI152-2mm standard space with dimensions of 91x109x91 (*fsl_anat* tool [106]). Data used to feed the models have also been normalized using the "min-max" scaling formula reported below.

For each input image x:

Table 5.1: Main demographic and clinical data for the three ADNI study cohorts. Age is reported as mean ± standard deviation values, gender as the number of males/females, while APOE4 refers to the number of $\varepsilon 4$ alleles (0, 1, or 2, respectively).

					Gender		APOE4		
AD	**Sampl**	**AD (%)**	**CN (%)**	**Age (avg ± std)**	**F**	**M**	**typ**	**type1**	**type2**
1	411	184 (44.77%)	227 (55.23%)	75.58±6.21	198	213	229	143	39
2	288	143 (49.65%)	145 (50.35%)	73.69±7.35	130	158	149	107	32
3	262	51 (19.47%)	211 (80.53%)	72.01±6.44	136	126	169	75	18

$$Scaled_x = \frac{x - min(x)}{max(x) - min(x)} \tag{5.19}$$

However, for applications in the *Bio-Medical*, accuracy only might not be enough to understand the goodness of a classifier. For a more in-depth analysis, we also calculated the f1-score for each experiment. This metric is defined as the harmonic mean of *Precision* and *Recall* and can give us a hint about the ability of the classifier to correctly identify TP cases. More precisely, the metrics are defined as follows:

$$Precision = TP/(TP + FP) \tag{5.20}$$

$$Recall = TP/(TP + FN) \tag{5.21}$$

$$f1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.22}$$

5.4.3 Resources

For FL pipelines, we adopted Open Federated Learning (OpenFL) [72, 212]: an open FL framework developed by the Intel Internet of Things Group (IOTG) and Intel Labs. To ensure a reliable and fair comparison among

Table 5.2: **Accuracy values for all models in each setting:** "Exp" describes the type of experiment which is also associated to the model used: "IMG" is the Image-Only model, "TAB" the Tabular-only model and "MI" the Multi-Input model. The table shows how accuracy changes depending on the type of experiment, whether is training or validation and on which setting: VFL, HFL or Centralized (CL).

	Exp	VFL	HFL	CL
VAL	IMG	0.79 ± 0.02	**0.8 ± 0.03**	0.88 ± 0.01
	TAB	**0.73 ± 0.03**	0.71 ± 0.04	0.73 ± 0.04
	MI	0.82 ± 0.03	**0.83 ± 0.03**	0.87 ± 0.01
TRAIN	IMG	1.0 ± 0.0	0.99 ± 0.0	1.0 ± 0.0
	TAB	0.72 ± 0.01	0.73 ± 0.01	0.74 ± 0.01
	MI	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0

the two settings, we avoided any networking path simulating all the federations on the same machine. FL experiments included two types of entities: Collaborators (institutions i of the federation that train a global model on their local dataset) and one Aggregator (that aggregates the model updates received from Collaborators), each of which ran on the same node.

The specific adopted hardware consisted of a 4 node cluster of dual socket machines equipped with Intel(R) Xeon(R) Platinum 8480+ CPU @ 2.30GHz, with 56 physical cores per socket.

5.5 Results

This section presents the results and addresses the answer to the two research questions posed in Section 5.1, with some additional considerations on the overall learning.

5.5.1 Q1: Which FL setting should be chosen when the context would allow both?

To address Q1, we first need to discuss the VFL and HFL differences in terms of performance.

Table 5.2 shows the training and validation accuracy obtained by the three different models, in each setting. At a first glance there are two aspects that confirm the validity of our approach:

Table 5.3: **F1 values for all models in each setting,** "Exp" describes the type of experiment which is also associated to the model used: "IMG" is the Image-Only model, "TAB" the Tabular-only model and "MI" the Multi-Input model. The table shows how F1-scores change depending on the type of experiment, whether is training or validation and on which setting: VFL, HFL or Centralized (CL).

	Exp	VFL	HFL	CL
VAL	IMG	0.58 ± 0.07	$\mathbf{0.64 \pm 0.03}$	0.76 ± 0.04
	TAB	0.55 ± 0.05	$\mathbf{0.57 \pm 0.04}$	0.57 ± 0.05
	MI	0.65 ± 0.08	$\mathbf{0.69 \pm 0.04}$	0.77 ± 0.04
TRAIN	IMG	0.92 ± 0.02	0.93 ± 0.01	0.93 ± 0.02
	TAB	0.57 ± 0.02	0.58 ± 0.02	0.57 ± 0.02
	MI	0.93 ± 0.02	0.93 ± 0.01	0.93 ± 0.02

1. the Centralized experiments (CL) outperform the VFL and HFL results, as expected from the state of the art [212];

2. results from our previous study [39], demonstrating the effectiveness of the Multi-Input (MI) model over the Tabular-only (TAB) or Image-only (IMG) models, are confirmed despite using a more performing Image model suitable for Alzehimer classification.

The table shows how HFL can achieve higher accuracy than VFL when images are involved. On the contrary, VFL with 0.73 ± 0.03 gets very close to the CL results with 0.73 ± 0.04 when dealing with Tabular data.

Since accuracy might not be the best metric to measure the effectiveness of a model in the Bio-Medical field, Table 5.3 provides the f1-scores obtained in the same conditions. In this case, HFL confirms to be more effective than VFL in Images-based experiments, and in addition outperforms VFL by achieving performance similar to CL setting when it comes to experiments based on Tabular data, respectively: 0.57 ± 0.04 and 0.57 ± 0.05. As we stated in Section 5.1, due to the high heterogeneity, the performance of the bare model is not the only parameter that matters. Execution time represents a critical factor that can help choose one setting over another.

In Table 5.4 the execution times are reported. For each setting, the Table reports three values in seconds: Epoch time (Ep.), computation time (CT), and communication time (NT). As we can see, HFL confirms having a lower and more consistent execution times for all the experiments. The

Table 5.4: **Execution time among settings** This table shows the total Epoch time (Ep.), the compute time (CT) as the time spent by all components to compute their part of the pipeline, and communication time (NT) of each setting. All timings are expressed in seconds. The best values are highlighted.

	HFL			VFL		
	Ep.	**CT**	**NT**	**Ep.**	**CT**	**NT**
IMG	55.73	53.46	2.27	1734.62	1730.65	3.96
TAB	5.86	3.85	2.01	130	112.26	17.74
MI	55.76	53.50	2.26	982.74	968.74	14

faster execution in the HFL setting can be explained by looking at how the pipeline is orchestrated (Fig. 5.1). In HFL each i can loop through all the training rounds (Epochs) independently, and then share the updates with the aggregator. This does not happen in VFL (Fig. 5.2) where at each training round, the process requires sending updates to the aggreator and waiting to receive updates in return. This step is essential to complete the backpropagation step which helps update the model weights. This path, increases the communication time as confirmed by the "NT" values of Fig. 5.4. Another important aspect highlighted by the same column is the delays difference: consistent and close to $2s$ in HFL and dynamic in the VFL. Taking a closer look to the "NT" column in VFL, we can see how the communication time seems to be inversely proportional to the Epoch time: the higher the Epoch time, the smaller the communication. This seems correct if we think that for non-compute intensive workloads, like training the Tabular-only model, we need more time to communicate and exchange information than is required to train the model. What might not be easily explainable is the big difference between Computation-time (CT) in the IMG and MI experiments of the VFL setting, taking respectively 1734.62 versus 982.74 seconds to complete the 200 rounds. The same values for the HFL show how the two models IMG and MI are fairly aligned with 55.73 and 55.76. The explanation to the gap in the VFL setting is to be searched in the tool we used to perform the experiments. With OpenFL 1.5 [72], the VFL feature at time of this work was still experimental and some process might not be fully optimized. We retrieved this answer by interacting directly with the team that developed the tool. Despite this issue, by combining what we know from the State of the Art regarding the two settings and the results reported in Table 5.2 and 5.3, we can see how HFL would be the ideal choice where performance and

Table 5.5: **Execution time among settings** This table shows the Accuracy (left) and f1-score (right) obtained in each experiment setting by the two aggregation functions FedAVG [154] and FedProx [128]. The best values are highlighted.

		Accuracy		f1-score	
	Exp	**FedAVG**	**FedProx**	**FedAVG**	**FedProx**
VAL	IMG	0.86 ± 0.04	**0.87 ± 0.01**	0.73 ± 0.08	**0.73 ± 0.06**
	TAB	0.73 ± 0.03	**0.74 ± 0.03**	**0.57 ± 0.03**	**0.57 ± 0.03**
	MI	**0.87 ± 0.03**	**0.87 ± 0.03**	0.74 ± 0.07	**0.75 ± 0.06**
TRAIN	IMG	1 ± 0.0	1 ± 0.0	0.94 ± 0.01	0.94 ± 0.01
	TAB	0.73 ± 0.01	0.73 ± 0.01	0.58 ± 0.02	0.57 ± 0.02
	MI	1 ± 0.0	1 ± 0.0	0.94 ± 0.01	0.94 ± 0.01

speed for training are the top priorities.

5.5.2 Q2) In what way does the previous answer differ, given that HFL permits the exploration of various aggregation functions compared to VFL?

As mentioned in Section 5.1, HFL has the additional advantage of allowing testing with different aggregation functions. This is a critical feature when we work with non i.i.d. data. To see if we could further improve our performance highlighted in Table 5.2 and 5.3, we ran additional experiments comparing two aggregation functions: FedAVG [154] and FedProx [128].

Results are reported in Table 5.5. As we can see for both, the accuracy and f1-scores, FedProx algorithm can further improve the performance over FedAVG.

Figure 5.4 shows the convergence curves for training loss, validation accuracy and validation f1-score on each fold, while training the MI model using both the aggregation functions. In most cases, FedProx [128] converges faster and to higher value than [154]. To answer Q2, this means that the HFL pipeline not only outperforms the VFL one, but thanks to the additional possibility of customizing the aggregation functions, performance can be improved to get closer to the CL values.

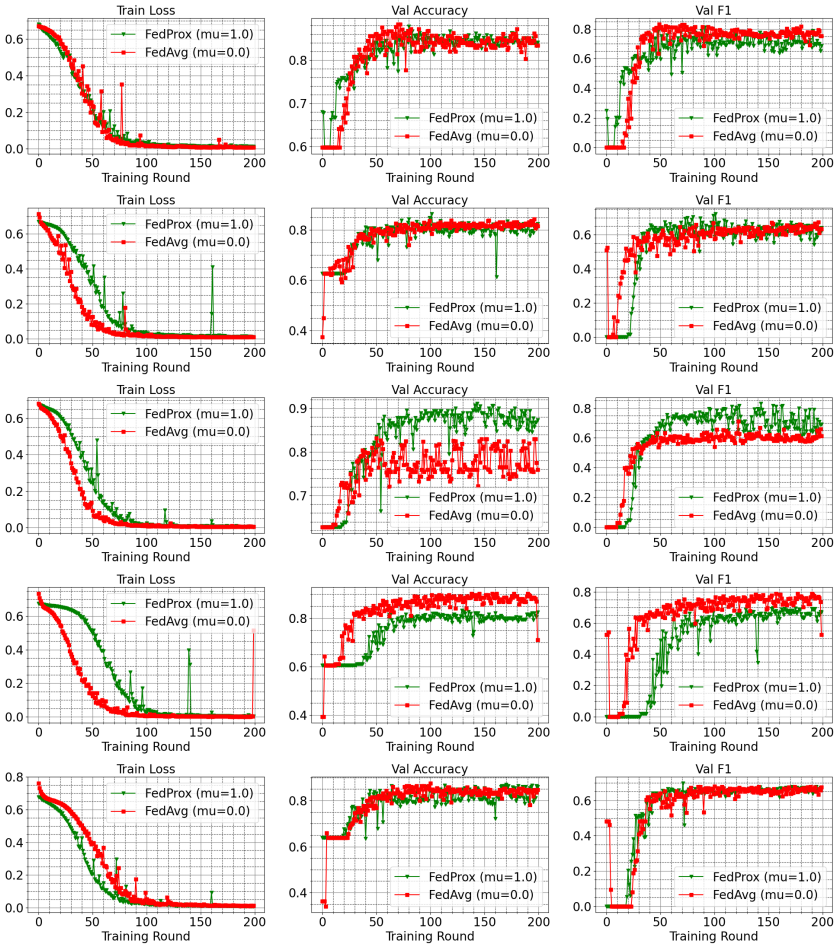


Figure 5.4: Aggregation functions compared in terms of Training Loss, Validation accuracy and validation F1 score. Each row represent the metrics obtained by running the training with the multi-input setting over the 5-folds cross-validation. As depicted, in most of cases the FedProx [128] algorithm seems to converge earlier and to higher values than FedAVG [154]

5.6 Conclusions

As discussed in Section 5.5, HFL seems to be the best choice in terms of validation performance, time-to-train and degree of customization that can be done by working on aggregation functions. An additional aspect that might play an important role when deciding which setting to adopt, is the Security. From this perspective, VFL is commonly known to be more robust [126]. An intuitive explanation can be found if we consider that during a training round, each i exchanges only a partial amount of information with the aggregator. That information alone would not be enough to understand the aggregated model architecture or the type of data used in input in each i. This differs from the HFL which shares a local and self-contained copy of the global model architecture at each step. Hence, attackers willing to steal data through reverse engineering or the model architecture, might have more challenges to address when studying a VFL pipeline.

5.6.1 Future Directions

This chapter proposes a comparison between HFL and VFL in terms of performance and time-to-train and degrees of freedom for aggregation functions. Furthermore, the proposed experimental setup enriches the contribution of our previous work [39]. Moving forward we would like to:

- Extend testing across multiple nodes to simulate a realistic production environment where all the i are hosted on different machines;
- Augment the number of institutions to include ablation studies, by disabling one or more collaborators at run-time, to better understand the behavior of the two settings when only a subset of the federation works.
- Replicate the experiments by using different FL tools [190], to add proof points to the performance scalability of VFL.
- Investigate how performance and results change when integrating confidential computing aspects in the pipeline.

Chapter 6

Other contributions

This chapter contains the abstract of the publications achieved through external collaborations, as anticipated in Chapter 1 in Section 1.3.8.

External collaborations refer to activities conducted with industrial or research partners for which the candidate does not appear as the first or second author despite providing relevant value to the context. The abstract of these contributions are provided below. In addition, given the first author's contribution, the abstract of the submitted patent is provided below.

6.1 OpenFL: the open federated learning library

6.1.1 Abstract [72]

Objective. Federated learning (FL) is a computational paradigm that enables organizations to collaborate on machine learning (ML) and deep learning (DL) projects without sharing sensitive data, such as patient records, financial data, or classified secrets. Approach. Open federated learning (OpenFL) framework is an open-source python-based tool for training ML/DL algorithms using the data-private collaborative learning paradigm of FL, irrespective of the use case. OpenFL works with training pipelines built with both TensorFlow and PyTorch, and can be easily extended to other ML and DL frameworks. Main results. In this manuscript, we present OpenFL and summarize its motivation and development characteristics, with the intention of facilitating its application to existing ML/DL model training in a production environment. We further provide recommendations to secure a federation using trusted execution environments to ensure explicit model security and integrity, as well as maintain data confidentiality. Finally, we de-

scribe the first real-world healthcare federations that use the OpenFL library, and highlight how it can be applied to other non-healthcare use cases. Significance. The OpenFL library is designed for real world scalability, trusted execution, and also prioritizes easy migration of centralized ML models into a federated training pipeline. Although OpenFL's initial use case was in healthcare, it is applicable beyond this domain and is now reaching wider adoption both in research and production settings. The tool is open-sourced at github.com/intel/openfl.

6.2 Model-Agnostic Federated Learning

6.2.1 Abstract [161]

Since its debut in 2016, Federated Learning (FL) has been tied to the inner workings of Deep Neural Networks (DNNs). On the one hand, this allowed its development and widespread use as DNNs proliferated. On the other hand, it neglected all those scenarios in which using DNNs is not possible or advantageous. The fact that most current FL frameworks only allow training DNNs reinforces this problem. To address the lack of FL solutions for non-DNN-based use cases, we propose MAFL (Model-Agnostic Federated Learning). MAFL marries a model-agnostic FL algorithm, AdaBoost.F, with an open industry-grade FL framework: Intel OpenFL. MAFL is the first FL system not tied to any specific type of machine learning model, allowing exploration of FL scenarios beyond DNNs and trees. We test MAFL from multiple points of view, assessing its correctness, flexibility and scaling properties up to 64 nodes. We optimised the base software achieving a 5.5x speedup on a standard FL scenario. MAFL is compatible with x86-64, ARM-v8, Power and RISC-V.

6.3 A Performance Analysis for Confidential Federated Learning

6.3.1 Abstract - PREPRINT version

Federated Learning (FL) has emerged as a solution to preserve data privacy by keeping the data locally on each participant's device. However, FL alone is still vulnerable to attacks that can cause privacy leaks. Therefore, additional security measures, at the cost of increasing runtimes, become necessary. The Trusted Execution Environment (TEE) approach offers the highest degree of

security during execution. However, TEEs suffer from memory limits which prevent safe end-to-end FL training of modern deep models. State-of-the-art approaches limit secure training to selected layers, failing to avert the full spectrum of attacks or adopt layer-wise training affecting model performance. We benchmark the usage of a library OS (LibOS) to run the full, unmodified end-to-end FL training inside the TEE. We extensively evaluate and model the overhead of the different security mechanisms needed to protect the data and model during computation (TEE), communication (TLS), and storage (disk encryption). The obtained results across three datasets and two models demonstrate that LibOSes are a viable way to seamlessly inject security into FL with limited overhead (at most 2x), offering valuable guidance for researchers and developers aiming to apply FL in data-security-focused contexts.

6.4 Submitted Patent

Today, Federated Learning allows distributed training of DL algorithms preserving data privacy. Once a federation is established, each institution trains a copy of the model using local data. What gets shared is the model and not the data.

This paradigm looks like a natural fit to leverage IoT devices. However, since a federation needs to be statically defined, it could introduce an obstacle to any efforts made to make new IoT seamlessly integrable in a given environment. Zigbee is a great example of such kind of devices. About Federated Learning, currently it is implemented as a synchronous or asynchronous process. In the first case, each collaborator performs a given number of local steps (epochs), then shares their locally trained model with a centralized server. In asynchronous Federated Learning, each collaborator can request a community update from the controller whenever they are ready to share updates. While both approaches are valid to regulate the execution flow of a federation, the problem on how to facilitate a seamless and dynamic integration of new devices in the federation remains open.

As Ph.D. student and employee of Intel Corporation, I started working on a solution to address the challenges described above.

The result of this effort is a patent that was submitted to the Intel internal review committee on August 22 with the title *Dynamic bootstrapping of new devices for learning federations in connected smart environments.*

Chapter 7

Conclusions and future work

This thesis shows how FL pipelines can be implemented for large-scale institutions (data centers) and shares some considerations on how to optimize model convergence when dealing with multimodal input data, both in HFL and VFL settings. As certified by Fig. 2.2 and Fig. 2.3 in Chapter 1, the community has just started exploring FL for the biomedical field, and although this thesis is a small and extremely limited contribution to the state of the art, there are some key learning that we can share from this experience. Before diving into the specific details, a summary of the contributions and research perspectives is provided.

7.1 Thesis status: what's been done

Whether for research or industry development, the community needs tools to build FL pipelines. As a first step, a direct contribution was provided to develop a tool that the community could leverage to conduct research and help advance the state of the art.

To help in this aspect, we started from zero assumption and performed a deep search of what the Web had to offer to help the development of the research environment. Chapter 3 explains how the rising interest in the research community and industrial *R&D* departments has influenced the development and evolution of a multitude of new tools for implementing FL pipelines. If from one perspective this aspect is encouraging, on the other hand, the availability of such a large amount of toolboxes requires clear indications on their main features, in particular usability, adoption, readiness (with respect to the features available for research experiments) and maturity (with respect to real life deployment in an industrial production environment). The outcome of this first research activity can be summarized

in four main contributions.

1. Provide an updated and fair list of tools publicly available for implementing FL pipelines;

2. Identify and describe the key aspects required by the research community and map them into a list of features that tools should include;

3. Provide a detailed comparison of the tools, based on objective metrics including commonly used indicators and ability to match the needs highlighted in the previous point.

After building the code to automatically retrieve content related to FL tools, a manual assessment of the outcome of the queries was performed. In order to keep the survey objective and inclusive, we proposed a few evaluation metrics to capture various aspects that can be consolidated into 3 main branches: Usability, Portability, and Flexibility.

More in detail, each category would include:

- Usability: Documentation, User Interface (UI), Programming Language(s)

- Portability: Templates/Examples availability, Distribution channels, Multi-node mode, Open-Source

- Flexibility: Containers/Virtualization, Modular architecture, Privacy and Security, Easy integration with other tools

We provide an objective assessment of the tools to allow the research community to be able to accelerate its activities, promote fairness by proposing a clear framework for comparative studies, and help tool providers identify ways to improve their products. The availability of a ranking of the FL tools will also boost their exploitation for production environment, where such tools are still largely unexplored.

This initial effort allowed us to identify one tool, OpenFL [187], which was then used to implement the remaining exploitation activities. Thanks to the collaboration with University of Turin, we were able to stress the tool and propose a multi-input model that can handle different types of data, namely images and tabular text, trained by following the orchestration paradigm of the HFL schema. Multi-input FL tasks are usually implemented as VFL by design. In this case, multiple models were specifically tailored to the different data sources.

The idea beyond the proposed solution was to solve the trade-off between using diverse data to train input-specific models (VFL) and collecting homogeneous data from different populations (HFL), better reflecting the needs encountered in systems hterogeneity studies, defined in Section 2.2.3. In addition, we demonstrated how the proposed approach is helpful for medical image classification tasks and provide evidence of how our federated multi-input model can outperform the typical single-input models. To do this, we focused on two case studies addressing core issues in the field, i.e., COVID-19 prognosis and classification of Alzheimer's disease patients, relying on the CoViD-19 CXR [220] (2D data) and the ADNI study [239] (3D data), respectively.

The scientific contributions of this activity, expanded by Chapter 4, can be summarized as follows:

4. We proposed an approach to translate FL multi-input classification tasks from a complex VFL to a more straightforward HFL orchestration.

5. We demonstrated how the proposed approach can effectively capture knowledge from multiple data sources, even when exploiting standard (i.e., not custom) models.

We experimentally evaluated our approach on two different datasets: the COVID-CXR dataset [220] (2D) and ADNI dataset [239] (3D). In line with open science initiatives, to help reproduce our results and facilitate the customization of the model, we made the code and instructions on how to use it available at [39].

In summary, we selected an example of a federated multi-input model for classification orchestrated following an HFL schema. Thanks to this achievement, research can now leverage the proposed framework and us it with different neural network architectures and training algorithms. Our findings stimulated the appetite for more investigation of the different settings, in order to better understand the role of FL heterogeneity. More precisely, thanks to the contribution described in Chapter 4, we wanted to better understand how the data heterogeneity, model heterogeneity, and system heterogeneity, captured by the two HFL and VFL settings, affect model convergence.

To have a more objective view of the problem, in Chapter 5 we have proposed a mathematical formalization of the context, by providing the definition of the HFL and VFL scenarios considering the respective ways of partitioning the training datasets. We then asked ourselves the following two research questions:

- Q1) Which FL setting should be chosen when the context would allow for both HFL or VFL?
- Q2) In what way does the previous answer differ, given that HFL permits the exploration of various aggregation functions compared to VFL?

To address them, we further investigated the performance of a biomedical scenario for both FL settings as initiated in Chapter 4, observing the convergence behavior of three different Deep Learning (DL) models trained on a multimodal non-i.i.d. dataset with 3D images and tabular data. Each of the six settings is evaluated in terms of model accuracy, f1-score, and time-to-train.

By combining what we know from the state of the art regarding the two settings and the results proposed in Chapter 5, our finding suggests that HFL pipelines represent the ideal choice where performance and speed for training are the top priorities as opposed to VFL which instead looks like a much better fit when privacy and security are the must-have feature. This also means that the HFL pipeline not only outperforms the VFL, but thanks to the additional possibility to customize the aggregation functions, performance can be even further improved to get closer to the CL values.

Overall, the contribution of Chapter 5, can be summarized as follow:

6. Propose a qualitative evaluation of the HFL and VFL settings in terms of model accuracy, f1-score, time-to-train and degree of customization for model convergence strategies.
7. Share guidelines on how to select the most appropriate FL setting where context would allow for both options
8. Add an additional proof-point to confirm the validity of our previous work [39]

In summary, we conducted a qualitative evaluation of the HFL and VFL settings to share guidance on how to choose one over the other and provide a better understanding of the performance expectations of the two settings.

7.2 Adherence to RQs and Objectives

In the previous Section, we discussed the key contribution of the research activity of this thesis. To draw a contour on the actual contribution of this work and have a more realistic understanding of the depth and breadth of

this work, few considerations are shared with respect to the initial objective and adherence of the research activities performed.

In Chapter 1, we define two main research questions addressed in this work. The questions are reported here to improve readability:

- **RQ 1)** How can the development and analysis of federated learning tools and methodologies enhance the efficiency and efficacy of multimodal data integration in biomedical applications?
- **RQ 2)** How to exploit the advantages and address the challenges of Horizontal and Vertical federated learning approaches in the same context?

All activities have been carried out with the adherence to the research question as a key requirement. More precisely, for the first item, based on the published results reported in Chapter 3, it is clear how the choice of the proper tools to implement FL pipelines is crucial to the outcome of research or federations in general. Effective tools can handle the complexity of distributing the data across multiple nodes while ensuring security and privacy and facilitating development to focus on the model and data preparation routines. All of these aspects play a direct role in the overall level of heterogeneity of a federation. As highlighted in Chapter 3, all the various tools retrieved and investigated within a window of time of one year appeared to converge to a similar list of features with different degrees of flexibility. Tools that can handle system complexity contribute to keeping system heterogeneity low (defined in Chapter 2, Section 2.2.3). This has a positive impact on developers and researchers that can focus on the other types of heterogeneity, respectively, related to data and models. By combining this insight with the results outlined in Chapter 4, we know that the ability to experiment with multimodal data, and the reliability of the results, is highly dependent on the level of maturity of the tools. In Chapter 5, for example, we experienced critical delays when running VFL experiments. After a thorough investigation, we found that the problem belongs to the adopted tool more than to the complexity of the proposed pipeline. This may seem trivial; however, it is important to note that the tool we adopted is one of the most used tools in the world, supported and provisioned directly by Linux Foundation. The Linux Foundation is a nonprofit organization established in 2000 to support and promote the growth of Linux and collaborative software development. It encompasses more than 900 open source projects, more than 777 thousands of developers contributing with an estimate of 51 million lines of code per week, and more than 17 thousand contributing organizations around the world. However, despite the prestigious source, the level of limitations found

in the described tool should help paint a more realistic picture of the current level of maturity of the tools available to the FL research community and set the expectations accordingly. The direct implication of this growing environment is that method exploitation is more simplified for HFL pipelines than it is for VFL. This affects both research and industry.

Thanks to the mature experience, we could further investigate the heterogeneity aspects in multimodal settings, sharing some considerations about the role of aggregation functions in the two settings. These experimental phases helped in addressing the second research question. In Chapter 4 we have seen how to build a custom federation with the VFL ability to draw together different data modalities while preserving a (relatively) low-complexity orchestration process typical of HFL settings. This approach highlighted the advantages of the two schemas, but also emphasized the limitations, such as the need to have multimodal data for a unique list of patients hosted in the same collaborating institution instead of seeing it distributed across the federation. On the other hand, Chapter 5 took the opposite direction by preserving the standard orchestration schema of two distinct settings (HFL, VFL) and emphasizing their advantages and drawbacks. More in depth, we have learned how the HFL setting can lead to better model convergence than VFL due to the additional tuning power represented by the various aggregation strategies. This twofold approach consisting of:

- combining the settings to study their combined strength and weaknesses (Chapter 4) and the pros and cons, and
- comparing the HFL and VFL different schema to highlight respective pros and cons in depth

has resulted in helpful considerations that address the second research question.

7.3 Key Learnings

After highlighting what has been done in Section 7.1, the key learnings that this work has produced are provided below as a general three-year journey. This section draws considerations on the whole experience, not focusing only on what worked well but sharing light on secondary activities and attempts that did not produce direct results but better expertise.

7.3.1 Federated Dataset

As part of the execution plan detailed in Table 1.1 (Chapter 1, Section 1.3.7), WP2 focused on having a properly configured working infrastructure to facilitate experiments and research for the upcoming WPx. This is why we have started investigating the performance of a classification algorithm using a centralized data setting, while configuring an FL environment. The purpose of this first investigation was to implement a working code and assess centralized performance. As a second step, we would have ported the centralized code to a federated setting to proceed with further extensive examination.

This initial experiment would be based on Medical Imaging data, more specifically, leveraging the Alzheimer's Disease Neuroimaging Initiative[1] (ADNI) database (adni.loni.usc.edu). The data we had access to was a subgroup of the ADNI3 package, more specifically, we were dealing with 3D MRI-T1 scan of a total of 565 subjects grouped into different categories depending on the severity of the disease. The categories and the respective number of subjects are reported below:

- Cognitively normal (CN), made of 306 samples
- Subjective Memory Complaints (SMC), made of 50 samples
- Mild cognitive impairment (MCI), made of 70 samples
- Early mild cognitive impairment (EMCI), made of 82 samples
- Late mild cognitive impairment (LMCI), made of 37 samples
- Alzheimer disease (AD), made of 20 samples

Initially, we wanted to use this exercise as a software development vehicle to have a classification system we could use for reference. We assessed performance of a multiclass classification approach, to gather initial insights.

Knowing that the development environment would be based on specific hardware resources, we had to select a classification model that would consider these aspects (i.e. memory bound preferred versus compute bound models) and with good performances reported in the state-of-the-art for similar use cases.

We performed various runs by changing the learning rate (lr) and the number of epochs (e) and keeping the batch size set to the same number of

[1]Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative

classes (bs = 2). To compare performance, we recorded the loss, accuracy, precision, recall, and f1 score for both the training and the validation phase.

Given the medical environment, we ranked the scores according to f1 scores with the best scores the highest. The initial performance was not good as both the f1 score and the accuracy could barely pass the 50% threshold. Among possible causes was the amount of data available. Hence, we enhanced the original dataset with additional (coherent) data retrieved from the ADNI repository. This activity took a few months to complete but resulted in the addition of more than 1000 new patients distributed throughout ADNI1 and ADNI2. It is important to note how most of the published works base their conclusions on experiments conducted on limited datasets or non-deep learning models. Most of the original and currently adopted aggregation functions (discussed in 2.4 were actually proposed on standard ML models [116, 128, 235].

Thanks to this initial activity, we built a realistic experimental environment, with natively heterogeneous data and a mixture of standard and custom DL models.

7.3.2 Experimental environment

Building an experimental environment to test federated learning pipelines in a realistic way poses several significant challenges. Some aspects have already been highlighted in the previous section, where we discussed the critical role of mature tools. In addition to that are the challenges of realistic scenarios. Most of the current state-of-the-art studies refer to simulated FL pipelines. These settings are typically hosted on one machine, and the various institutions are represented by different processes running on the same host and exchanging information. Although conclusions drawn around statistical model convergence and data contribution can be relevant, the simulation introduces a level of abstraction that does not allow proper evaluation on communication efficiency or cost of security measures.

For this work, we created a small distributed infrastructure composed of four nodes (servers) connected through an ethernet network. This allowed us to reproduce a more authentic federated environment that can simplify and stimulate investigation on a bigger scale. However, this realistic scenario preserved all the heterogeneity layers that can be found in typical FL settings (as described in Chapter 2, Section 2.2.3). Therefore, we had to prioritize the items on which to focus our research, to avoid roadblocks associated with an extreme complex environment.

- **Security and privacy features**: Our FL setting did not consider

security aspects, including memory encryption for protected run-time execution. Messages exchanged among participants in each setting were encoded as *.zip* files, to have more realistic measurements of execution timings. Security and privacy applied to FL represents a specific research branch itself that we did not explore. The only contribution we provided was done through the collaboration with University of Turin by investigating the cost of encryption mechanism applied to protect runtime protection 6.

- **Distributed compute**: as mentioned in the beginning, the maturity level of the adopted tools is instrumental and directly impacts the reliability of results and conclusions drawn from any experimental phase performed on the specific tool. In this case, the tool was lacking for effective multinode scalability for VFL. So, all the experiments reported in Chapters 4 and 5 are simulated using a single machine. On the other hand, all the HFL conclusions are reliable and the proposed approaches can be easily scaled and applied to more nodes.

- **Modular tool architecture**: a direct implication of the previous aspect is the ability to isolate data pre-processing procedures. In realistic FL scenarios among data-center institutions it is fair to expect each one of them to define private pre-processing routines tailored for the data owned by the respective institution. In our case this feature was implemented and each ADNI would benefit from this isolation. This is another critical learning that can help produce more realistic federations.

7.3.3 Scientific aspects

When working with FL, the variance of the pipelines due to the different types of heterogeneity can add layers of complexity to a standard AI pipeline which already includes several challenges. FL can be seen as a connection of multiple simpler steps such as dataset preprocessing, model training, validation, and model optimization typical of an AI pipeline, but scaled to I institutions in parallel. The first key learning is that centralized settings are a great starting point to have for ensuring the experiments run and setting the expectations in terms of performance. This holds for the research environment and for production, although real-life deployment might not always allow this opportunity. Having a centralized dataset is not realistic in real-life scenarios, and research is directly applied to distributed datasets without a point of reference. Is the situation experienced by [179] where the authors

built a real-life federation, merging contributions from more than 71 clinical institutions worldwide to advance the state of the art in brain tumor segmentation. In these types of studies, there might be unpredictable aspects, for example the existence of *data poisoning*. Data poisoning is an attack method that aims to corrupt a machine learning model by introducing malicious data into its training set. Although the real thread is represented when external agents try to impact training routines, there are situations where data poisoning is the result of weak coordination or simple errors. What would happen in case an institution uses wrong data or not-well preprocessed data to train its local copy of the global mode? Data poisoning can be difficult to locate even when it is genuine and internal.

Additional key learnings are related to what has already been said regarding model convergence studies and the reliability of the conclusions drawn so far. Most of the foundational studies, such as [128, 235], propose aggregation functions that are tested with standard ML algorithms or are incomplete. The problem of model convergence and the formalization of the mathematical background is still very fragmented. As we learned from Chapter 5, HFL seems to be the best choice in terms of validation performance, training time, and degree of customization that can be done by working on aggregation functions. However, from a security perspective, VFL represents a more robust option. This is due to the nature of VFL that aggregates parts of a larger model, making it difficult for an attacker to reconstruct unseen parts of information. This means that VFL is less customizable from the aggregation function perspective to HFL. Regardless, finding the best strategy or rule-of-thumbs on how to fuse together contributions, whether to work in a latent space or other encoded representation, is yet to be explored. This is why despite the research progress for HFL, the question marks seem to grow when it comes to VFL settings, which is still largely unexplored. Ideas for possible future directions are provided in Section 7.4.

In summary, this thesis emphasizes the exploitation of the two settings HFL and VFL with the aim of having a better understanding of *how* and *what* might affect the performance of a federation. The extensive experiments performed on 2D and 3D multimodal dataset contributed to achieve the goal from a data perspective as well as on the model convergence behavior when dealing with multiple single or merged models.

7.4 Future directions

As described in many sections and chapters, this work represents a small contribution to what can and needs to be done in FL. We have helped the

researcher and developer community with suitable tools and practical guidelines and validated and best practices to improve model convergence and FL adoption in both settings. In this section, we provide a few considerations on how this work can be further extended to address some of the challenges faced and open question marks.

7.4.1 Stress multiclass

All the experiments performed in this thesis involve a binary classification of healthy patients, composed of CN patients and affected patients, which captures the AD category. More research is needed to assess the quality of our results when dealing with multiple classes. On one side, adding classes to the problem would increase the data heterogeneity and hence the complexity, but on the other hand, it would allow us to have more perspectives on model convergence and perhaps help draw more conclusions on finding best ways to aggregate models. This would apply to both HFL and VFL.

Deeper investigation with multiclass classification task should also be performed in multi-input setting, where each class would contribute with multiple modalities. A first batch of experiments could work to validate the results obtained with the binary setting. Successively, the focus could be oriented towards the hyperparameters tuning and inner-validation loops. During a standard step of a federation, each collaborator performs local training and then sends the updated model to the aggregator hub. Each collaborator could perform as many training routines as they prefer and also include some validation rounds to select the best local model. Today, this step is often skipped and the state-of-the-art does not share a clear indication on this aspect: for example, [154] suggests increasing the number of rounds for local training to reduce communication costs, while [38] did not observe any significant impact on having more rounds before sending the local model to the aggregator. Both studies refer to training rounds with no specific indications on the importance of local validation.

7.4.2 Formalize model convergence problem

Formalizing FL pipelines can be extremely challenging due to the high heterogeneity typical of federated settings. However, few efforts have been made to better understand the convergence behavior. Most of the aggregation functions described in Chapter 2, Section 2.4 provide their own formalization of the concept. Among the key studies, two of them rise two problems, which we clearly connected:

- the *Client-Drift problem* [116]: It occurs when the local updates to a machine learning model made by clients (or nodes) are diverse because of nonidentically distributed data across clients. This can lead to a situation where the aggregated model does not perform well universally or converges more slowly, as the updates from different clients may not be aligned and can cancel each other out to some extent.

- the *Objective-Inconsistency problem* [235]: the objective-inconsistency problem arises due to the differences in objectives between the central server (aggregator) and the local clients (collaborators). Each client may optimize its local model based on its own dataset, which can be diverse and non-iid. This can lead to inconsistencies when the central server aggregates these local models because the local objectives may not align with the global objective. The problem manifests as decreased performance or slower convergence of the aggregated global model.

What these problems suggest is that the high statistical heterogeneity affects the updates of each client resulting in slow and unstable convergence. This means that the path to a stable and optimal convergence is yet to be found and would benefit from further investigation.

This applies primarily to HFL scenarios, where all the collaborators share a local copy of the global model. For what concerning VFL, next Section discusses the few aspects that need to be considered when proposing future directions.

7.4.3 Deep dive on VFL

VFL is extremely fascinating. If HFL can unlock new possibilities to train models on very large distributed dataset with a certain degree of complexity and heterogeneity, VFL can push both boundaries even further. The VFL concept is relatively new and the research community is expanding on a daily basis. However, there is a particular VFL scenario that can be associated with an older technique called *Split Neural Networks (SNN)* [41]. SNNs refer to a type of architecture in which the computation is divided into multiple segments that can be executed on different devices or locations. From a practical standpoint, in SNNs, various layers of a DL models are split and distributed on multiple machines. The training process requires all of these machines to complete their partial computation in order to proceed with the next group of layers hosted by another machine. As happens in VFL, SNNs are especially useful for scenarios where computational resources are distributed or for privacy concerns where different parts of the data are available at different nodes. Arguably, a discriminant factor between the

two approaches is in the way data are distributed. While in VFL vertically partitioned data are a must-have, the same does not represent a requirement for SNNs which are instead more focused on reducing the computational load for training the model. The two approaches also differ in the execution principles since SNNs often involve sequential execution of segments of the network, whereas VFL typically involves parallel training instances. Despite these differences, if we look at model convergence or strategies to combine various parts of a distributed model to maximize training outcomes, we believe the two approaches might share some behavioral aspect. Considering that the SNN concept has been more investigated than VFL, we think that by exploring the overlap between the two paradigms, we can have a faster route to better understand VFL convergence aspects.

For scenarios where VFL cannot be reduced to SNNs, other directions can be explored. For example, in cases where each participant can have various types of data set, such as images, tabular features, or text or audio reports, there is a need to explore alternative ways of concatenating the different input sources. This can be done in the aggregation center by allowing each institution to provide a n-dimensional vector of the same or different length, or a combination of multiple information. Preliminary results [39] show that aggregating only the identical architectural layers of different networks (particularly the classifier of two different CNNs shown in Chapter 4) leads to performance comparable to the typical case in which all model parameters are aggregated. Despite this new aggregation technique presenting the same typical limitations of FL (i.e., layers must be identical to be aggregated), further investigation is required to analyze the importance of the input features of the classifier.

Bibliography

[1] Website: Data protection definition from european general data privacy regulation (gdpr). Available at https://www.edps.europa.eu/data-protection/data-protection_en.

[2] Website: Fate ai. Available at https://fate.fedai.org/. Accessed: March 28, 2022.

[3] Website: Google scholar. Available at https://scholar.google.com/. Accessed: March 28, 2022.

[4] Website: Protection of personal information act (popi act) - popia. Available at https://popia.co.za.

[5] Website: Semantic scholar, a free, ai-powered research tool for scientific literature. Available at https://www.semanticscholar.org/. Accessed: March 28, 2022.

[6] Website: Tensorflow federated. Available at https://www.tensorflow.org/federated. Accessed: September 28, 2022.

[7] Website: Top500 hpc centers of the world. Available at https://top500.org/.

[8] Abdelmoniem, A. M., Ho, C., Papageorgiou, P., and Canini, M. A comprehensive empirical study of heterogeneity in federated learning. *IEEE Internet Things J. 10*, 16 (2023), 14071–14083.

[9] Aggarwal, P., Mishra, N. K., Fatimah, B., Singh, P., Gupta, A., and Joshi, S. D. COVID-19 image classification using deep learning: Advances, challenges and opportunities. *Comput. Biol. Medicine 144* (2022), 105350.

[10] Al-Issa, Y., and Alqudah, A. M. A lightweight hybrid deep learning system for cardiac valvular disease classification. *Scientific Reports 12*, 1 (2022), 1–20.

[11] Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., and Taha, K. Efficient machine learning for big data: A review. *Big Data Res. 2*, 3 (2015), 87–93.

[12] Alam, M. U., and Rahmani, R. Fedsepsis: A federated multimodal deep learning-based internet of medical things application for early detection of sepsis from electronic health records using raspberry pi and jetson nano devices. *Sensors 23*, 2 (2023).

[13] Aldinucci, M., Rabellino, S., Pironti, M., Spiga, F., Viviani, P., Drocco, M., Guerzoni, M., Boella, G., Mellia, M., Margara, P., Drago, I., Marturano, R., Marchetto, G., Piccolo, E., Bagnasco, S., Lusso, S., Vallero, S., Attardi, G., Barchiesi, A., Colla, A., and Galeazzi, F. HPC4AI, an AI-on-demand federated platform endeavour. In *ACM Computing Frontiers* (May 2018).

[14] Ali, M. S., Ahsan, M. M., Tasnim, L., Afrin, S., Biswas, K., Hossain, M. M., Ahmed, M. M., Hashan, R., Islam, M. K., and Raman, S. Federated learning in healthcare: Model misconducts, security, challenges, applications, and future research directions - A systematic review. *CoRR abs/2405.13832* (2024).

[15] Angelou, N., Benaissa, A., Cebere, B., Clark, W., Hall, A. J., Hoeh, M. A., Liu, D., Papadopoulos, P., Roehm, R., Sandmann, R., Schoppmann, P., and Titcombe, T. Asymmetric private set intersection with applications to contact tracing and private vertical federated machine learning. *CoRR abs/2011.09350* (2020).

[16] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics* (2020), PMLR, pp. 2938–2948.

[17] Baldi, P., and Sadowski, P. J. A theory of local learning, the learning channel, and the optimality of backpropagation. *Neural Networks 83* (2016), 51–74.

[18] Barroso, N. R., Stipcich, G., Jiménez-López, D., Ruiz-Millán, J. A., Martínez-Cámara, E., González-Seco, G., Luzón, M. V., Veganzones, M. A., and Herrera, F. Federated learning and differential privacy: Software tools analysis, the

sherpa.ai FL framework and methodological guidelines for preserving data privacy. *Inf. Fusion 64* (2020), 270–292.

[19] Barstugan, M., Ozkaya, U., and Öztürk, S. Coronavirus (covid-19) classification using CT images by machine learning methods. In *Proceedings of the 4th International Conference on Recent Trends and Applications in Computer Science and Information Technology, Tirana, Albania, May 21st - to - 22nd, 2021* (2021), E. Xhina and K. Hoxha, Eds., vol. 2872 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 29–35.

[20] Bekker, A. J., Greenspan, H., and Goldberger, J. A multi-view deep learning architecture for classification of breast microcalcifications. In *13th IEEE International Symposium on Biomedical Imaging, ISBI 2016, Prague, Czech Republic, April 13-16, 2016* (2016), IEEE, pp. 726–730.

[21] Bernecker, T., Peters, A., Schlett, C. L., Bamberg, F., Theis, F. J., Rueckert, D., Weiss, J., and Albarqouni, S. Fednorm: Modality-based normalization in federated learning for multi-modal liver segmentation. *CoRR abs/2205.11096* (2022).

[22] Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., and Lane, N. D. Flower: A friendly federated learning research framework. *CoRR abs/2007.14390* (2020).

[23] Bhatt, P., Sahoo, A. K., Chattopadhyay, S., and Bhatt, C. A study on deep learning models for medical image segmentation. In *Artificial Intelligence in Industrial Applications*. Springer, 2022, pp. 161–174.

[24] Böhle, M., Eitel, F., Weygandt, M., and Ritter, K. Layer-wise relevance propagation for explaining deep neural network decisions in mri-based alzheimer's disease classification. *Frontiers in aging neuroscience 11* (2019), 194.

[25] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), pp. 1175–1191.

[26] Bonawitz, K. A., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Overveldt, T. V., Petrou, D., Ramage, D., and Roselander, J. Towards federated learning at scale: System design. In *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019* (2019), A. Talwalkar, V. Smith, and M. Zaharia, Eds., mlsys.org.

[27] Briggs, C., Fan, Z., and Andras, P. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020* (2020), IEEE, pp. 1–9.

[28] Briggs, C., Fan, Z., and Andras, P. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020* (2020), IEEE, pp. 1–9.

[29] Burlachenko, K., Horváth, S., and Richtárik, P. Fl_pytorch: optimization research simulator for federated learning. *CoRR abs/2202.03099* (2022).

[30] Caldas, S., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A benchmark for federated settings. *CoRR abs/1812.01097* (2018).

[31] Calzada, I. Citizens' data privacy in china: The state of the art of the personal information protection law (pipl). *Smart Cities 5*, 3 (2022), 1129–1150.

[32] Cao, G., Wang, Y., Zhang, M., Zhang, J., Kang, G., and Xu, X. Multiview long-short spatial contrastive learning for 3d medical image analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022* (2022), IEEE, pp. 1226–1230.

[33] Cao, H., Zhang, Y., Baumbach, J., Burton, P., Dwyer, D., Koutsouleris, N., Matschinske, J., Marcon, Y., Rajan, S., Rieg, T., Ryser-Welch, P., Späth, J., Schwarz, E., Alnæs, D., Andreasssen, O., Chen, J., Degenhardt, F., Doncevic,

D., Eils, R., and Meyer-Lindenberg, A. dsmtl - a computational framework for privacy-preserving, distributed multi-task machine learning. *Bioinformatics 38* (09 2022).

[34] Carneiro, T., Medeiros Da NóBrega, R. V., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., and Filho, P. P. R. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access 6* (2018), 61677–61685.

[35] Carpov, S., Gama, N., Georgieva, M., and Jetchev, D. Genoppml - a framework for genomic privacy-preserving machine learning. In *IEEE 15th International Conference on Cloud Computing, CLOUD 2022, Barcelona, Spain, July 10-16, 2022* (2022), C. A. Ardagna, N. L. Atukorala, R. Buyya, C. K. Chang, R. N. Chang, E. Damiani, G. B. Dasgupta, F. Gagliardi, C. Hagleitner, D. S. Milojicic, T. M. H. Trong, R. Ward, F. Xhafa, and J. Zhang, Eds., IEEE, pp. 532–542.

[36] Casella, B., Colonnelli, I., Mittone, G., Birke, R., Riviera, W., Sciarappa, A., Cavazzoni, C., and Aldinucci, M. A performance analysis for confidential federated learning. In *IEEE Security and Privacy, SP 2024 - Workshops, San Francisco, CA, USA, May 23, 2024* (2024), IEEE, pp. 40–47.

[37] Casella, B., Esposito, R., Cavazzoni, C., and Aldinucci, M. Benchmarking fedavg and fedcurv for image classification tasks. In *Proc. of th 1st Italian Conference on Big Data and Data Science, ITADATA22, Milan, September 20-21, 2022* (2022).

[38] Casella, B., Esposito, R., Sciarappa, A., Cavazzoni, C., and Aldinucci, M. Experimenting with normalization layers in federated learning on non-iid scenarios. *CoRR abs/2303.10630* (2023).

[39] Casella, B., Riviera, W., Aldinucci, M., and Menegaz, G. MERGE: A model for multi-input biomedical federated learning. *Patterns 4*, 11 (2023), 100856.

[40] Casella, B., Riviera, W., Aldinucci, M., and Menegaz, G. Protocol for training merge: A federated multi-input neural network for covid-19 prognosis. *STAR Protocols 5*, 1 (2024), 102812.

[41] Ceballos, I., Sharma, V., Mugica, E., Singh, A., Roman, A., Vepakomma, P., and Raskar, R. Splitnn-driven vertical partitioning. *CoRR abs/2008.04137* (2020).

[42] Cha, D., Sung, M., Park, Y.-R., et al. Implementing vertical federated learning using autoencoders: Practical application, generalizability, and utility study. *JMIR medical informatics 9*, 6 (2021), e26598.

[43] Chandiramani, K., Garg, D., and Maheswari, N. Performance analysis of distributed and federated learning models on private data. *Procedia Computer Science 165* (2019), 349–355. 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019.

[44] Chang, K., Balachandar, N., Lam, C. K., Yi, D., Brown, J. M., Beers, A., Rosen, B. R., Rubin, D. L., and Kalpathy-Cramer, J. Distributed deep learning networks among institutions for medical imaging. *J. Am. Medical Informatics Assoc. 25*, 8 (2018), 945–954.

[45] Che, S., Kong, Z., Peng, H., Sun, L., Leow, A. D., Chen, Y., and He, L. Federated multi-view learning for private medical data integration and analysis. *ACM Trans. Intell. Syst. Technol. 13*, 4 (2022), 61:1–61:23.

[46] Chen, C., Zhou, J., Wang, L., Wu, X., and All. When homomorphic encryption marries secret sharing: Secure large-scale sparse logistic regression and applications in risk control. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021* (2021), F. Zhu, B. C. Ooi, and C. Miao, Eds., ACM, pp. 2652–2662.

[47] Chen, D. Openfed: An open-source security and privacy guaranteed federated learning framework. *CoRR abs/2109.07852* (2021).

[48] Chen, P., Zou, B., Belkacem, A. N., Lyu, X., Zhao, X., Yi, W., Huang, Z., Liang, J., and Chen, C. An improved multi-input deep convolutional neural network for automatic emotion recognition. *Frontiers in Neuroscience* (2022).

[49] Chen, T., Jin, X., Sun, Y., and Yin, W. VAFL: a method of vertical asynchronous federated learning. *CoRR abs/2007.06081* (2020).

[50] Chen, Y., Ning, Y., Slawski, M., and Rangwala, H. Asynchronous online federated learning for edge devices with non-iid data. In *2020 IEEE International Conference on Big Data (IEEE BigData*

2020), Atlanta, GA, USA, December 10-13, 2020 (2020), X. Wu, C. Jermaine, L. Xiong, X. Hu, O. Kotevska, S. Lu, W. Xu, S. Aluru, C. Zhai, E. Al-Masri, Z. Chen, and J. Saltz, Eds., IEEE, pp. 15–24.

[51] Chen, Y., Qin, X., Wang, J., Yu, C., and Gao, W. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intell. Syst. 35*, 4 (2020), 83–93.

[52] Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D., and Yang, Q. Secureboost: A lossless federated learning framework. *IEEE Intell. Syst. 36*, 6 (2021), 87–98.

[53] Cheng, Y., Liu, Y., Chen, T., and Yang, Q. Federated learning for privacy-preserving AI. *Commun. ACM 63*, 12 (2020), 33–36.

[54] Cruciani, F., Altmann, A., Lorenzi, M., Menegaz, G., and Galazzo, I. B. What PLS can still do for imaging genetics in alzheimer's disease. In *IEEE-EMBS International Conference on Biomedical and Health Informatics, BHI 2022, Ioannina, Greece, September 27-30, 2022* (2022), IEEE, pp. 1–4.

[55] Cruciani, F., Brusini, L., Zucchelli, M., Pinheiro, G. R., Setti, F., Galazzo, I. B., Deriche, R., Rittner, L., Calabrese, M., and Menegaz, G. Interpretable deep learning as a means for decrypting disease signature in multiple sclerosis. *Journal of Neural Engineering 18*, 4 (jul 2021), 0460a6.

[56] da Silva, F. R., Camacho, R., and Tavares, J. M. R. S. Federated learning in medical image analysis: A systematic survey. *Electronics 13*, 1 (2024).

[57] Damschroder, L. J., Pritts, J. L., Neblo, M. A., Kalarickal, R. J., Creswell, J. W., and Hayward, R. A. Patients, privacy and trust: patients' willingness to allow researchers to access their medical records. *Social science & medicine 64*, 1 (2007), 223–235.

[58] Dang, T. K., Lan, X., Weng, J., and Feng, M. Federated learning for electronic health records. *ACM Trans. Intell. Syst. Technol. 13*, 5 (2022), 72:1–72:17.

[59] D'Ascenzo, F., De Filippo, O., Gallone, G., Mittone, G., et al. Machine learning-based prediction of adverse events following an acute coronary syndrome (PRAISE): a modelling study of pooled datasets. *The Lancet 397*, 10270 (2021), 199–207.

[60] Dayan, I., Roth, H. R., Zhong, A., Harouni, A., Gentili, A., Abidin, A. Z., Liu, A., Costa, A. B., Wood, B. J., Tsai, C.-S., et al. Federated learning for predicting clinical outcomes in patients with covid-19. *Nature medicine 27*, 10 (2021), 1735–1743.

[61] Dimitriadis, D., Garcia, M. H., Diaz, D. M., Manoel, A., and Sim, R. FLUTE: A scalable, extensible framework for high-performance federated learning simulations. *CoRR abs/2203.13789* (2022).

[62] Dolci, G., Rahaman, M. A., Chen, J., Duan, K., Fu, Z., Abrol, A., Menegaz, G., and Calhoun, V. D. A deep generative multimodal imaging genomics framework for alzheimer's disease prediction. In *22nd IEEE International Conference on Bioinformatics and Bioengineering, BIBE 2022, Taichung, Taiwan, November 7-9, 2022* (2022), IEEE, pp. 41–44.

[63] Dwork, C. Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II* (2006), M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of *Lecture Notes in Computer Science*, Springer, pp. 1–12.

[64] Ek, S., Portet, F., Lalanda, P., and Vega, G. A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison. In *19th IEEE International Conference on Pervasive Computing and Communications, PerCom 2021, Kassel, Germany, March 22-26, 2021* (2021), IEEE, pp. 1–10.

[65] Ekmefjord, M., Ait-Mlouk, A., Alawadi, S., Åkesson, M., Singh, P., Spjuth, O., Toor, S., and Hellander, A. Scalable federated machine learning with fedn. In *22nd IEEE International Symposium on Cluster, Cloud and Internet Computing, CCGrid 2022, Taormina, Italy, May 16-19, 2022* (2022), IEEE, pp. 555–564.

[66] Escudero, D. An introduction to secret-sharing-based secure multiparty computation. *IACR Cryptol. ePrint Arch.* (2022), 62.

[67] Fan, Z., Su, J., Gao, K., Hu, D., and Zeng, L. A federated deep learning framework for 3d brain MRI images. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021* (2021), IEEE, pp. 1–6.

[68] Fauzi, M. A., Yang, B., and Blobel, B. Comparative analysis between individual, centralized, and federated learning for smartwatch based stress detection. *Journal of Personalized Medicine 12*, 10 (2022), 1584.

[69] Feki, I., Ammar, S., Kessentini, Y., and Muhammad, K. Federated learning for COVID-19 screening from chest x-ray images. *Appl. Soft Comput. 106* (2021), 107330.

[70] Feng, Y., Yu, M., Xiong, W., Guo, X., Huang, J., Chang, S., Campbell, M., Greenspan, M. A., and Zhu, X. extending pysyft with N.-IID federated learning benchmarkby houda bouraqqadi, ayoub berrag, mohamed mhaouach, afaf bouhoute, khalid fardousse, and ismail berrada: Learning to recover reasoning chains for multi-hop question answering via cooperative games. In *Proceedings of the 34th Canadian Conference on Artificial Intelligence, Canadian AI 2021, online, May 2021* (2021), L. Antonie and P. M. Zadeh, Eds., Canadian Artificial Intelligence Association.

[71] Florescu, L. M., Streba, C. T., Şerbănescu, M.-S., Mămuleanu, M., Florescu, D. N., Teică, R. V., Nica, R. E., and Gheonea, I. A. Federated learning approach with pre-trained deep learning models for covid-19 detection from unsegmented ct images. *Life 12*, 7 (2022), 958.

[72] Foley, P., Sheller, M. J., Edwards, B., Pati, S., Riviera, W., Sharma, M., Moorthy, P. N., Wang, S.-h., Martin, J., Mirhaji, P., et al. Openfl: the open federated learning library. *Physics in Medicine & Biology 67*, 21 (2022), 214001.

[73] French, R. M., and Chater, N. Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting. *Neural Comput. 14*, 7 (2002), 1755–1769.

[74] Fricke, S. Semantic scholar. *Journal of the Medical Library Association: JMLA 106*, 1 (2018), 145.

[75] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. Gan-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing 321* (2018), 321–331.

[76] Fu, X., Zhang, B., Dong, Y., Chen, C., and Li, J. Federated graph machine learning: A survey of concepts, techniques, and applications. *SIGKDD Explor. 24*, 2 (2022), 32–47.

[77] Gadekallu, T. R., Pham, Q., Huynh-The, T., Bhattacharya, S., Maddikunta, P. K. R., and Liyanage, M. Federated learning for big data: A survey on opportunities, applications, and future directions. *CoRR abs/2110.04160* (2021).

[78] Galtier, M. N., and Marini, C. Substra: a framework for privacy-preserving, traceable and collaborative machine learning. *CoRR abs/1910.11567* (2019).

[79] Gamal, A., Elattar, M. A., and Selim, S. Does deep learning require image registration for early prediction of alzheimer's disease? A comparative study using ADNI database. In *Advances in Model and Data Engineering in the Digitalization Era - MEDI 2022 Short Papers and DETECT 2022 Workshop Papers, Cairo, Egypt, November 21-24, 2022, Proceedings* (2022), P. Fournier-Viger, A. H. Yousef, L. Bellatreche, A. Awad, A. A. Wakrime, Y. Ouhammou, and I. Aït-Sadoune, Eds., vol. 1751 of *Communications in Computer and Information Science*, Springer, pp. 3–11.

[80] Gao, X., Shi, F., Shen, D., and Liu, M. Task-induced pyramid and attention GAN for multimodal brain image imputation and classification in alzheimer's disease. *IEEE J. Biomed. Health Informatics 26*, 1 (2022), 36–43.

[81] Gawehn, E., Hiss, J. A., and Schneider, G. Deep learning in drug discovery. *Molecular informatics 35*, 1 (2016), 3–14.

[82] Geng, J., Mou, Y., Li, F., Li, Q., Beyan, O., Decker, S., and Rong, C. Towards general deep leakage in federated learning. *CoRR abs/2110.09074* (2021).

[83] Ghose, P., Alavi, M., Tabassum, M., Ashraf Uddin, M., Biswas, M., Mahbub, K., Gaur, L., Mallik, S., and Zhao, Z. Detecting covid-19 infection status from chest x-ray and ct scan via single transfer learning-driven approach. *Frontiers in Genetics 13* (2022), 980338.

[84] Gosselin, R., Vieu, L., Loukil, F., and Benoit, A. Privacy and security in federated learning: A survey. *Applied Sciences 12*, 19 (2022), 9901.

[85] Gostin, L. O., Levit, L. A., and Nass, S. J. Beyond the hipaa privacy rule: enhancing privacy, improving health through research.

[86] Gratton, C., Venkategowda, N. K. D., Arablouei, R., and Werner, S. Distributed ridge regression with feature partitioning. In *52nd Asilomar Conference on Signals, Systems, and Computers, ACSSC 2018, Pacific Grove, CA, USA, October 28-31, 2018* (2018), M. B. Matthews, Ed., IEEE, pp. 1423–1427.

[87] Guendouzi, S. B., Ouchani, S., Assaad, H. E., and El-Zaher, M. A systematic review of federated learning: Challenges, aggregation methods, and development tools. *J. Netw. Comput. Appl. 220* (2023), 103714.

[88] Gunesli, G. N., Bilal, M., Raza, S., and Rajpoot, N. M. A federated learning approach to tumor detection in colon histology images. *J. Medical Syst. 47*, 1 (2023), 99.

[89] Guo, P., Wang, P., Zhou, J., Jiang, S., and Patel, V. M. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021* (2021), Computer Vision Foundation / IEEE, pp. 2423–2432.

[90] Hao, M., Li, H., Luo, X., Xu, G., Yang, H., and Liu, S. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Trans. Ind. Informatics 16*, 10 (2020), 6532–6542.

[91] Hao, M., Li, H., Xu, G., Liu, S., and Yang, H. Towards efficient and privacy-preserving federated deep learning. In *2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, May 20-24, 2019* (2019), IEEE, pp. 1–6.

[92] Harman, L. B., Flite, C. A., and Bond, K. Electronic health records: privacy, confidentiality, and security. *AMA journal of ethics 14*, 9 (2012), 712–719.

[93] Hatamizadeh, A., Yin, H., Molchanov, P., Myronenko, A., Li, W., Dogra, P., Feng, A., Flores, M. G., Kautz, J., Xu, D., and Roth, H. R. Do gradient inversion attacks make federated learning unsafe? *IEEE Trans. Medical Imaging 42*, 7 (2023), 2044–2056.

[94] He, C., Balasubramanian, K., Ceyani, E., Rong, Y., Zhao, P., Huang, J., Annavaram, M., and Avestimehr, S. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *CoRR abs/2104.07145* (2021).

[95] He, C., Li, S., So, J., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M., and Avestimehr, S. Fedml: A research library and benchmark for federated machine learning. *CoRR abs/2007.13518* (2020).

[96] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (2016), IEEE Computer Society, pp. 770–778.

[97] Hjerppe, K., Ruohonen, J., and Leppänen, V. The general data protection regulation: Requirements, architectures, and constraints. In *27th IEEE International Requirements Engineering Conference, RE 2019, Jeju Island, Korea (South), September 23-27, 2019* (2019), D. E. Damian, A. Perini, and S. Lee, Eds., IEEE, pp. 265–275.

[98] Hu, H., Zhang, X., Salcic, Z., Sun, L., Choo, K. R., and Dobbie, G. Source inference attacks: Beyond membership inference attacks in federated learning. *IEEE Trans. Dependable Secur. Comput. 21*, 4 (2024), 3012–3029.

[99] Hu, Y., Fang, L., and He, G. Privacy-preserving SVM classification on vertically partitioned data without secure multi-party computation. In *Fifth International Conference on Natural Computation, ICNC 2009, Tianjian, China, 14-16 August 2009, 6 Volumes* (2009), H. Wang, K. S. Low, K. Wei, and J. Sun, Eds., IEEE Computer Society, pp. 543–546.

[100] Huang, S., Shi, W., Xu, Z., Tsang, I. W., and Lv, J. Efficient federated multi-view learning. *Pattern Recognit. 131* (2022), 108817.

[101] Huang, Y., Gupta, S., Song, Z., Li, K., and Arora, S. Evaluating gradient inversion attacks and defenses in federated learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual* (2021), M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., pp. 7232–7241.

[102] Huang, Z., Hu, Y., Liu, R., Xue, X., Zhu, Z., Song, L., and Tan, K. C. Federated multi-task learning for joint diagnosis of multiple mental disorders on MRI scans. *IEEE Trans. Biomed. Eng. 70*, 4 (2023), 1137–1149.

[103] Hunt, T., Song, C., Shokri, R., Shmatikov, V., and Witchel, E. Chiron: Privacy-preserving machine learning as a service. *CoRR abs/1803.05961* (2018).

[104] Imteaj, A., Thakker, U., Wang, S., Li, J., and Amini, M. H. A survey on federated learning for resource-constrained iot devices. *IEEE Internet Things J. 9*, 1 (2022), 1–24.

[105] Jatain, D., Singh, V., and Dahiya, N. A contemplative perspective on federated machine learning: Taxonomy, threats & vulnerability assessment and challenges. *J. King Saud Univ. Comput. Inf. Sci. 34*, 9 (2022), 6681–6698.

[106] Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., and Smith, S. M. Fsl. *Neuroimage 62*, 2 (2012), 782–790.

[107] Jiang, D., Song, Y., Tong, Y., Wu, X., Zhao, W., Xu, Q., and Yang, Q. Federated topic modeling. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds., ACM, pp. 1071–1080.

[108] Jiang, Z., Wang, W., Li, B., and Li, B. Pisces: efficient federated learning via guided asynchronous training. In *Proceedings of the 13th Symposium on Cloud Computing, SoCC 2022, San Francisco, California, November 7-11, 2022* (2022), A. Gavrilovska, D. Altinbüken, and C. Binnig, Eds., ACM, pp. 370–385.

[109] Jin, X., Chen, P., Hsu, C., Yu, C., and Chen, T. CAFE: catastrophic data leakage in vertical federated learning. *CoRR abs/2110.15122* (2021).

[110] Joshi, M., Pal, A., and Sankarasubbu, M. Federated learning for healthcare domain - pipeline, applications and challenges. *ACM Trans. Comput. Heal. 3*, 4 (2022), 40:1–40:36.

[111] Ju, C., Gao, D., Mane, R., Tan, B., Liu, Y., and Guan, C. Federated transfer learning for EEG signal classification. In *42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC 2020, Montreal, QC, Canada, July 20-24, 2020* (2020), IEEE, pp. 3040–3045.

[112] Kaissis, G., Ziller, A., Passerat-Palmbach, J., Ryffel, T., Usynin, D., Trask, A., Lima, I., Mancuso, J., Jungmann, F., Steinborn, M., Saleh, A., Makowski, M. R., Rueckert, D., and Braren, R. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nat. Mach. Intell. 3*, 6 (2021), 473–484.

[113] Kaissis, G., Ziller, A., Passerat-Palmbach, J., Ryffel, T., Usynin, D., Trask, A., Lima, I., Mancuso, J., Jungmann, F., Steinborn, M., Saleh, A., Makowski, M. R., Rueckert, D., and Braren, R. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nat. Mach. Intell. 3*, 6 (2021), 473–484.

[114] Kang, H., Xia, L., Yan, F., Wan, Z., Shi, F., Yuan, H., Jiang, H., Wu, D., Sui, H., Zhang, C., and Shen, D. Diagnosis of coronavirus disease 2019 (COVID-19) with structured latent multi-view representation learning. *IEEE Trans. Medical Imaging 39*, 8 (2020), 2606–2614.

[115] Kang, Y., Liu, Y., and Liang, X. Fedcvt: Semi-supervised vertical federated learning with cross-view training. *ACM Trans. Intell. Syst. Technol. 13*, 4 (2022), 64:1–64:16.

[116] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. SCAFFOLD: stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event* (2020), vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 5132–5143.

[117] Kholod, I., Yanaki, E., Fomichev, D., Shalugin, E., Novikova, E., Filippov, E., and Nordlund, M. Open-source federated learning frameworks for iot: A comparative review and analysis. *Sensors 21*, 1 (2021), 167.

[118] Kim, H., Park, J., Bennis, M., and Kim, S. Blockchained on-device federated learning. *IEEE Commun. Lett. 24*, 6 (2020), 1279–1283.

[119] Kim, J., Park, Y., Kim, G., and Hwang, S. J. Splitnet: Learning to semantically split deep networks for parameter reduction and model parallelization. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017* (2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 1866–1874.

[120] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. Jupyter notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (2016), F. Loizides and B. Schmidt, Eds., IOS Press, pp. 87 – 90.

[121] Konečnỳ, J., McMahan, B., and Ramage, D. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575* (2015).

[122] Lai, C., Zhou, S., and Trayanova, N. A. Optimal ecg-lead selection increases generalizability of deep learning on ecg abnormality classification. *Philosophical Transactions of the Royal society A 379*, 2212 (2021), 20200258.

[123] Lai, X., Zhou, J., Wessely, A., Heppt, M., Maier, A., Berking, C., Vera, J., and Zhang, L. A disease network-based deep learning approach for characterizing melanoma. *International Journal of Cancer 150*, 6 (2022), 1029–1044.

[124] Li, Q., Diao, Y., Chen, Q., and He, B. Federated learning on non-iid data silos: An experimental study. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022* (2022), IEEE, pp. 965–978.

[125] Li, Q., He, B., and Song, D. Model-contrastive federated learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021* (2021), Computer Vision Foundation / IEEE, pp. 10713–10722.

[126] Li, Q., Thapa, C., Ong, L., Zheng, Y., and All. Vertical federated learning: Taxonomies, threats, and prospects. *CoRR abs/2302.01550* (2023).

[127] Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag. 37*, 3 (2020), 50–60.

[128] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020* (2020), I. S. Dhillon, D. S. Papailiopoulos, and V. Sze, Eds., mlsys.org.

[129] Li, W., Qi, F., Tang, M., and Yu, Z. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing 387* (2020), 63–77.

[130] Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[131] Li, X., Jiang, M., Zhang, X., Kamp, M., and Dou, Q. Fedbn: Federated learning on non-iid features via local batch normalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (2021), OpenReview.net.

[132] Li, X., Qu, Z., Tang, B., and Lu, Z. Fedlga: Toward system-heterogeneity of federated learning via local gradient approximation. *IEEE Trans. Cybern. 54*, 1 (2024), 401–414.

[133] Liang, X., Nguyen, D., and Jiang, S. B. Generalizability issues with deep learning models in medicine and their potential solutions: illustrated with cone-beam computed tomography (CBCT) to computed tomography (CT) image conversion. *Mach. Learn. Sci. Technol. 2*, 1 (2021), 15007.

[134] Liang, Y., Guo, Y., Gong, Y., Luo, C., Zhan, J., and Huang, Y. Flbench: A benchmark suite for federated learning. In *BenchCouncil International Federated Intelligent Computing and Block Chain Conferences* (2020), Springer, pp. 166–176.

[135] Linardos, A., Kushibar, K., Walsh, S., Gkontra, P., and Lekadir, K. Federated learning for multi-center imaging diagnostics: A study in cardiovascular disease. *CoRR abs/2107.03901* (2021).

[136] Liu, B., Tan, C., Wang, J., Zeng, T., Shan, H., Yao, H., Huang, H., Dai, P., Bo, L., and Chen, Y. Fedlearn-algo: A flexible open-source privacy-preserving machine learning platform. *CoRR abs/2107.04129* (2021).

[137] Liu, J., Liang, X., Yang, R., Luo, Y., Lu, H., Li, L., Zhang, S., and Yang, S. Federated learning-based vertebral body segmentation. *Eng. Appl. Artif. Intell. 116* (2022), 105451.

[138] Liu, S., Liu, S., Cai, W., Che, H., Pujol, S., Kikinis, R., Feng, D., and Fulham, M. J. Multimodal neuroimaging feature learning for multiclass diagnosis of alzheimer's disease. *IEEE Trans. Biomed. Eng. 62*, 4 (2015), 1132–1140.

[139] Liu, X., Shi, T., Xie, C., Li, Q., Hu, K., Kim, H., Xu, X., Li, B., and Song, D. Unifed: A benchmark for federated learning frameworks. *CoRR abs/2207.10308* (2022).

[140] Liu, X., Song, L., Liu, S., and Zhang, Y. A review of deep-learning-based medical image segmentation methods. *Sustainability 13*, 3 (2021), 1224.

[141] Liu, Y., Kang, Y., Zou, T., Pu, Y., and All. Vertical federated learning. *CoRR abs/2211.12814* (2022).

[142] Liu, Y., Zhang, X., Kang, Y., Li, L., Chen, T., Hong, M., and Yang, Q. Fedbcd: A communication-efficient collaborative learning framework for distributed features. *IEEE Trans. Signal Process. 70* (2022), 4277–4290.

[143] Long, G., Tan, Y., Jiang, J., and Zhang, C. Federated learning for open banking. *CoRR abs/2108.10749* (2021).

[144] Lu, M. Y., Chen, R. J., Kong, D., Lipková, J., Singh, R., Williamson, D. F. K., Chen, T. Y., and Mahmood, F. Federated learning for computational pathology on gigapixel whole slide images. *Medical Image Anal. 76* (2022), 102298.

[145] Lu, M. Y., Chen, R. J., Kong, D., Lipková, J., Singh, R., Williamson, D. F. K., Chen, T. Y., and Mahmood, F. Federated learning for computational pathology on gigapixel whole slide images. *Medical Image Anal. 76* (2022), 102298.

[146] Ludwig, H., Baracaldo, N., Thomas, G., Zhou, Y., Anwar, A., Rajamoni, S., Ong, Y. J., Radhakrishnan, J., Verma, A., Sinn, M., Purcell, M., Rawat, A., Minh, T. N., Holohan, N., Chakraborty, S., Witherspoon, S., Steuer, D., Wynter, L., Hassan, H., Laguna, S., Yurochkin, M., Agarwal, M., Chuba, E., and Abay, A. IBM federated learning: an enterprise framework white paper V0.1. *CoRR abs/2007.10987* (2020).

[147] Lugan, S., Desbordes, P., Tormo, L. X. R., Legay, A., and Macq, B. Secure architectures implementing trusted coalitions for blockchained distributed learning (tclearn). *CoRR abs/1906.07690* (2019).

[148] Luo, B., Xiao, W., Wang, S., Huang, J., and Tassiulas, L. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In *IEEE INFOCOM 2022-IEEE conference on computer communications* (2022), IEEE, pp. 1739–1748.

[149] Lyu, L., Yu, J., Nandakumar, K., Li, Y., Ma, X., Jin, J., Yu, H., and Ng, K. S. Towards fair and privacy-preserving federated deep models. *IEEE Trans. Parallel Distributed Syst. 31*, 11 (2020), 2524–2541.

[150] Mandal, P. K. Vertical federated alzheimer's detection on multimodal data. *CoRR abs/2312.10237* (2023).

[151] Mansour, R. F., Alfar, N. M., Abdel-Khalek, S., Abdelhaq, M. S., Saeed, R. A., and Alsaqour, R. A. Optimal deep learning based fusion model for biomedical image classification. *Expert Syst. J. Knowl. Eng. 39*, 3 (2022).

[152] Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C., and Buckner, R. L. Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *J. Cogn. Neurosci. 19*, 9 (2007), 1498–1507.

[153] Mårtensson, G., Ferreira, D., Granberg, T., Cavallin, L., Oppedal, K., Padovani, A., Rektorová, I., Bonanni, L., Pardini, M., Kramberger, M. G., Taylor, J., Hort, J., Snædal, J., Kulisevsky, J., Blanc, F., Antonini, A., Mecocci, P., Vellas, B., and Westman, E. The reliability of a deep learning model in clinical out-of-distribution MRI data: A multicohort study. *Medical Image Anal. 66* (2020), 101714.

[154] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA* (2017), A. Singh and X. J. Zhu, Eds., vol. 54 of *Proceedings of Machine Learning Research*, PMLR, pp. 1273–1282.

[155] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA* (2017), A. Singh and X. J. Zhu, Eds., vol. 54 of *Proceedings of Machine Learning Research*, PMLR, pp. 1273–1282.

[156] McMahan, H. B., Moore, E., Ramage, D., and y Arcas, B. A. Federated learning of deep networks using model averaging. *CoRR abs/1602.05629* (2016).

[157] Mendieta, M., Yang, T., Wang, P., Lee, M., Ding, Z., and Chen, C. Local learning matters: Rethinking data heterogeneity in federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022* (2022), IEEE, pp. 8387–8396.

[158] Mengistu, T. M., Kim, T., and Lin, J. A survey on heterogeneity taxonomy, security and privacy preservation in the integration of iot, wireless sensor networks and federated learning. *Sensors 24*, 3 (2024), 968.

[159] Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Medical Imaging 34*, 10 (2015), 1993–2024.

[160] Messner, E., Fediuk, M., Swatek, P., Scheidl, S., Smolle-Jüttner, F., Olschewski, H., and Pernkopf, F. Multi-channel lung sound classification with convolutional recurrent neural networks. *Comput. Biol. Medicine 122* (2020), 103831.

[161] Mittone, G., Riviera, W., Colonnelli, I., Birke, R., and Aldinucci, M. Model-agnostic federated learning. *CoRR abs/2303.04906* (2023).

[162] Moncada-Torres, A., Martin, F., Sieswerda, M., van Soest, J., and Geleijnse, G. VANTAGE6: an open source privacy preserving federated learning infrastructure for secure insight exchange. In *AMIA 2020, American Medical Informatics Association Annual Symposium, Virtual Event, USA, November 14-18, 2020* (2020), AMIA.

[163] Morsbach, F., and Toor, S. Decfl: An ubiquitous decentralized model training protocol and framework empowered by blockchain. In *BSCI '21: Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure, Virtual Event, Hong Kong, June 7, 2021* (2021), K. Gai and K. R. Choo, Eds., ACM, pp. 61–70.

[164] Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., and Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst. 115* (2021), 619–640.

[165] Naehrig, M., Lauter, K. E., and Vaikuntanathan, V. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM Cloud Computing Security Workshop, CCSW 2011, Chicago, IL, USA, October 21, 2011* (2011), C. Cachin and T. Ristenpart, Eds., ACM, pp. 113–124.

[166] Narayanan, A., Bonneau, J., Felten, E. W., Miller, A., and Goldfeder, S. *Bitcoin and Cryptocurrency Technologies - A Comprehensive Introduction.* Princeton University Press, 2016.

[167] Nasirigerdeh, R., Torkzadehmahani, R., Matschinske, J. O., Baumbach, J., Rueckert, D., and Kaissis, G. Hyfed: A hybrid federated framework for privacy-preserving machine learning. *CoRR abs/2105.10545* (2021).

[168] Nasr, M., Shokri, R., and Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference

attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019* (2019), IEEE, pp. 739–753.

[169] Nguyen, D., Kay, F., Tan, J., Yan, Y., Ng, Y. S., Iyengar, P., Peshock, R., and Jiang, S. B. Deep learning-based COVID-19 pneumonia classification using chest CT images: Model generalizability. *Frontiers Artif. Intell. 4* (2021), 694875.

[170] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., and Poor, H. V. Federated learning for internet of things: A comprehensive survey. *IEEE Commun. Surv. Tutorials 23*, 3 (2021), 1622–1658.

[171] Nguyen, D. C., Pham, Q.-V., Pathirana, P. N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., and Hwang, W.-J. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (Csur) 55*, 3 (2022), 1–37.

[172] Niedziela, P., Danilenka, A., Kolasa, D., Ganzha, M., Paprzycki, M., and Nalinaksh, K. Sunday-fl–developing open source platform for federated learning. In *2021 Emerging Trends in Industry 4.0 (ETI 4.0)* (2021), IEEE, pp. 1–6.

[173] Nuding, F., and Mayer, R. Data poisoning in sequential and parallel federated learning. In *IWSPA@CODASPY 2022: Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics, Baltimore, MD, USA, April 27, 2022* (2022), A. H. Sung, R. M. Verma, and R. H. C. Yap, Eds., ACM, pp. 24–34.

[174] O'Shea, K., and Nash, R. An introduction to convolutional neural networks. *CoRR abs/1511.08458* (2015).

[175] Ozkaya, U., Öztürk, S., and Barstugan, M. Coronavirus (COVID-19) classification using deep features fusion and ranking technique. *CoRR abs/2004.03698* (2020).

[176] Palihawadana, C., Wiratunga, N., Wijekoon, A., and Kalutarage, H. K. Fedsim: Similarity guided model aggregation for federated learning. *Neurocomputing 483* (2022), 432–445.

[177] Pan, L., Ji, B., Wang, H., Wang, L., Liu, M., Chongcheawchamnan, M., and Peng, S. MFDNN: multi-channel

feature deep neural network algorithm to identify COVID19 chest x-ray images. *Health Inf. Sci. Syst. 10*, 1 (2022), 4.

[178] Pappas, C., Chatzopoulos, D., Lalis, S., and Vavalis, M. IPLS: A framework for decentralized federated learning. In *IFIP Networking Conference, IFIP Networking 2021, Espoo and Helsinki, Finland, June 21-24, 2021* (2021), Z. Yan, G. Tyson, and D. Koutsonikolas, Eds., IEEE, pp. 1–6.

[179] Pati, S., Baid, U., Edwards, B., Sheller, M. J., Wang, H. S., Reina, G. A., Foley, P., Gruzdev, A., Karkada, D., Davatzikos, C., Sako, C., Ghodasara, S., Bilello, M., Mohan, S., Vollmuth, P., Brugnara, G., Preetha, C. J., Sahm, F., Maier-Hein, K. H., Zenk, M., Bendszus, M., Wick, W., Calabrese, E., Rudie, J. D., Villanueva-Meyer, J. E., Cha, S., Ingalhalikar, M., Jadhav, M., Pandey, U., Saini, J., Garrett, J., Larson, M., Jeraj, R., Currie, S., Frood, R., Fatania, K., Huang, R. Y., Chang, K., Quintero, C. B., Capellades, J., Puig, J., Trenkler, J., Pichler, J., Necker, G., Haunschmidt, A., Meckel, S., Shukla, G., Liem, S., Alexander, G. S., and et al. Federated learning enables big data for rare cancer boundary detection. *CoRR abs/2204.10836* (2022).

[180] Pfohl, S. R., Dai, A. M., and Heller, K. A. Federated and differentially private learning for electronic health records. *CoRR abs/1911.05861* (2019).

[181] Poria, S., Cambria, E., Bajpai, R., and Hussain, A. A review of affective computing: From unimodal analysis to multimodal fusion. *Inf. Fusion 37* (2017), 98–125.

[182] Qayyum, A., Ahmad, K., Ahsan, M. A., Al-Fuqaha, A., and Qadir, J. Collaborative federated learning for healthcare: Multimodal covid-19 diagnosis at the edge. *IEEE Open Journal of the Computer Society 3* (2022), 172–184.

[183] Qayyum, A., Ahmad, K., Ahsan, M. A., Al-Fuqaha, A. I., and Qadir, J. Collaborative federated learning for healthcare: Multimodal COVID-19 diagnosis at the edge. *IEEE Open J. Comput. Soc. 3* (2022), 172–184.

[184] Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G., and Piccialli, F. Model aggregation techniques in federated learning: A

comprehensive survey. *Future Gener. Comput. Syst. 150* (2024), 272–293.

[185] R., S. S. S., Altmann, A., Gutman, B., and Lorenzi, M. Fedbiomed: A general open-source frontend framework for federated learning in healthcare. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning - Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4-8, 2020, Proceedings* (2020), S. Albarqouni, S. Bakas, K. Kamnitsas, M. J. Cardoso, B. A. Landman, W. Li, F. Milletari, N. Rieke, H. Roth, D. Xu, and Z. Xu, Eds., vol. 12444 of *Lecture Notes in Computer Science*, Springer, pp. 201–210.

[186] Rahman, S. A., Tout, H., Ould-Slimane, H., Mourad, A., Talhi, C., and Guizani, M. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet Things J. 8*, 7 (2021), 5476–5497.

[187] Reina, G. A., Gruzdev, A., Foley, P., Perepelkina, O., Sharma, M., Davidyuk, I., Trushkin, I., Radionov, M., Mokrov, A., Agapov, D., Martin, J., Edwards, B., Sheller, M. J., Pati, S., Moorthy, P. N., Wang, H. S., Shah, P., and Bakas, S. Openfl: An open-source framework for federated learning. *CoRR abs/2105.06413* (2021).

[188] Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K. H., Ourselin, S., Sheller, M. J., Summers, R. M., Trask, A., Xu, D., Baust, M., and Cardoso, M. J. The future of digital health with federated learning. *npj Digit. Medicine 3* (2020).

[189] Riviera, W., Galazzo, I. B., and Menegaz, G. Felebrities: A user-centric assessment of federated learning frameworks. *IEEE Access 11* (2023), 96865–96878.

[190] Riviera, W., Galazzo, I. B., and Menegaz, G. Felebrities: A user-centric assessment of federated learning frameworks. *IEEE Access 11* (2023), 96865–96878.

[191] Ro, J. H., Suresh, A. T., and Wu, K. FedJAX: Federated learning simulation with JAX. *arXiv preprint arXiv:2108.02117* (2021).

[192] Romanini, D., Hall, A. J., Papadopoulos, P., Titcombe, T., Ismail, A., Cebere, T., Sandmann, R., Roehm, R., and Hoeh, M. A. Pyvertical: A vertical federated learning framework for multi-headed splitnn. *CoRR abs/2104.00489* (2021).

[193] Rooijakkers, T. Convinced—enabling privacy-preserving survival analyses using multi-party computation.

[194] Rosenbloom, S. T., Smith, J. R. L., Bowen, R., Burns, J., and other. Updating HIPAA for the electronic medical record era. *J. Am. Medical Informatics Assoc. 26*, 10 (2019), 1115–1119.

[195] Roth, H. R., Chang, K., Singh, P., Neumark, N., Li, W., Gupta, V., Gupta, S., Qu, L., Ihsani, A., Bizzo, B. C., Wen, Y., Buch, V., Shah, M., Kitamura, F., Mendonça, M., Lavor, V., Harouni, A., Compas, C., Tetreault, J., Dogra, P., Cheng, Y., Erdal, S., White, R. D., Hashemian, B., Schultz, T. J., Zhang, M., McCarthy, A., Yun, B. M., Sharaf, E., Hoebel, K. V., Patel, J. B., Chen, B., Ko, S., Leibovitz, E., Pisano, E. D., Coombs, L., Xu, D., Dreyer, K. J., Dayan, I., Naidu, R. C., Flores, M., Rubin, D. L., and Kalpathy-Cramer, J. Federated learning for breast density classification: A real-world implementation. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning - Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4-8, 2020, Proceedings* (2020), S. Albarqouni, S. Bakas, K. Kamnitsas, M. J. Cardoso, B. A. Landman, W. Li, F. Milletari, N. Rieke, H. Roth, D. Xu, and Z. Xu, Eds., vol. 12444 of *Lecture Notes in Computer Science*, Springer, pp. 181–191.

[196] Roth, H. R., Chang, K., Singh, P., Neumark, N., Li, W., Gupta, V., Gupta, S., Qu, L., Ihsani, A., Bizzo, B. C., Wen, Y., Buch, V., Shah, M., Kitamura, F., Mendonça, M., Lavor, V., Harouni, A., Compas, C., Tetreault, J., Dogra, P., Cheng, Y., Erdal, S., White, R. D., Hashemian, B., Schultz, T. J., Zhang, M., McCarthy, A., Yun, B. M., Sharaf, E., Hoebel, K. V., Patel, J. B., Chen, B., Ko, S., Leibovitz, E., Pisano, E. D., Coombs, L., Xu, D., Dreyer, K. J., Dayan, I., Naidu, R. C., Flores, M., Rubin, D. L., and Kalpathy-Cramer, J. Federated learning for breast density classification: A

real-world implementation. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning - Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4-8, 2020, Proceedings* (2020), S. Albarqouni, S. Bakas, K. Kamnitsas, M. J. Cardoso, B. A. Landman, W. Li, F. Milletari, N. Rieke, H. Roth, D. Xu, and Z. Xu, Eds., vol. 12444 of *Lecture Notes in Computer Science*, Springer, pp. 181–191.

[197] Roth, H. R., Cheng, Y., Wen, Y., Yang, I., Xu, Z., Hsieh, Y., Kersten, K., Harouni, A., Zhao, C., Lu, K., Zhang, Z., Li, W., Myronenko, A., Yang, D., Yang, S., Rieke, N., Quraini, A., Chen, C., Xu, D., Ma, N., Dogra, P., Flores, M., and Feng, A. NVIDIA FLARE: federated learning from simulation to real-world. *CoRR abs/2210.13291* (2022).

[198] Ryu, M., Kim, Y., Kim, K., and Madduri, R. K. APPFL: open-source software framework for privacy-preserving federated learning. In *IEEE International Parallel and Distributed Processing Symposium, IPDPS Workshops 2022, Lyon, France, May 30 - June 3, 2022* (2022), IEEE, pp. 1074–1083.

[199] Sadat, S. U., Shomee, H. H., Awwal, A., Amin, S. N., Reza, M. T., and Parvez, M. Z. Alzheimer's disease detection and classification using transfer learning technique and ensemble on convolutional neural networks. In *2021 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2021, Melbourne, Australia, October 17-20, 2021* (2021), IEEE, pp. 1478–1481.

[200] Sakib, S., Tazrin, T., Fouda, M. M., Fadlullah, Z. M., and Guizani, M. Dl-crc: deep learning-based chest radiograph classification for covid-19 detection: a novel approach. *Ieee Access 8* (2020), 171575–171589.

[201] Salameh, j.-p., Bossuyt, P., McGrath, T., Thombs, B., Hyde, C., Macaskill, P., Deeks, J., Leeflang, M., Korevaar, D. A., Whiting, P., Takwoingi, Y., Reitsma, J., Cohen, J., Frank, R., Hunt, H., Hooft, L., Rutjes, A., Willis, B., Gatsonis, C., and McInnes, M. Preferred reporting items for systematic review and meta-analysis of diagnostic test accuracy studies (prisma-dta): Explanation, elaboration, and checklist. *BMJ 370* (08 2020), m2632.

[202] Sanabria, R., Caglayan, O., Palaskar, S., Elliott, D., Barrault, L., Specia, L., and Metze, F. How2: A large-scale dataset for multimodal language understanding. *CoRR abs/1811.00347* (2018).

[203] Sarma, K. V., Harmon, S. A., Sanford, T., Roth, H. R., Xu, Z., Tetreault, J., Xu, D., Flores, M. G., Raman, A. G., Kulkarni, R., Wood, B. J., Choyke, P. L., Priester, A., Marks, L. S., Raman, S. S., Enzmann, D., Turkbey, B., Speier, W., and Arnold, C. W. Federated learning improves site performance in multicenter deep learning without data sharing. *J. Am. Medical Informatics Assoc. 28*, 6 (2021), 1259–1264.

[204] Sattler, F., Wiedemann, S., Müller, K., and Samek, W. Robust and communication-efficient federated learning from non-iid data. *CoRR abs/1903.02891* (2019).

[205] Savazzi, S., Nicoli, M., and Rampa, V. Federated learning with cooperating devices: A consensus approach for massive iot networks. *IEEE Internet Things J. 7*, 5 (2020), 4641–4654.

[206] Schmidt, A. M., Desai, A. D., Watkins, L. E., Crowder, H. A., Black, M. S., Mazzoli, V., Rubin, E. B., Lu, Q., MacKay, J. W., Boutin, R. D., et al. Generalizability of deep learning segmentation algorithms for automated assessment of cartilage morphology and mri relaxometry. *Journal of Magnetic Resonance Imaging* (2022).

[207] Schneebeli, C., Kalloori, S., and Klingler, S. A practical federated learning framework for small number of stakeholders. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021* (2021), L. Lewin-Eytan, D. Carmel, E. Yom-Tov, E. Agichtein, and E. Gabrilovich, Eds., ACM, pp. 910–913.

[208] Sedik, A., Iliyasu, A. M., Abd El-Rahiem, B., Abdel Samea, M. E., Abdel-Raheem, A., Hammad, M., Peng, J., Abd El-Samie, F. E., and Abd El-Latif, A. A. Deploying machine and deep learning models for efficient data-augmented detection of covid-19 infections. *Viruses 12*, 7 (2020), 769.

[209] Sello, P., Bagula, A. B., and Ajayi, O. Laws and regulations on big data management: The case of south africa. In *e-Infrastructure and e-Services for Developing Countries - 11th EAI International*

Conference, AFRICOMM 2019, Porto-Novo, Benin, December 3-4, 2019, Proceedings (2019), R. Zitouni, M. Agueh, P. Houngue, and H. Soude, Eds., vol. 311 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer, pp. 169–179.

[210] Shahid, O., Pouriyeh, S., Parizi, R. M., Sheng, Q. Z., Srivastava, G., and Zhao, L. Communication efficiency in federated learning: Achievements and challenges. *CoRR abs/2107.10996* (2021).

[211] Shamir, O., Srebro, N., and Zhang, T. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014* (2014), vol. 32 of *JMLR Workshop and Conference Proceedings*, JMLR.org, pp. 1000–1008.

[212] Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R. R., et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports 10*, 1 (2020), 1–12.

[213] Sheller, M. J., Reina, G. A., Edwards, B., Martin, J., and Bakas, S. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries - 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I* (2018), A. Crimi, S. Bakas, H. J. Kuijf, F. Keyvan, M. Reyes, and T. van Walsum, Eds., vol. 11383 of *Lecture Notes in Computer Science*, Springer, pp. 92–104.

[214] Shoeibi, A., Khodatars, M., Jafari, M., Ghassemi, N., Sadeghi, D., Moridian, P., Khadem, A., Alizadehsani, R., Hussain, S., Zare, A., et al. Automated detection and forecasting of covid-19 using deep learning techniques: A review. *Neurocomputing* (2024), 127317.

[215] Shokri, R., and Shmatikov, V. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015* (2015), I. Ray, N. Li, and C. Kruegel, Eds., ACM, pp. 1310–1321.

[216] Silva R., S. S., Gutman, B. A., Romero, E., Thompson, P. M., Altmann, A., and Lorenzi, M. Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In *16th IEEE International Symposium on Biomedical Imaging, ISBI 2019, Venice, Italy, April 8-11, 2019* (2019), IEEE, pp. 270–274.

[217] Singh, P., Singh, M. K., Singh, R., and Singh, N. *Federated Learning: Challenges, Methods, and Future Directions*. Springer International Publishing, Cham, 2022, pp. 199–214.

[218] Sleeman, W. C., Kapoor, R., and Ghosh, P. Multimodal classification: Current landscape, taxonomy and future directions. *ACM Comput. Surv. 55*, 7 (2023), 150:1–150:31.

[219] So, J., Hsieh, K., Arzani, B., Noghabi, S. A., Avestimehr, S., and Chandra, R. Fedspace: An efficient federated learning framework at satellites and ground stations. *CoRR abs/2202.01267* (2022).

[220] Soda, P., D'Amico, N. C., Tessadori, J., Valbusa, G., Guarrasi, V., Bortolotto, C., Akbar, M. U., Sicilia, R., Cordelli, E., Fazzini, D., Cellina, M., Oliva, G., Callea, G., Panella, S., Cariati, M., Cozzi, D., Miele, V., Stellato, E., Carrafiello, G. P., Castorani, G., Simeone, A., Preda, L., Iannello, G., Del Bue, A., Tedoldi, F., Alì, M., Sona, D., and Papa, S. Aiforcovid: predicting the clinical outcomes in patients with covid-19 applying ai to chest-x-rays. an italian multicentre study. *Medical Image Analysis* (2021).

[221] Sohan, M. F., and Basalamah, A. A systematic review on federated learning in medical image analysis. *IEEE Access 11* (2023), 28628–28644.

[222] Song, Y., Ren, S., Lu, Y., Fu, X., and Wong, K. K. L. Deep learning-based automatic segmentation of images in cardiac radiography: A promising challenge. *Comput. Methods Programs Biomed. 220* (2022), 106821.

[223] Song, Y., Xie, Y., Zhang, H., Liang, Y., Ye, X., Yang, A., and Ouyang, Y. Federated learning application on telecommunication-joint healthcare recommendation. In *21st International Conference on Communication Technology, ICCT 2021, Tianjin, China, October 13-16, 2021* (2021), IEEE, pp. 1443–1448.

[224] Spairani, E., Daniele, B., Signorini, M. G., and Magenes, G. A deep learning mixed-data type approach for the classification of fhr signals. *Frontiers in Bioengineering and Biotechnology 10* (2022).

[225] Sportisse, A., Schmutz, H., Humbert, O., Bouveyron, C., and Mattei, P.-A. Are labels informative in semi-supervised learning? – estimating and leveraging the missing-data mechanism.

[226] Stripelis, D., Ambite, J. L., Lam, P., and Thompson, P. M. Scaling neuroscience research using federated learning. In *18th IEEE International Symposium on Biomedical Imaging, ISBI 2021, Nice, France, April 13-16, 2021* (2021), IEEE, pp. 1191–1195.

[227] Taylor, P. J., Dargahi, T., Dehghantanha, A., Parizi, R. M., and Choo, K. R. A systematic literature review of blockchain cyber security. *Digit. Commun. Networks 6*, 2 (2020), 147–156.

[228] Tong, T., Gray, K. R., Gao, Q., Chen, L., and Rueckert, D. Multi-modal classification of alzheimer's disease using nonlinear graph fusion. *Pattern Recognit. 63* (2017), 171–181.

[229] Tran, T., Le, T., Pham, D., Hoang, V., Khong, V., Tran, Q., Nguyen, T., and Pham, C. A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality. In *24th International Conference on Pattern Recognition, ICPR 2018, Beijing, China, August 20-24, 2018* (2018), IEEE Computer Society, pp. 1947–1952.

[230] Tripathi, S., and Beigi, H. S. M. Multi-modal emotion recognition on IEMOCAP dataset using deep learning. *CoRR abs/1804.05788* (2018).

[231] Venkateswaran, P., Isahagian, V., Muthusamy, V., and Venkatasubramanian, N. Fedgen: Generalizable federated learning. *CoRR abs/2211.01914* (2022).

[232] Wan, S., Lu, J., Fan, P., Shao, Y., Peng, C., Letaief, K. B., and Chuai, J. How global observation works in federated learning: Integrating vertical training into horizontal federated learning. *IEEE Internet Things J. 10*, 11 (2023), 9482–9497.

[233] Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D. S., and Khazaeni, Y. Federated learning with matched averaging. In

8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020 (2020), OpenReview.net.

[234] Wang, H., Zhou, Y., Zhang, C., Peng, C., Huang, M., Liu, Y., and Zhang, L. Xfl: A high performace, lightweighted federated learning framework. *arXiv preprint arXiv:2302.05076* (2023).

[235] Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds.

[236] Wang, Y., Gu, X., Hou, W., Zhao, M., Sun, L., and Guo, C. Dual semi-supervised learning for classification of alzheimer's disease and mild cognitive impairment based on neuropsychological data. *Brain Sciences 13*, 2 (2023).

[237] Wei, K., Li, J., Ma, C., Ding, M., Wei, S., Wu, F., Chen, G., and Ranbaduge, T. Vertical federated learning: Challenges, methodologies and experiments. *CoRR abs/2202.04309* (2022).

[238] Wei, W., Liu, L., Wu, Y., Su, G., and Iyengar, A. Gradient-leakage resilient federated learning. In *41st IEEE International Conference on Distributed Computing Systems, ICDCS 2021, Washington DC, USA, July 7-10, 2021* (2021), IEEE, pp. 797–807.

[239] Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Green, R. C., Harvey, D., Jack Jr, C. R., Jagust, W., Morris, J. C., et al. The alzheimer's disease neuroimaging initiative 3: Continued innovation for clinical trial improvement. *Alzheimer's & Dementia 13*, 5 (2017), 561–571.

[240] Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proc. IEEE 78*, 10 (1990), 1550–1560.

[241] Weston, A. D., Korfiatis, P., Kline, T. L., Philbrick, K. A., Kostandy, P., Sakinis, T., Sugimoto, M., Takahashi, N., and Erickson, B. J. Automated abdominal segmentation of ct scans for body composition analysis using deep learning. *Radiology 290*, 3 (2019), 669–679.

[242] Wu, Q., Chen, X., Zhou, Z., and Zhang, J. Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. *IEEE Transactions on Mobile Computing 21*, 8 (2020), 2818–2832.

[243] Wu, Q., He, K., and Chen, X. Personalized federated learning for intelligent iot applications: A cloud-edge based framework. *IEEE Open J. Comput. Soc. 1* (2020), 35–44.

[244] Xia, G., Chen, J., Yu, C., and Ma, J. Poisoning attacks in federated learning: A survey. *IEEE Access 11* (2023), 10708–10722.

[245] Xu, J., and Wang, H. Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective. *IEEE Transactions on Wireless Communications 20*, 2 (2020), 1188–1200.

[246] Yan, X., Hu, S., Mao, Y., Ye, Y., and Yu, H. Deep multi-view learning methods: A review. *Neurocomputing 448* (2021), 106–129.

[247] Yang, D., Xu, Z., Li, W., Myronenko, A., Roth, H. R., Harmon, S. A., Xu, S., Turkbey, B., Turkbey, E., Wang, X., Zhu, W., Carrafiello, G., Patella, F., Cariati, M., Obinata, H., Mori, H., Tamura, K., An, P., Wood, B. J., and Xu, D. Federated semi-supervised learning for COVID region segmentation in chest CT using multi-national data from china, italy, japan. *Medical Image Anal. 70* (2021), 101992.

[248] Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. 10*, 2 (2019), 12:1–12:19.

[249] Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., and Yu, H. *Federated Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2019.

[250] Yang, Q., Matsutani, H., and Kondo, M. A selective model aggregation approach in federated learning for online anomaly detection.

[251] Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D., and Beaufays, F. Applied federated learning: Improving google keyboard query suggestions, 2018.

[252] Yang, Y., Ye, X., and Sakurai, T. Multi-view federated learning with data collaboration. In *ICMLC 2022: 14th International Conference on Machine Learning and Computing, Guangzhou, China, February 18 - 21, 2022* (2022), ACM, pp. 178–183.

[253] Ye, M., Fang, X., Du, B., Yuen, P. C., and Tao, D. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Comput. Surv. 56*, 3 (2024), 79:1–79:44.

[254] Ye, R., Ni, Z., Xu, C., Wang, J., Chen, S., and Eldar, Y. C. Fedfm: Anchor-based feature matching for data heterogeneity in federated learning. *IEEE Trans. Signal Process. 71* (2023), 4224–4239.

[255] Yu, Y., Zeng, M., Qiu, Z., Luo, L., and Chen, H. A data protection-oriented design procedure for a federated learning framework. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, October 21-23, 2020* (2020), IEEE, pp. 968–974.

[256] Yuan, B., Ge, S., and Xing, W. A federated learning framework for healthcare iot devices. *arXiv preprint arXiv:2005.05083* (2020).

[257] Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K. H., Hoang, T. N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (2019), K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 7252–7261.

[258] Zeng, D., Liang, S., Hu, X., and Xu, Z. Fedlab: A flexible federated learning framework. *CoRR abs/2107.11621* (2021).

[259] Zhang, D., Wang, Y., Zhou, L., Yuan, H., and Shen, D. Multimodal classification of alzheimer's disease and mild cognitive impairment. *NeuroImage 55*, 3 (2011), 856–867.

[260] Zhang, J., Zhang, J., Chen, J., and Yu, S. GAN enhanced membership inference: A passive local attack in federated learning. In *2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020* (2020), IEEE, pp. 1–6.

[261] Zhang, L., Tan, J., Han, D., and Zhu, H. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug discovery today 22*, 11 (2017), 1680–1685.

[262] Zhang, W., Yin, Z., Sheng, Z., Li, Y., et al. Graph attention multi-layer perceptron. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC,*

USA, August 14 - 18, 2022 (2022), A. Zhang and H. Rangwala, Eds., ACM, pp. 4560–4570.

[263] Zhang, W., Zhou, T., Lu, Q., Wang, X., Zhu, C., Sun, H., Wang, Z., Lo, S. K., and Wang, F. Dynamic-fusion-based federated learning for COVID-19 detection. *IEEE Internet Things J. 8*, 21 (2021), 15884–15891.

[264] Zhang, Y., Wang, S., Xia, K., Jiang, Y., and Qian, P. Alzheimer's disease multiclass diagnosis via multimodal neuroimaging embedding feature selection and fusion. *Inf. Fusion 66* (2021), 170–183.

[265] Zhao, A., Balakrishnan, G., Durand, F., Guttag, J. V., and Dalca, A. V. Data augmentation using learned transformations for one-shot medical image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019* (2019), Computer Vision Foundation / IEEE, pp. 8543–8553.

[266] Zhao, Y., Barnaghi, P. M., and Haddadi, H. Multimodal federated learning on iot data. In *Seventh IEEE/ACM International Conference on Internet-of-Things Design and Implementation, IoTDI 2022, Milano, Italy, May 4-6, 2022* (2022), IEEE, pp. 43–54.

[267] Zhao, Z., Birke, R., Kunar, A., and Chen, L. Y. Fed-tgan: Federated learning framework for synthesizing tabular data. *CoRR abs/2108.07927* (2021).

[268] Zhu, H., Zhang, H., and Jin, Y. From federated learning to federated neural architecture search: A survey. *CoRR abs/2009.05868* (2020).

[269] Ziller, A., Trask, A., Lopardo, A., Szymkow, B., Wagner, B., Bluemke, E., Nounahon, J.-M., Passerat-Palmbach, J., Prakash, K., Rose, N., Ryffel, T., Reza, Z. N., and Kaissis, G. *PySyft: A Library for Easy Federated Learning*. Springer International Publishing, Cham, 2021, pp. 111–139.

Ringraziamenti

Ringrazio la Vita per tutte le lezioni con cui mi ha reso ciò che sono. La ringrazio perchè non mi ha mai messo primo e mi ha spesso lasciato ultimo. La ringrazio per avermi insegnato ad essere l'ultimo a mollare.

Grazie a *Lui*, per esser stato il mio miglior amico. Per l'integrità, il coraggio e la consistenza investita. Per averci creduto nonostante gli ostacoli. Questo è il ricordo.

Grazie alla mia famiglia ed a tutti coloro che in qualche modo mi hanno supportato, sopportato ed hanno tifato per me.

Grazie alla straordinarietà della Scienza. Che si concede a chi lascia sfogo alla curiosità. Scienza che apre, non chiude. Scienza che accoglie, non giudica. Scienza che trasforma sforzi ed impegno in conoscenza che arricchisce.

$TiRicordiDiMe$?