A Comprehensive Comparison of Federated Learning Frameworks

Jonas Boekhoff[1[0009-0008-3374-0810]], Jan-Philipp Awick[1[0009-0008-1031-2119]], Lars Steffens[2], Michael Karl[2], and Jorge Marx Gómez[1[0000-0002-7833-7549]]

[1] Carl von Ossietzky Universität Oldenburg, Dept. of Business Informatics, Germany
[2] German Aerospace Center (DLR), Institute for AI Safety and Security, Sankt Augustin, Germany

Abstract. Federated Learning is a Machine Learning paradigm that enables institutions to train models collaboratively on decentralized data without exchanging it. Training is performed locally on each participant's device, and only model updates are shared and aggregated, rather than raw data. The selection of a suitable framework is a critical step in designing a Federated Learning System, as this choice significantly influences the capabilities and limitations of the system. While previous research has primarily focused on comparing Federated Learning functionalities, other important criteria, such as usability, technical performance, and legal aspects have often been addressed only partially or overlooked entirely. To address this gap and support informed framework selection, we conducted a comprehensive comparison of Federated Learning frameworks. Frameworks were identified through a structured literature review and an analysis of GitHub repositories. The comparison covers five relevant comparison criteria: (1) Federated Learning functionalities, (2) user-friendliness, (3) technical aspects, (4) legal aspects, and (5) performance evaluation.

Keywords: Federated Learning · Framework Comparison · Benchmark.

1 Introduction

The growing interest in automating business processes has led to the widespread adoption of Artificial Intelligence (AI) solutions by companies aiming to improve efficiency and optimize the use of human resources [7]. However, training Machine Learning (ML) models requires large amounts of high-quality training data to achieve robust model performance. The data volumes necessary for training are often either unavailable within a single organization or of insufficient quality [39]. The exchange of data between organizations remains a challenge, and even intra-organizational data sharing is frequently impeded by competitive thinking, data protection concerns, and administrative complexities [39]. A promising approach to overcome these challenges and to enable model training without exchanging data is Federated Learning (FL) [32]. The general idea is to train a model collaboratively with multiple devices or organizations while keeping the data localized. Therefore, each participant trains a model with local data, and only the model parameters of the participants are used for aggregation of a global model. The process is repeated iteratively until the desired model performance is achieved [2,20]. This enables collaborative training without exposing sensitive information, thereby ensuring privacy protection. To implement a Federated Learning System (FLS), organizations can utilize frameworks to streamline the development by enhancing code reusability and increasing productivity, leading to shorter development time and high quality applications [27]. Although numerous FL frameworks are currently available, they vary considerably in terms of functionality, performance, and usability [16]. Without systematic and up-to-date framework comparisons, developers are often required to conduct time-consuming manual evaluations of the available options. This challenge is further intensified by the growing number of existing frameworks and functional diversity. A comprehensive comparison of FL frameworks can therefore offer valuable insights into their capabilities and support developers in selecting an appropriate framework for their specific use case. Therefore, this paper presents a current and comprehensive comparison of FL frameworks and extends previous comparative studies, e.g. [10], [18], and [28]. The key contributions of this paper are:

- Identification and description of 29 existing FL frameworks
- Comparative analysis of five prominent FL frameworks based on a comprehensive criteria catalog, including FL-functionalities, user-friendliness, technical aspects, and legal aspects
- Technical performance comparison of the selected FL frameworks using a benchmark dataset
- Support for informed selection of suitable FL frameworks, based on the findings of this paper

This paper is structured as follows: Section 2 presents related research based on a structured literature review. Section 3 provides an overview of currently available frameworks. Section 4 details the framework selection process, the definition of the criteria catalog, and the corresponding results. The technical performance evaluation is described in Section 5. Section 6 covers a summary and discussion of the findings, and Section 7 gives a conclusion and outlines possible directions for future work.

2 Related Work

This section describes the identification and analysis of related work. The methodology of the conducted literature review is described in Section 2.1 and the related studies are discussed in Section 2.2.

2.1 Literature Review Methodology

In order to identify related studies and existing FL frameworks in the literature, a structured literature review was conducted according to the approach outlined in

[36]. The methodology followed a three-step process: *(1) identification of relevant articles*, *(2) backward search* within the identified studies and *(3) forward search*. An overview of the search strategy and the results is illustrated in Figure 1.
To identify potentially relevant studies, the following literature databases were used: ACM Digital Library, arXiv, and IEEE Xplore. The search terms shown in Figure 1 had to be included in the title of the scientific papers.

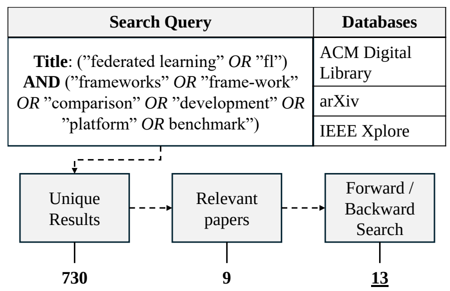

Fig. 1. Search strategy and filtering of identified literature.

The search strategy resulted in 730 unique papers. These were subjected to a sequential filtering process based on predefined inclusion criteria: first, a title and abstract review, followed by a full-text analysis of the remaining candidates. This process yielded nine relevant papers that met the requirement of systematically analyzing and comparing at least three FL frameworks in a minimum of three characteristics. The subsequent forward and backward search of the nine relevant papers resulted in a final corpus of 13 relevant studies.

2.2 Overview of Existing Studies

Related research covers studies that perform a comparison of FL frameworks. The main difference in existing literature is the coverage of comparative aspects. To systematically compare existing related studies, we consider six different comparison aspects: *(1) functional features*, *(2) user-friendliness*, *(3) technical aspects*, *(4) legal aspects*, *(5) performance evaluation*, and *(6) publication year*.
The concept matrix, covering 13 studies, is presented in Table 1. The reviewed studies were published between 2020 and 2024, highlighting the need for up-to-date comparisons to include the evolution of FL frameworks. While most studies address functional features, user-friendliness, and technical aspects, legal aspects and performance evaluations are often not included. The only study that covers all six comparison aspects is [18]. However, as the study was published in 2021, its findings may no longer reflect the current state. Consequently, this paper fills this gap by providing an up-to-date framework comparison that includes all categories.

Table 1. Concept-matrix of related work compared to this paper.

Source	**Functional features**	**User-friendliness**	**Technical aspects**	**Legal aspects**	**Performance evaluation**	**Year**
[29]	✔	✔	✖	✖	✖	2020
[18]	✔	✔	✔	✔	✔	2021
[5]	✔	✖	✔	✖	✖	2022
[28]	✔	✔	✔	✖	✖	2023
[16]	✔	✖	✖	✖	✖	2023
[37]	✔	✖	✖	✖	✖	2023
[22]	✔	✔	✔	✖	✔	2023
[21]	✔	✔	✔	✖	✖	2023
[17]	✔	✖	✖	✖	✖	2023
[12]	✔	✔	✔	✖	✖	2023
[24]	✔	✔	✖	✖	✖	2024
[3]	✔	✖	✖	✖	✖	2024
[10]	✖	✖	✖	✖	✔	2024
This paper	✔	✔	✔	✔	✔	2025

3 Available Federated Learning Frameworks

To identify the currently available FL frameworks, we examined two sources: (1) frameworks referenced in the relevant literature and (2) frameworks published in public GitHub repositories. Note, that frameworks from literature were only considered, if the code is publicly available, e.g. via a repository. The identification of frameworks on GitHub has been conducted by a GitHub search, using the following search string: *"federated learning framework"*. The final list consists of 29 publicly available frameworks. The following overview briefly characterizes the investigated frameworks. The characterization is based on findings from the research literature or, due to a lack of available literature, on information in the associated GitHub repositories and documentations:

APPFL is a modular framework for applying privacy-preserving FL. It allows users to implement their own algorithms, privacy techniques, communication protocols, and neural network structures [31].

EdgeFL is a framework for Decentralized Federated Learning (DFL) that is characterized by a simple implementation with minimal code. It consists of two types of nodes: FL edge nodes for model training and registration nodes for coordinating the participating edge nodes [41].

FederatedScope aims to address heterogeneity in FL scenarios, such as heterogeneity in training data, participant resources, and goals [38].

FedLab is a flexible framework that focuses on communication efficiency and optimization of the FL process. It supports both synchronous and asynchronous FL training [40].

FedML supports various computational paradigms, including on-device training, distributed computing, and single-device simulations [13].

FedStellar focuses on DFL but also enables semi-decentralized and centralized FL. It supports training in a simulation on simulated devices, as well as training on distributed devices [4].

FLEX is designed to simplify scientific experiments with FL. It offers the ability to use different data distributions, privacy parameters, and communication strategies [14].

FLGo includes over 40 implemented benchmarks for simulating FL in various environments and datasets [35].

Flower is designed for experiments with many participants and non-heterogeneous systems. It supports both simulation and execution on different devices. In addition, many different ML frameworks are supported [6].

FS-Real is a FL framework that is focused on the execution on edge devices like Smartphones. The framework offers possibilities like personalization, compression or asynchronous aggregation [11].

GFL focuses on DFL and implements a ring-shaped communication strategy. The authors implemented a new algorithm that is focused on communication efficiency and performance in decentralized FL. The framework is based on the use of blockchain [15].

HyFed primarily aims to provide a privacy-enhancing framework and a generic API for developers to implement their own FL algorithms [25].

IBM Federated Learning aims to enable easy implementation of FL on distributed devices [23].

OpenFed builds on the PyTorch framework and supports various other frameworks such as Hugging Face, MONAI, or MMCV. It supports different FL paradigms like Split Learning, Vertical Federated Learning (VFL), Horizontal Federated Learning (HFL) or DFL [8].

OpenFL aims to strengthen the use of FL in production environments. It is designed to be independent of the use case, industry, and ML framework [26].

Plato can simulate an unlimited number of clients on one or more GPUs. It also offers the possibility to apply FL on distributed devices [19].

PyVertical is designed for VFL using neural networks [30].

VFlair is designed for the use of VFL. It provides several implemented models, datasets, protocols, and modules for evaluating attacks and defenses [43].

XFL enables FL training on multiple devices and implements various privacy techniques such as homomorphic encryption, differential privacy, and secure multi-party computation [34].

AsyncFL supports synchronous, asynchronous, semi-synchronous, and personalized FL training. It offers various modules for implementing custom aggregation algorithms [3].

[3] https://github.com/NUAA-SmartSensing/async-FL

FEDn is a FL framework designed for research purposes as well as for productive use in real environments. The framework is designed to scale with a large number of training participants as well as a large model size[9].

MetisFL emphasizes modularity, extensibility, and adaptability. It is use case independent and designed for real-world use [33].

TensorFlow Federated aims to simplify the conversion of implemented TensorFlow models into federated models [4].

XayNet aims to enable FL on millions of edge devices. The frameworks implements asynchronous aggregation, and offers different interfaces to make use of it on edge devices like smartphones [5].

PySyft aims to enable FL with common ML frameworks. It supports the use of differential privacy, secure multi-party computation, and homomorphic encryption [42].

FATE is designed for use in industrial environments and implements various privacy techniques such as homomorphic encryption and secure multi-party computation [6].

PaddleFL is based on PaddlePaddle and aims to enable the implementation and comparison of different FL algorithms. It supports HFL as well as VFL [7].

p2pfl focuses on DFL using peer-to-peer networks and the gossip protocol. It is the only frameworks that is splitted into a paid and a free version [8].

4 Functional Comparison of Federated Learning Frameworks

This section presents the functional comparison of FL frameworks. It first details the selection of frameworks for comparison (Section 4.1), presents the creation of a criteria catalog used for the comparison, encompassing key aspects like supported FL architectures, communication strategies, and privacy techniques (Section 4.2). In Section 4.3, the results of the comparison are presented, highlighting the strengths and weaknesses of each framework based on the defined criteria.

4.1 Selection of Frameworks for Comparison

Given the large number of available frameworks, only the most capable, up-to-date and widely-used frameworks were considered for comparison. To ensure a systematic selection, the following process was applied:

1. **Actuality**: Frameworks without an update in the last six months (as of June 2024) were excluded, reducing the initial set from 29 to 19 frameworks.

[4] https://github.com/google-parfait/tensorflow-federated
[5] https://github.com/xaynetwork/xaynet
[6] https://github.com/FederatedAI/FATE
[7] https://github.com/PaddlePaddle/PaddleFL
[8] https://github.com/p2pfl/p2pfl

2. **Real-world applicability**: To assess real-world applicability, we examined whether the frameworks were designed exclusively for simulations or supported deployment on distributed devices. Of the 19 frameworks examined, 14 were designed for device deployment.
3. **Popularity**: To assess widespread adoption, we measured the Weighted Total Popularity Score (WTPS) developed by [1]. This metric evaluates popularity based on stars and forks of GitHub repositories and the development of those metrics over time. The WTPS score represents the prevalence of the framework within the developer community. To ensure relevance, only widely used frameworks are taken into account. Frameworks with a WTPS value below 750 were considered insufficiently widespread and thus excluded.

Following this selection process, five frameworks were chosen for further analysis: PySyft, FATE, FedML, Flower and OpenFL.

4.2 Creation of a Criteria Catalog

To define the criteria for comparing FL frameworks, we first examined the core features of FL, such as the FL architecture, and FL paradigms. Additionally, we analyzed related work to identify further relevant comparison criteria. To ensure a comprehensive evaluation, we also included criteria that, while maybe only met by a few frameworks, could provide valuable insights for developers, allowing for a more detailed assessment of their functionalities.
The selected criteria can be categorized into four areas: **(1) FL functionalities**, **(2) user-friendliness**, **(3) technical aspects**, and **(4) legal aspects**. The first category focuses on the core functionalities of FL frameworks, such as architecture, communication strategies, and optimization algorithms. The second category addresses aspects that influence ease of use, deployment, and integration, such as package managers and installation possibilities. The technical aspects consider implementation details like the used communication protocols or the supported programming language. Finally, legal aspects are represented by the type of license under which the framework is distributed. Table 2 presents the final criteria catalog, including a description for each criterion:

Table 2. Comparison criteria catalog for Federated Learning frameworks.

FL functionalities	
FL architecture	Defines whether the framework follows a centralized or decentralized communication scheme.
FL paradigms	Specifies supported FL paradigms, such as Horizontal FL (HFL), Vertical FL (VFL), and Federated Transfer Learning (FTL).
Network topology	Specifies the possible network topologies in decentralized settings, such as star or ring topologies.

Communication strategy	Determines whether synchronous or asynchronous communication is supported.
Optimization algorithms	Lists implemented optimization algorithms for global model aggregation, such as FedAvg.
Data privacy techniques	Lists supported privacy-enhancing technologies like Differential Privacy (DP) or Homomorphic Encryption.
Attack simulation	Indicates whether attacks can be simulated in a FL system.
Blockchain integration	Determines whether blockchain technology is supported for security or transparency.
Simulation capabilities	Specifies whether the frameworks implement a simulation engine.
Installation on edge-devices	Specifies whether the frameworks can be installed on edge devices to use these for distributed settings.
User-friendliness	
Extensibility	Allows easy modification, customization, and extension of the framework.
Examplary implementation	Provides ready-to-use example implementations for quick deployment.
Tutorial availability	Assesses whether learning resources are available.
Technical aspects	
Communication protocols	Specifies the supported communication protocols (e.g., gRPC, MQTT, HTTP).
Supported operating systems	Lists compatible OS environments, such as Windows, Linux, or macOS.
Installation options	Indicates the supported installation options, like package management systems (e.g., pip, conda) that can be used for installation.
Programming language support	Lists programming languages the framework supports for development.
ML framework compatibility	Lists supported ML frameworks, such as TensorFlow, PyTorch, or Scikit-learn.
Legal aspects	
License	Specifies the licensing model (e.g., Apache 2.0).

4.3 Results of Functional Comparison

To evaluate each framework based on the defined criteria catalog, multiple source were considered. The research literature was analyzed as the primary source for all frameworks to identify which features from the criteria catalog were explicitly addressed. If there is no information about a certain criterion, information from GitHub repositories were utilized next, including the respective README files of the frameworks and e.g., GitHub issues were examined. If no information regarding the criterion was found, the documentation of the frameworks was consulted. In the event that no information could be found in the mentioned sources, an explicit search was conducted within the code of the frameworks using the GitHub search. If no mention of a feature was found, it was considered as not supported by the framework.

Table 3. Comparison results of selected Federated Learning frameworks.

	PySyft (v. 0.9)	**FATE (v. 2.1.0)**	**FedML (v. 0.8.5)**	**Flower (v. 1.10.0)**	**OpenFL (v. 1.5.0)**
FL functionalities					
FL architecture	✖	CFL	CFL, DFL	CFL	CFL
FL paradigms	✖	HFL, VFL, FTL	HFL, VFL	HFL, VFL	HFL, VFL, FTL
Network topologies	✖	✖	Decentralized, Hierarchical, Vertical, Split	✖	✖
Communication strategy	✖	synchronous, asynchronous	synchronous, asynchronous	synchronous	synchronous, asynchronous
Optimization algorithms	✖	FedAVG	FedAvg, FedOpt, FedNova, FedGKT, FedProx, FedDyn, FedGan, FedNas, FedSeg	FedAdagrad, FedAdam, FedAvg, FedAvgM, FedMedian, FedOpt, FedProx, FedTrimmedAvg, FedXGB, FedYogi	FedAvg, FedProx, FedOpt, FedCurv
Data privacy techniques	DP, SMPC (SPDZ), HE (CKKS, Paillier)	DP, HE, SMPC (ABY-/SPDZ)	TEE, HE, SMPC	DP, Secure Aggregation	DP, PKI certificates, TEE
Attack simulation	✖	✖	✔	✖	✖

Blockchain integration	✖	✖	✖	✖	✖
Simulation capabilities	✖	✔	✔	✔	✔
Installation on edge-devices	✔	✔	✔	✔	✔
User-friendliness					
Extensibility	✔	✔	✔	✔	✔
Examplary implementation	✖	✔	✔	✔	✔
Tutorial availability	✖	✔	✔	✔	✔
Technical aspects					
Communication protocols	Protocol Buffers, Websockets	gRPC	MQTT+S3, MQTT, PyTorch RPC, gRPC, MPI	gRPC	gRPC
Supported operating systems	Linux, RHEL, MacOS, Windows	CentOS and other Linux distributions	Ubuntu, CentOS, Android, MacOS, Windows	Linux, Windows, macOS, Android, iOS	Linux
Installation options	Pip, Conda, Docker, Podman, Kubernetes	Pip, Conda, Docker, Installation Packages, Ansible	Pip, Conda, Docker, Kubernetes	Pip, Conda, Mamba, Docker	Pip, Docker
Programming language support	Python	Python	Python, Java	Python, Java	Python

ML framework compatibility	PyTorch	PyTorch	PyTorch, TensorFlow, Keras, MXNet, sklearn	PyTorch, TensorFlow, HuggingFace, PyTorch Lightning, scikitlearn, JAX, TFLite, MONAI, fastai, MLX, XGBoost	PyTorch, TensorFlow
Legal aspects					
License	Apache 2.0	Apache 2.0	Apache 2.0	Apache 2.0	Apache 2.0

Table 3 presents an overview of the supported features of each compared framework. A cross (✖) indicates that a criterion is not supported, while a checkmark (✔) indicates support of a criterion. If information regarding a criterion is unavailable or the criterion is not applicable due to missing features, a cross (✖) is used as an indicator.

Regarding the supported FL architectures, only the FedML frameworks supports DFL scenarios through various algorithms and different network structures. Except for Flower and PySyft, each frameworks implements synchronous as well as asynchronous communication. HFL and VFL are supported by all frameworks except PySyft, while FTL is only supported by Fate and OpenFL. The frameworks implement similar data protection techniques but with different compositions. FedML and Flower offer the most implemented optimization algorithms. All of the frameworks can be installed on edge devices. Most frameworks have been developed for Linux. FedML and Flower allow the use of mobile devices, primarily on Android. None of the frameworks supports blockchain FL. Only FedML implements different simulations of attacks on FLS. All of the Frameworks are based on Python and most of them have a rich support for multiple ML frameworks. FedML and Flower stand out for their compatibility. The frameworks are extensible and beginner-friendly, except PySyft. Each framework is licensed under the Apache 2.0 license.

5 Technical Comparison of Federated Learning Frameworks

This section delves into the technical performance of different FL frameworks through a practical experiment. The experimental setup is described in Section 5.1 and the results are discussed in Section 5.2.

5.1 Experimental Setup

In order to establish a basis for a comparison across the frameworks, we defined a standardized FL training configuration. We implemented the training simulation using Python version 3.10 and PyTorch as the underlying ML framework. Note that the PySyft framework is excluded from the technical comparison, as it no longer supports FL functionalities from version 0.5 onwards. The MNIST dataset[9], comprising grayscale images of handwritten digits with corresponding numerical labels, was employed for supervised learning in a classification scenario. This dataset was partitioned equally among 10 simulated clients, each performing 10 local training epochs. The global training process consisted of 20 rounds. A batch size of 32 and a learning rate of 0.001 were used. In each global round, all clients participated in the training process. The hardware used for the simulations consisted of a Intel Core i7-8086K, a NVIDIA GeForce RTX 2080 Ti and 64GB of RAM. We employed a simple convolutional neural network for image classification, consisting of two convolutional layers with ReLU activation and max-pooling, followed by three fully connected layers with ReLU activations.

5.2 Results of Technical Comparison

Figure 2 visualizes the development of accuracy and loss of the evaluated frameworks across 20 training epochs. The OpenFL framework achieves the best performance, demonstrating the highest accuracy and lowest loss. The values for FedML and Flower show a similar trend but the models generally converge more slowly than OpenFL. The framework FATE performs worse, with accuracy and loss stagnating for several epochs before improving, yet remaining significantly below the other frameworks. An analysis of CPU and GPU utilization revealed no hardware-related causes for the inferior performance. The gradual accuracy improvement from round 14 onward suggests that implementation errors are unlikely the primary factor, as this indicates some convergence capability of the model. However, a more detailed investigation of this anomaly should be conducted in future work to better understand its underlying causes. In contrast to the findings of other authors, like [18] or [22], who did not address this specific challenge, FATE consistently delivered results on par with, or exceeding, those of other FL frameworks.

6 Summary & Discussion

The results show that there is a high degree of overlap regarding the supported FL features among the evaluated frameworks. However, for some categories, there are notable distinctions. For instance, the DFL paradigm is exclusively supported by FedML. In addition, the variety of optimization algorithms implemented differs significantly, with only Flower and FedML offering a diverse list

[9] https://yann.lecun.com/exdb/mnist/

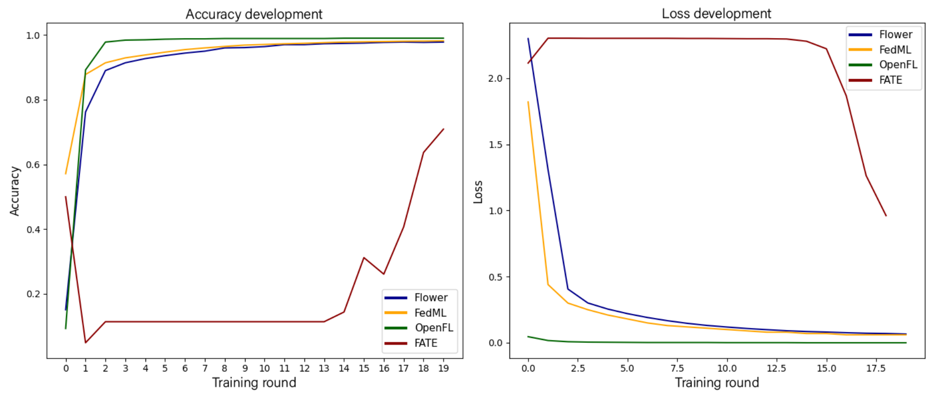

Fig. 2. Progression of model accuracy and loss for the frameworks Flower, FedML, OpenFL, and FATE over 20 training epochs. PySyft was excluded from this performance comparison.

of approaches. Further, FedML is the only framework that supports attack simulations. It is notable, that none of the tested frameworks include a blockchain integration. Regarding user-friendliness, most frameworks, except of PySyft, provide similar levels of support for learning and implementation. All frameworks are licensed under Apache 2.0, allowing free use, code modification, derivative work creation, distribution, and commercial use. However, license terms require preserving copyright notices, documenting modifications, and licensing derivative works under Apache 2.0.

A notable deviation is PySyft, which has shifted focus from FL to remote data science. To the best of our knowledge, FL features are no longer natively supported in versions beyond 0.5. This shift underscores the need for up-to-date framework comparisons, as older studies classified PySyft as an FL framework. To support developers in selecting the most suitable framework for their use case, comprehensive and up-to-date comparisons are essential. Beyond framework comparison, an important limitation in existing comparative studies is the lack of transparency regarding the framework selection criteria. Many related studies identify and analyze a set of FL frameworks without explicitly detailing the selection process. To address this issue, we applied a set of selection criteria for framework selection, ensuring reproducibility and transparency.

7 Conclusion & future work

The selection of a suitable FL framework is a crucial step in implementing an Federated Learning System, as the frameworks vary significantly in their supported functionalities. Depending on the specific use case, different requirements must be met. Given the continuous updates of existing frameworks and the emergence of new ones, the need for up-to-date comparisons arises. To support the

framework selection process, we propose a comprehensive comparison of current FL frameworks, evaluating their FL functionalities, user-friendliness, technical aspects, legal aspects, and technical performance. Additionally, we provide an overview of existing comparative studies, a list of published frameworks and a derived criteria catalog for comparison.
Potential directions for future work include extending the comparison to incorporate additional frameworks and conducting a performance evaluation in distributed environments. As the development of frameworks remains a dynamic field, automatic approaches for framework comparison are a promising direction towards a continuous evaluation.

References

1. Al-Rubaye, A., Sukthankar, G.: Scoring popularity in GitHub. In: 2020 International Conference on Computational Science and Computational Intelligence (CSCI). pp. 217–223
2. Awick, J.P., Schumann, G., Gómez, J.M.: Exploring federated learning for data integration: A structured literature review. In: 2023 International Conference on Big Data, Knowledge and Control Systems Engineering (BdKCSE). pp. 1–8. IEEE (2023)
3. Belal, Y., Mokhtar, S.B., Haddadi, H., Wang, J., Mashhadi, A.: Survey of Federated Learning Models for Spatial-Temporal Mobility Applications
4. Beltrán, E.T.M., Gómez, L.P., Feng, C., Sánchez, P.M.S., Bernal, S.L., Bovet, G., Pérez, M.G., Pérez, G.M., Celdrán, A.H.: Fedstellar: A platform for decentralized federated learning **242**, 122861
5. Beltrán, E.T.M., Pérez, M.Q., Sánchez, P.M.S., Bernal, S.L., Bovet, G., Pérez, M.G., Pérez, G.M., Celdrán, A.H.: Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges
6. Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K.H., Parcollet, T., de Gusmão, P.P.B., Lane, N.D.: Flower: A friendly federated learning research framework
7. Bughin, J.: Artificial intelligence, its corporate use and how it will affect the future of work. In: Capitalism, Global Change and Sustainable Development. pp. 239–260. Springer (2020)
8. Chen, D., Tan, V., Lu, Z., Hu, J.: OpenFed: A Comprehensive and Versatile Open-Source Federated Learning Framework
9. Ekmefjord, M., Ait-Mlouk, A., Alawadi, S., Åkesson, M., Singh, P., Spjuth, O., Toor, S., Hellander, A.: Scalable federated machine learning with FEDn
10. Fonio, S.: Benchmarking Federated Learning Frameworks for Medical Imaging Tasks. In: Foresti, G.L., Fusiello, A., Hancock, E. (eds.) Image Analysis and Processing - ICIAP 2023 Workshops. pp. 223–232. Springer Nature Switzerland
11. Gao, D., Chen, D., Li, Z., Xie, Y., Pan, X., Li, Y., Ding, B., Zhou, J.: FS-Real: A Real-World Cross-Device Federated Learning Platform **16**(12), 4046–4049
12. Guendouzi, B.S., Ouchani, S., EL Assaad, H., EL Zaher, M.: A systematic review of federated learning: Challenges, aggregation methods, and development tools **220**, 103714
13. He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Zhu, X., Wang, J., Shen, L., Zhao, P., Kang, Y., Liu, Y.,

Raskar, R., Yang, Q., Annavaram, M., Avestimehr, S.: FedML: A research library and benchmark for federated machine learning
14. Herrera, F., Jiménez-López, D., Argente-Garrido, A., Rodríguez-Barroso, N., Zuheros, C., Aguilera-Martos, I., Bello, B., García-Márquez, M., Luzón, M.V.: FLEX: FLEXible federated learning framework
15. Hu, Y., Zhou, Y., Xiao, J., Wu, C.: GFL: A Decentralized Federated Learning Framework Based On Blockchain
16. Karimireddy, S.P., Veeraragavan, N.R., Elvatun, S., Nygård, J.F.: Federated Learning Showdown: The Comparative Analysis of Federated Learning Frameworks. In: 2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC). pp. 224–231
17. Khan, A., family=Thij, given=Marijn, p.u., Wilbik, A.: Vertical Federated Learning: A Structured Literature Review
18. Kholod, I., Yanaki, E., Fomichev, D., Shalugin, E., Novikova, E., Filippov, E., Nordlund, M.: Open-Source Federated Learning Frameworks for IoT: A Comparative Review and Analysis **21**(1), 167
19. Li, B., Su, N., Ying, C., Wang, F.: Plato: An Open-Source Research Framework for Production Federated Learning. In: Proceedings of the ACM Turing Award Celebration Conference - China 2023. pp. 1–2. ACM TURC '23, Association for Computing Machinery
20. Li, L., Fan, Y., Tse, M., Lin, K.Y.: A review of applications in federated learning. Computers & Industrial Engineering **149**, 106854 (2020)
21. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., He, B.: A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection **35**(4), 3347–3366
22. Liu, X., Shi, T., Xie, C., Li, Q., Hu, K., Kim, H., Xu, X., Vu-Le, T.A., Huang, Z., Nourian, A., Li, B., Song, D.: UniFed: All-In-One Federated Learning Platform to Unify Open-Source Frameworks
23. Ludwig, H., Baracaldo, N., Thomas, G., Zhou, Y., Anwar, A., Rajamoni, S., Ong, Y., Radhakrishnan, J., Verma, A., Sinn, M., Purcell, M., Rawat, A., Minh, T., Holohan, N., Chakraborty, S., Whitherspoon, S., Steuer, D., Wynter, L., Hassan, H., Laguna, S., Yurochkin, M., Agarwal, M., Chuba, E., Abay, A.: IBM Federated Learning: An Enterprise Framework White Paper V0.1
24. Luzón, M.V., Rodríguez-Barroso, N., Argente-Garrido, A., Jiménez-López, D., Moyano, J.M., Del Ser, J., Ding, W., Herrera, F.: A Tutorial on Federated Learning from Theory to Practice: Foundations, Software Frameworks, Exemplary Use Cases, and Selected Trends **11**(4), 824–850
25. Nasirigerdeh, R., Torkzadehmahani, R., Matschinske, J., Baumbach, J., Rueckert, D., Kaissis, G.: HyFed: A Hybrid Federated Framework for Privacy-preserving Machine Learning
26. 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, S.h., Shah, P., Bakas, S.: OpenFL: An open-source framework for Federated Learning **67**(21), 214001
27. Riehle, D., Gross, T.: Role model based framework design and integration pp. 117–133
28. Riviera, W., Galazzo, I.B., Menegaz, G.: FeLebrities: A User-Centric Assessment of Federated Learning Frameworks **11**, 96865–96878
29. Rodríguez-Barroso, N., Stipcich, G., Jiménez-López, D., Ruiz-Millán, J.A., Martínez-Cámara, E., González-Seco, G., Luzón, M.V., Veganzones, M., Herrera, F.: Federated Learning and Differential Privacy: Software tools analysis, the

Sherpa.ai FL framework and methodological guidelines for preserving data privacy **64**, 270–292
30. Romanini, D., Hall, A.J., Papadopoulos, P., Titcombe, T., Ismail, A., Cebere, T., Sandmann, R., Roehm, R., Hoeh, M.A.: PyVertical: A Vertical Federated Learning Framework for Multi-headed SplitNN
31. Ryu, M., Kim, Y., Kim, K., Madduri, R.K.: APPFL: Open-source software framework for privacy-preserving federated learning
32. Schumann, G., Awick, J.P., Gómez, J.M.: Natural language processing using federated learning: a structured literature review. In: 2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings). pp. 1–7. IEEE (2023)
33. Stripelis, D., Anastasiou, C., Toral, P., Asghar, A., Ambite, J.L.: MetisFL: An Embarrassingly Parallelized Controller for Scalable & Efficient Federated Learning Workflows
34. Wang, H., Zhou, Y., Zhang, C., Peng, C., Huang, M., Liu, Y., Zhang, L.: XFL: A High Performace, Lightweighted Federated Learning Framework
35. Wang, Z., Fan, X., Peng, Z., Li, X., Yang, Z., Feng, M., Yang, Z., Liu, X., Wang, C.: FLGo: A fully customizable federated learning platform
36. Webster, J., Watson, R.: Analyzing the Past to Prepare for the Future: Writing a Literature Review **26**
37. Witt, L., Heyer, M., Toyoda, K., Samek, W., Li, D.: Decentral and Incentivized Federated Learning Frameworks: A Systematic Literature Review **10**(4), 3642–3663
38. Xie, Y., Wang, Z., Gao, D., Chen, D., Yao, L., Kuang, W., Li, Y., Ding, B., Zhou, J.: FederatedScope: A flexible federated learning platform for heterogeneity
39. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated Machine Learning: Concept and Applications **10**(2), 12:1–12:19
40. Zeng, D., Liang, S., Hu, X., Wang, H., Xu, Z.: FedLab: A flexible federated learning framework **24**(100), 1–7
41. Zhang, H., Bosch, J., Olsson, H.H.: EdgeFL: A lightweight decentralized federated learning framework
42. 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., Kaissis, G.: PySyft: A Library for Easy Federated Learning. In: family=Rehman, given=Muhammad Habib, p.u., Gaber, M.M. (eds.) Federated Learning Systems: Towards Next-Generation AI, pp. 111–139. Springer International Publishing
43. Zou, T., Gu, Z., He, Y., Takahashi, H., Liu, Y., Zhang, Y.Q.: VFLAIR: A Research Library and Benchmark for Vertical Federated Learning