FeLebrities: a user-centric assessment of Federated Learning frameworks

Walter Riviera [1], Gloria Menegaz [1], and Ilaria Boscolo Galazzo [1]

[1]Affiliation not available

October 31, 2023

Abstract

Federated Learning (FL) is a new paradigm that aims at solving the data access problem. It is gaining an increasing interest in a variety of research fields, including the Biomedical and Financial environments, where lots of valuable data sources are available but not often directly accessible due to the regulations that protect sensitive information. FL provides a wayout enabling the processing and sharing of data modeling solutions moving the focus from data to models. The FL paradigm involves different entities (institutions) holding proprietary datasets, contributing with each other to locally train a copy of a shared Artificial Intelligence (AI) model. Although there are different studies in the literature that suggest how to conceptually implement and orchestrate a federation, fewer efforts have been made on practical implications. With the ambition of helping accelerating the exploitation of FL frameworks, this paper proposes a survey of public tools that are currently available, an objective ranking based on current state of user preferences and the assessment of the growth trend of the tool popularity over a six months time window. Finally, a ranking of the tools maturity is derived based on key aspects to consider when building a FL pipeline.

FeLebrities: a user-centric assessment of Federated Learning frameworks

Walter Riviera, Ilaria Boscolo Galazzo, *Member, IEEE*, Gloria Menegaz, *Senior Member, IEEE*

Abstract—Federated Learning (FL) is a new paradigm aiming to solve the data access problem. It is gaining an increasing interest in a variety of research fields, including the Biomedical and Financial environments, where lots of valuable data sources are available but not often directly accessible due to the regulations that protect sensitive information. FL provides a solution by moving the focus from sharing data to sharing models. The FL paradigm involves different entities (institutions) holding proprietary datasets, contributing with each other to train a global Artificial Intelligence (AI) model using their own locally available data. Although several studies propose ways to distribute the computation or aggregate results, fewer efforts have been made on how to implement it. With the ambition of helping accelerate the exploitation of FL frameworks, this paper 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 the assessment of the growing trend of the tool's popularity over a six months time window. Finally, a ranking of the maturity of the tools is derived based on keyaspects to consider when building an FL pipeline.

Index Terms—Federated Learning tools, Distributed systems, AI at scale.

1 Introduction

FEDERATED LEARNING FL is the paradigm that aims at solving the data access problem. In the Artificial Intelligence (AI) domain, data represent the starting point for many research and development activities. With rising attention given to the field, data have also grown in demand and appreciation, redefining the list of priorities in designing and building solutions for real-world applications. A clear demonstration of this growing importance is represented by the creation of dedicated laws - such as General Data Privacy Regulations (GDPR) [1] in place in the European Union, the Protection of Personal Information Act (POPIA)footnotehttps://popia.co.za/, and the Health Insurance Portability and Accountability Act (HIPAA)[1] in the USA, which is specific for accessing clinical data and medical records. From the AI perspective, this reflects the need to access data for advancing the State of the Art (SOA) in a given environment while fully complying with the regulations. FL is an effective way to satisfy all those requirements. In a federation of collaborating institutions, what is shared is a common global model, partially trained by every single collaborator using local data. Historically, the approach of training AI models would assume 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[2]) centers are great examples of this approach, as illustrated in Figure 1. Contrarily, 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 the institutions taking part in a federation. A generic example is provided in Figure 2.

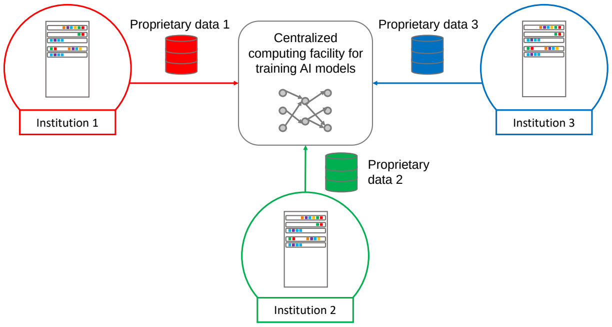


Fig. 1. Data-to-model: example of legacy approach where data would move to a centralized training facility. Here the AI model is represented as a graph or neural network.

The research community has already started investigating this emerging topic either for its privacy-compliant aspects [2] [3] as well as a viable tool for addressing AI challenges in critical domains such as the biomedical context [4] [5] [6]. Despite the domain being still relatively new, the literature can already provide helpful surveys on how the concept works and complies with privacy aspects [2], how it can be transferred in the *Internet-of-things (IoT)* world [7] and what are the steps to implement it from a protocol, software and hardware standpoint [7]. 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

- *W. Riviera, I. Boscolo Galazzo and G. Menegaz are with the Department of Computer Science, University of Verona, Verona, Italy.*

  E-mail: walter.riviera, gloria.menegaz, ilaria.boscologalazzo@univr.it
- *W. Riviera is with Intel Corporation.*
  E-mail: walter.riviera@intel.com

[1] https://www.cdc.gov/phlp/publications/topic/hipaa.html
[2] https://www.top500.org/

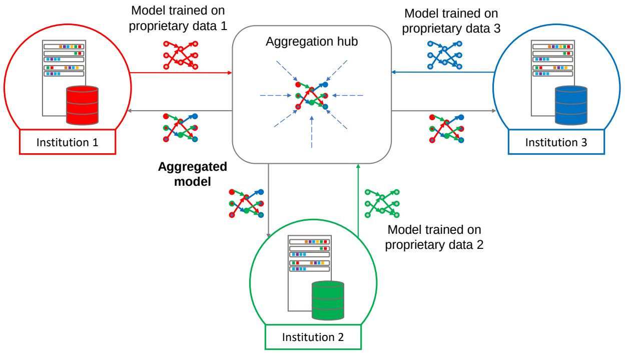

Model trained on proprietary data 1
Model trained on proprietary data 3
Aggregation hub
Institution 1
Institution 3
Aggregated model
Model trained on proprietary data 2
Institution 2

Fig. 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 coming from different institutions and restart the whole process by sending out the latest aggregated version.

aspect is encouraging, on the other, it reflects the need to get 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 paper 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;
3) Identify and describe the key aspects required by the research community 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 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 the 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.

1.1 Paper organization

This paper is composed of six Sections. In section II, we discuss the SOA for FL implementations. Section III focuses on the list of tools currently available to the community, sharing a high-level overview of their popularity and level of adoption. Section IV will augment the retrieved list of tools with the current state of adoption, including the growth trend observed over six months, and Section V discusses the key aspects that would need to 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 lead to a ranking presented as a table. Ultimately, we draw some conclusions and share future directions in Section VI.

2 Related works

FL is a distributed machine learning (ML) approach that enables organizations to collaborate on projects without sharing sensitive data [8], such as patient records [9], [10] or financial data [11], or not easily accessible data, like the ones stored in remote locations as satellites or space stations from high-resolution sensors [12]. The basic premise behind FL [2] [13] is that the model `moves` to meet the data rather than the data `moving` to meet the model. Therefore, the minimum data movement needed across the federation is solely the model parameters and their updates.

2.1 FL settings

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

In the synchronous scenario, the idea is to have a central unit, often identified as aggregator [8] [**?**] [9], acting as a central pivot and determining when to start a new iteration. The aggregator would be responsible for cloning the initial model to each collaborating institution, waiting to receive the locally trained copies coming back, and finally merging them, as the name suggests. This type of FL pipeline is usually implemented among big data centers (cross-silo), like the ones involved in medical environments [14] [15]. Data centers can store vast amounts of data and provide the required computational power to process them. On top of this, big computing infrastructures like 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 the data centers towards the edge devices, new challenges arise due to the high variance in products and manufacturers. Devices with different latency, working frequency, and hardware features can lead to different computation times [16] [17]. These are some reasons motivating the need to have an asynchronous FL pipeline. In this scenario, each collaborating institution can share its update at any time either to a unique aggregator [18], [16], [19] or to the other participants in an ”all-to-all” setup [20], [21].

Another critical point to address is the difference between Horizontal (HFL) and Vertical (VFL) federated learning. To understand this difference, we need to consider the features' space and the model type. In the examples shared so far, we were implicitly referring 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 of institutions with offices distributed across different locations that would like to train a common model leveraging the local data stored in each facility in a privacy-compliant way. In the Vertical FL, each collaborator is expected to contribute by providing different bits of information of the same samples. This leads to a scenario where the feature space accessed by every collaborator might be different from the others. Because of this, each collaborator might be training a different model in the Vertical configuration. The 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 [22], [23]. To give an example, in a typical VHL setting, we could see a life-insurance agency collaborating with hospitals to build a decision model to get more precise estimations for their affiliates. In this case, the expectation would be 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 the model and how the federation gets orchestrated. While in the HFL, there is only one model, and all the collaborators are responsible for ensuring that data gets 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. On top of this, there must be a federated model, which takes all the outputs of the various local models as input. As illustrated by Chen et al. [23], the procedure for training DL models based on back-propagation [24], [25], needs to deal with the two-level training procedure represented by the different models that need to be managed: one at the collaborator level, the other at the aggregation point. This complexity is also reflected in the challenges that might arise in finding a satisfying converging point for the adopted DL model.

2.2 FL challenges

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

1) Aggregation functions and model convergence starting from different data distributions;
2) Privacy aspects and ways to build a secure FL pipeline for protecting IP during the 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 [26] [27]. Data distribution can severely impact the training performance by affecting the total accuracy [28], the convergence capability, the authentication processes (especially in the case of different devices), and the speed of the process intended as total time-to-train [29]. In a nutshell, under this setting, the performance of the training process may vary significantly according to the unbalance of local data samples and the particular statistical distribution of the training examples (i.e., features and labels) stored at the local nodes [13].

In the past few years, institutions have introduced FL deployments to answer the need for training AI models. Sectors like healthcare and finance would benefit from having a setting with greater access to more extensive and more diverse datasets without violating privacy laws [30] [31], such as HIPAA, GDPR [1] and POPIA[3]. While on one side, FL has been designed with security in mind [28], the setup is just the beginning. Securing execution environments brings a lot of open challenges for the research field [32]. Key questions include finding a consolidated method to guarantee secure execution (encryption, key exchange, 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 participate in a federation, the more critical the ability to scale. As mentioned in the previous Section, to the best of our knowledge, a consolidated way for detecting "poor" training contributions (coming from institutions with corrupted or redundant data) is still missing. Aggregation functions are also currently being evaluated by the research community [26] [31] [33]. Another implication when talking about big scales is represented by the infrastructure and the connectivity chosen by the institutions for communication [13].

2.3 Study relevance

Several works have proposed surveys to illustrate the advancement in the field [2] [7], however, to the best of our knowledge, no one is providing a ranked list based on adhoc quality assessment criteria of all the (possible) tools available to the community to implement FL experiments. In [34], a comparison of five tools is provided, some of which are accessible through a licensed service, without clarifications on why or how those tools were precisely selected. Another work [35] provides an attractive comparison table. Still, the main focus of that work is to promote an alternative tool specifically for FL benchmarks instead of giving a complete list of the available options to boost the exploitation of FL across the community. Even in this related work, it was unclear why and how the discussed tools were selected. On the same line, [36] proposes a complete benchmarking suite with a helpful decision tree to help users choose a tool based on their needs. Their recommended ranking also includes some of the evaluation metrics proposed in this work 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 the end users. In-fact [36] centers its evaluation around nine tools, but the criteria for which those tools were identified and selected are not clear. As we discovered in this work, the list of open-source FL tools can go beyond 30, and it is interesting to note how the most popular tool to date was not considered in their decision tree.

3 Federated Learning tools

3.1 Methods and premises

This article aims at provide an inclusive and informative list of the current FL tools available to the community for implementing research pipelines in any environment where accessing distributed data is a challenge. To better understand

[3] https://popia.co.za/

the present scenario, we performed two literature searches: one carried out on March 28th and another on September 28th. The activity was inspired by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines. More specifically, we decided to follow the *Preferred reporting items for systematic review and meta-analysis of diagnostic test accuracy studies (PRISMA-DTA): explanation, elaboration, and checklist* [37]. In particular, the guidelines we followed are a selection of the ones 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 we identified as applicable to this collection. The number reported below is a direct reference 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 considered to frame this work. The following map illustrates how the single guideline contributed to shaping what section:

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

3.2 Exploring tools

To objectively build the list of tools, we performed two reviews (harvests), with roughly six months (184 days) as a time gap. Capturing the tool lists in one survey would have been enough to draw a general overview of the current environment at that time, but having two observational points for each of the tools was helpful to understand better the level of commitment from the engineering team and the maturity and popularity growth rate beyond each tool. The collection methods were the same and are described below. We relied on three different search engines: Google Scholar[4], Semantic Scholar[5] and standard Google website[6].

For the first two, we developed a script to automatically query the search engines with 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 the related works where a given keyword (or part of it) was mentioned in the paper and not only in the title. We could identify a cumulative list of 420 related articles, of which 217 were unique. Despite the encouraging numbers, we soon encountered a bottleneck represented by the API service. For each keyword, the system would only return maximum 20 results. Furthermore, after an indefinite but limited amount of searches, a CAPTCHA request would rise, blocking any automatic API interrogation. The service provider wants to avoid free unlimited searches performed by automated systems to ensure enough resources for manual searches performed by real users. For this reason, the referred numbers belong to the article harvest completed in March only. We could not run any new queries during the September harvest using this engine.

To build a more robust and consistent set of related works, we leveraged the Semantic Scholar service [38]. As defined on their website, *”Semantic Scholar is a free, AI-powered research tool for scientific literature, based at the Allen Institute for AI.”* Using this tool, we were able to increase the number of results obtained for each keyword search to 100 and access more accurate content given the semantic nature of the search engine. The website (and its API) allows users to perform queries and sort the outcome according to four metrics: ”Relevance”, ”Citations-count”, ”Most Influential Paper” and by ”Recency”. Among these options, a user could also select articles based on where they were published (e.g., conferences, journals, books) and the field of studies and applications (e.g., Medicine, Geology,

[4] https://scholar.google.com/
[5] https://www.semanticscholar.org/
[6] https://www.google.com

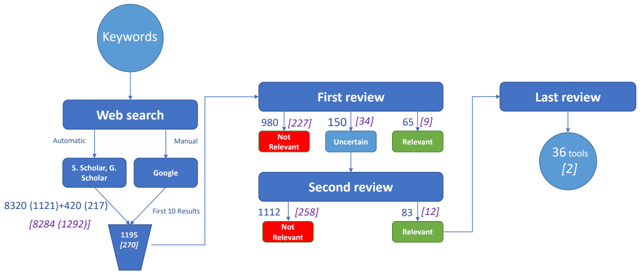
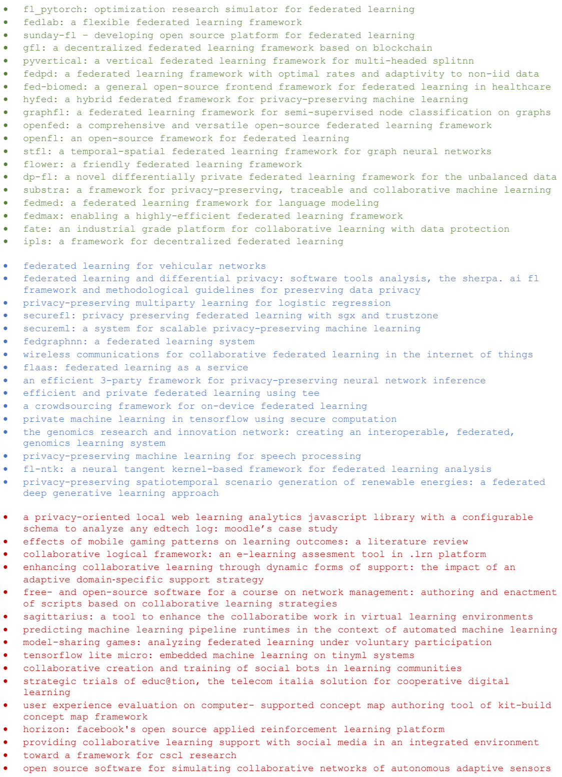


Fig. 3. Collection pipeline implemented for the harvests. On each arrow is outlined the number of articles retrieved and filtered. Numbers between squared brackets refer to the outcome of the September harvest. The output numbers of the second review are obtained by summing the numbers of a given category 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.



- 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
- pyvertical: a vertical federated learning framework for multi-headed splitnn
- fedpd: a federated learning framework with optimal rates and adaptivity to non-iid data
- fed-biomed: a general open-source frontend framework for federated learning in healthcare
- hyfed: a hybrid federated framework for privacy-preserving machine learning
- graphfl: a federated learning framework for semi-supervised node classification on graphs
- openfed: a comprehensive and versatile open-source federated learning framework
- openfl: an open-source framework for federated learning
- stfl: a temporal-spatial federated learning framework for graph neural networks
- flower: a friendly federated learning framework
- dp-fl: a novel differentially private federated learning framework for the unbalanced data
- substra: a framework for privacy-preserving, traceable and collaborative machine learning
- fedmed: a federated learning framework for language modeling
- fedmax: enabling a highly-efficient federated learning framework
- fate: an industrial grade platform for collaborative learning with data protection
- ipls: a framework for decentralized federated learning

- federated learning for vehicular networks
- federated learning and differential privacy: software tools analysis, the sherpa. ai fl framework and methodological guidelines for preserving data privacy
- privacy-preserving multiparty learning for logistic regression
- securefl: privacy preserving federated learning with sgx and trustzone
- secureml: a system for scalable privacy-preserving machine learning
- fedgraphnn: a federated learning system
- wireless communications for collaborative federated learning in the internet of things
- flaas: federated learning as a service
- an efficient 3-party framework for privacy-preserving neural network inference
- efficient and private federated learning using tee
- a crowdsourcing framework for on-device federated learning
- private machine learning in tensorflow using secure computation
- the genomics research and innovation network: creating an interoperable, federated, genomics learning system
- privacy-preserving machine learning for speech processing
- fl-ntk: a neural tangent kernel-based framework for federated learning analysis
- privacy-preserving spatiotemporal scenario generation of renewable energies: a federated deep generative learning approach

- a privacy-oriented local web learning analytics javascript library with a configurable schema to analyze any edtech log: moodle’s case study
- effects of mobile gaming patterns on learning outcomes: a literature review
- collaborative logical framework: an e-learning assesment tool in .lrn platform
- enhancing collaborative learning through dynamic forms of support: the impact of an adaptive domain-specific support strategy
- free- and open-source software for a course on network management: authoring and enactment of scripts based on collaborative learning strategies
- sagittarius: a tool to enhance the collaboratibe work in virtual learning environments
- predicting machine learning pipeline runtimes in the context of automated machine learning
- 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
- strategic trials of educ@tion, the telecom italia solution for cooperative digital learning
- user experience evaluation on computer- supported concept map authoring tool of kit-build concept map framework
- horizon: facebook's open source applied reinforcement learning platform
- providing collaborative learning support with social media in an integrated environment
- toward a framework for cscl research
- open source software for simulating collaborative networks of autonomous adaptive sensors

Fig. 4. A subset of examples of selected articles for each category: "relevant" green, "uncertain" blue and "non-relevant" red.

Economics). For our collection, we did not leverage any of these options as we wanted to be as inclusive as possible.

By repeating the research of all the keywords for all the four sorting types mentioned above, we obtained a cumulative list of 8320 articles, of which 1121 were unique during the March harvest, and a cumulative list of 8284 articles, of which 1292 were unique, during the September harvest. The number of new articles retrieved in September that were unavailable in March is 270.

The fact that among the Google Scholar search results, we obtained 51% of unique contributions versus the 14% of the ones identified through Semantic Scholar is an indicator of the goodness of the research. Through Semantic Scholar, we found more related and consistent articles, which we interpreted as a reliable capability of the tool to capture the semantic aspects of the research.

To ensure we would capture all the relevant FL tools not yet described in a published paper, we also decided to perform a manual search on the standard Google search engine. 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 (Clara) [7], Tensorflow Federated [8] and IBM Federated [9].

Once we obtained the three lists of unique titles described above, we finally merged them, resulting in a total of 1195 unique articles discovered in March and a list of 1292 retrieved in September. 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 utterly unrelated to the topic (e.g., work mentioning ML methods or collaborative learning platforms for schools) were discarded from the whole collection. After the first labeling cycle, we had 65 R, 980 NR, and 150 TBD for the March harvest, and 9 R, 227 NR, and 34 TBD for the September harvest.

An example of articles being captured by the three categories is available in figure 4.

The ”uncertain” category required us to conduct a deeper review of the work. All the articles in this list went through a second round of labeling. The objective was to review the 150 papers belonging to the TBD list to allocate them either to the R or NR. As resulting cardinalities, we obtained: 83 R and 1112 NR for the March harvest and 12 R and 258 NR for the September harvest.

A summary of the research pipeline adopted and the results collected during each Harvest is shown in Figure 3.

Ultimately, a deeper understanding of the relevant papers was performed to draw the final list of FL tools. In general, for both the harvests, Many articles used the word ”framework” to suggest methods and approaches to addressing specific FL tasks but were not proposing toolkits or open-source products that the community could leverage to implement FL pipelines. This final review allowed us to identify 36 suitable tools during the March harvest and additional two tools during the September harvest. The complete list of tools retrieved in March with the indicators from the Github and Gitlab repositories is reported in Table 1.

[7] https://blogs.nvidia.com/blog/2021/11/29/federated-learning-ai-nvidia-flare/
[8] https://www.tensorflow.org/federated
[9] https://ibmfl.mybluemix.net/

	TOOL	Watch	Fork	Star	ETA (days)	SCORES
1	PySyft [39]	211	1800	8000	1714	1.95
2	FATE https://fate.fedai.org/	134	1200	4100	956	1.89
3	FedML [40]	37	331	1100	614	0.80
4	Tensorflow Federated https://www.tensorflow.org/federated	66	447	1800	1301	0.59
5	Flower [41]	20	15	825	770	0.37
6	OpenFL [42]	10	76	286	437	0.28
7	IBM-Federated [43]	20	85	292	647	0.20
8	FedLab [44]	6	30	180	383	0.19
9	LEAF [45]	18	187	470	1249	0.18
10	FedGraphNN [46]	9	33	147	383	0.16
11	Fedlearn-algo [47]	8	34	80	255	0.16
12	FedJAX [48]	11	28	172	461	0.15
13	PyVertical [49] [50]	12	35	101	661	0.07
14	PriMIA [51]	8	18	102	707	0.06
15	Substra [52]	8	24	143	1252	0.05
16	Fedn [53]	7	20	59	622	0.05
17	FedBioMed (GitLab) [54]	NA	23	3	332	0.04
18	OpenFED [35]	2	1	17	300	0.02
19	APPFL [55]	2	1	7	172	0.02
20	HyFed [56]	4	3	9	361	0.01
21	PyFed [57]	3	2	5	544	0.01
22	dsMTL [58]	2	2	0	266	0.01
23	Sunday FL [59]	1	4	2	524	0.00
24	DecFL [60]	2	2	3	717	0.00
25	MTC-ETH [61]	1	1	2	881	0.00
26	Vantage6 [62]	5	2	0	1835	0.00
27	Sherpa ai [63]	3	0	0	0	-
28	FL-Pytorch (Pytorch Federated) [64]	NA	NA	NA	NA	NA
29	Chiron [65]	NA	NA	NA	NA	NA
30	FedHealth [66]	NA	NA	NA	NA	NA
31	FAE [67]	NA	NA	NA	NA	NA
32	GENO [68]	NA	NA	NA	NA	NA
33	FedTGan [69]	NA	NA	NA	NA	NA
34	FL-Bench [70]	NA	NA	NA	NA	NA
35	IPLS [71]	NA	NA	NA	NA	NA
36	Nvidia-Clara https://docs.nvidia.com/clara/	NA	NA	NA	NA	NA

TABLE 1

Tool popularity table, March harvest. 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.

4 Tools popularity and level of adoption

After retrieving the list of tools, our goal was to understand each item's popularity and level of adoption from the community perspective. Each Git repository has public indicators like the number of Watch (W), Fork (F), and Stars(S). The Watch indicator can capture the number of users actively watching the repository. These users will receive updates when new actions are taken on the repository. The number of Fork indicates the number of times a repository has been forked. It can tell how many interested users might develop code to extend the tool. Finally, the number of Stars indicates the number of likes the repository has received. This final indicator might need to be more accurate in capturing actual users, but it can give a reasonable estimation of the reach in terms of how many people have seen the tool at least once. For practicality, we wanted to combine these three aspects into one consolidated score that we could use for providing a *popularity* driven ranking of the tools. Since the popularity would also depend on the time a given repository was made available to the commu-

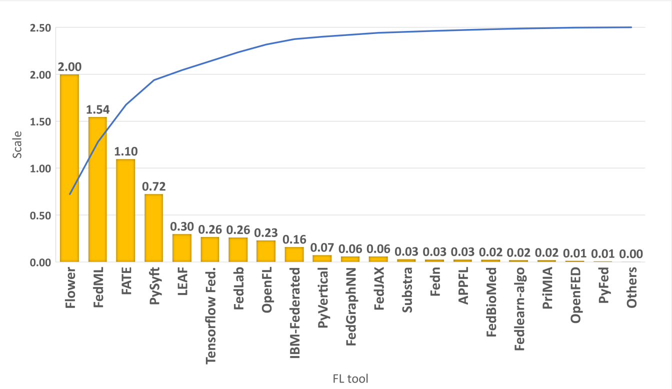


Fig. 5. Popularity growth rate: this graph illustrates which tools have been gaining more popularity in the community over an observation window of 184 days (roughly six months). Tools that did not have a repository available in March were excluded from this chart.

nity, we wanted to normalize all the values with a timing factor. This step would ensure that newer repositories with less exposure to the community would not be affected by a low score. To achieve this goal, we combined these three factors into a consolidated score calculated as follows:

$$Score_{\mathrm{t}} = \frac{\frac{1}{3}(W_t + F_t + S_t)}{ETA_{\mathrm{fr}}} \tag{1}$$

Where ETA_{fr} is the time elapsed between the day of the tool's first release and the harvest date. We used the date of the first commit if the date of the first release was unavailable. Table 1 shows the scores associated with the tools retrieved in 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 the Git indicators can share important insights about user preferences in a given time frame, they do not necessarily capture the community trends from a popularity growth rate perspective.

To access this information, we observed the list of tools over a time window to check which tools were being considered by the community at a higher pace. Thanks to the second harvest (September 28th), 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 in March and in September, we computed the growth rate for each of the tools as follows:

$$gr_{\mathrm{t}} = \frac{\frac{1}{3}((W_t + F_t + S_t)^{September} - (W_t + F_t + S_t)^{March})}{ETA} \tag{2}$$

Where ETA, in this case, corresponds to 184 days. The outcome of this computation can be appreciated in Figure 5.

Interestingly, the order of the tools based on the popularity level observed in March and captured by Table 1 does not match the growth rate highlighted in Figure 5.

5 Priorities in FL tools for Research

Previous Sections illustrated how we could retrieve a list of relevant tools for building FL pipelines and which ones are preferred by the community. In this Section, our goal is to provide a new ranking of the tools to identify the most mature ones. We will focus on the specific features of each tool, regardless of any popularity aspects as outlined in the previous Sections.

The task consists in retrieving and evaluating the FL tools that could be adopted to boost exploitability. To perform this classification, we defined a set of measures based on the different needs and expectations 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 leveraging the agile nature of IoT devices, FL pipelines have to be shaped according to the needs, goals, and constraints to consider. However, coherent 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.

On the other side, 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 [12], the applicability of FL is not only related to the need to access data complying with regulations. It can also refer to data that is not readily retrievable, like the ones on a satellite or space station. Furthermore, the Medical [14] or Geo-spatial [12] environments are usually sources of high-resolution data acquired by machines manufactured by different companies, which could map in need of having dedicated preprocessing 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 as it enhances the reliability of the conclusions when investigating privacy and communication efficiency aspects.

In addition to what has been outlined so far, other practical considerations related to the research environment might apply. The easier the path to results in a research environment is, the fastest could be the path to deploying 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 (like Python) and AI platforms (like 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.

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
   - Privacy and Security independent module? (PVC/SEC)
   - Easy integration with other tools

We used these parameters to build an ”Evaluation Table” [2] for the tools identified in the previous Section. The table has been populated with information retrieved from publicly available resources for each tool. As one can easily see, there is a mismatch between the tools listed in Table 1 and those reported in Table 2. This is mainly due to the following four reasons:

1) Tools not open-source, like Sherpa-ai [63]
2) Missing repositories: is the case of tools that have not yet released their codes after the paper publication: Chiron [65], FedHealth [66], FAE [67], GENO [68], FedTGan [69] and IPLS [71].
3) Coherent but not suitable: is the case of LEAF [45], FL-Bench [70], and PyFed [57], which are positioned for benchmarking purposes and, therefore, might lack essential features for conducting more extensive research activities. FedGraphNN [46] is a sub-project of the more significant initiative called FedML [40] already included in this survey.
4) New tools or new openings: is the case of Nvidia-Flare (a sub-project of Nvidia-Clara https://docs.nvidia.com/clara/) and FL-Pytorch [64] which opened their repositories at some point after March harvest, as well as FLUTE [72], and PLATO [73] which have been retrieved (and added) during the September harvest.

After pruning the 12 tools that did not qualify, adding the 2 (despite the substitution with Nvidia-Flare, Nvidia-Clara was already captured by Table 1) from the September harvest, we ended up with a list of 27 total tools.

In a second instance, a score would be associated with each cell based on a quantitative assessment. Aiming at an objective classification of the tools, we captured qualitative aspects in a very inclusive way, rewarding tools that demonstrated additional development effort for the community through the available material without penalizing newer promising tools that might still be under development. More in detail, we adopted a simple approach to assign a score to each cell and designed the ”Score Table” 3:

- Documentation: we considered having a Paper P and-or a public repository of the tool Gh (Github) or Gl (Gitlab) to be 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 (like 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.
- User Interface (UI): 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 [10] or google collab [11] to mention two). We gave 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 the information about the supported version was not clearly outlined in the documentation. 1 point was given to the tools supporting at least one language (or one version), and 1.5 points were given to all the tools supporting two languages (or two versions of a language). Finally, 2 points were given to all the tools where the engineering team made 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 framework supported. When the number of different supported frameworks exceeds 2, we assign the maximum score of 2.5 points.
- Type of AI: 0 points if not mentioned in the documentation, 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 0 points to all the tools respecting this minimum requirement, 1 point to all the tools that had at least 1 additional way to access the software package (e.g., Pipy or Anaconda), finally, 1.5 points were given to 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. 1 point to all the tools that allow imple-

[10] https://jupyter.org/

[11] https://colab.research.google.com/

TOOL	Docs.	DX	Lang.	AI frame-works	AI type	Examples	Dist. channels	Multinode?	C/V	H/V	PVC/SEC	Tools integration
APPFL	P, Gh, Pipy, W	N.M.	Python = 3.6	Pythorc	DL	Yes	Gh, Pipy	Yes	Docker	H	N.M.	MPI orchestration
DecFL	P, Gh	N.M.	N.M.	TF	DL	Yes	Gh	Yes	Required	H	Yes	N.M.
dsMTL	P, Gh,	N.M.	R only	R-based	ML	Yes	Gh	Yes	N.M.	H	Yes	N.M.
FATE	P, Gh, W	N.M.	3.8, 3.9	N.M.	ML, DL	Yes	Gh	Yes	Docker	H,V	Yes	KubeFATE, flake, Spark
FedBioMed (Gl)	P, Gh	Jupyter	python	Pythorc, TF	DL	Yes	Gh	Yes	Required	H	Yes	Flake, Sklearn
FedJAX	P, Gh, W	G. Colab, Jupyter	Python	Pythorc, TF	DL	Yes	Gh, Pipy	Simulated	N.M.	H	N.M.	N.M.
FedLab	P, Gh, W	N.M.	python = 3.6	Pytorch	DL	Yes	Gh, Pipy	Yes	Docker	H	Yes	N.M.
Fedlearn-algo	P, Gh	N.M.	Python 3.6, 3.7	Pythorc, TF	ML, DL	Yes	Gh	Yes	Docker	H,V	Yes	Hugging Face, Sklearn
FedML	P, Gh, W, dedicated links, Slack	N.M.	Python	Pytorch, TF, JAX, mxnet	ML, DL	Yes	Gh, Pipy	Yes	Docker	H	N.M.	MPI, NCCL, MQTT, gRPC, Pytorch RPC
Fedn	P, Gh, dedicated docs	Yes	Python 3.8	Pytorch, TF	DL	Yes	Gh	Yes	Required	H,V	Yes	C++, scikit-learn
FLOM	Missing repo											
Flower	P,	Jupyter, colab	Python = 3.6	Pytorch, TF, MxNet, Jax	ML, DL	Yes	Gh, Pipy	Yes	Yes	H	Yes	Android, flake, hugging Face
FLUTE	P, Gh	N.M.	3.8	Pytorch	DL	Yes	Gh	Azure Cloud	N.M.	H	N.M.	HuggingFace
HyFed	P, Gh	WebAPP	N.M.	N.M.	N.M.	Yes	Gh	Yes	N.M.	H	Yes	N.M.
IBM-Federated	P, Gh, W, video tutorials	Jupyter	Python 3.6	Pytorch, TF	ML, DL	Yes	Gh, wheel	Yes	Docker	H	Yes	Ray, Openshift
IPLS												
MTC-ETH	P, Gh	N.M.	N.M.	N.M.	N.M.	N.M.	Gh	Yes, but low number	Required	H	Yes	N.M.
Nvidia Flare	Gh, Website	N.M.	3.8	Pytorch, TF	DL, ML	Yes	Gh, pip	Yes	Docker	H	Yes	MONAI
OpenFED	P, Gh, W	Jupyter	Python 3.6	Pytorch	DL	Yes	Gh, Pipy	Simulated	N.M.	H	Yes	N.M.
OpenFL	P, Gh, slack, W, videos	Jupyter, Colab	Python 3.6, 3.7, 3.8	Pytorch, TF	DL	Yes	Gh, Pipy	Yes	Docker	H	Yes	MonAI, Hugging Face
PLATO	P, Gh, Website	N.M.	3.9	Pytorch, TF, Mindspore	DL	Yes	Gh, pip	Yes	Docker	H	N.M.	Catalyst, Mindspore
PriMIA	P, Gh, W	N.M.	N.M.	Pytorch	DL	Yes	Gh	Simulated	N.M.	H	Yes	PySyft, K8s
PySyft	P, Gh, Slack,	Jupyter	3.8	Pythorc, TF	DL	Yes	Gh, Pipy	Yes	Docker, VMs. +	H	Yes	PySyft, k8s
FL-Pytorch	P, Gh, W, slack, videos	Custom GUI	3.9	Pytorch	DL	Yes	Gh, pipy	Simulated	N.M.	H	N.M.	N.M.
PyVertical	P, Gh	Jupyter	3.6,3.7,3.8	N.M.	DL	Yes	Gh	Yes	Docker	V	Yes	Syft
Substra	P, Gh, W, Slack	N.M.	N.M.	N.M.	N.M.	Yes	Gh, Pipy	yes	docker	H	N.M.	N.M.
Sunday FL	P, Gh, youtube	N.M.	Python, java	N.M.	N.M.	Yes	Gh	Yes	Docker	H	N.M.	Azure
Tensorflow Federated	P, Gh, W	Colab	Python 3	TF	DL	Yes	Gh, Pipy	Simulated	N.M.	H	N.M.	N.M.
Vantage6	P, Gh, W, youtube	N.M.	Python 3.7	N.M.	N.M.	N.M.	Gh, Pipy	Yes	Docker	H	N.M.	N.M.

TABLE 2

Tool evaluation table. Legend: 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. The NM label refers to ”Not Mentioned”, meaning that the information did not appear as mentioned in the available documentation.

menting 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. 1 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 tools' repositories, we trust that all of them respect this parameter. More in details, all of them have proven to have spearate entities (like client-server and orchestration process) that can be launched independently.

#	TOOL	Docs.	DX	Lang.	AI frameworks	AI type	Examples	Dist. channels	Multinode?	C/V	H/V	PVC/SEC	Tools integration	TOTAL
1	Flower	1	1.5	2	2.5	2	1	1	1	1	0	1	1.5	15.5
2	OpenFL	1.5	1.5	2	2	1	1	1	1	1	0	1	1.5	14.5
3	IBM-Federated	1.5	1	1	2	2	1	1	1	1	0	1	1.5	14
4	PySyft	1	1	1	2	1	1	1	1	1.5	0	1	1.5	13
5	FedML	1.5	0	1	2.5	2	1	1	1	1	0	0	1.5	12.5
6	Fedn	1	1	1	2	1	1	0	1	0.5	1	1	1.5	12
7	Fedlearn-algo	0	0	1.5	2	2	1	0	1	1	1	1	1.5	12
8	PLATO	1	0	1	2.5	1	1	1	1	1	0	0	1.5	11
9	Nvidia Flare	0	0	1	2	2	1	1	1	1	0	1	1	11
10	FATE	1	0	1.5	0	2	1	0	1	1	1	1	1.5	11
11	APPFL	1	0	2	1	1	1	1	1	1	0	0	1	10
12	FedLab	1	0	2	1	1	1	1	1	1	0	1	0	10
13	FedBioMed (GitLab)	0	1	1	2	1	1	0	1	0.5	0	1	1.5	10
14	OpenFED	1	1	2	1	1	1	1	0.5	0	0	1	0	9.5
15	FedJAX	1	1.5	1	2	1	1	1	0.5	0	0	0	0	9
16	PyVertical	0	1	2	0	1	1	0	1	1	0	1	1	9
17	Tensorflow Federated	1	1	2	1	1	1	1	0.5	0	0	0	0	8.5
18	FL-Pytorch	1.5	1	1	1	1	1	1	0.5	0	0	0	0	8
19	PriMIA	1	0	0	1	1	1	0	0.5	0	0	1	1.5	7
20	Sunday FL	1	0	1.5	0	0	1	0	1	1	0	0	1	6.5
21	dsMTL	0.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	6
22	FLUTE	0	0	1	1	1	1	0	0.5	0	0	0	1	5.5
23	DecFL	0	0	0	1	1	1	0	1	0.5	0	1	0	5.5
24	Vantage6	1.5	0	1	0	0	0	1	1	1	0	0	0	5.5
25	Substra	1	0	0	0	0	1	1	1	1	0	0	0	5
26	HyFed	0	1	0	0	0	1	0	1	0	0	1	0	4
27	MTC-ETH	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.5	0	1.0	0.0	2.5

TABLE 3
Tool scoring table

- Horizontal or Vertical: a tool must implement at least one of the two. Therefore we rewarded 1 point only to the tools that would allow executions in both settings.
- 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, etc.). 1 point to all the tools that included at least one of the two.
- Easy integration with other tools: same process applied for evaluating the containers and virtualization mechanism.

6 Results discussion and future directions

6.1 Discussion

We first performed extensive and inclusive semi-automated research to identify suitable tools for implementing FL pipelines. Then we built a popularity ranking based on a Score that we calculated by combining the Stars, Forks, and Watch indicators from each Git repository. Following the same Scoring function, we could also draw the growth rate for each tool over a six months time window.

We ultimately defined a scoring matrix and evaluated each tool to calculate a ”maturity table” as in Table 3. Based on the resulting ranking, the most mature tools are Flower [41], OpenFL [42], and IBM-Federated [43]. While the last two are fairly close to each other, the table leader does not appear to have a solid distance. PySyft [39] is following alone despite FedML [40], and Fedn [53] being very close behind.

This is just an initial observation, but things change when we integrate into the equation the popularity results highlighted in Table 1 and the growth rate outlined in Figure 5. In Table 1, PySyft [39] and FATE https://fate.fedai.org/ are the two most popular tools according to the developer's community, while Flower [41], OpenFL [42] and IBM-federated [43], cover the 5th, 6th, and 7th placement, respectively, with a considerable distance from the first two. An interesting aspect is that there is a clear gap between what the community awarded as the most popular tools and what this work outlined as the most mature ones. On the same line, another essential element is the results highlighted by the growth rates reported in Figure 5. Leading that ranking is Flower [41], followed by FedML [40], FATE https://fate.fedai.org/ and PySyft [39]. OpenFL [42] is in the 8th placement, with a growth rate of 0.26, which is approximately 10 times smaller than the Table leader, which has a value of 2.

One interesting aspect is that regardless of which scoring we decide to consider, the top 5 placements seem to be occupied by the same names, re-shuffled a bit. In fact, out of 15 possible different names, we can only count up to 8 different tools. Out of those eight different names, four are the more dominant as they appear in at least two rankings: Flower [41], FedML [40], FATE https://fate.fedai.org/, and PySyft [39]. The remaining 4 are LEAF [45], Tensorflow-Federated (tff) https://www.tensorflow.org/federated, OpenFL [42] and IBM-federated [43].

Despite our best efforts to adopt objective scoring when building Table 3, we are aware that other valid alternatives might exist. For example, a more in-depth analysis of all the functional features provided by each tool (such as communication protocols) 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 elements in the lists, we would not expect significant shifts in the main order. This consideration arises when looking at what defines the current ranking. The success of the top two tools is mainly justified by the high score obtained in the ”Usability” and ”Portability” factors outlined in Section V. This might suggest 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 could be critical for the community. As confirmed when looking at the lower part of Table 3, low scoring for the worse-ranked tools might not necessarily be related to a lack of critical features but more to insufficient documentation that might have compromised the exploitation. On the other side, we noticed a discrepancy in Table 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 have identified three main ones: Participation in more significant international projects involving multiple institutions Tool adoption in various application fields More dissemination and marketing activities by the respective engineering teams The suggestion is to revise the proposed criteria to account for these arguments and potentially other factors to get closer to a comprehensive measure harmonizing the overall results.

6.2 Conclusions

Several tools for implementing FL pipelines could accelerate the community's research activities in this field. In this paper, we provided a survey of all open-source solutions and a ranking based on tool popularity and readiness with the ambition to guide users (including non-experts) in adopting FL solutions, boosting their exploitation, and accelerating their research and development. Through this work, we learned that tools primarily adopted by the community are not necessarily the most mature tool. Although summarizing the results of the three tables might be difficult, we can say that if somebody does not know where to start with FL, tools like Flower [41] or PySyft [39] represent a good compromise between maturity and popularity. At the same time, we recognize that more in-depth benchmarking with dedicated tools like [36], LEAF [45], or FL-bench [70] might be needed to correctly asses the different peculiarities of each tool.

Acknowledgments

References

[1] K. Hjerppe, J. Ruohonen, and V. Leppänen, "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*, D. E. Damian, A. Perini, and S. Lee, Eds. IEEE, 2019, pp. 265–275.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[3] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, and K. S. Ng, "Towards fair and privacy-preserving federated deep models," *IEEE Trans. Parallel Distributed Syst.*, vol. 31, no. 11, pp. 2524–2541, 2020.

[4] M. Y. Lu, R. J. Chen, D. Kong, J. Lipková, R. Singh, D. F. K. Williamson, T. Y. Chen, and F. Mahmood, "Federated learning for computational pathology on gigapixel whole slide images," *Medical Image Anal.*, vol. 76, p. 102298, 2022.

[5] H. R. Roth, K. Chang, P. Singh, N. Neumark, W. Li, V. Gupta, S. Gupta, L. Qu, A. Ihsani, B. C. Bizzo, Y. Wen, V. Buch, M. Shah, F. Kitamura, M. Mendonça, V. Lavor, A. Harouni, C. Compas, J. Tetreault, P. Dogra, Y. Cheng, S. Erdal, R. D. White, B. Hashemian, T. J. Schultz, M. Zhang, A. McCarthy, B. M. Yun, E. Sharaf, K. V. Hoebel, J. B. Patel, B. Chen, S. Ko, E. Leibovitz, E. D. Pisano, L. Coombs, D. Xu, K. J. Dreyer, I. Dayan, R. C. Naidu, M. Flores, D. L. Rubin, and J. Kalpathy-Cramer, "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*, ser. Lecture Notes in Computer Science, 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. Springer, 2020, pp. 181–191.

[6] S. S. S. R., B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "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*. IEEE, 2019, pp. 270–274.

[7] S. A. Rahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, 2021.

[8] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "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*, ser. Lecture Notes in Computer Science, A. Crimi, S. Bakas, H. J. Kuijf, F. Keyvan, M. Reyes, and T. van Walsum, Eds., vol. 11383. Springer, 2018, pp. 92–104.

[9] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[10] D. Stripelis, J. L. Ambite, P. Lam, and P. M. Thompson, "Scaling neuroscience research using federated learning," in *18th IEEE International Symposium on Biomedical Imaging, ISBI 2021, Nice, France, April 13-16, 2021*. IEEE, 2021, pp. 1191–1195.

[11] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," *CoRR*, vol. abs/2108.10749, 2021.

[12] J. So, K. Hsieh, B. Arzani, S. A. Noghabi, S. Avestimehr, and R. Chandra, "Fedspace: An efficient federated learning framework at satellites and ground stations," *CoRR*, vol. abs/2202.01267, 2022.

[13] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[14] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen *et al.*, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[15] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai *et al.*, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[16] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "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*, 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, 2020, pp. 15–24.

[17] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 1–24, 2022.

[18] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent iot applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, 2020.

[19] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," 2018. [Online]. Available: http://arxiv.org/abs/1812.02903

[20] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive iot networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, 2020.

[21] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[22] K. Wei, J. Li, C. Ma, M. Ding, S. Wei, F. Wu, G. Chen, and T. Ranbaduge, "Vertical federated learning: Challenges, methodologies and experiments," *CoRR*, vol. abs/2202.04309, 2022.

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

[24] P. Baldi and P. J. Sadowski, "A theory of local learning, the learning channel, and the optimality of backpropagation," *Neural Networks*, vol. 83, pp. 51–74, 2016.

[25] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[26] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "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*. OpenReview.net, 2020.

[27] F. Sattler, S. Wiedemann, K. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *CoRR*, vol. abs/1903.02891, 2019.

[28] C. Briggs, Z. Fan, and P. Andras, "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*. IEEE, 2020, pp. 1–9.

[29] K. A. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "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*, A. Talwalkar, V. Smith, and M. Zaharia, Eds. mlsys.org, 2019.

[30] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, I. Ray, N. Li, and C. Kruegel, Eds. ACM, 2015, pp. 1310–1321.

[31] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "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.

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

[33] Q. Yang, H. Matsutani, and M. Kondo, "A selective model aggregation approach in federated learning for online anomaly detection," 11 2020.

[34] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, and M. Nordlund, "Open-source federated learning frameworks for iot: A comparative review and analysis," *Sensors*, vol. 21, no. 1, p. 167, 2021.

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

[36] X. Liu, T. Shi, C. Xie, Q. Li, K. Hu, H. Kim, X. Xu, B. Li, and D. Song, "Unifed: A benchmark for federated learning frameworks," *CoRR*, vol. abs/2207.10308, 2022.

[37] j.-p. Salameh, P. Bossuyt, T. McGrath, B. Thombs, C. Hyde, P. Macaskill, J. Deeks, M. Leeflang, D. A. Korevaar, P. Whiting, Y. Takwoingi, J. Reitsma, J. Cohen, R. Frank, H. Hunt, L. Hooft, A. Rutjes, B. Willis, C. Gatsonis, and M. McInnes, "Preferred reporting items for systematic review and meta-analysis of diagnostic test accuracy studies (prisma-dta): Explanation, elaboration, and checklist," *BMJ*, vol. 370, p. m2632, 08 2020.

[38] S. Fricke, "Semantic scholar," *Journal of the Medical Library Association: JMLA*, vol. 106, no. 1, p. 145, 2018.

[39] A. Ziller, A. Trask, A. Lopardo, B. Szymkow, B. Wagner, E. Bluemke, J.-M. Nounahon, J. Passerat-Palmbach, K. Prakash, N. Rose *et al.*, "Pysyft: A library for easy federated learning," *Federated Learning Systems: Towards Next-Generation AI*, pp. 111–139, 2021.

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

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

[42] G. A. Reina, A. Gruzdev, P. Foley, O. Perepelkina, M. Sharma, I. Davidyuk, I. Trushkin, M. Radionov, A. Mokrov, D. Agapov, J. Martin, B. Edwards, M. J. Sheller, S. Pati, P. N. Moorthy, H. S. Wang, P. Shah, and S. Bakas, "Openfl: An open-source framework for federated learning," *CoRR*, vol. abs/2105.06413, 2021.

[43] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. J. Ong, J. Radhakrishnan, A. Verma, M. Sinn, M. Purcell, A. Rawat, T. N. Minh, N. Holohan, S. Chakraborty, S. Witherspoon, D. Steuer, L. Wynter, H. Hassan, S. Laguna, M. Yurochkin, M. Agarwal, E. Chuba, and A. Abay, "IBM federated learning: an enterprise framework white paper V0.1," *CoRR*, vol. abs/2007.10987, 2020.

[44] D. Zeng, S. Liang, X. Hu, and Z. Xu, "Fedlab: A flexible federated learning framework," *CoRR*, vol. abs/2107.11621, 2021.

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

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

[47] B. Liu, C. Tan, J. Wang, T. Zeng, H. Shan, H. Yao, H. Huang, P. Dai, L. Bo, and Y. Chen, "Fedlearn-algo: A flexible open-source privacy-preserving machine learning platform," *CoRR*, vol. abs/2107.04129, 2021.

[48] J. H. Ro, A. T. Suresh, and K. Wu, "FedJAX: Federated learning simulation with JAX," *arXiv preprint arXiv:2108.02117*, 2021.

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

[50] D. Romanini, A. J. Hall, P. Papadopoulos, T. Titcombe, A. Ismail, T. Cebere, R. Sandmann, R. Roehm, and M. A. Hoeh, "Pyvertical: A vertical federated learning framework for multi-headed splitnn," *CoRR*, vol. abs/2104.00489, 2021.

[51] G. Kaissis, A. Ziller, J. Passerat-Palmbach, T. Ryffel, D. Usynin, A. Trask, I. Lima, J. Mancuso, F. Jungmann, M. Steinborn, A. Saleh, M. R. Makowski, D. Rueckert, and R. Braren, "End-to-end privacy preserving deep learning on multi-institutional medical imaging," *Nat. Mach. Intell.*, vol. 3, no. 6, pp. 473–484, 2021.

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

[53] M. Ekmefjord, A. Ait-Mlouk, S. Alawadi, M. Åkesson, P. Singh, O. Spjuth, S. Toor, and A. Hellander, "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*. IEEE, 2022, pp. 555–564.

[54] S. S. S. R., A. Altmann, B. Gutman, and M. Lorenzi, "Fed-biomed: 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*, ser. Lecture Notes in Computer Science, 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. Springer, 2020, pp. 201–210.

[55] M. Ryu, Y. Kim, K. Kim, and R. K. Madduri, "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*. IEEE, 2022, pp. 1074–1083.

[56] R. Nasirigerdeh, R. Torkzadehmahani, J. O. Matschinske, J. Baumbach, D. Rueckert, and G. Kaissis, "Hyfed: A hybrid federated framework for privacy-preserving machine learning," *CoRR*, vol. abs/2105.10545, 2021.

[57] Y. Feng, M. Yu, W. Xiong, X. Guo, J. Huang, S. Chang, M. Campbell, M. A. Greenspan, and X. Zhu, "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*, L. Antonie and P. M. Zadeh, Eds. Canadian Artificial Intelligence Association, 2021.

[58] H. Cao, Y. Zhang, J. Baumbach, P. Burton, D. Dwyer, N. Koutsouleris, J. Matschinske, Y. Marcon, S. Rajan, T. Rieg, P. Ryser-Welch, J. Späth, E. Schwarz, D. Alnæs, O. Andreasssen, J. Chen, F. Degenhardt, D. Doncevic, R. Eils, and A. Meyer-Lindenberg, "dsmtl - a computational framework for privacy-preserving, distributed multi-task machine learning," *Bioinformatics*, vol. 38, 09 2022.

[59] P. Niedziela, A. Danilenka, D. Kolasa, M. Ganzha, M. Paprzycki, and K. Nalinaksh, "Sunday-fl–developing open source platform for federated learning," in *2021 Emerging Trends in Industry 4.0 (ETI 4.0)*. IEEE, 2021, pp. 1–6.

[60] F. Morsbach and S. Toor, "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*, K. Gai and K. R. Choo, Eds. ACM, 2021, pp. 61–70.

[61] C. Schneebeli, S. Kalloori, and S. Klingler, "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*, L. Lewin-Eytan, D. Carmel, E. Yom-Tov, E. Agichtein, and E. Gabrilovich, Eds. ACM, 2021, pp. 910–913.

[62] A. Moncada-Torres, F. Martin, M. Sieswerda, J. van Soest, and G. Geleijnse, "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*. AMIA, 2020.

[63] N. R. Barroso, G. Stipcich, D. Jiménez-López, J. A. Ruiz-Millán, E. Martínez-Cámara, G. González-Seco, M. V. Luzón, M. A. Veganzones, and F. Herrera, "Federated learning and differential privacy: Software tools analysis, the sherpa.ai FL framework and methodological guidelines for preserving data privacy," *Inf. Fusion*, vol. 64, pp. 270–292, 2020.

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

[65] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, "Chiron: Privacy-preserving machine learning as a service," *CoRR*, vol. abs/1803.05961, 2018. [Online]. Available: http://arxiv.org/abs/1803.05961

[66] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, 2020.

[67] Y. Yu, M. Zeng, Z. Qiu, L. Luo, and H. Chen, "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*. IEEE, 2020, pp. 968–974.

[68] S. Carpov, N. Gama, M. Georgieva, and D. Jetchev, "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*, 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, 2022, pp. 532–542.

[69] Z. Zhao, R. Birke, A. Kunar, and L. Y. Chen, "Fed-tgan: Federated learning framework for synthesizing tabular data," *CoRR*, vol. abs/2108.07927, 2021.

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

[71] C. Pappas, D. Chatzopoulos, S. Lalis, and M. Vavalis, "IPLS: A framework for decentralized federated learning," in *IFIP Networking Conference, IFIP Networking 2021, Espoo and Helsinki, Finland, June 21-24, 2021*, Z. Yan, G. Tyson, and D. Koutsonikolas, Eds. IEEE, 2021, pp. 1–6.

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

[73] Z. Jiang, W. Wang, B. Li, and B. Li, "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*, A. Gavrilovska, D. Altinbüken, and C. Binnig, Eds. ACM, 2022, pp. 370–385.