References

1. "Communication-Efficient Learning of Deep Networks from Decentralized Data"

   H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Agüera y Arcas
   Link: https://arxiv.org/abs/1602.05629

2. "A Comparative Evaluation of FedAvg and Per-FedAvg Algorithms for Dirichlet Distributed Heterogeneous Data"

   Hamza Reguieg1 , Mohammed El Hanjri2 , Mohamed El Kamili1 , Abdellatif Kobbane2
   Link: https://arxiv.org/abs/2309.01275

3. "Heterogeneous Federated Learning: State-of-the-art and Research Challenges"
MANG YE, XIUWEN FANG, and BO DU, PONG C. YUEN, ,DACHENG TAO

   Link: https://dl.acm.org/doi/10.1145/3625558

15-03-2024 **PROJECT DISCUSSION -3(MID UPDATE)**

Discussion was about the existing algorithms for aggregating the parameters from local model to global model.

Existing algorithms Research Paper:

- Federated Averaging(FedAvg) :https://arxiv.org/abs/1602.05629
- FedProx: https://arxiv.org/abs/1812.06127
- Q-FedAvg: https://arxiv.org/abs/1905.10497
- Per-FedAvg: https://proceedings.neurips.cc/paper/2020/file/24389bfe4fe2eba8bf9aa9203a44cdad-Paper.pdf

FEDERATED LEARNING FRAMEWORKS

1. Flower (Federated Learning Framework). Flower is an open-source Python library that allows model training across decentralized devices or servers. Flower works with popular deep learning frameworks like TensorFlow and PyTorch. It is designed to scale to a large number of participating clients and supports diverse FL device scenarios. Flower follows the **client-server architecture** where the central server manages the global model, and the clients perform local training. It supports optimizations for enhancing the efficiency and speed of the FL processes.

- Flower is highly customizable to be adapted to each individual use case
- Extendability: Many components can be extended and overridden to build new state-of-the-art systems

https://flower.ai/docs/framework/tutorial-series-get-started-with-flower-pytorch.html

2. The FedML platform. FedML is an open research library and benchmark that supports diverse FL computing paradigms: on-device training for edge devices, distributed computing, and single-machine simulation. FedML also supports flexible and generic API design and provides comprehensive reference baseline implementations (optimizer, models, and datasets). FedML utilizes virtual nodes and central server architecture.

https://doc.fedml.ai/federate

3. Substra. Substra is an open source framework which provides distributed learning that guarantees privacy such that the data never leaves the nodes, and the exchange within the network involves only the predictive models, algorithms, and non-sensitive metadata. It supports a variety of computation models, including the parallel computation plan used in FL. Substra is focused on the medical field with the purpose of data ownership and privacy.

- Privacy: Substra uses trusted execution environments (also called enclaves) that enables setting aside private regions for code and data
- Traceability: Substra writes all operations on the platform to an immutable ledger
- Security: Substra encrypts model updates, data on rest, and network communication

https://github.com/Substra/substra/tree/main/docs

4. The Open Federated Learning (OpenFL) framework. OpenFL is an open-source Python-based tool designed for training machine learning algorithms through the collaborative learning paradigm of FL. It is compatible with training pipelines using both TensorFlow and PyTorch and can be extended to other ML frameworks. The significance of OpenFL lies in its scalability, emphasis on trusted execution, and the consistent migration of centralized ML models to a federated training pipeline.

The OpenFL design philosophy is based on the Federated Learning (FL) Plan. It is a YAML file that defines required collaborators, aggregators, connections, models, data, and

any required configuration. OpenFL runs on docker containers to isolate federation environments.

https://openfl.readthedocs.io/en/latest/index.html

5. **Nvidia Flare:** NVIDIA FLARE (NVIDIA Federated Learning Application Runtime Environment) is a domain-agnostic, open-source, extensible SDK that allows researchers, data scientists and data engineers to adapt existing ML/DL and compute workflows to a federated paradigm. With the FLARE platform, developers can create a secure and privacy-preserving solution for decentralized data computing, facilitating distributed multi-party collaboration. FLARE makes it easy to use models from MONAI and Hugging Face and enables ML Engineers to easily connect to existing ML workflows (PyTorch, RAPIDS, Nemo, TensorFlow).

https://nvflare.readthedocs.io/en/main/flare_overview.html

6. **PySyft:** PySyft is an open-source Python 3 based library that enables federated learning for research purposes and uses FL, differential privacy, and encrypted computations. It was developed by the OpenMined community and works mainly with deep learning frameworks such as PyTorch and TensorFlow.

https://openmined.github.io/PySyft/

7. **TensorFlow Federated:** TensorFlow Federated (TFF) is a Python 3 open-source framework for federated learning developed by Google. The main motivation behind TFF was Google's need to implement mobile keyboard predictions and on-device search. TFF is actively used at Google to support customer needs.

https://www.tensorflow.org/federated

Fig. Most used Federated Learning frameworks
https://medium.com/elca-it/flower-pysyft-co-federated-learning-frameworks-in-python-b1a8eda68b0d

Framework	Number of Clients	Total Training Time (Sec)	Loss	Accuracy	CPU Usage %	RAM Usage %
Flower	2	4192	4.582	0.9931	80.08	45.43
	15	6008	2.742	0.9913	76.14	47.92
	30	5712	3.642	0.9891	75.69	47.85
	60	5044	3.906	0.9874	89.65	52.11
	80	4248	5.298	0.9835	86.61	46.03
	100	4539	5.314	0.9809	85.93	49.02
FedML	2	271	1.109	0.664	90.48	30.49
	15	707	1.038	0.6579	93.45	30.28
	30	1,431	1.869	0.5212	96.85	30.43
	60	2,633	1.035	0.6819	98.24	30.62
	80	3,269	0.687	0.7662	98.62	31.14
	100	4,346	0.615	0.7913	98.97	31.58
Substra	2	180	0.278	0.9148	42.18	20.71
	15	1330	0.166	0.9537	42.88	21.05
	30	2,760	0.243	0.9098	42.15	20.64
	60	6,320	0.194	0.9467	40.73	21.27
	80	9,590	0.239	0.9365	38.46	23.21
	100	13,630	0.202	0.9406	35.81	21.38
OpenFL	2	2246	0.35	0.9743	67.19	42.76
	15	2816	0.122	0.9738	65.26	42.29
	30	3128	0.145	0.9674	64.99	45.89
	60	4269	0.169	0.9594	63.1	53.13
	80	4985	0.177	0.9567	62.14	58.34
	100	5876	0.193	0.949	64.13	63.11

Performance Metrics for the Different Frameworks with Different Client Counts
Citation: https://doi.org/10.21203/rs.3.rs-3934159/v1

Flower Framework Architecture

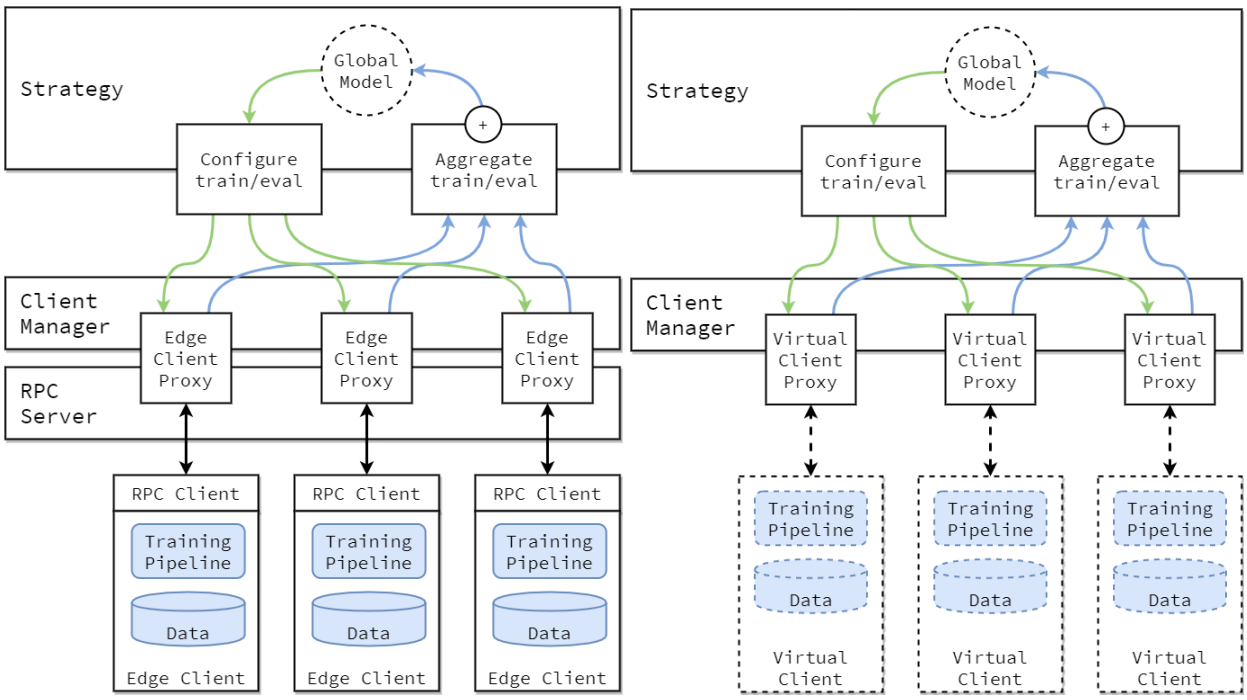


Fig: Edge Client Engine and Virtual Client Engine

Categorization of Federated Learning based on Type of Participating Client:

Federated Learning can be categorized into two categories based on the capabilities, count, and type of participating clients: cross-device FL and cross-silo FL. In cross-device FL, the participating clients are compact distributed entities such as smartphones, wearables, and edge devices. Each client generally possesses a relatively limited amount of local data. Consequently, the success of cross-device FL often hinges on the involvement of a substantial number (potentially millions) of edge devices in the training process. On the other hand, cross-silo FL involves clients that are typically companies or organizations like hospitals and banks. In this scenario, the number of participants is modest, ranging from two to a hundred, and each client is expected to engage in the entire training process.

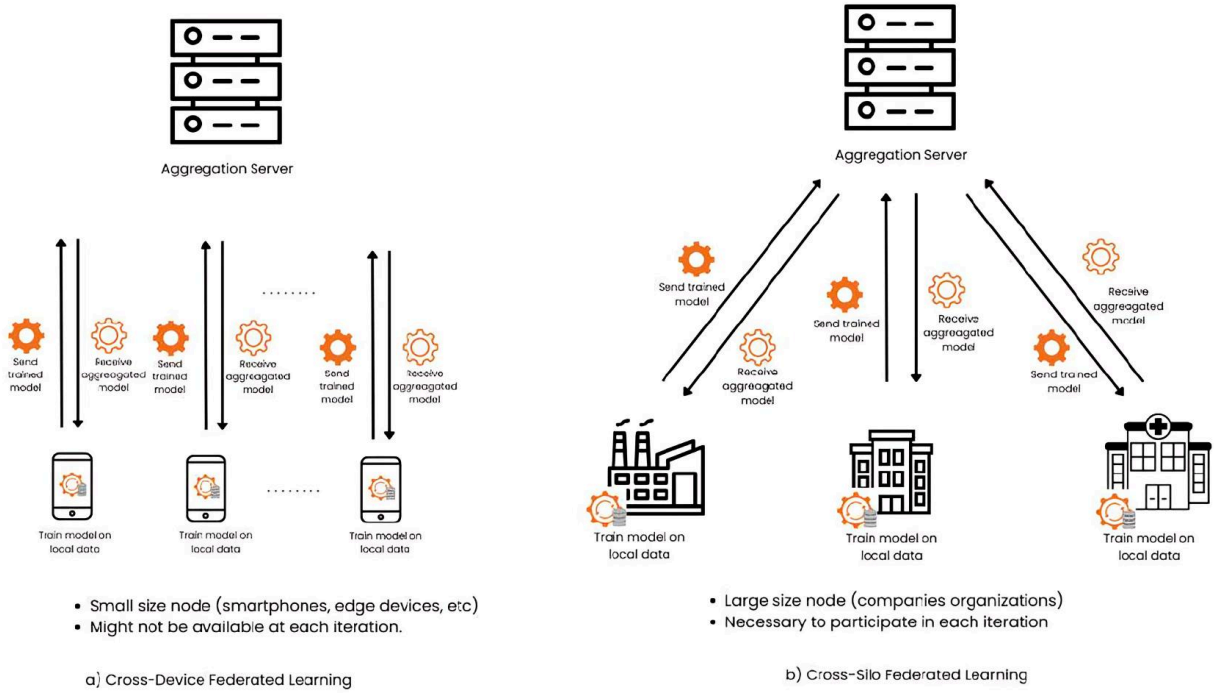


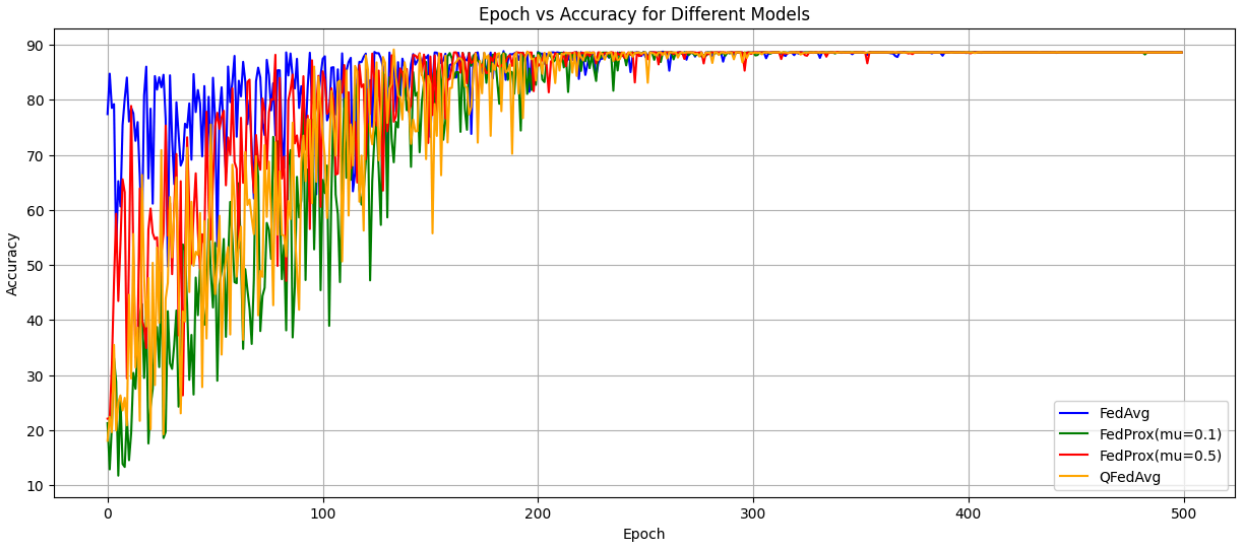
Fig Cross Divice FL and Cross Silo FL

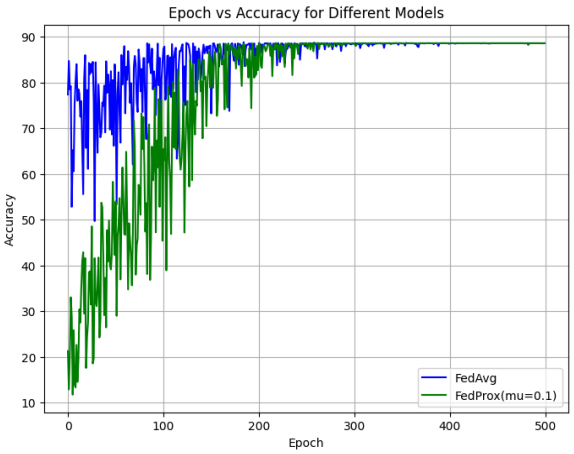
Experimental setup

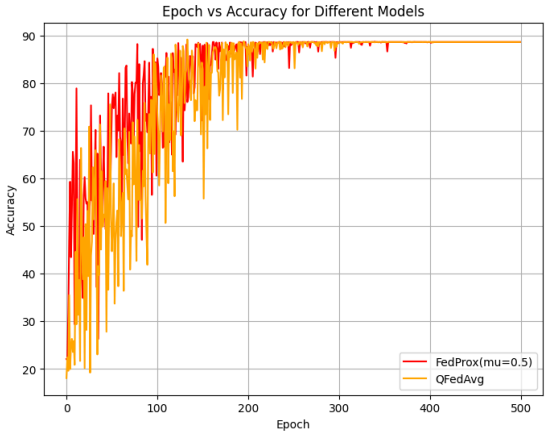
Water Quality Dataset:

- Dataset - Classification Problem(Two class)(20 features)(7999 data points)
- Number of clients : 3
- Client 1 - Features {'aluminium', 'ammonia', 'arsenic', 'barium', 'cadmium', 'chloramine', 'chromium' , ‘is_safe’}
- Client 2 - Features {'copper', 'flouride', 'bacteria', 'viruses', 'lead', 'nitrates', 'nitrites','is_safe’}
- Client 3 -Features{‘mercury’ ‘perchlorate’ ‘radium’ ‘selenium’ ‘silver’ ‘uranium’,’is_safe’}

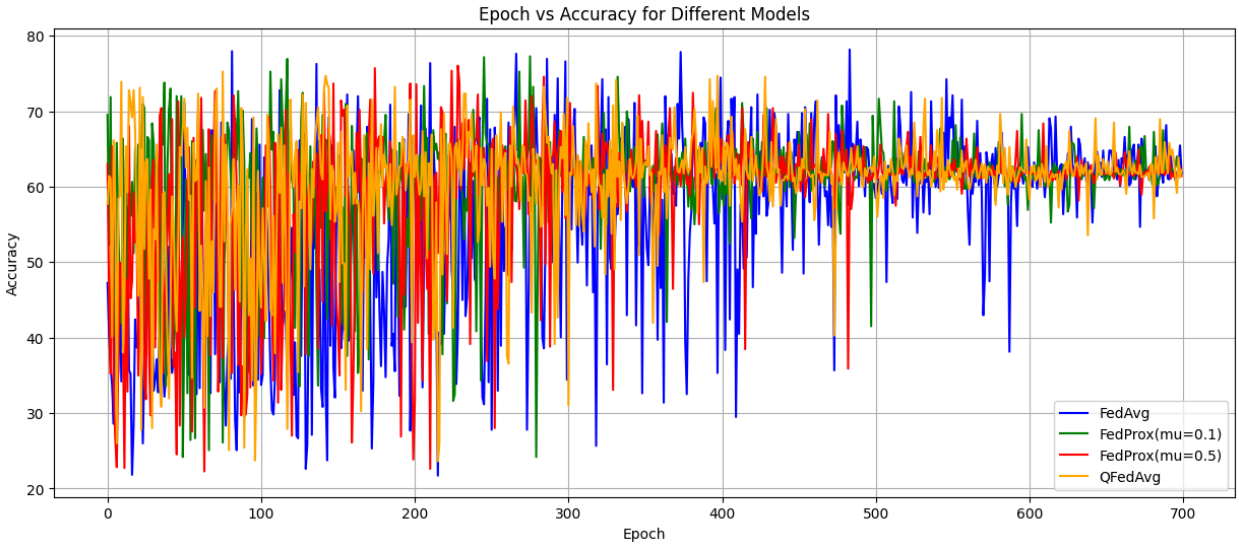
Results & Discussion:

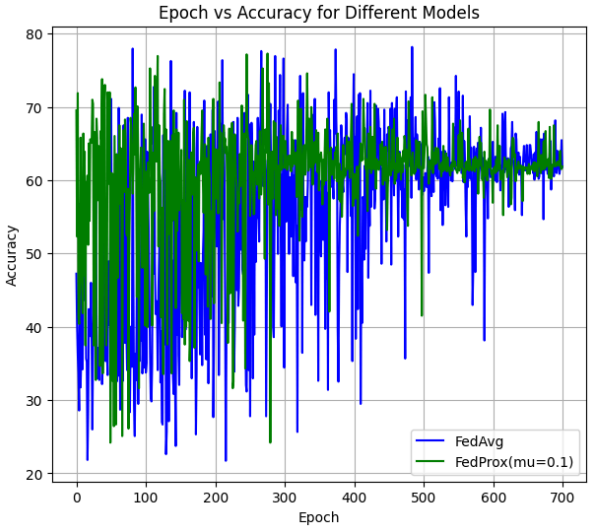


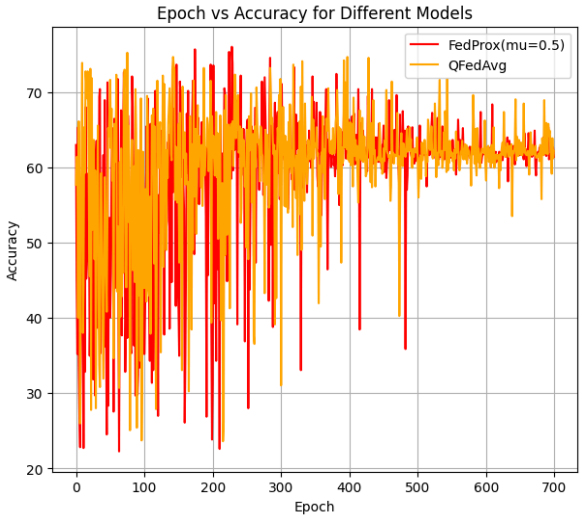




Titanic Dataset:







Security in Federated Learning

Federated learning can employ a variety of security approaches to further ensure secure aggregation and privacy preservation during the federated training process.

- **Secure Aggregation** ensures that individual updates from clients are aggregated in a way that the server can only access the aggregated results and not the individual contributions. An example of secure aggregation is the use of cryptographic techniques similar to Homomorphic Encryption to safeguard model updates during aggregation, preventing unauthorized access. Another example is Secure Multi-Party Computation (SMPC) where the computations are performed on encrypted data, ensuring that no party has access to the raw data.

- **Differential Privacy** introduces controlled noise into the updates, making it difficult to infer any individual's data from the shared information. In the context of Federated Learning, each client has the ability to inject randomly generated values into its model weights prior to transmission. As a result, even if the data is subjected to reverse engineering, it does not correspond precisely to the data of any individual user.

References:

[1] Scalability and Performance Evaluation of Federated Learning Frameworks: A Comparative Analysis
Bassel Soudan, Sohail Abbas, Ahmed Kubba, Manar Abu Wasif Talib, Qassim Nasir

Link: https://doi.org/10.21203/rs.3.rs-3934159/v1

[2] A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection
Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, Bingsheng He

Link: https://arxiv.org/pdf/1907.09693#cite.voigt2017eu