A Systematic Comparison of Federated Machine Learning Libraries

Ahmed Saidani

16.01.2023, Master's Thesis Final Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

Outline

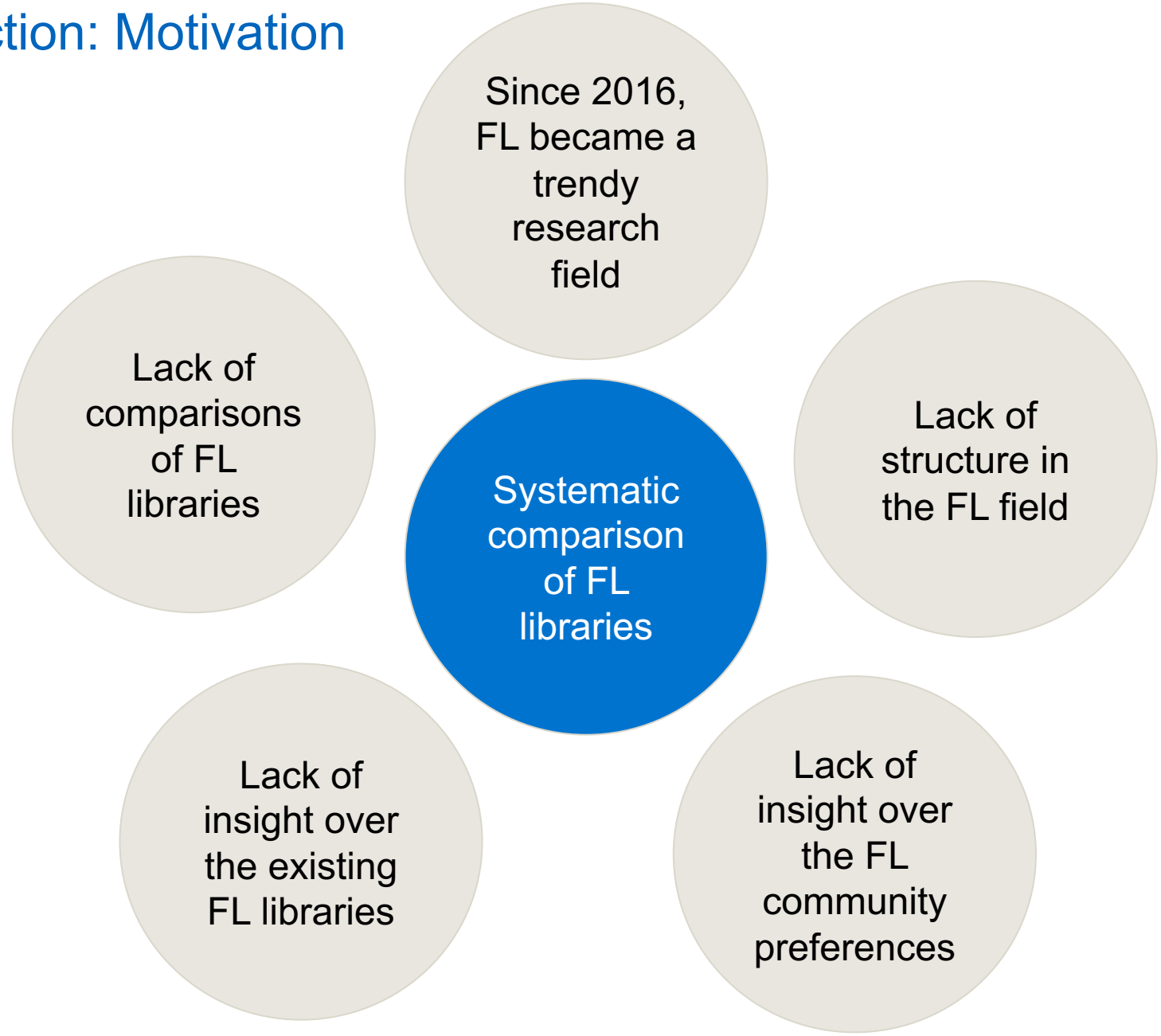
- Introduction

- Research Methodology and Artifacts

- Results
  - Functional and Non-functional Requirements for FL Libraries
  - Federated Learning Libraries
  - Functional Comparison between the FL Libraries
  - The Federated Machine Learning Benchmark
  - Non-functional Comparison between the FL Libraries

- Conclusion

Introduction: Motivation



Introduction: Motivation

1. There are more than 12 FL libraries
2. Each has its own features and functionality. For instance, some support both traditional ML and DL models, while others only support DL models.
3. Each is in its own maturity stage. Some are production-ready, while others are not.
4. Each support different ML frameworks. Some support TF, some support PyTorch, and others support both.
5. They function totally differently. Some logically separate the client from the server logically, while others not.

Introduction: Objective

A Systematic Comparison of Federated Learning Libraries

1. Identify the FR and NFR for FL libraries
2. Research the available FL libraries
3. Compare the features of FL libraries
4. Develop a Benchmark for FL libraries
5. Compare the results of the benchmark

Introduction: Research Questions

RQ1 What are the functional and non-functional requirements relevant for a federated learning library, and what are the most important metrics to benchmark them?

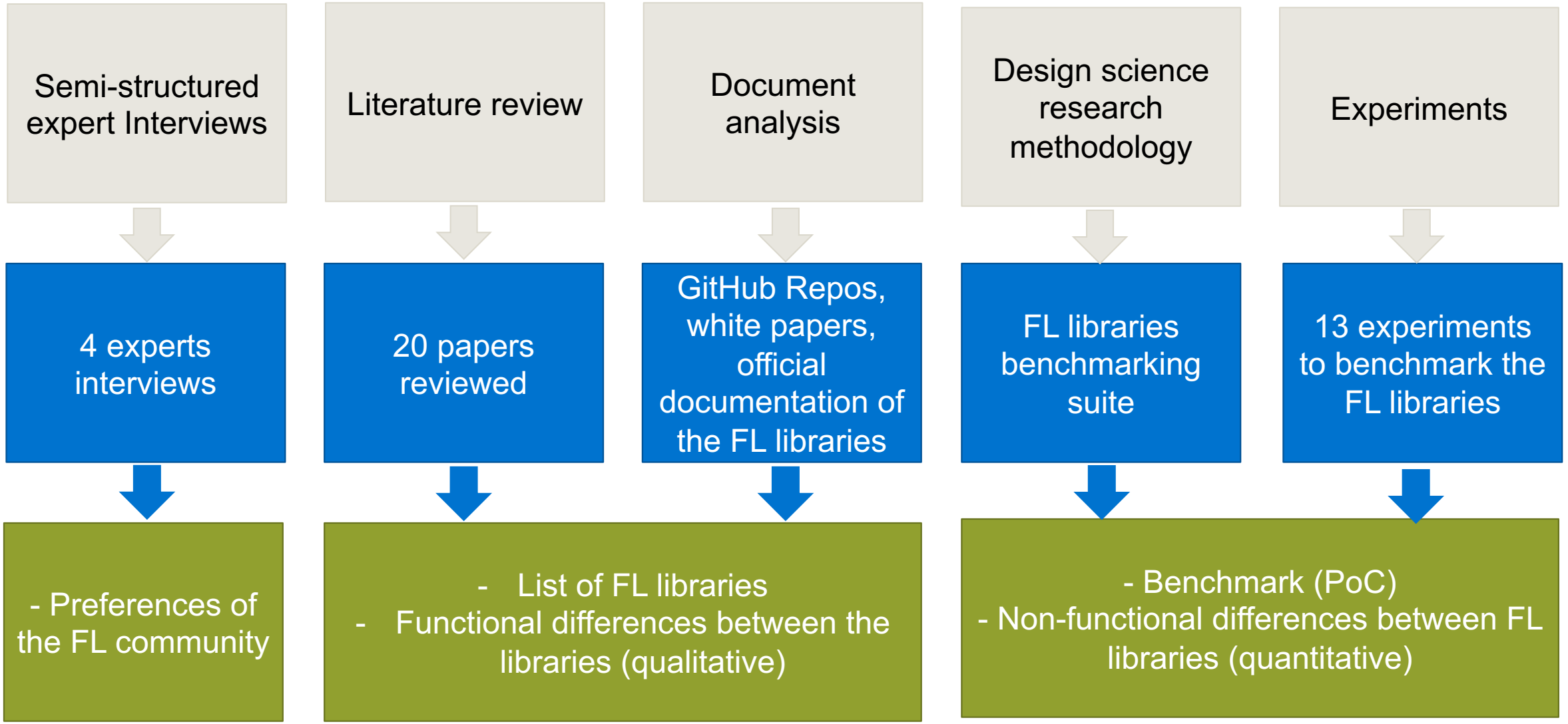
RQ2 What are the different federated learning libraries available, and how do they differ in terms of functionality?

RQ3 How could a modular software application that benchmarks the different federated learning libraries using the metrics be developed?

Outline

- Introduction
- Research Methodology and Artifacts
- Results
  - Functional and Non-functional Requirements for FL Libraries
  - Federated Learning Libraries
  - Functional Comparison between the FL Libraries
  - The Federated Machine Learning Benchmark
  - Non-functional Comparison between the FL Libraries
- Conclusion

Research Methodology & Artifacts



Outline

- Introduction
- Research Methodology and Artifacts
- Results
  - Functional and Non-functional Requirements for FL Libraries
  - Federated Learning Libraries
  - Functional Comparison between the FL Libraries
  - The Federated Machine Learning Benchmark
  - Non-functional Comparison between the FL Libraries
- Conclusion

Results: Functional and Non-functional Requirements for FL Libraries (RQ1)

Semi-structured interviews

Invited 20 experts	5 accepted the invitations	4 showed up to the interviews	Asked each participant 17 questions about FL	Thematically encoded and analysed the data

1. The questions covered: general information about the interviewees and their experience with FL, the functionalities that they use and deem important for FL libraries, the important non-functional requirements for FL libraries and the metrics to measure them.
2. The importance of a functionality was measured through the count of the interviewees that deemed a functionality important and the interviewee gave each NFR an importance factor (1-5). The importance of the non-functional requirements is the sum of all NFR importance factors.
3. At the end the interviewees had the opportunity to add anything they want. They all spoke about the potential of FL and the possible use cases for it, as well as the bottlenecks for its adoption.

Results: Functional and Non-functional Requirements for FL Libraries (RQ1)

Functionality	Feature	Importance
Network topology	Decentralized federated learning	Not so important
	Automatic clients orchestration	Somewhat important
Data partition support	Vertical data	Somewhat important
	Non-i.i.d data	Important
Deployment support	Simulation	Important
	Cross-silo	Important
	Cross-device	Not important
ML models	Traditional ML models	Somewhat important
	Deep learning models	Very important
Security mechanism	Encryption	Very important
	Differential privacy	Somewhat important
Data aggregation algorithm	FedAvg	Very important
	SecBoost	Not important
Other features	C++ support	Not important
	GPU support	Somewhat important
	State management	Not important
	Native tests and benchmarks	Not important

Non-functional quality dimension	Metrics	Importance
Fairness	Variance	very important
Accuracy	hit rate, precision, recall,F1	very important
Scalability	max number of supported clients	very important
Efficiency	RAM, Network, CPU, GPU consumption	important
Performance	Execution time	somewhat important
Usability/Interoperability	ease of ML framework integration, number of compatible ML frameworks	somewhat important
Accountability	% of logged operations	not important
Robustness	% of time the system is running	not important

Outline

- Introduction
- Research Methodology and Artifacts
- Results
  - Functional and Non-functional Requirements for FL Libraries
  - Federated Learning Libraries
  - Functional Comparison between the FL Libraries
  - The Federated Machine Learning Benchmark
  - Non-functional Comparison between the FL Libraries
- Conclusion

Results: Federated Learning Libraries (RQ2)

Library	Pysyft	TFF	FedML	Flower	IBM FL
Contributor	Openmi-nded	Google	FedML Inc.	Adap Gmbh	IBM
ML framework	Pytorch	TF	Pytorch, TF	Pytorch, TF, Libtorch, JAX …	SciLearn, Pytorch, TF, Keras
Environm-ent	Windows, Mac, Linux, Docker	Windows, Mac, Linux, Docker	Windows, Mac, Linux, Docker	Windows, Mac, Linux, Docker	Windows, Mac, Linux, Docker
Number of Github stars	8300	1900	1400	1200	339
Number of Github forks	1800	482	406	316	106
Number of contributors	+250	+90	+50	+50	+10

Results: Federated Learning Libraries (RQ2)

1. **FedLearner:** developed by byteDance. It uses tensorflow as a ML framework.
2. **FATE:** developed by weBank. It has an entire ecosystem (KubeFATE, FATE-Cloud, and FATEBoard…). It is production-ready.
3. **EasyFL:** developed by Smietanka, M., et al. . It is designed to be lightweight and easy to use. It is more suited for learning about federated learning.
4. **Flute:** developed by Microsoft. It offers native benchmarks and tests, and it is more suited for experimentation.
5. **OpenFL:** developed by Intel. It is designed for the IoT usecase.
6. **FedTree:** developed by Li, Q, et al.. It only supports decision trees.
7. **PaddleFL:** developed by Baidu. It uses PaddlePaddle as an ML framework. It is production-ready.

Outline

- Introduction
- Research Methodology and Artifacts
- Results
  - Functional and Non-functional Requirements for FL Libraries
  - Federated Learning Libraries
  - Functional Comparison between the FL Libraries
  - The Federated Machine Learning Benchmark
  - Non-functional Comparison between the FL Libraries
- Conclusion

Results: Functional Comparison between the FL Libraries (RQ2)

Features/Framework		Pysyft	Flower	IBM FL	TFF	FedML
Architecture	**Data Partitioning**	Vertical Horizontal	Vertical Horizontal	Horizontal	Horizontal	Vertical Horizontal
	Datatypes	Numbers, Text, Image, Time-ser-ies	Numbers, Text, Image, Time-ser-ies	Numbers, Text, Image, Time-ser-ies	Numbers, Text, Image, Time-ser-ies	Numbers, Text, Image, Time-ser-ies
	Privacy & Security	HE,MPC, DP	SecAgg	Multiple cryptographi c methods	DP	Secret sharing key agreement,
	Communi-cation scheme	gRPC	gRPC	gRPC	gRPC, Custom Protocol	MPI, MQTT, gRPC…
	FL Strategy	FedAVG, FedSGD	FedAVG,F ed, qffedavg …	FedAVG, FedProx, FedAVG+, …	FedAVG, FedSGD	FedAVG, FedNOV, FedNAS…

Results: Functional Comparison between the FL Libraries (RQ2)

Features/Framework		Pysyft	Flower	IBM FL	TFF	FedML
Engineering	**Customization**	topology, exchange message	exchange message	none	none	topology, exchange message, message flow
	Deployment	single simulation Multi-host (<16 clients) Cross- device (>100 clients)	single simulation Multi-host (<16 clients) Cross- device (>100 clients)	single simulation Multi-host (<16 clients)	single simulation	single simulation Multi-host (<16 clients) Cross- device (>100 clients)
	Documentation	Detailed tutorial, Code Snippets, and API documentation	Detailed tutorial, Code Snippets, and API documentation	API documentation	Detailed tutorial, Code Snippets, and API documentation	Detailed tutorial, Code Snippets
	GPU support	yes	yes	yes	yes	yes
	Native tests & Benchmark	yes	yes	no	yes	yes
FL paradigms	**Vertical FL**	yes	yes	no	no	yes
	FTL	no	yes	no	no	yes
	Simulation	yes	yes	yes	yes	yes
	Cross device	yes	yes	no	no	yes
	Cross silo	yes	yes	yes	no	yes
	Hetero-task learning	no	yes	yes	no	yes
	Decentralized FL	no	no	no	no	yes

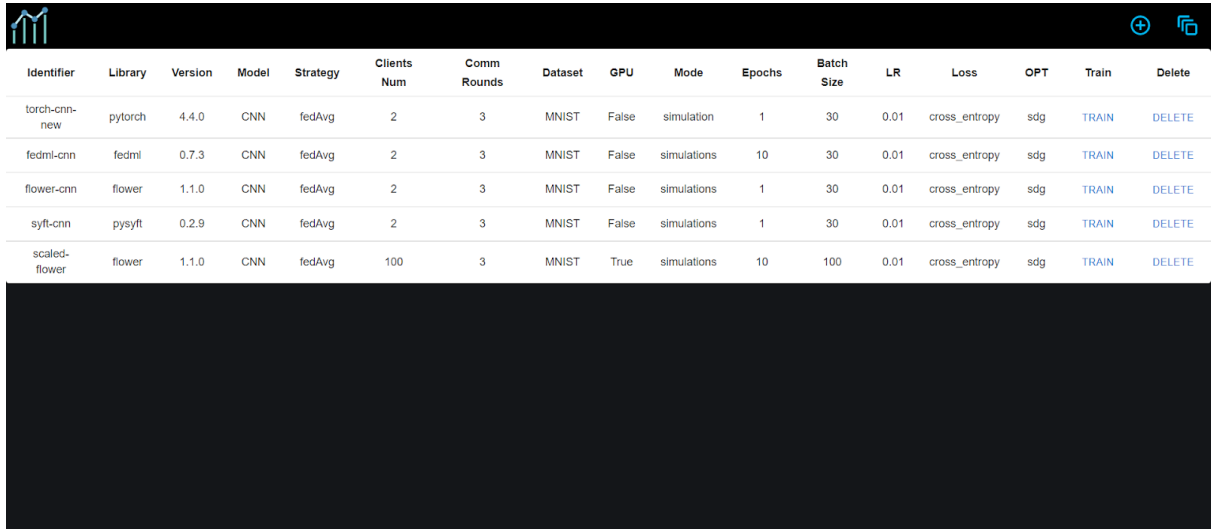
Results: Functional Comparison between the FL Libraries (RQ2)

Features/Framework		Pysyft	Flower	IBM FL	TFF	FedML
ML Models	Regression	yes	yes	yes	no	yes
	Clustering	no	yes	yes	no	no
	Trees	no	no	yes	no	no
	SVM	no	no	yes	no	no
	Bayes networks	no	no	yes	no	no
	NN	yes	yes	yes	yes	yes
	DNN	yes	yes	yes	yes	yes
	CNN	yes	yes	yes	yes	yes
	RNN	yes	yes	yes	yes	yes
Computing paradigms	Distributed computing	yes	no	no	no	yes
	Edge computing	yes	yes	no	yes	yes
	Split learning	yes	no	no	no	yes
	On-device training	yes	yes	no	no	yes

Outline

- Introduction
- Research Methodology and Artifacts
- Results
  - Functional and Non-functional Requirements for FL Libraries
  - Federated Learning Libraries
  - Functional Comparison between the FL Libraries
  - The Federated Machine Learning Benchmark
  - Non-functional Comparison between the FL Libraries
- Conclusion

Results: The Federated Machine Learning Benchmark (RQ3)



Identifier	Library	Version	Model	Strategy	Clients Num	Comm Rounds	Dataset	GPU	Mode	Epochs	Batch Size	LR	Loss	OPT	Train	Delete
torch-cnn-new	pytorch	4.4.0	CNN	fedAvg	2	3	MNIST	False	simulation	1	30	0.01	cross_entropy	sdg	TRAIN	DELETE
fedml-cnn	fedml	0.7.3	CNN	fedAvg	2	3	MNIST	False	simulations	10	30	0.01	cross_entropy	sdg	TRAIN	DELETE
flower-cnn	flower	1.1.0	CNN	fedAvg	2	3	MNIST	False	simulations	1	30	0.01	cross_entropy	sdg	TRAIN	DELETE
syft-cnn	pysyft	0.2.9	CNN	fedAvg	2	3	MNIST	False	simulations	1	30	0.01	cross_entropy	sdg	TRAIN	DELETE
scaled-flower	flower	1.1.0	CNN	fedAvg	100	3	MNIST	True	simulations	10	100	0.01	cross_entropy	sdg	TRAIN	DELETE

- The user can perform CRUD operations on FL settings
- The benchmark includes both CNN and Logistic Regression models on the MNIST and CIFAR-10 datasets

Results: The Federated Machine Learning Benchmark (RQ3)

Library	Version	Environment	Folder	Script
pysyft	0.2.9	syft	pysyft	image_classifier_cnn.py

Model

Model	Federated Strategy	Communication Rounds	GPU	Mode	Epochs	Batches	Learning Rate	Loss	Optimizer
CNN	fedAvg	2	false	simulation	10	30	0.01	cross_entropy	sdg

Dataset

Dataset	Number of Clients	Number of Datapoints	Datasize	Datapoint / Client	Datasize / Client
MNIST	2	70000	30 MB	35000	15 MB

Performance and Efficiency

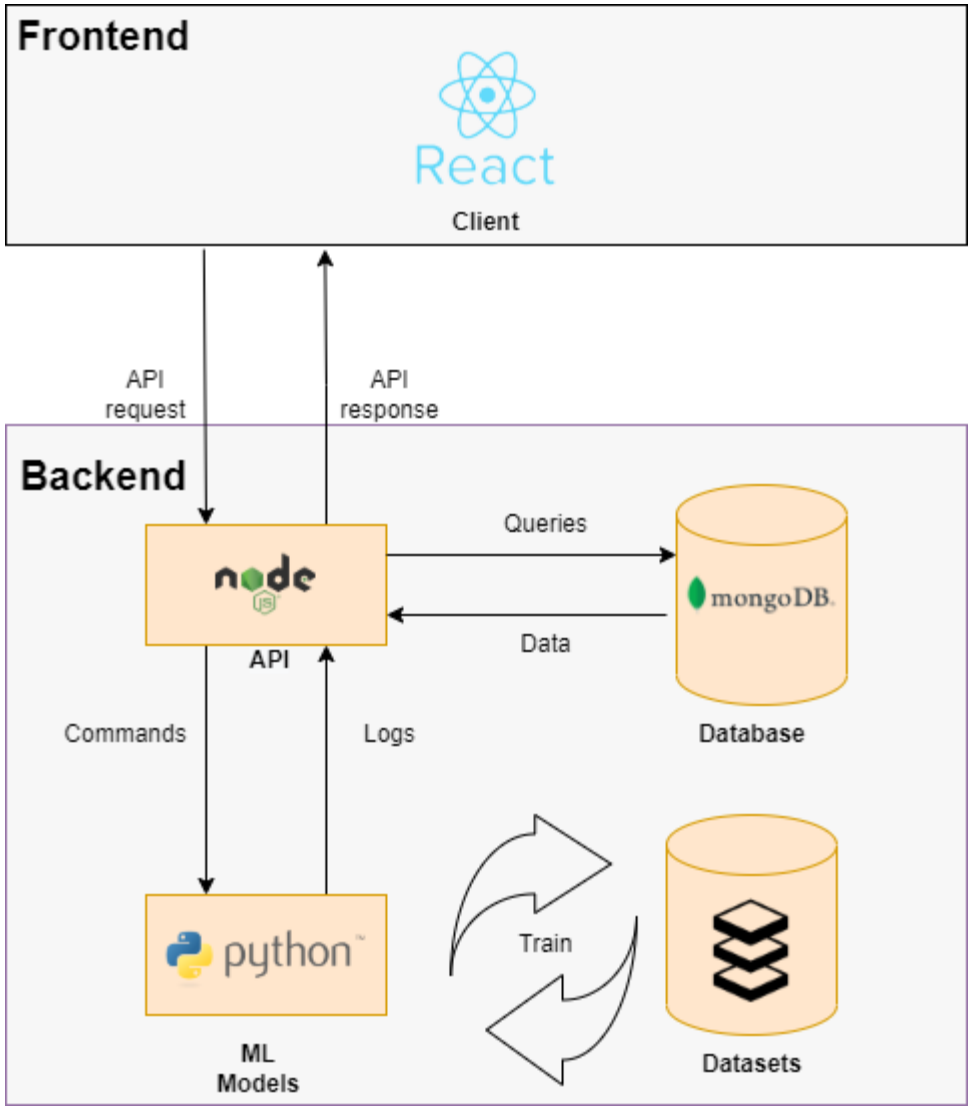
Time of Execution	CPU	GPU	Memory	Network
71.0276 s	99.7 %	0	446.785 MB	21.7841 KB

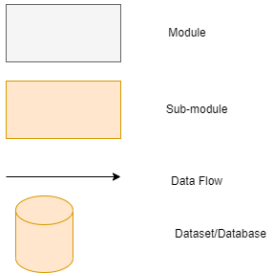
Accuracy

Accuracy	Loss	Precision	Recall	F1
78.95 %	0.00734904	79.63 %	78.46 %	78.11 %

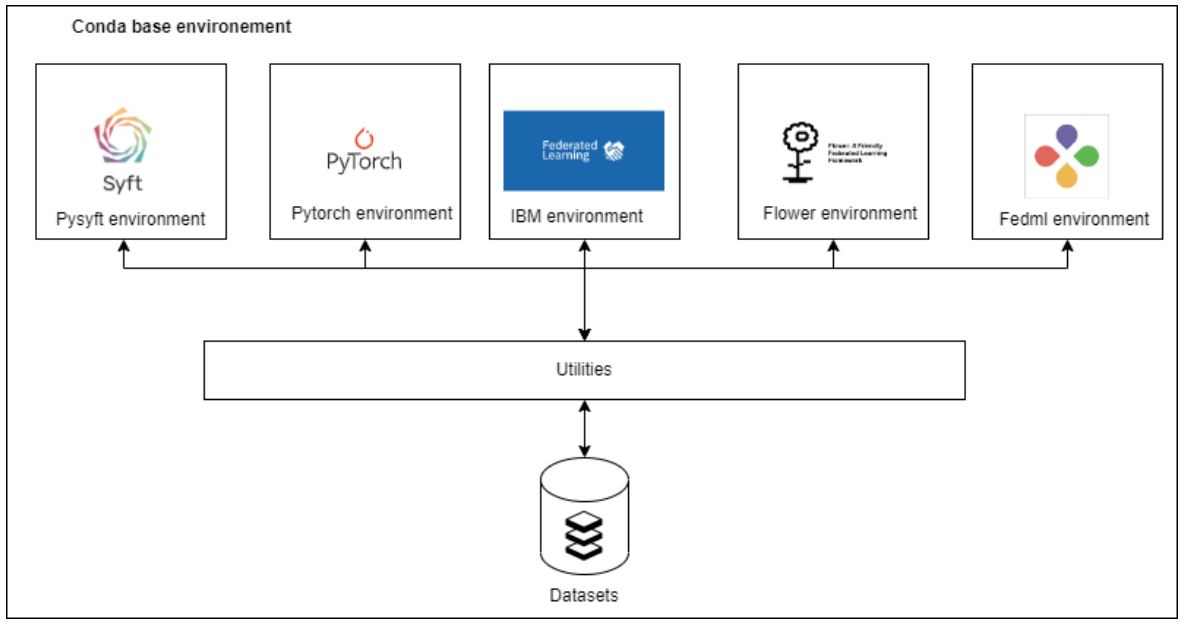
- Training page takes a lot of time to load since the experiments are done in real-time
- For the metrics scraping PSUtil, GPUtil, Times, and Sk-learn-metrics were used

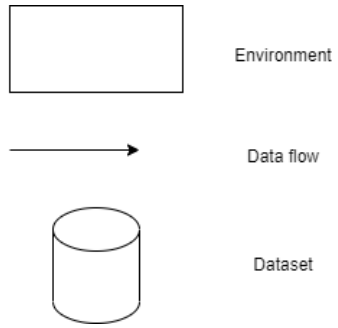
Results: The Federated Machine Learning Benchmark (RQ3)





Results: The Federated Machine Learning Benchmark (RQ3)





Outline

- Introduction
- Research Methodology and Artifacts
- Results
  - Functional and Non-functional Requirements for FL Libraries
  - Federated Learning Libraries
  - Functional Comparison between the FL Libraries
  - The Federated Machine Learning Benchmark
  - Non-functional Comparison between the FL Libraries
- Conclusion

Results: Non-Functional Comparison between the FL Libraries (RQ3)

- Experiment Description:
  - Mode: Simulation
  - Model: CNN
  - Dataset: MNIST
  - Batch size: 100
  - Epochs: 10
  - Learning rate: 0.01
  - Number of communication rounds: 3
  - Optimizer: SDG
  - Loss function: Cross entropy
  - No GPU
- PyTorch Results:
  - Accuracy: 98.69%
  - Precision: 98.69%
  - Recall: 98.67%
  - F1: 98.68%
  - Loss: 0.0004
  - Time of execution: 3 minutes and 46 seconds
  - CPU: 52.0%
  - GPU: 1.0%
  - RAM: 267.91 MB
  - Network: 148.35 KB

Results: Non-functional Comparison between the FL Libraries (RQ3)

2 clients	Pysyft	Fedml	Flower	IBM federated learning
Accuracy	97.48 %	10.51 %	99.03%	99.26%
Precision	97.45 %	N/A	99.02%	99.25%
Recall	97.48 %	N/A	99.02%	99.25%
F1	97.46 %	N/A	99.02%	99.25%
Loss	0.0008	4.810 5	0.0003	0.0003
Time of execution	22m 19s 890ms	1m 13s 670m s	20mins 23s 600ms	37m 21s 50ms
CPU consumption	91.8%	34.5 %	33.9%	99.9%
RAM consumption	604.01 MB	620.9 6MB	1GB 232.97M B	856.94MB
Network consumption	136.21 MB	25.64 MB	465.14M B	1025,68M B

16 clients	Pysyft	Fedml	Flower	IBM federated learning
Accuracy	97.23%	27.43 %	99.31%	99.0%
Precision	97.22%	N/A	99.33%	99.98%
Recall	97.20%	N/A	99.24%	99.99%
F1	97.20%	N/A	99.29%	99.99%
Loss	0.0009	1.95	0.0003	0.0006
Time of execution	22mins 23s	1mins 43s	1hours 27mins 10s	4 hours 18 mins 32s
CPU consumption	99.0%	40.0%	99.3%	99.9%
RAM consumption	858.48 MB	670.1 7MB	4GB 321.37MB	4GB 410.64MB
Network consumption	592.3M B	105.2 3MB	974,94MB	2332.45MB

100 clients	Pysyft	Fedml	Flower	IBM federated learning
Accuracy	96.82%	80.35 %	99.98%	99.22%
Precision	96.8%	N/A	99.98%	99.11%
Recall	96.82%	N/A	99.98%	99.11%
F1	96.81%	N/A	99.98%	99.11%
Loss	0.0009	0.624 9	0.0002	0.0003
Time of execution	23m 46s	4mins 40s	3hours 47mins 3s	25hours 3mins 45s
CPU consumption	97.8%	60.7%	99.4%	99.9%
RAM consumption	861.1M B	749.4 3MB	25.68GB	24.37GB
Network consumption	748.86 MB	543.0 4MB	3345.85M B	3543.04 MB

Results: Non-functional Comparison between the FL Libraries (RQ3)

- FedML is the fastest and is the least resources intensive. However, it comes with an accuracy trade-off. It is more suited for quick experimentations
- Pysyft is fast, and not resources intensive. It is more customizable than FedML. Thus, It is more suited for high-fidelity experiments.
- Flower is highly scalable but resources and time intensive. It has a high accuracy. It can be used in production. The consumption of resources makes it more suitable for a cross-silo use case but according to its documentation It is good in a cross-device settings too.
- IBM federated learning can be used in cross-sillo settings since it comes with the most features and it is the most resources intensive.

Outline

- Introduction
- Research Methodology and Artifacts
- Results
  - Functional and Non-functional Requirements for FL Libraries
  - Federated Learning Libraries
  - Functional Comparison between the FL Libraries
  - The Federated Machine Learning Benchmark
  - Non-functional Comparison between the FL Libraries
- Conclusion

Conclusion: Reflection, Implications, and Future work

Reflection	Implications	Future work
FR and NFR for FL libraries	Guide for FL practionners and researchers	Include more libraries and ML models in the benchmark
FL libraries and their functional and non-functional differences	Overview of the expectations of the FL community	Inspect the differences between the libraries
FL libraries Benchmark	Benchmarking tool for FL libraries	More realistic cross-silo settings

B. Sc.

Ahmed Saidani

Technical University of Munich (TUM)
TUM School of CIT
Department of Computer Science (CS)
Chair of Software Engineering for Business Information Systems (sebis)

Boltzmannstraße 3
85748 Garching bei München

+49.89.289.17132
matthes@in.tum.de
wwwmatthes.in.tum.de

Backup

DSRM

Step	Question	Description
Problem identification and motivation	What is the problem that the artifact is solving?	Qualitatively compare different quality dimensions of the FL libraries using different metrics. The quality dimensions are scalability, performance, efficiency, and accuracy.
Definition of solution objectives	How is the artifact going to solve that problem?	A benchmarking suite that allows multiple experiments to be conducted using different ML models implemented with the different federated learning libraries. It will collect the logs for the different metrics from the libraries and display them on an admin dashboard.
Design and development	How are the solutions going to be implemented?	The benchmarking suite is constituted of multiple modules. Namely, a module for each FL library that has the implementation of the different ML models and FL strategies in it, a module for a web application that communicates with the different libraries modules and acts as an admin panel to configure and conduct the different experiments using the tool, and a module for the different datasets.
Demonstration	What is the efficacy of the solution?	The benchmarking suite needs to present fair, verifiable, and reproducible results.
Contribution	What is the contribution of the solution to the current research?	An easy-to-use benchmarking suite that is modular and extensible.

Interview Questions

Interview Question	RQs
Background of the Interviewee	
1. Please introduce yourself and your role in this company/organization.	N/A
2. Do you consider yourself more of an academic person or an industry-related person?	N/A
3. Total years of experience in the industry and how long have you been in your current position?	N/A
4. Please describe your responsibilities in your organization (e.g., Product owner, developer, Software Architect).	N/A

Interview Question	RQs
Background of the Interviewee	
5. Please describe your experience working with FL.	N/A
6. For what use cases do you use FL?	N/A
7. Which FL libraries do you know?	RQ 1
FR-related questions	
8. What features are the most important for FL libraries?	RQ 1
9. Which aggregation algorithms do you usually use?	RQ 1
10. Which ML models do you usually use?	RQ 1
11. Do you use security mechanisms? if yes, do you prefer encryption-based security or Anonymisation-based security?	RQ 1
12. How often do you work with vertically partitioned data?	RQ 1
13. How often do you work with non-IID data (heterogeneous data)?	RQ 1
NFR-related questions	
14. Do you think NFRs play an important role in the success of FL systems? If yes, how?	RQ 1
15. What non-functional requirements do you think are important for FL systems?	RQ 1
16. Do you measure NFRs over FL-enabled software?	RQ 1
NFR Measurement questions	
17. What are the most important metrics for NFRs in an FL context?	RQ 1
18. How do you capture NFRs and their measurement for FL?	RQ 1
19. What are the challenges you face measuring NFRs for FL?	RQ 1
20. Do you have anything else you would like to add?	N/A

Interviewees Profiles

Participant	Field	Role	Experience	Experience with FL	Responsibility	FL Projects
P1	Industry research	Researcher	3 years	1 year	Research, design, and implement federated learning sector projects	One public sector project
P2	Industry research	Senior researcher	18 years	1 year	Research, design, and implement privacy enhancing technologies	One Edge computing project
P3	FL	AI software developer	3 years	1 year	Develop an FL library	Building an FL library
P4	Industry research	Research assistant (Ph.D.)	4 years	1 year	Research about security & privacy in the aerospace industry	two projects (in robotics and automotive)

Importance Factors for FR and NFR

FOR FR

If c = 4 → The FR is very important

If c = 3 → The FR is important

If c = 2 → The FR is somewhat important

If c = 1 → The FR is not so important

FOR NFR

R-factor = Counts of mention * Average importance

15 < r-factor<20→ The NFR is very important

10 < r-factor<15 → The NFR is important

5 < r-factor<10 → The NFR is somewhat important

1 < r-factor<5→ The NFR is not so important

R-FACTOR OF NFR

Fairness (R-factor=17

Accuracy (R-factor=15)

Scalability (R-factor=15)

Efficiency (R-factor=14)

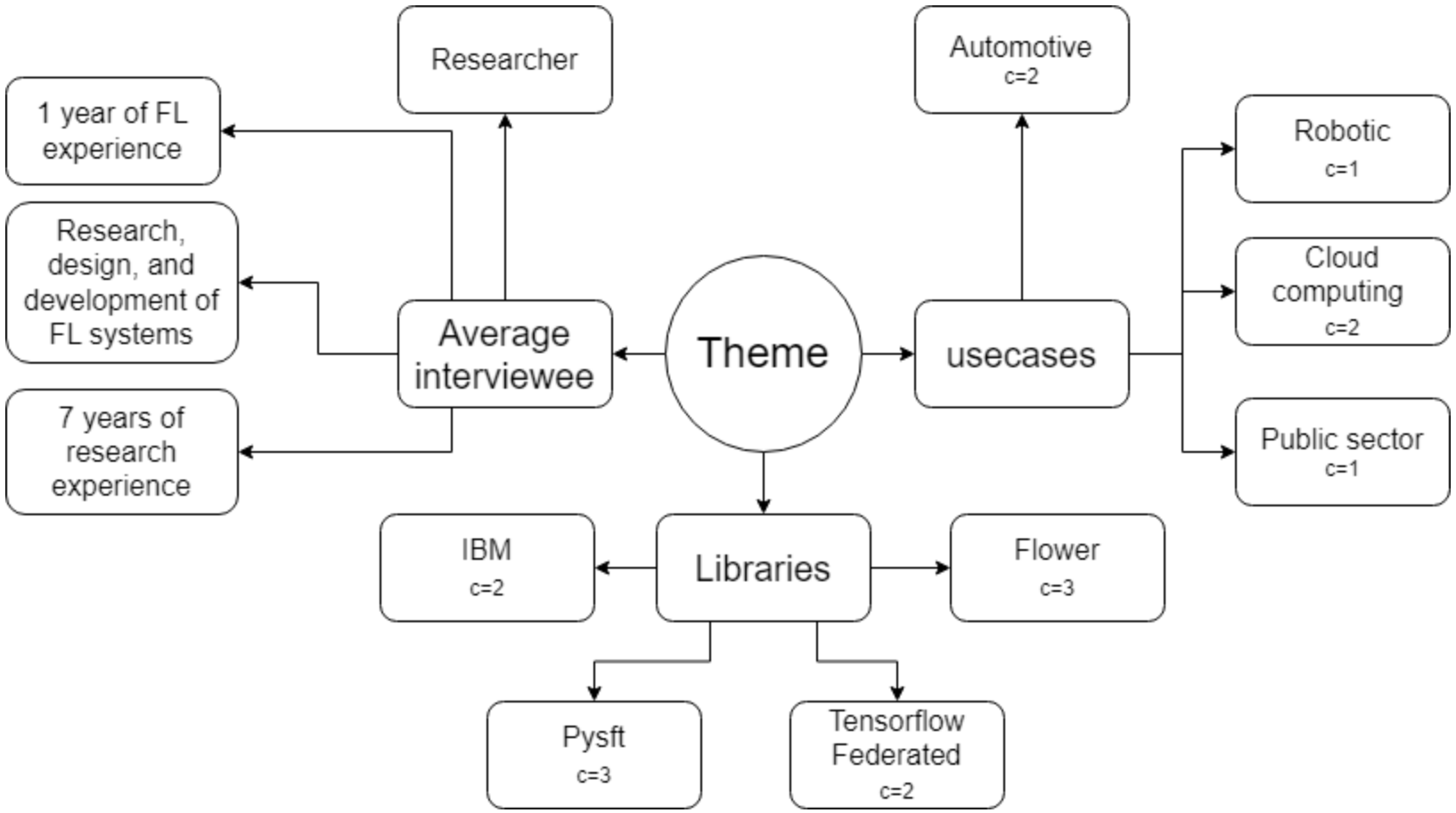
Performance (R-factor=7)

Interoperability/Usability (R-factor=6)

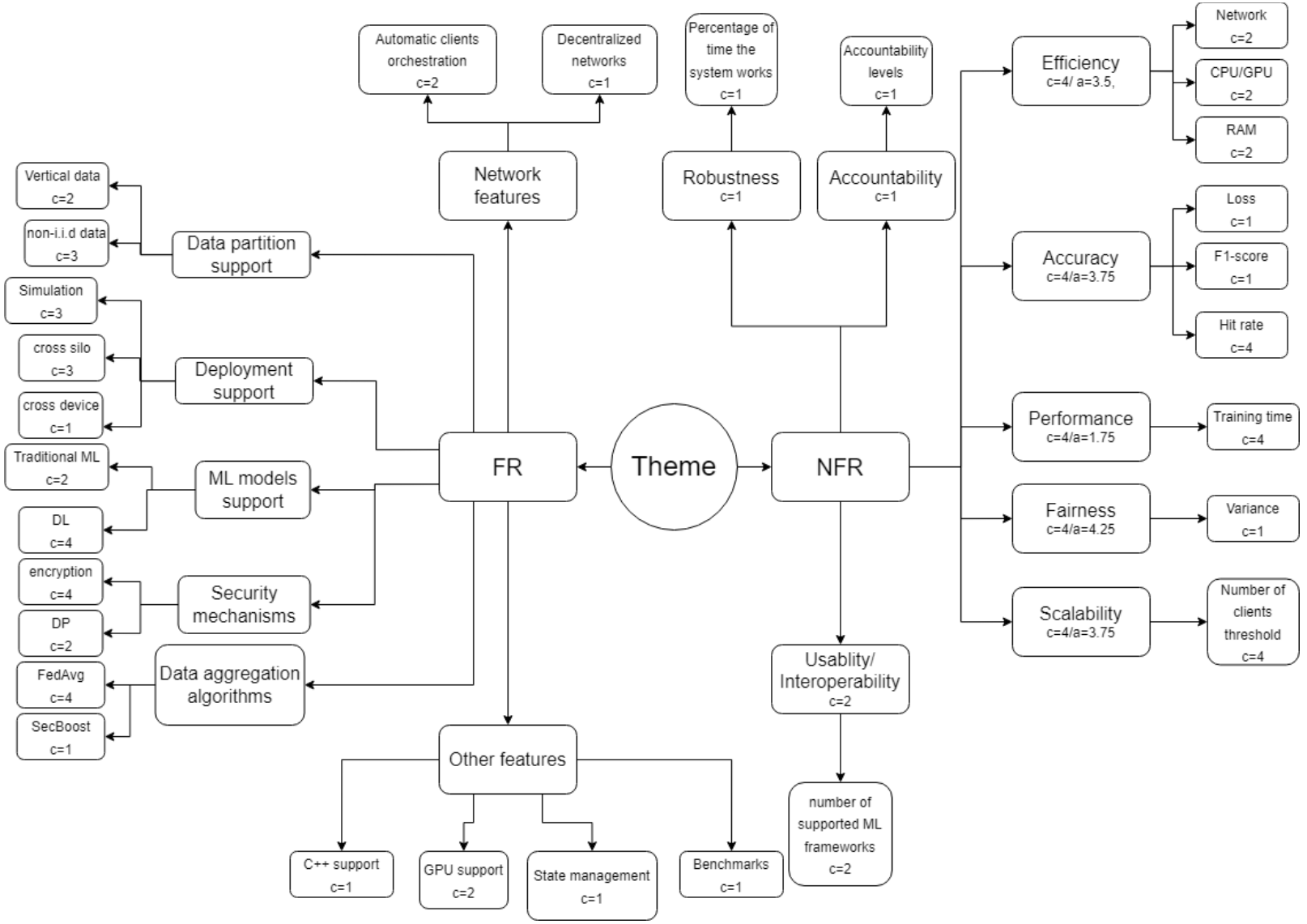
Accountability (R-factor=5)

Robustness (R-factor=4)

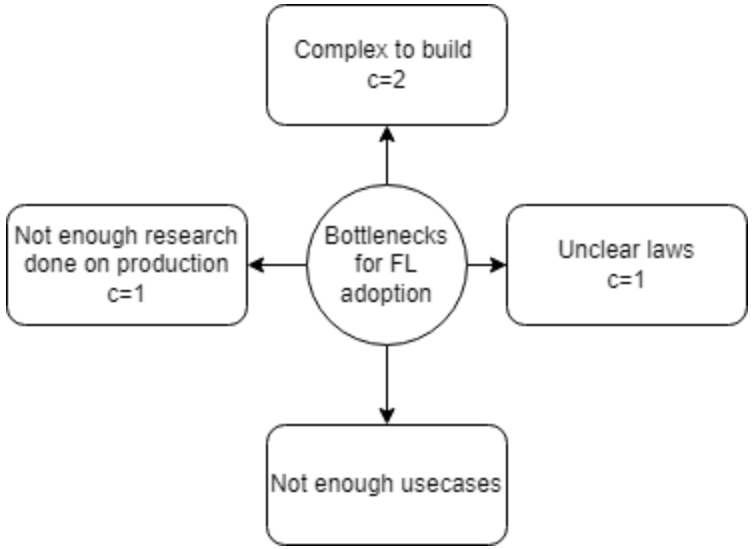
Thematic Encoding: Meta-information about the Interviews



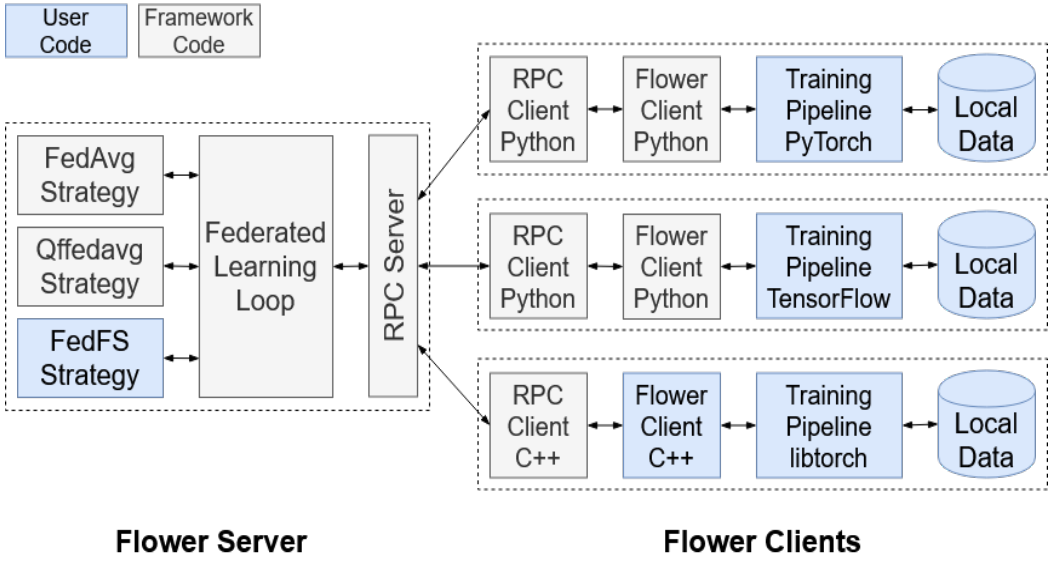
Thematic Encoding: FR and NFR for FL libraries



Thematic Encoding: FL Bottlenecks



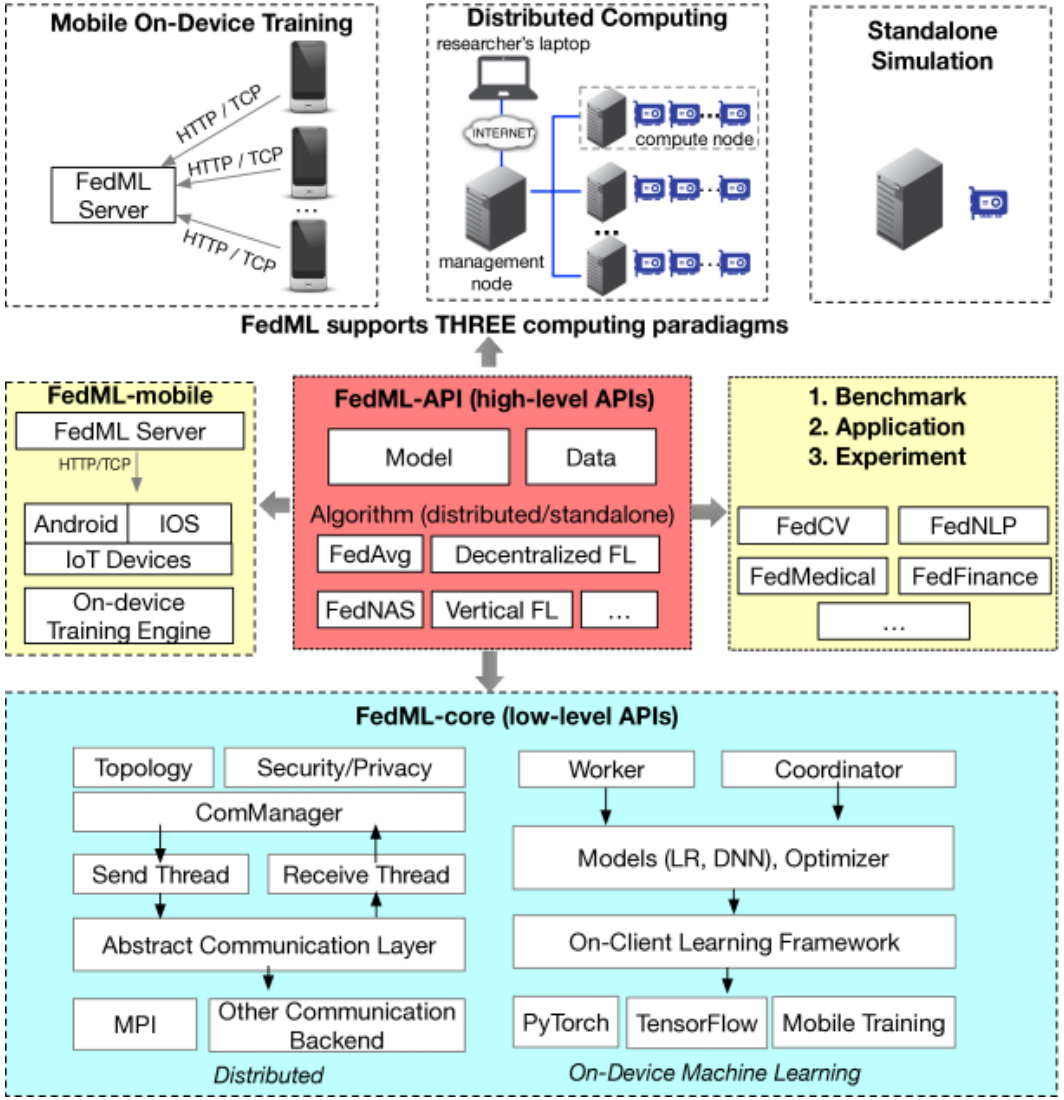
Results: Federated Learning libraries (Flower)



- It has 1400 stars on GitHub.
- 50 Contributors.
- It supports a wide variety of ML models and ML frameworks.
- It has a logical separation between the client and the server.
- It is highly scalable.
- Built for customization

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., ... & Lane, N. D. (2022). Flower: A friendly federated learning framework.

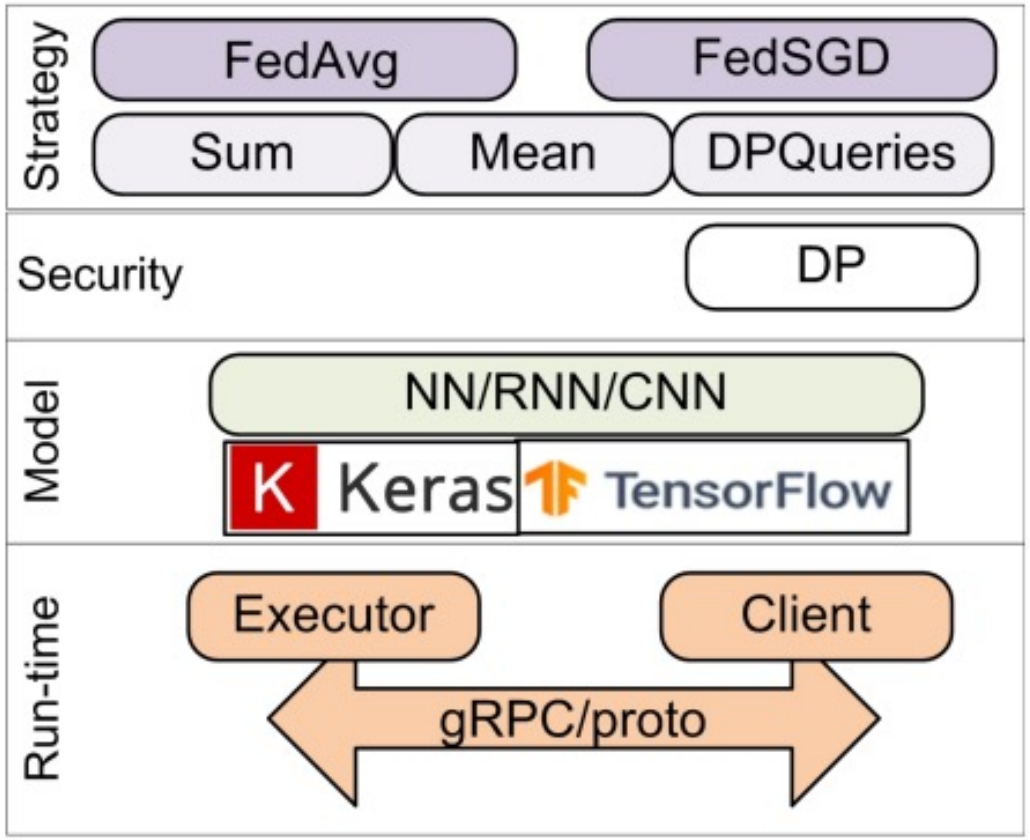
Results: Federated Learning libraries (FedML)



- It has 1400 stars on GitHub.
- 50 Contributors.
- It has an entire ecosystem (Parrot, Octopus, Cheetah).
- It has built-in models and datasets.
- It has its own built-in benchmark.
- It supports many ML frameworks, communication protocols, and FL paradigms.

He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., ... & Avestimehr, S. (2020). Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518.

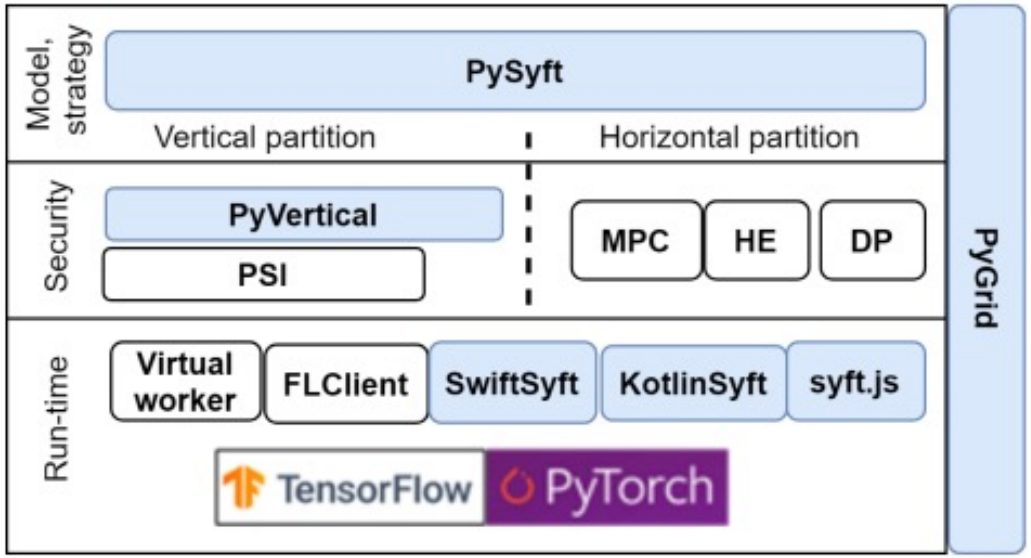
Results: Federated Learning libraries (TFF)



- It has 1900 stars on GitHub.
- 90 Contributors.
- It has native built-in differential privacy functions(Sum, Mean, DPQueries).
- It can run in simulation or cross-silo.
- It can only be used to train deep learning models.

Tensorflow Federated documentation page, https://www.tensorflow.org/federated , last accessed 03.09.2022.

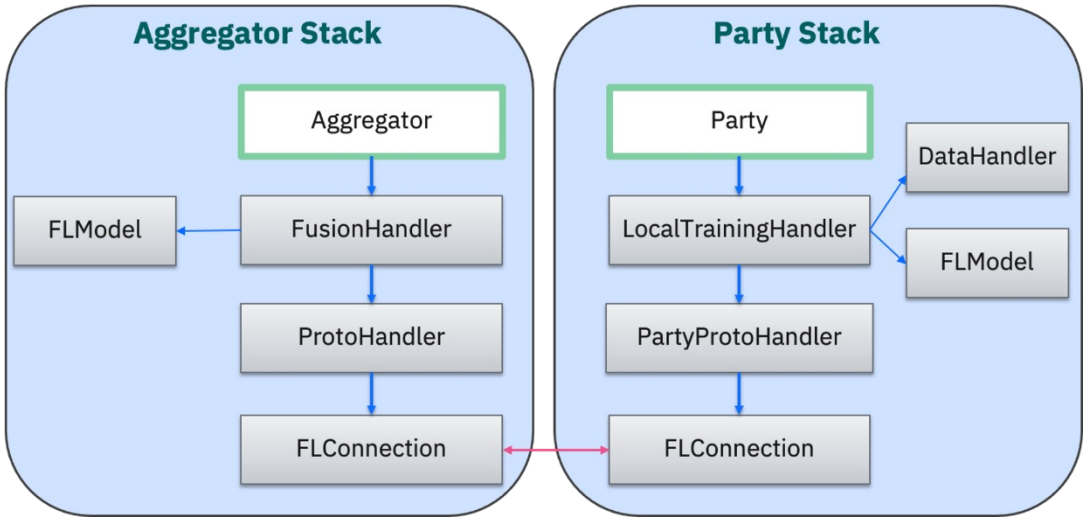
Results: Federated Learning libraries (PySyft)



- It has 8300 stars on GitHub.
- 250 Contributors.
- It works in simulation mode only.
- It can be extended with PyGrid, PyVertical or syft.js.
- Supports only Deep Learning models.

Ziller, A., Trask, A., Lopardo, A., Szymkow, B., Wagner, B., Bluemke, E., ... & Kaissis, G. (2021). Pysyft: A library for easy federated learning. In Federated Learning Systems (pp. 111-139). Springer, Cham.

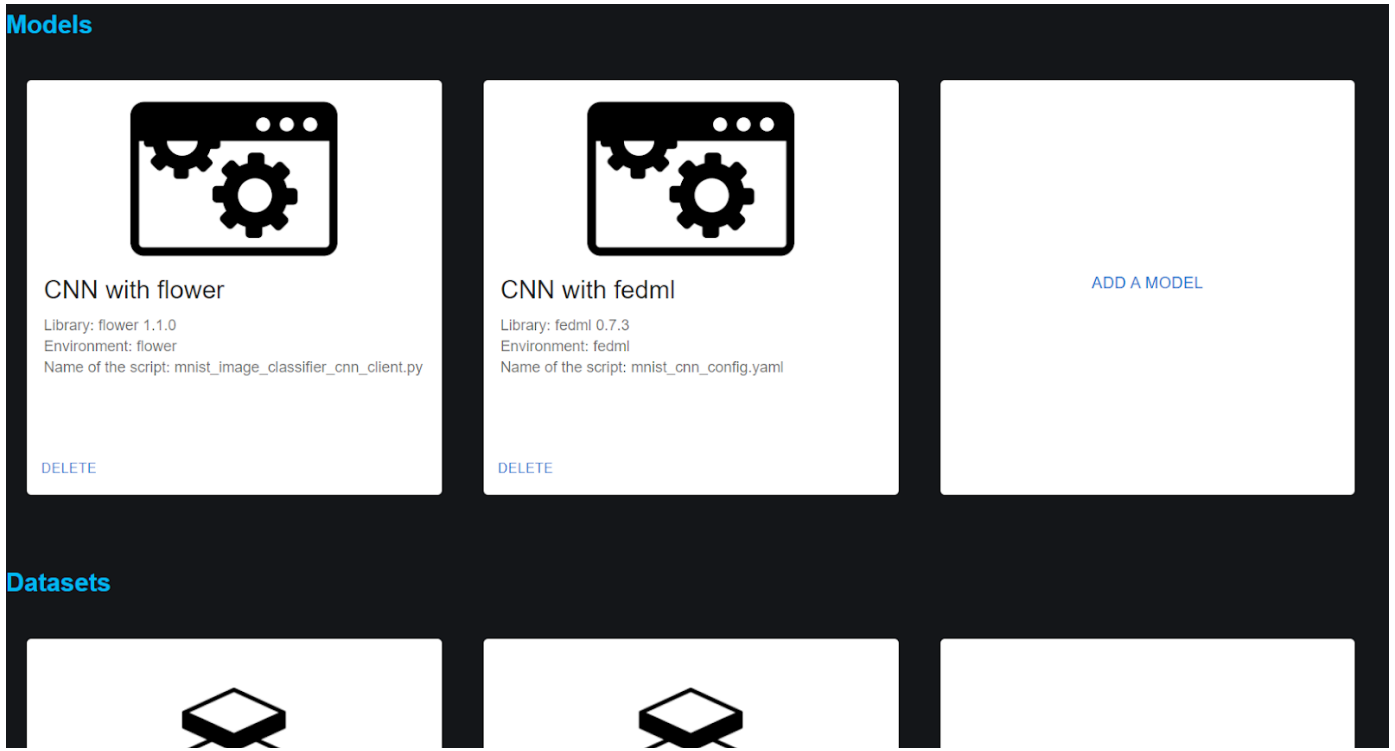
Results: Federated Learning libraries (IBM FL)

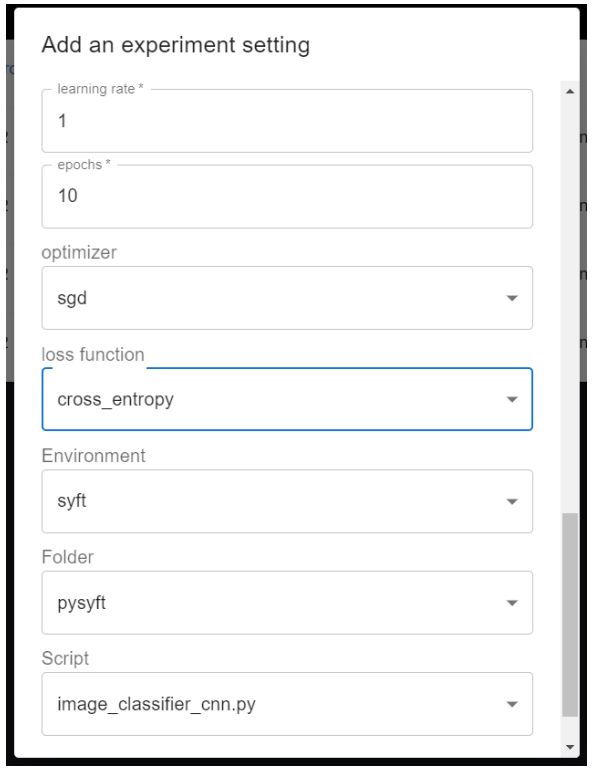


- It has 340 stars on GitHub.
- 10 Contributors.
- It supports a wide variety of ML models and ML frameworks.
- It has a logical separation between the client and the server.
- It supports both cross-silo and stand-alone simulation.

Ludwig, H., Baracaldo, N., Thomas, G., Zhou, Y., Anwar, A., Rajamoni, S., ... & Abay, A. (2020). Ibm federated learning: an enterprise framework white paper v0. 1. arXiv preprint arXiv:2007.10987.

Addittional Benchmark Screenshots





FedML additional results

2 clients	
Precision	4.26%
Recall	1.78%
F1	2.56%

16 clients	
Precision	4.26%
Recall	2.42%
F1	3.09%

100 clients	
Precision	4.11%
Recall	4.08%
F1	4.09%

Conclusion: Answering the Research Questions

RQ1: What are the functional and non-functional requirements relevant for a federated learning library, and what are the most important metrics to benchmark them?

The FL expert interviewed expect an FL library to support with the basic functionalities (communication, encryption, and data aggregation). They think that the most important NFR are fainess, scalability, accuracy, and efficiency.

RQ2: What are the different federated libraries available, and how do they differ in terms of functionality?

There are currently 12 libraries referenced in the litterature. They all differ to eachother in terms of architecture, maturity, functionality, and usecases

RQ3: How could a modular software application that benchmarks the different federated learning libraries using the metrics be developed?

The benchmark includes a fullstack web application that sends CLI commands to python scripts to train the FL models. The results are then scraped and displayed on the application. The experiments conducted with the benchmark showed that each library is suitable for a different usecase