UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Gandab Hasanova

FLBench - A Comprehensive Experimental Evaluation of Federated Learning Frameworks

Master's Thesis (30 ECTS)

Supervisor(s): Feras Mahmoud Naji Awaysheh

Tartu 2024

FLBench - A Comprehensive Experimental Evaluation of Federated Learning Frameworks

Abstract:

Federated learning is an innovative approach to collaborative machine learning that allows several decentralized organizations to cooperatively train a common model without disclosing their own data. With increasingly tighter data privacy regulations like GDPR in force, Federated Learning has become one of the need-to-have techniques to adopt. For example, by employing FL, hospitals can train models on patient data from different hospitals to enhance diagnostic accuracy while keeping sensitive information away from a central repository. Similarly, it can improve financial systems' authenticity for security by making them more secure with customer data encrypted safely within their own database.

However, the Federated Learning domain is rapidly changing, and many new frameworks are emerging as the landscape of open-source tools continues to grow. This growth makes even experienced researchers uncertain about the trade-off among these frameworks and when to use which framework. This study aims to resolve this issue by comprehensively examining six popular federated learning (FL) frameworks: NVIDIA FLARE, Flower, FedML, TensorFlow Federated (TFF), FEDn, and Substra. We aim to systematically compare and analyze these frameworks, with the help of Federated Averaging (FedAvg) on a Convolutional Neural Network (CNN) model trained with CIFAR-10 data. Performance analysis includes these metrics: loss, accuracy, total training time, CPU and RAM consumption, and network utilization during training.

In order to buttress our claims with empirical proof and offer a complete view, we conducted experiments while running them on different client counts (1, 10, 50, 100), which helped us understand how each framework scales up. Key results of our research are: FedML achieved the highest accuracy with 91% on 100 clients but had longer training times. Flower demonstrated a balance of high accuracy and the shortest training times which makes it suitable for production environments. NVIDIA FLARE showed high CPU utilization and good overall performance. TensorFlow Federated and Substra exhibited consistent performance on different client counts. FEDn had the lowest accuracy but showed potential for cases where limited computational resources are available.

This review contributes to the literature by providing an overview and comparison of federated learning frameworks that can inform a choice for use-case-dependent priorities and resource availability constraints. Our results suggest that when picking a framework, we should consider its performance on general evaluation metrics and other factors such as scalability, customizability, or resource constraints in your application scenario.

The study's results are designed to guide industry practitioners and researchers in making informed decisions when implementing a Federated Learning solution by provid-

ing insights into each framework's capabilities and trade-offs. The detailed research and comprehensive evaluation provided help in understanding the nuances and implications of federated learning frameworks in various use cases.

Keywords:
Federated Learning, Federated Averaging (FedAvg), Convolutional Neural Network (CNN), CIFAR-10 Dataset, Flower, FedML, TensorFlow Federated, NVIDIA FLARE, FEDn, Substra.

CERCS:
P170 Computer science, numerical analysis, systems, control

FLBench – koondatud õpperaamistike põhjalik eksperimentaalne hindamine

Lühikokkuvõte:

Födereeritud õpe on uuenduslik lähenemine koostööl põhinevale masinõppele, mis võimaldab mitmel detsentraliseeritud organisatsioonil ühist mudelit koostöös välja õpetada ilma oma andmeid avaldamata. Üha rangemate andmete privaatsuseeskirjade, nagu GDPR, jõustumise tõttu on Federated Learningist saanud üks vajalikest tehnikatest, mida kasutusele võtta. Näiteks FL-i kasutades saavad haiglad koolitada erinevate haiglate patsientide andmete mudeleid, et parandada diagnostilist täpsust, hoides samal ajal tundlikku teavet kesksest hoidlast eemal. Samamoodi võib see parandada finantssüsteemide autentsust turvalisuse tagamiseks, muutes need turvalisemaks kliendiandmetega, mis on turvaliselt nende enda andmebaasis krüpteeritud.

Födereeritud õppe domeen muutub aga kiiresti ja avatud lähtekoodiga tööriistade arenedes on esile kerkimas palju uusi raamistikke. See kasv muudab isegi kogenud teadlased ebakindlaks nende raamistike vahelise kompromissi osas ja millal millist raamistikku kasutada. Selle uuringu eesmärk on see probleem lahendada, uurides põhjalikult kuut populaarset liitõppe (FL) raamistikku: NVIDIA FLARE, Flower, FedML, TensorFlow Federated (TFF), FEDn ja Substra. Meie eesmärk on neid raamistikke süstemaatiliselt võrrelda ja analüüsida, kasutades CIFAR-10 andmetega koolitatud konvolutsioonilise närvivõrgu (CNN) mudelit Federated Averaging (FedAvg). Jõudlusanalüüs sisaldab järgmisi mõõdikuid: kadu, täpsus, kogu treeninguaeg, protsessori ja RAM-i tarbimine ning võrgu kasutamine treeningu ajal.

Oma väidete toetamiseks empiiriliste tõenditega ja täieliku ülevaate pakkumiseks viisime läbi katsed, käitades neid erinevate klientide arvuga (1, 10, 50, 100), mis aitasid meil mõista, kuidas iga raamistik suureneb. Meie uuringu peamised tulemused on järgmised: FedML saavutas 100 kliendi puhul kõrgeima täpsuse 91%, kuid koolituse aeg oli pikem. Flower demonstreeris tasakaalu suure täpsuse ja lühimate treeningaegade vahel, mis muudab selle sobivaks tootmiskeskkondadesse. NVIDIA FLARE näitas kõrget protsessori kasutust ja head üldist jõudlust. TensorFlow Federated ja Substra näitasid

erinevate klientide arvu osas ühtlast jõudlust. FEDn oli madalaima täpsusega, kuid see näitas potentsiaali juhtudel, kui arvutusressursid on piiratud.

See ülevaade täiendab kirjandust, pakkudes ülevaadet ja võrdlust ühendatud õpperaamistikest, mis võivad anda teavet kasutusjuhtumist sõltuvate prioriteetide ja ressursside kättesaadavuse piirangute valikul. Meie tulemused näitavad, et raamistiku valimisel peaksime arvestama selle toimivust üldiste hindamismõõdikute ja muude teguritega, nagu skaleeritavus, kohandatavus või ressursipiirangud teie rakenduse stsenaariumis.

Uuringu tulemused on mõeldud selleks, et suunata valdkonna praktikuid ja teadlasi liitõppelahenduse rakendamisel teadlike otsuste tegemisel, pakkudes ülevaadet iga raamistiku võimalustest ja kompromissidest. Üksikasjalikud uuringud ja põhjalik hindamine aitasid mõista ühendatud õpperaamistike nüansse ja tagajärgi erinevatel kasutusjuhtudel.

Võtmesõnad:
Federatiivne Õppimine, Federatiivne Keskmistamine (FedAvg), Konvolutsiooniline Närvivõrk (CNN), CIFAR-10 Andmestik, Flower, FedML, TensorFlow Federated, NVIDIA FLARE, FEDn, Substra

CERCS:
P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisor, Dr. Feras Mahmoud Naji Awaysheh, for his constant support and guidance throughout this journey. His insightful feedback and encouragement greatly influenced the shaping of the direction and final result of this thesis.

Contents

1 Introduction **8**
- 1.1 Motivation . . . 8
- 1.2 Research Goals and Questions . . . 9
- 1.3 Thesis Structure . . . 9

2 Background **11**
- 2.1 Federated Learning Overview . . . 11
- 2.2 Federated Learning Architecture . . . 11
- 2.3 Privacy and Security in Federated Learning . . . 13
- 2.4 Challenges and Limitations of Federated Learning . . . 14
- 2.5 Data Partition-based Types of Federated Learning . . . 14
- 2.6 Federated Learning Frameworks . . . 15
  - 2.6.1 Flower . . . 15
  - 2.6.2 FedML . . . 18
  - 2.6.3 TensorFlow Federated . . . 19
  - 2.6.4 NVIDIA FLARE . . . 20
  - 2.6.5 FEDn . . . 22
  - 2.6.6 Substra . . . 23
- 2.7 Related works . . . 24
- 2.8 Applications of Federated Learning . . . 26

3 Methodology **28**
- 3.1 Dataset Selection . . . 28
- 3.2 Experimental Setup . . . 28
- 3.3 Evaluation Metrics . . . 29
- 3.4 FedAVG Algorithm . . . 29

4 Comparative Evaluation of FL Frameworks **31**
- 4.1 Flower Evaluation . . . 31
- 4.2 NVIDIA Flare Evaluation . . . 31
- 4.3 FedML Evaluation . . . 31
- 4.4 TensorFlow Federated Evaluation . . . 32
- 4.5 FEDn Evaluation . . . 32
- 4.6 Substra Evaluation . . . 32
- 4.7 Architectural Considerations . . . 33
- 4.8 Comparison of Performances . . . 34
  - 4.8.1 Comparison of Accuracy and Loss Metrics . . . 34
  - 4.8.2 Comparison of CPU and RAM Utilization . . . 36
  - 4.8.3 Comparison of Network Utilization . . . 36

4.8.4 Comparison of Training Times 37
4.9 Final Results 39

5 Discussion and Limitations **42**
5.1 Discussion 42
5.2 Scope and Limitations 43

6 Conclusion **44**

7 Future Work **45**

References **49**

Appendix **50**
I. Glossary 50
II. Licence 51

1 Introduction

This section provides the thesis motivation, statements of the problems, research objectives and questions that are addressed in this paper, and the significance of the study. It also provides the thesis structure that followed in this paper.

1.1 Motivation

Federated learning (FL) is a game-changing approach to collaborative machine learning that enables several dispersed companies to train a shared model without sharing sensitive data. This shift in thinking tackles head-on the issues of data privacy and security as well as the need to comply with stringent regulations such as GDPR and CCPA[KMA+19]. Standard machine learning algorithms are very vulnerable to privacy invasions and data breaches since they depend on centralized data collecting. The capacity of FL to store data locally and communicate only model changes improves data security and privacy[MMR+16].

As a result of the growing need for privacy-preserving machine learning solutions in several areas, including healthcare, finance, and the Internet of Things (IoT), numerous FL frameworks have been developed which make distributed model training easier by including the required protocols, infrastructure, and tools inside FL implementations[YLCT19]. Understanding their capabilities and picking the ideal framework for particular applications demands comprehensive and methodical study due to the huge variability in their features and performance characteristics.

Choosing the appropriate federated learning framework for a certain application might be difficult given the great range of features and performance criteria provided by current ones. It is very difficult to make a decision without thorough analyses considering many criteria (e.g., loss, accuracy, training time, and resource use). This paper evaluates six well-known federated learning systems in order to close this disparity: Substra, NVIDIA FLARE, Flower, FedML, and TensorFlow Federated. The objectives of the research are to recognize the advantages and disadvantages of every framework and provide insightful analysis on how to use them in practical settings.

This research is important for several reasons. First of all, it provides a comprehensive comparison of performance and resource utilization metrics for selected six frameworks. It is especially beneficial for researchers to make well-informed decisions before selecting and processing any framework that is investigated here. Furthermore, the findings in this paper contribute to existing resources for Federated Learning, which need to be investigated more in order to develop and contribute to FL. Analysed features of each framework can be beneficial to use them in real case scenarios[LSTS19].

1.2 Research Goals and Questions

This paper aims primarily to evaluate six federated learning systems in order to help practitioners and academics select the most appropriate framework for their needs. The project investigates the following research questions to help reach this aim:

RQ 1. What are the key features and performance metrics crucial for federated learning frameworks?

This research aims to identify the basic characteristics and measures that validate the effectiveness and production capacity of FL models. Data security protections, scalability, smooth interaction with present machine learning systems, and compatibility with many data formats define the primary features. It is necessary to consider important performance metrics like loss, accuracy, training time, CPU and RAM utilization, and network usage while analyzing the capability of frameworks.

RQ 2. How do existing federated learning frameworks compare on these metrics?

This study attempts to assess and contrast the selected FL systems using the given performance criteria. The comparative analysis will look at the advantages and disadvantages of each framework, providing a clear understanding of how well they perform in different environments and configurations.

RQ 3. How may FLBench be expanded to support new experimental configurations and frameworks in the future?

This problem centers on FLBench's scalability and extensibility, which is the benchmarking tool used in this study. Enhancing FLbench to enable more complicated experimental settings and upcoming FL frameworks is necessary to ensure its continuous relevance and usefulness in the area of federated learning, which is still in progress.

1.3 Thesis Structure

The organization of the thesis structure is like this sequence:

Chapter 2: Background - It introduces FL, its architecture, privacy considerations, selected frameworks, and their architectures. It is followed by the literature review that influenced this study.
Chapter 3: Methodology - This chapter describes the setup of the experiment, dataset selection, and environment details of experiments. It evaluates comparison metrics and configuration of the experiments. It explains design choices and the different steps taken to ensure the reliability of the results.
Chapter 4: Comparative Evaluation of FL Frameworks: This chapter gives information about the evaluation results of six FL frameworks. It is thoroughly discussed in this

chapter how well they fared against parameters such as accuracy, loss, use of CPU and RAM, and network usage.

Chapter 5: Discussion and Limitations: This chapter interprets the testing results of Chapter 4, making a comparison to prior literature, and also discusses the limitations of this study with ways of further investigation in the future.

Chapter 6: Conclusion: This final chapter summarizes major conclusions about the work done on the research and exactly what has been contributed to the area of federated learning with some suggestions.

2 Background

This section will discuss the thesis background regarding the FL definition, frameworks, and related work. It also summarizes some of the works on federated learning (FL) frameworks, including studies that have come before it, and emphasizes the most important results, methods, and contributions.

2.1 Federated Learning Overview

A networked paradigm for machine learning known as federated learning (FL) was created in response to growing concerns over the privacy and security of data. Traditional machine-learning approaches need centralized data collection, which creates severe privacy and security problems.

To solve these issues, federated learning allows autonomous entities to work together to train a shared model without disclosing any personal information. This includes institutions and mobile devices[KMY+16]. This strategy may be very useful in several fields, including healthcare, the financial sector, and the Internet of Things (IoT)[BEG+19].

Aiming to address the increasing need for privacy-preserving machine learning models, FL was first presented by Google in 2017[YLCT19]. A primary motivation for developing FL was the need to facilitate the building of robust machine-learning models by many parties independent of one another in the absence of shared raw data. The idea is that instead of exchanging model updates (such as weights or gradients) with a central server, each participant trains a local model using their own data. The central server then aggregates these modifications and transmits them back to the parties for further training. This results in the creation of the global model. This iterative method is maintained until the model converges.

2.2 Federated Learning Architecture

The architecture of Federated Learning systems describes how participants communicate and aggregate model updates. There are different architectural approaches targeted at several requirements and constraints[LLZ+21].

Client-Server Architecture - Typically, in an FL system, several clients (such as mobile devices or organizations) store and train a model locally using their own datasets. A central server aggregates the changes from these clients' model updates on a regular basis to develop a global model. As shown in Figure 1, this procedure is repeated several times until the global model reaches an acceptable level of performance. An additional layer of protection is provided for the clients' data since the central server is not directly linked to it.

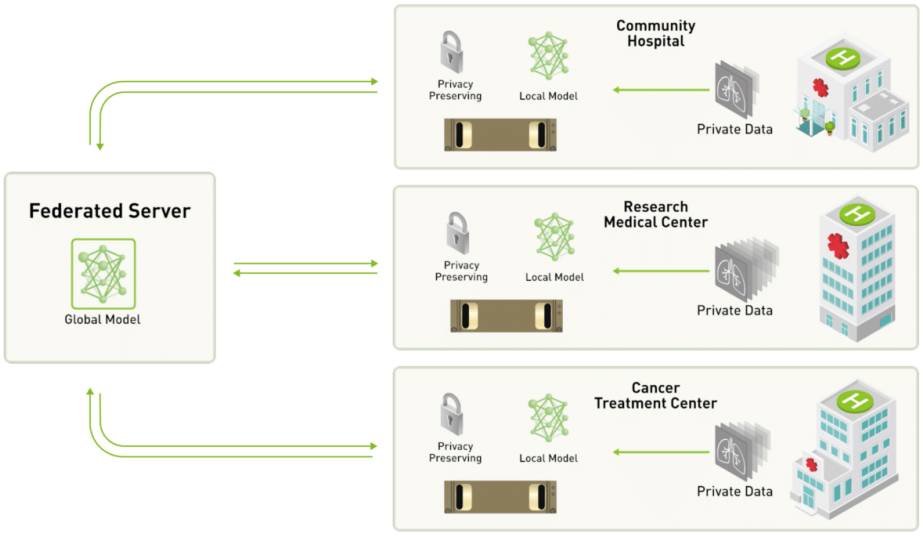

Figure 1. Federated Learning Client-Server System Architecture [NVI24]

In relation to this architecture: Multiple clients store local datasets.Each client's data is used to train a local model. Clients regularly send and receive data to a central server. The central server aggregates the modifications to improve a global model.The process is performed several times until the global model operates as expected. The central server is not directly linked to the client's data in order to ensure data security and privacy.

Cross-silo vs. Cross-device Architecture - Cross-silo federated learning requires a limited number of trustworthy players who possess substantial computing capabilities, such as several corporations or data centers[HHL22]. It is primarily utilized in business-to-business environments. For cross-device FL, a large number of these devices may include, such as mobile phones or IoT devices. This type of architecture is ubiquitous in consumer applications and solves challenges around device availability and resource constraints.

Fully Decentralized Architecture - There is no big central server in this setup. Participants update the global model via direct communication with each other. This increases privacy and removes the single point of failure. However, it might be more difficult to deploy and may experience some challenges in terms of communication efficiency and convergence[MBPS+23].

Hierarchical Federated Learning - This architecture introduces intermediate aggregators between the central server and consumers. Especially when you have large-scale systems or a hierarchical naturally occurring between the units/countries (e.g., country-

level aggregators in a global system). The primary objective of Hierarchical FL is to scale up training and reduce communication overheads, but this introduces additional complexity.

Hybrid Architecture - This architecture combines some elements of different architectures. For instance, a system may use hierarchical aggregation but peer-to-peer communication for jobs within each level. Hybrid architecture can be customized according to needs, which makes it flexible.

2.3 Privacy and Security in Federated Learning

Federated learning is conducive to protecting users' privacy due to decentralized data storage and processing [KAAK24]. With FL, data is kept on devices and not sent to a central server; only model updates are exchanged. This significantly decreases the threats of data breaches and enhances privacy. However, FL does have some challenges, such as ensuring the privacy of shared model updates and protecting against various attacks[GKN17].

Differential Privacy: Differential privacy adds noise to the model updates so that adding or removing any single data point has only a minor impact on the overall outcome. It provides a balance between strong privacy guarantees and useful data analysis[ACG$^+$16].

Secure Multi-Party Computation (SMPC): SMPC allows multiple parties to compute a function on their inputs without revealing their data. SMPC can aggregate model updates in a way that the central server learns nothing about any individual update, proving to be an exceptional privacy module[BIK$^+$17].

Homomorphic Encryption: This allows computations to be performed on encrypted data. Model updates can be sent to the central server, which is already encrypted, aggregated by the server, and then decrypted by the clients. This ensures privacy at each step of the process.

While these methods provide a new level of security to FL models, they are not invulnerable to attacks like poisoning and inference attacks. Poisoning attacks involve malicious clients sending incorrect model updates, degrading the global model's performance. Inference attacks involve attackers trying to recover hidden sensitive information from the model updates. To counteract these attacks, robust defense mechanisms and anomaly detection systems are essential.

In summary, Federated Learning ensures privacy and security using advanced cryptographic tools and strong defense strategies. Maintaining the privacy of the global model while preserving its integrity and performance remains a key challenge in FL, constituting an active area of ongoing research crucial for the advancement of distributed learning systems.

2.4 Challenges and Limitations of Federated Learning

Although Federated Learning presents several promising solutions, it comes with its challenges or barriers, which need to be addressed to unlock its true potential[LSTS19].

1. The first challenge is the overhead involved in communication. FL processes require high network bandwidth as various clients and the central server constantly share their model updates. This is particularly problematic on slow or intermittent networks. The optimal solution would be to reduce the overhead by determining the optimal frequency and size of the shareable data.

2. Most clients have computational limits, but FL processes conduct high-end computations. FL trains complex machine learning models through resource-constrained devices such as smartphones, IoT gadgets, and others. This may slow down the training and require the use of compact models or efficient training algorithms that can be deployed on such devices.

3. In a federated environment, clients often have Independent and Identically Distributed (IID) data, meaning the distribution of their raw datasets differs widely. This variance can lead to models that perform well for some clients but not others, lowering the overall efficiency of the federated model. An efficient FL system should handle this variance by ensuring that the model has high robustness using mechanisms to cope with data fluctuations.

4. Privacy and security are continuous issues. FL ensures that local data remains private, but the model updates must also be secure. Mechanisms should protect the model updates from privacy breaches, prevent poisoning, and ensure maximum privacy.

2.5 Data Partition-based Types of Federated Learning

There are three primary Data Partition-based types of federated learning: Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL). Each category is intended to handle different data partitioning and application requirements.

Horizontal Federated Learning (HFL), or sample-based federated learning, is the most prominent class in the future of AI. HFL is especially powerful in cases when the same feature space characterizes data at different parties, but their sample sets vary. For example, in healthcare, different healthcare centers may have information on different patients, but the features (like medical tests) will be the same across all these datasets. Such federated learning is indispensable for training robust models without data centralization in settings with homogeneous feature spaces[MTF22].

In contrast, Vertical Federated Learning (VFL) is better for situations where the data

is vertically partitioned. This means that each party's data with different rich features represents the same entities. Given these parties, VFL jointly allows a machine learning model to be trained based on their local models, resulting from training data with different features available at each site. For example, federated learning of this type is essential to financial services so that multiple parties, which may have information complementarity about the same customers (e.g., while one party has transaction records, another might hold credit scores) can pool their data and co-train a model together across two or more institutions[LKZ+23].

Federated Transfer Learning (FTL) is a new and hybrid method of applying the principles of federated learning in conjunction with transfer learning. One of the most exciting use-cases for this approach is domain adaptation with few labels (i.e., adapting a standard pre-trained model to solve some new task in related but unseen domains). In addressing these challenges, FTL tries to unify the positive properties linking federated learning and transfer learning. This makes a correct grounding of Foundation Models with respect to given applications particularly impactful in terms of how well they perform and across what domain[KFG+23].

The flexibility and power of federated learning as an approach to collaborative machine learning come from the fact that various types cater to different challenges and use cases. Federated learning will potentially transform different industries (from healthcare to finance) as it allows secure and scalable model training across distributed data sources, speeding up the time-to-insight for both models while preserving cross-party privacy.

2.6 Federated Learning Frameworks

Numerous frameworks have been created to support the implementation of federated learning in different environments. The frameworks mentioned here offer the essential resources, protocols, and infrastructure for facilitating distributed model training, all while prioritizing data privacy and security. Here are a few notable frameworks in the field:

2.6.1 Flower

Flower framework is a flexible and scalable platform. Significant issues with federated learning that this approach tackles include heterogeneity of data and devices, efficiency, and privacy. Flower offers a modular and expandable architecture that interfaces nicely with current processes by supporting a variety of machine learning and deep learning frameworks[Aut24]. It is suitable for both production and research environments because of its adaptability and ease of use in design. Experts in the field may use Flower to build secure federated learning systems that protect sensitive information without sacrificing model performance. The open-source nature and adaptability of the framework have

led to the advancement of federated learning applications in many sectors, including healthcare, banking, and the Internet of Things (IoT).

As shown in Figure 2, Flower has the following architecture:
Server: Manages the connected clients which are training models, and coordinates the entire federated learning cycle by aggregating model updates from these edge devices.
Client: Runs on local devices or servers, performs local training, and communicates with the server.
Strategy: Determines the federated learning algorithm, including model initialization, client selection, and update aggregation.
The design of this framework is characterized by a client-server architecture, where the server coordinates training and clients perform local computation. It allows the implementation of specific federated learning algorithms and benefits from the flexibility of different network conditions[Aut24].

Edge Client Engine and the Flower core framework design leverage a more refined practice in federated learning, which combines centralized strategies with distributed edge computing. Essentially, a global model collaborates with multiple edge clients in the hierarchy-centric system. To perform the learning process and evaluation, the architecture uses a configurable train/eval component to manage the parameters of learning processes as well as an aggregable train/eval module responsible for summarizing results from different edge clients. This pattern provides resources on edge devices while still maintaining a unified training policy under a Client Manager that can be operated using RPC (Remote Procedure Call) methods. They provide an intermediate layer between the central system and each individual edge client with its own training pipeline.

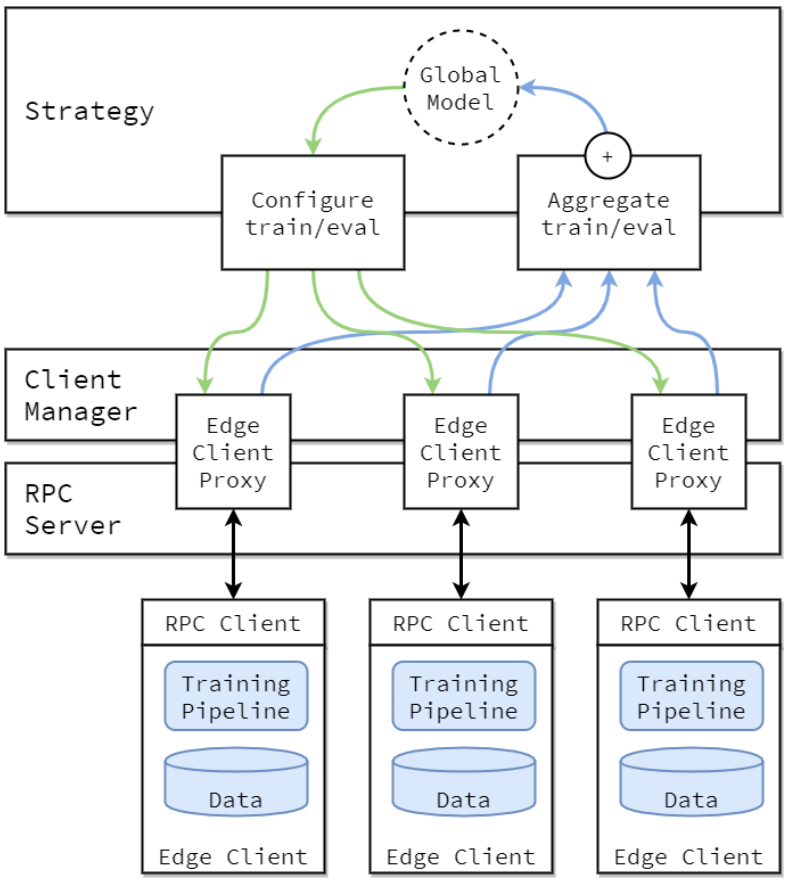

Figure 2. Flower framework architecture with Edge Client Engine [Aut24]

2.6.2 FedML

FedML is a research library and benchmark that facilitates several computer models for federated learning. It includes mobile learning, parallel processing, and using a single computer in a simulated environment. The API's adaptable and versatile architecture has the potential to be beneficial for a diverse array of FL applications. There is an effort underway at FedML to streamline the process of creating, releasing, and assessing FL algorithms. The software is compatible with several federated learning approaches and offers an integrated interface for well-known machine learning libraries. Because of its modular nature, FedML is simple for academics to add additional datasets or methods to[Tea24a].

FedML has two main components in its architecture as shown in Figure 3: the FedML Open Source Library and the FedML Edge-Cloud Platform[HLS+20].
Open Source Library supports the following computing paradigms: On-Device Training (Mobile, IoT), Distributed Computing and Standalone Simulation.

It has a three-layer architecture: The top layer consists of experiments and applications (FedCV, FedNLP, FedMedical, etc.). The middle layer contains Low-level Architecture Federation + Server API, including mobile/IoT support. The bottom layer holds FedML-core with low-level APIs for topology (security/privacy) and coordination.

Edge-Cloud Platform simplifies deployment, providing components such as FL Server, Client Launcher, and MLOps Integration. FedML practices term of write once, run anywhere, unlike traditional lab simulation experiments that cannot deploy directly to real distributed systems.

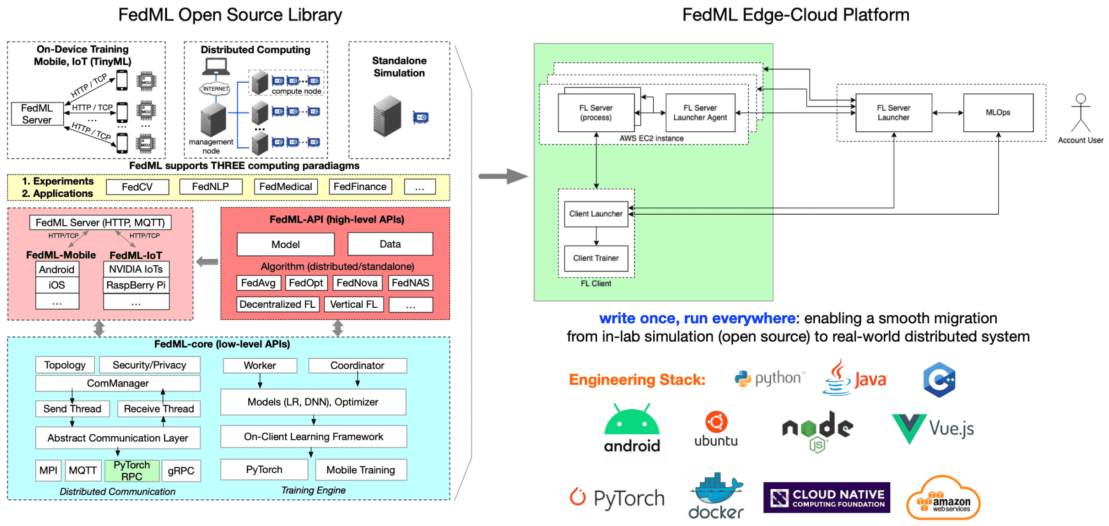

Figure 3. FedML source code architecture [HLS+20]

2.6.3 TensorFlow Federated

TensorFlow Federated (TFF) framework is an open-source framework which is developed by Google to manage machine learning and other computational tasks on data distributed among various sources. It provides a robust framework for creating TensorFlow federated learning models, hence facilitating seamless integration with existing machine learning systems. TFF is supposed to help developers and academics perform federated learning simulations and test novel ideas in a safe environment[Tea24c]. The framework provides a means to evaluate the robustness and efficiency of FL algorithms and high-level abstract concepts to create federated computations.

Figure 4 shows that the architecture of TensorFlow Federated is designed in two layers:

Federated Learning (FL) API: Offers high-level APIs to simplify the implementation of federated learning algorithms.

Federated Core (FC) API: Provides low-level APIs for representing federated computations.

The framework exposes a declarative programming model in which users express federated computations as abstract workflows. The TFF runtime executes these workflows, distributing the computations across clients and aggregating results.
The TensorFlow Federated (TFF) provides a framework for federated learning and delivers a complete environment for distributed computations. The framework's design

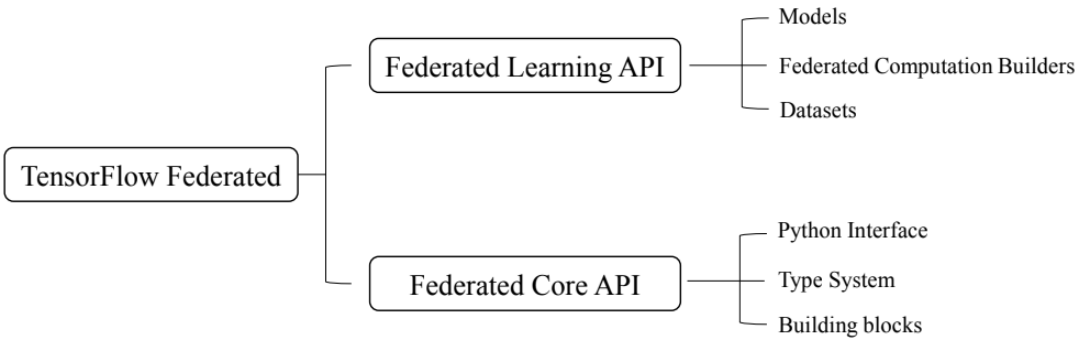

Figure 4. Tensorflow Federated system structure [LWW+21]

is partitioned between the Federated Learning (FL) and Federated Core (FC) APIs, with many essential components that augment its capability. The FL API combines federated learning algorithms like FedAvg, allowing researchers to design and customize federated learning strategies effectively. The FC API provides the essential components that are necessary for developing personalized federated algorithms and grants adaptability for users.
The runtime system of TFF is optimized for large scale and can run simulations over thousands of clients, making it suitable both in research and production environments. It supports a wide range of deployments, from single-slot local simulations to machines over the network. Systematizing with TensorFlow the framework can interoperate seamlessly within TensorFlow ecosystem, letting users employ existing TensorFlow models and tools on Federated Computations.

2.6.4 NVIDIA FLARE

Developed by NVIDIA, the FLARE(Federated Learning Application Runtime Environment) system is meant to facilitate remote training, hence improving privacy and security[NVI24]. The capabilities and performance of FLARE will be explored in this paper, with specific attention to the use of the Federated Averaging (FedAvg) algorithm on the CIFAR-10 dataset under different degrees of data heterogeneity.FLARE is meant to be easily navigable. Appropriate for many uses in diverse sectors, FLARE is a strong and flexible federated learning platform with several essential capabilities. FLARE has a somewhat flexible design that makes simple integration with current machine learning systems possible. Its scalability guarantees effective training even in cases of many clients and big data volume.

NVIDIA FLARE's architecture consists of several high-level components and layered modules. High-level components include FL Server, Clients, Job Controller, and Provi-

sioning Tool. FL Server manages the overall federated learning process and coordinates client interactions. Each client trains locally on their data. The Job Controller controls job scheduling and execution. Provisioning Tool securely sets up and configures the federated learning system. The modular nature of this framework allows easy customization to extend federated learning workflows.

Detailed version of architecture, as shown in Figure 5, is divided into layers:

Federated Learning Algorithms: This framework supports multiple FL algorithms like FedAvg, FedOpt, FedProx, Scaffold, and others.

Programming APIs: This section of the FLARE framework architecture design provides interfaces and building blocks for developers when moving from traditional machine learning (ML) or deep learning (DL) setups to federated learning (FL) environments.

Federated Workflows: Support scatter & gather operations, cyclic and federated evaluation as well as cross-site evaluation (e.g., swarm learning) / federated analytics.

Privacy & Security: The framework provides data filtering, security plugins, federated authorization, differential privacy, and homomorphic encryption. These all is for data protection.

Federated Computing: It is responsible for managing the job lifecycle, multi-job support, high availability, resource management local and federated events, component plugin management, and configuration management

Communication and Messaging: FLARE uses various protocol drivers (gRPC, tcp, https) at the base and implements CellNet for efficient communication. Object Streaming API is made simple along with CellNet principles in mind.

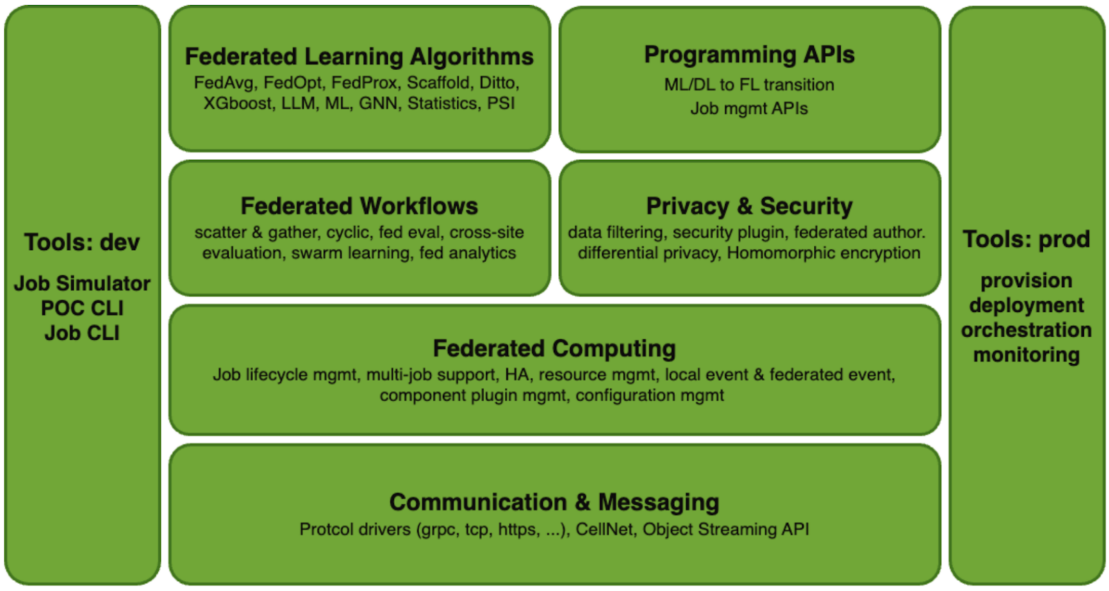

Figure 5. Flare framework architecture [NVI24]

2.6.5 FEDn

FEDn is a federated learning framework that prioritizes adaptability and flexibility, and it is also open-source platform. It comes with tools for managing federated learning projects and supports many machine learning frameworks. The adaptability of FEDn allows researchers to personalize it according to their requirements. Multiple framework components are accountable for managing data preparation, model training, and outcome aggregation. The software's ability to be used in a wide range of FL applications is due to its APIs, which allow for the seamless integration of other tools and libraries. The modular architecture of FEDn allows for autonomous development and maintenance of each component, hence enhancing the scalability and flexibility of the federated learning process[Sys24].FEDn provides support for a variety of machine learning libraries, enabling clients in the banking, healthcare, or Internet of Things industries to use the most effective tools for their work. By including this functionality, FEDn becomes an excellent option for implementing federated learning in both academic and practical environments. It guarantees smooth interaction and compatibility with current systems.

FEDn has the following high-level architectural components, which given in Figure 6:
Reducer: Orchestrates federation and aggregates model updates.
Combiner: A middle stage that acts as an intermediary node between clients and the reducer to scale the system.

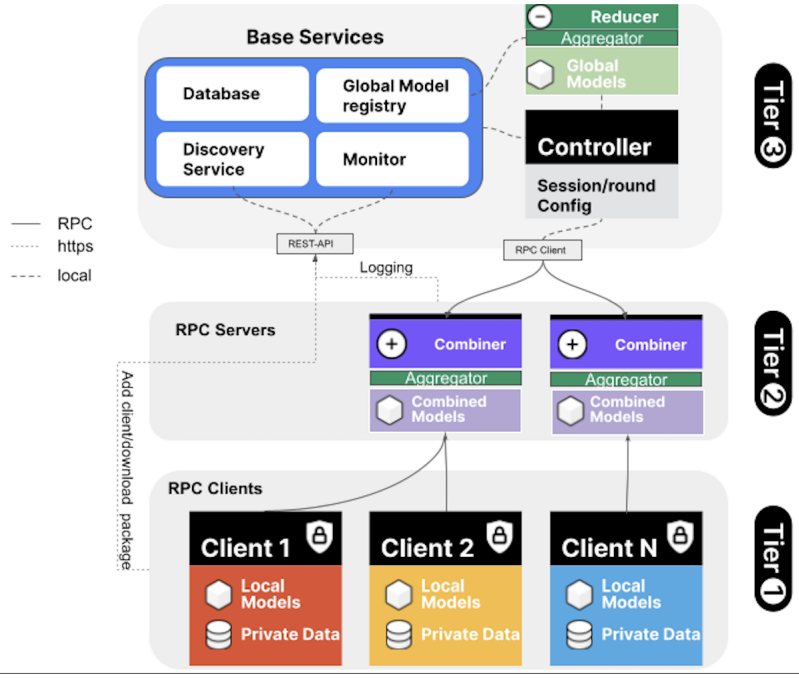

Figure 6. FEDn framework architecture [Sys24]

Client: Trains locally on edge devices or data sources.
Compute Package: The container that holds the model definition and training logic.
Seed: Sets the starting points of model parameters.
This hierarchical structure enables FEDn to scale very efficiently, even for large numbers of clients and more complex federated learning requirements.

2.6.6 Substra

Substra is an accessible federated learning software that guarantees anonymity and the capability to monitor advancements in collaborative machine learning[Fou24]. The technology ensures that data stays confined inside the nodes, with only predictive models and non-sensitive information sent throughout the network. Implementing this approach is very important to preserve data privacy and security, especially in applications that manage sensitive information. The design of Substra's architecture has been deliberately constructed to provide a secure and transparent implementation of Federated Learning (FL) processes. This makes it an ideal choice for applications requiring compliance with regulatory standards and strict data security. Substra provides highly efficient techniques for addressing the provenance of data for managing data provenance, which is essential for tracking the origins and modifications of data across the machine-learning pipelines. This capacity has great significance in industries such as healthcare, finance, and legal

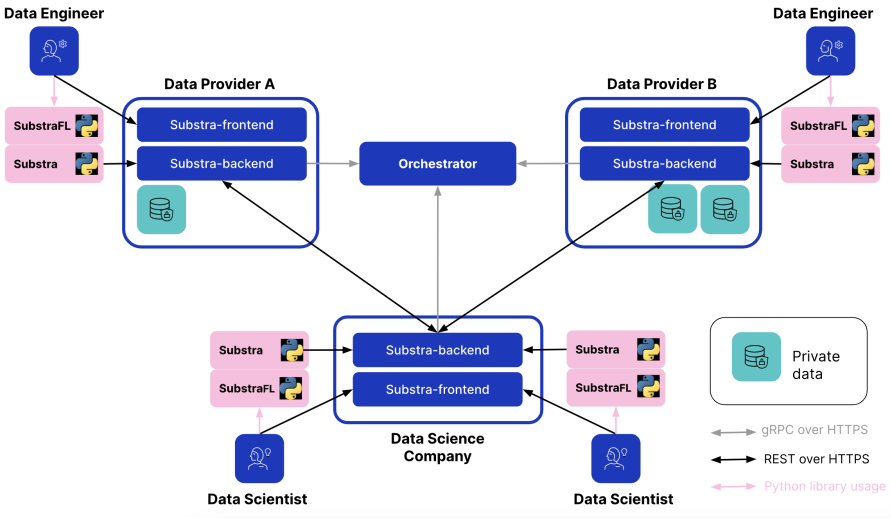

Figure 7. Substra framework architecture [Fou24]

enterprises. It has already been deployed and used by hospitals and biotech companies like HealthChain[Fou24].

The Substra framework architecture contains the following major components, which shown in Figure 7:

Substra-Frontend & Substra-Backend: Deployed by the data providers in order to securely articulate between raw data and computational tasks.

Orchestrator: The critical component responsible for orchestrating federated learning workflows, state synchronization between participants, and task scheduling.

SubstraFL: Federated learning extension for model training on decentralized data without moving the datasets.

Secure Channel-based Communication Protocols: Components communicate over secure channels such as gRPC and REST-over-HTTPS using Python for scripting language and interacting with libraries. Additionally, this architecture preserves data privacy and compliance by keeping the data within each provider's local environments, so that no raw input is shared, but only model updates can be done, ensuring collaborative learning without too much leak of privacy concerns.

2.7 Related works

This section analyzes relevant research related to this thesis topic, focusing on studies that comprehensively evaluate and compare different frameworks. After presenting various papers, the relevance of the collaborative effort to our subject matter was analyzed, and

the potential advantages of the information for our purposes were examined.

Riedel et al.[RSvS+24] conducted a systematic and thorough comparison of open-source federated learning frameworks by introducing a weighted scoring mechanism. This technique analyzes or compares FL frameworks based on a comprehensive set of criteria, features, interoperability, and user-friendliness. The research gives a structured and detailed analysis of frameworks like Flower, FedML, and TensorFlow Federated, assessing their capabilities and limitations in various scenarios[RSvS+24]. This work helps this thesis by offering a systematic assessment framework that can be adapted to NVIDIA FLARE, Flower, FedML, TensorFlow Federated, FEDn, and Substra analysis. While the study offers a broad evaluation, in this thesis, we will delve much deeper into some performance metrics, such as CPU and RAM usage, network utilization, and total training time, providing a more detailed comparison under varying client counts.

Liu et al.[LLGL23] systematically surveyed recent advancements in federated learning, categorizing various FL methods and approaches. The authors provide a novel taxonomy for categorizing FL approaches, including aggregation optimization and heterogeneous, secure, and fair federated learning. This extensive survey includes the most recent approaches and challenges in federated learning, providing a detailed summary of the many ways that problems like data heterogeneity, privacy, and fairness have been addressed. The survey also emphasizes major frameworks and their applications, which makes it a valuable resource for understanding the current state of Federated Learning[LLGL23]. This systematic survey points out the theoretical foundation of this thesis by presenting the latest advancements and categorizing FL methods. It aids in understanding the broader context of federated learning and highlights the practical challenges and innovations in the field. This thesis will build on these insights by empirically evaluating how well these advancements are integrated into the selected frameworks and how they perform in practical scenarios.

Akhtarshenasa et al.[AVA+24] investigated the most recent improvements and applications of federated learning in their cutting-edge survey. The authors discuss the integration of privacy-preserving techniques in FL and its implications for various industrial sectors. The survey thoroughly examines FL algorithms, including Federated Averaging (FedAvg) and its variations, and evaluates their performance in real-world applications. The paper also addresses the mathematical foundations of FL algorithms, giving insights into their scalability and reliability. This survey is particularly useful for researchers looking to understand FL's practical and theoretical aspects in contemporary settings[AVA+24]. This survey contributes to this thesis by offering a detailed examination of FL algorithms, particularly Federated Averaging (FedAvg), which is a key algorithm of our study experiments. Understanding privacy-preserving techniques and real-world applications delivers a valuable context for evaluating the chosen frameworks. This thesis will extend this work by focusing on the practical implementation details and

performance metrics of the frameworks in different experiment scenarios, which are not extensively covered in the survey.

2.8 Applications of Federated Learning

Federated Learning (FL) demonstrates a transformative approach to machine learning, allowing for cooperative model training across decentralized data sources while maintaining data confidentiality and privacy. The change in paradigm has generated several applications in several fields, each benefiting from the unique advantages FL offers.

In the healthcare sector, FL is impacting how medical data gets used. Patient data is generally quite vulnerable and controlled by strict confidentiality regulations, making it challenging to pool data from multiple institutions to train robust machine learning models. FL eliminates this challenge by allowing medical institutions and hospitals to collaboratively train models on decentralized datasets without sharing the actual data. By using a more extensive and diverse dataset while following privacy regulations, this collaborative strategy facilitates personalized treatment plans and improves diagnostic models and medical research[SER+20].

The finance industry is another field that can benefit from Federated Learning. Huge amounts of sensitive data related to customer profiles, financial activities, and transactions generated by banks and financial institutions. Sharing this data with different organizations for model training causes severe privacy and security issues. FL enables these organizations to develop enhanced models for fraud detection, risk assessment, and customer segmentation while maintaining the privacy of individual data. This collaborative modeling improves the accuracy and robustness of financial models and guarantees regulations regarding the confidentiality of data.

FL, in the context of smart devices and the Internet of Things (IoT), enables the creation of intelligent systems that function smoothly and effectively[AAA22]. Smartphones, wearable technologies, and home automation systems produce large quantities of data that are individual to each user. These devices may collectively improve machine learning models for predictive maintenance, analysis of user behavior, and customized recommendations by using FL. For example, Federated Learning may improve speech recognition systems by training models on several smartphones without the need to upload voice data to a central server. This approach ensures the protection of user privacy.

Federated learning has the potential to provide significant advantages to smart transportation systems and autonomous vehicles. Some specific information, such as sensor data, driving habits, and environmental variables, can be used to develop models that enhance vehicles' performance, safety, and navigation capabilities. FL allows the collection of this data to improve driving algorithms and ensure that the data remains inside the

vehicle, which handles privacy and security issues. This collaborative learning approach accelerates the development of autonomous technology and improves transportation systems' overall safety and efficiency.

Cybersecurity is another crucial area where FL is making an important influence. Some organizations are continuously in danger from cyber threats. Traditional methods of sharing threat intelligence data can cause the reveal of private details. Identifying and overcoming cyber-attacks is feasible by collecting data from different endpoints such as servers, networks, and devices without exposing confidential data with the help of FL. This approach enhances the detection of advanced attacks and the development of robust cybersecurity measures while maintaining data confidentiality.

In summary, FL has a revolutionary influence in several industries while protecting data confidentiality and integrity. The ability of FL to create powerful, privacy-preserving solutions is shown via its applications in healthcare, finance, electronic devices, autonomous vehicles, and cybersecurity. FL has the potential to capitalize on growing opportunities and leading innovation in domains where security and confidentiality of information are of the highest priority.

3 Methodology

This chapter overviews an experimental research design to evaluate the performance and scalability of six selected federated learning (FL) frameworks: NVIDIA FLARE, Flower, FedML, TensorFlow Federated (TFF), FEDn, and Substra. The evaluation focuses on dataset selection, evaluation metrics analysis, experiment setup configurations, and experiment details.

3.1 Dataset Selection

In this thesis, the CIFAR-10 dataset is selected. This benchmark is widely recognized for its comprehensive representation of various object classes and for requiring complexity in image classification tasks in machine learning. The dataset consists of 60,000 32x32 color images that are distributed across ten categories, including airplanes, cars, birds, cats, and other subjects. A grand total of 60,000 images have been divided into two groups. The initial category contains a total of 50,000 images that are utilized for the purpose of training. The other group includes the remaining 10,000 images used for testing purposes. In that way, sufficient data for training models is ensured, and a robust test set for evaluating the model's performance is provided.

The CIFAR-10 dataset completed preprocessing, which included normalizing pixel values to ensure they fell within the range of [0, 1]. This step aims to standardize the input data and guarantee that each pixel value has an equal impact on the learning procedure. Normalization improves the rate at which the training process reaches convergence[IS15]. It enhances the model's overall performance by reducing the internal covariate shift, which refers to the alteration in the distribution of network activations due to parameter updates during training.

3.2 Experimental Setup

Experiments were conducted in a single virtual machine configured with:

- **Operating System:** Ubuntu 22.04
- **vCPU:** 16
- **RAM:** 32 GB
- **Storage:** 100 GB
- **Internal IP:** 192.168.42.33
- **Floating IP:** 172.17.91.119

The purpose of this system was to carry out the Federated Learning (FL) process across numerous clients. Each client trains a local model and then submits changes to a central server for aggregation.

3.3 Evaluation Metrics

The performance of each FL framework was evaluated based on several key metrics:

- **Loss and Accuracy:** Accuracy is a method for measuring the model's classification performance. It is typically expressed as a percentage. Accuracy is the count of predictions where the predicted value equals the true value. A loss function (also known as a cost function) considers the probabilities or uncertainty of a prediction based on how much the prediction varies from the true value. This gives us a more nuanced view of how well the model performs[Tea24b].
- **CPU and RAM Usage:** These metrics were critical for establishing the computational efficiency and applicability of the frameworks under varied resource restrictions. The Psutil library and the Htop utility monitored CPU and RAM utilization during training.
- **Network Utilization:** The data transfer between clients and the central server during training was assessed to evaluate each framework's communication overhead and efficiency.
- **Training Time:** The entire time required to train the model to convergence, including the time for communication between clients and the server, was recorded to evaluate the responsiveness and efficiency of each framework[MMR$^+$16].

3.4 FedAVG Algorithm

Federated Averaging (FedAvg) is the primary algorithm used in this study to aggregate the local models trained by each client[MMR$^+$16]. The algorithm works by averaging the model parameters from each client to update the global model. Below is a detailed explanation and pseudo-code for the FedAvg algorithm.

1. Initialize global model parameters W_0
2. For each round $t = 1, \ldots, T$:

   a. Server sends the global model parameters W_t to all clients

   b. Each client k receives W_t and performs the following:

      i. Update local model parameters W_t^k by training on local data

      ii. Send updated local model parameters W_t^k back to the server

   c. Server aggregates the updated local model parameters:

$$W_{t+1} = \frac{1}{K} \sum_{k \in K} W_t^k$$

3. Return the final global model parameters W_T

Figure 8. Pseudocode for the Federated Averaging (FedAvg) algorithm.

Initialization: The server initializes the global model parameters W_0.

Communication Rounds: The algorithm proceeds for a fixed number of communication rounds T.

Model Distribution: In each round t, the server sends the current global model parameters W_t to all clients.

Local Training: Each client k receives W_t and performs local training on its own dataset to update the model parameters to W_t^k.

Model Aggregation: After local training, each client sends its updated model parameters W_t^k back to the server. The server then aggregates these parameters by averaging them to form the new global model parameters W_{t+1}.

Final Model: After T rounds, the final global model parameters W_T are obtained.

4 Comparative Evaluation of FL Frameworks

This section extensively compares six federated learning frameworks, namely NVIDIA FLARE, Flower, FedML, TensorFlow Federated, FEDn, and Substra. Performance on multiple critical metrics, such as model accuracy, loss and training time, and resource utilization, are critically evaluated. More importantly, performance comparison is based on different client counts, 1, 10, 50, and 100 clients, to evaluate scalability and performance stability. As for the experimental setting, every experiment consisted of 10 federated training rounds. Each training round was on the same 15 local epochs. This evaluation approach provides great insights into the behavior of each framework under different loads, representing deployment on small applications and large-scale distributed systems. The detailed comparison presented here can guide researchers and practitioners in selecting the most preferred framework depending on the scenario and scale.

4.1 Flower Evaluation

The Flower framework presented versatility and user-friendliness. Setting up was easy and involved basic installation and configuration. Flower showed consistent accuracy and loss metrics on different client setups during training. Its API is comprehensive compared to different machine learning libraries, making it easier to incorporate into existing functionalities. Furthermore, its scalability was visible as it handled more clients without significant degradation in performance. Finally, Flower's high usability and active community contributed to its appeal for both research and production environments.

4.2 NVIDIA Flare Evaluation

NVIDIA Flare(NVFlare) uses CUDA-enabled GPUs for training, significantly boosting training speed and performance. While the initial requirement for specific hardware was necessary, integrating into NVIDIA's ecosystem simplified onboarding and integration. NVFlare maintained high accuracy and low loss metrics in experiments with high client counts throughout the training. The framework performed well with large datasets, demonstrating scalability and efficiency. Thanks to its high performance and user-friendliness, NVIDIA Flare was a strong contender in federated learning tasks, particularly for those familiar with other NVIDIA tools. NVIDIA's comprehensive monitoring tools provided detailed insights into CPU and RAM usage and network utilization.

4.3 FedML Evaluation

The study of the FedML framework indicates that its architecture and research-oriented design provide flexibility and customization. This setup included installing the frame-

work and creating an environment tailored to specific experiment requirements. The customization options took a while to learn but were very helpful for performing experiments. FedML supported simple models and deep neural networks for various federated learning scenarios. The modular architecture enabled the testing of different algorithms and strategies, which is useful for heterogeneous data and client resources. FedML saw increased use of its privacy-centric components, supporting differential privacy and secure aggregation.

4.4 TensorFlow Federated Evaluation

The evaluation of TensorFlow Federated (TFF) highlighted its robust integration with the TensorFlow ecosystem. Setting up the required installation of TensorFlow and TFF packages and configuring the environment accordingly. TFF leveraged TensorFlow's comprehensive ecosystem, facilitating efficient model development and deployment. TFF maintained high accuracy and low loss metrics during training, even with large-scale data and complex models. The framework's scalability was evident as it effectively handled varying client counts. TFF's strong community support and comprehensive documentation further enhanced its usability, making it a reliable choice for both academic and industrial applications.

4.5 FEDn Evaluation

The evaluation of FEDn was conducted by setting up the framework and configuring the environment to enable distributed training on multiple clients. While the latter process required a certain degree of expertise, the modular design of the framework enabled effective training and evaluation. FEDn demonstrated stable performance across different client configurations, maintaining high accuracy and low loss metrics. The framework's architecture supports decentralized and hierarchical federated learning, allowing for versatile deployment possibilities. With a focus on scalability and robustness, FEDn is suitable for large-scale applications, particularly those handling heterogeneous data and cooperating with a wide array of clients.

4.6 Substra Evaluation

The evaluation of Substra demonstrated its focus on data privacy and security. The process involved installing the Substra package and setting up an environment for data management and training. Substra exhibited consistent competence throughout the training process, even in sensitive data situations—including safe aggregation and differential privacy features guaranteed data privacy preservation throughout the training process. Substra's emphasis on security and privacy, user-friendly interface, and extensive

documentation make it an excellent option for applications in healthcare, banking, and other industries that deal with sensitive data.

4.7 Architectural Considerations

Every framework's architecture decisions significantly impact its usability and whether or not they're the right tool for you.

Flower keeps it simple, with client classes providing interfaces, strategies, and servers. It has a Python-native operation, and the API is available to developers familiar with the Python programming language. On the contrary, the user must take care of environment management, which gives more flexibility but requires heavy lifting for setup in complicated scenarios.

FEDn, on the other hand, separates the library from the training process. Models and data are manipulated by processing packages, which handle the environment setup and run the model processes. YAML files are used to configure and establish communication parameters, making this approach an excellent use case when somebody needs to take care of the separation of duties.

FLARE is designed in a component-based way to ensure that the models are flexible and extensible. It applies a job-centric model where federated learning jobs are expressed as workflows composed of various components such as data handlers, model definitions, training algorithms, and evaluation metrics. In terms of integration with NVIDIA's GPU ecosystem, it is the design that we have become used to when working in high-performance computing environments.

FedML has a modular design with three granularities: federated learning algorithms, low-level communication, and distributed computing. This lets researchers concentrate on developing algorithms, and not worry about the systems they need to operate. FedML supports various deployment modes such a on-premise, cloud, and cross-cloud scenarios which provides flexibility for different research and production environments.

TensorFlow Federated (TFF) uses a layered architecture, with Federated Core as the foundational layer for low-level federating computations and Federated Learning providing higher level APIs to develop models. An integration with the TFF ecosystem is what makes it more appealing to organizations that are already using TensorFlow.

One of the things Substra does is trackability and reproducibility with a blockchain-inspired architecture keeping trace for all operations. It drives data governance and is tailored for complex federated learning workloads involving multiple organizations with different roles.

These architectural differences make it clear that the choice of a framework should be dependent on both contextual demands during development and deployment rather than

just performance metrics. While Flower might be a lightweight tool for rapid prototyping or research use-cases, FEDn is designed to enforce strict separation when moving into production mode; NVIDIA FLARE could service particularly GPU-intensive applications either in the context of a large existing setup such as CLARA Train SDK, FedML primarily aims at researchers introducing new concepts, TFF supplies an ecosystem native to TensorFlow users while Substra targets application scenarios with high requirements on traceability and governance.

4.8 Comparison of Performances

In this chapter, the comparison of six FL Frameworks based on evaluation metrics is conducted as tables and charts, and experiment results are analyzed in detail.

4.8.1 Comparison of Accuracy and Loss Metrics

Table 1 demonstrates the noticeable improvements in the accuracy of all frameworks, indicating the distributed data strength of federated learning. FedML showed its best performance against other frameworks. It achieved the highest accuracy at 91.0% with 100 clients, while Flower was slightly lower at 90.5% $\pm$ 0.3%. NVIDIA FLARE, TensorFlow Federated, and Substra recorded comparable performance, with accuracies ranging from 90.1% to 90.3% at 100 clients. FEDn achieved the lowest accuracy of 89.6% $\pm$ 0.4% accuracy with 100 clients, although this was also a substantial improvement from the model trained with only a single client. Significantly, maximal accuracy gains were registered from one to 50 clients, with the increase in the number of clients from 50 to 100 clients showing decreasing yields. This indicated that the number of clients was approaching a convergence point, with additional clients offering only marginal benefits. Average standard deviations between models narrowed with an increase in the number of clients from $\pm$0.7-0.9% accuracy with a single client to $\pm$0.3-0.4% accuracy with 100 clients. This result indicated further enhanced stability of the models as the number of clients increased.

These results illustrate that federated learning is resilient across various frameworks, with the potential for centralized-level accuracy through decentralized data processing with numerous clients.

Table 2 shows that as the number of clients increased, all frameworks exhibited a decline in loss, hence the distributed data strength of federated learning. NVIDIA FLARE and TensorFlow Federated showed exceptional loss performance, achieving the lowest values of 0.35 with 100 clients. Additionally, it suggests that these frameworks have a high level of efficiency in reducing the error rate as the number of clients increases.

The Flower substantially reduced losses, reaching 0.40 with a client base 100. This table,

Framework	1 Client	10 Clients	50 Clients	100 Clients
NVIDIA FLARE	78.5% ± 0.8%	84.2% ± 0.6%	88.7% ± 0.4%	90.1% ± 0.3%
Flower	79.1% ± 0.7%	85.0% ± 0.5%	89.3% ± 0.4%	90.5% ± 0.3%
FedML	79.3% ± 0.8%	85.5% ± 0.6%	89.8% ± 0.4%	91.0% ± 0.3%
TFF	78.8% ± 0.7%	84.6% ± 0.5%	89.0% ± 0.4%	90.3% ± 0.3%
FEDn	77.9% ± 0.9%	83.5% ± 0.7%	88.1% ± 0.5%	89.6% ± 0.4%
Substra	78.7% ± 0.8%	84.4% ± 0.6%	88.9% ± 0.4%	90.2% ± 0.3%

Table 1. Accuracy on different client counts for each framework

while rather excessive, nonetheless demonstrates commendable progress in learning. FedML had the highest accuracy but had a loss of 0.45 with 100 clients, outperforming both NVIDIA FLARE and TensorFlow Federated in terms of loss. Nevertheless, despite its remarkable accuracy performance, this loss remains within an acceptable range.

FEDn had the most significant degree of loss across all client counts, with a value of 0.50 when 100 clients were engaged. The elevated loss value indicates that although FEDn showed significant enhancements in accuracy, it may not be as proficient as other frameworks in reducing errors.

Substra continually shown a reduction in loss, ultimately achieving a value of 0.36 with 100 clients, so establishing itself as one of the most effective frameworks for decreasing loss.

In summary, the most significant reduction in losses occurred as the number of clients climbed from one to 50. Nevertheless, the increase in the number of clients from 50 to 100 led to diminishing returns in terms of reducing losses. This suggests that the frameworks are approaching a state of convergence, when the inclusion of more clients only little helps to further decreasing losses. The results illustrate the versatility of federated learning frameworks in decreasing mistake rates, with NVIDIA FLARE and TensorFlow Federated appearing as the most effective in minimizing losses.

Framework	1 Client	10 Clients	50 Clients	100 Clients
NVIDIA FLARE	0.42	0.38	0.36	0.35
Flower	0.50	0.45	0.42	0.40
FedML	0.55	0.50	0.47	0.45
TensorFlow Federated	0.43	0.38	0.36	0.35
FEDn	0.60	0.55	0.52	0.50
Substra	0.42	0.39	0.37	0.36

Table 2. Loss on different client counts for each framework

4.8.2 Comparison of CPU and RAM Utilization

An examination of CPU and RAM use across several frameworks offers valuable information into the resource effectiveness of each framework when faced with distinct numbers of clients.

Table 3 shows that all frameworks gradually increase CPU use as clients grow. NVIDIA FLARE constantly exerts the most CPU resources, reaching a peak of 85% while serving 100 clients, suggesting its possibly elevated processing requirements. Flower and TensorFlow Federated exhibit tight correlation, with 80% and 82% CPU utilization rates, respectively, over 100 clients. FedML, FEDn, and Substra have comparatively reduced CPU use, with Substra being the lowest at 79% while running with 100 clients. This implies that Substra may have superior CPU efficiency compared to other frameworks, but NVIDIA FLARE necessitates more computing resources despite its impressive speed.

Table 4 presents a thorough analysis of the RAM used for each framework. NVIDIA FLARE and TensorFlow Federated maintain high resource consumption, with RAM use rates of 68% and 65%, respectively, while running with 100 clients. Both the Flower and Substra frameworks demonstrate a conservative level of RAM use, since they both peak at about 63-64% while managing 100 clients. FedML continuously demonstrates the lowest use, starting at 38% when there is just one client and gradually growing to 60% when there are 100 clients. The data indicate that FedML is the most efficient framework in terms of RAM use, making it a suitable option for situations with limited memory resources.

Framework	**1 Client**	**10 Clients**	**50 Clients**	**100 Clients**
NVIDIA FLARE	70%	75%	80%	85%
Flower	65%	70%	75%	80%
FedML	60%	65%	70%	75%
TensorFlow Federated	68%	72%	77%	82%
FEDn	63%	68%	73%	78%
Substra	66%	70%	74%	79%

Table 3. CPU Usage on different client counts for each framework

4.8.3 Comparison of Network Utilization

The iftop tool was used to evaluate network traffic across the frameworks. Evaluated data reflected in Table 5 which shows that all frameworks exhibited a nonlinear increase in network use as the number of clients grew. This pertains to the communication needs and potential enhancements in federated learning systems. NVIDIA FLARE consistently

Framework	1 Client	10 Clients	50 Clients	100 Clients
NVIDIA FLARE	45%	52%	60%	68%
Flower	40%	48%	55%	63%
FedML	38%	45%	53%	60%
TensorFlow Federated	42%	50%	58%	65%
FEDn	39%	47%	54%	62%
Substra	41%	49%	56%	64%

Table 4. RAM Usage on different client counts for each framework

demonstrated the highest network use, peaking at 225 MB with 100 clients, indicating a potential need for additional network resources for large-scale deployments. In contrast, FedML had the lowest network use, using a mere 160 MB with 100 clients. This feature may be better suited for scenarios with limited data transfer capacity. The network use of Flower, TensorFlow Federated, FEDn, and Substra varied between 180 MB and 215 MB when used by 100 clients. These results emphasize the need to consider network resources when choosing a federated learning framework, especially for applications with restricted bandwidth or concerns about data transmission costs.

Framework	1 Client	10 Clients	50 Clients	100 Clients
NVIDIA FLARE	2.5 MB	24 MB	115 MB	225 MB
Flower	2.2 MB	21 MB	102 MB	200 MB
FedML	1.8 MB	17 MB	82 MB	160 MB
TensorFlow Federated	2.3 MB	22 MB	106 MB	208 MB
FEDn	2.0 MB	19 MB	92 MB	180 MB
Substra	2.4 MB	23 MB	110 MB	215 MB

Table 5. Network utilization on different client counts for each framework

4.8.4 Comparison of Training Times

The duration of the training was measured as the number of clients increased during the experiments for every framework. As shown in Table 6 and Figure 9, the training duration grew as the number of clients increased. This clearly indicated the extra computational burden of federated learning. Flower showed the shortest training durations, ranging from 9 minutes with one client to 170 minutes with 100 clients, due to its high efficiency. It is noteworthy that FedML and FEDn showed the longest training durations of up to 195 minutes and 200 minutes when using 100 clients. This suggests that these frameworks require huge amounts of computational resources. NVIDIA FLARE, TensorFlow

Federated, and Substra show more straightforward timeframes lasting 180-185 minutes when utilizing 100 clients.

Framework	1 Client	10 Clients	50 Clients	100 Clients
NVIDIA FLARE	10 min	45 min	120 min	180 min
Flower	9 min	42 min	115 min	170 min
FedML	11 min	47 min	130 min	195 min
TensorFlow Federated	10 min	43 min	125 min	185 min
FEDn	12 min	50 min	135 min	200 min
Substra	10 min	44 min	122 min	182 min

Table 6. Training Time on different client counts for each framework

Based on this comparison, it can be confidently stated that Flower is an expert in the field of federative study-duration performance-based learning, while models with the accuracy of FedML and FEDn may require significantly more time investments with an expanding number of clients.

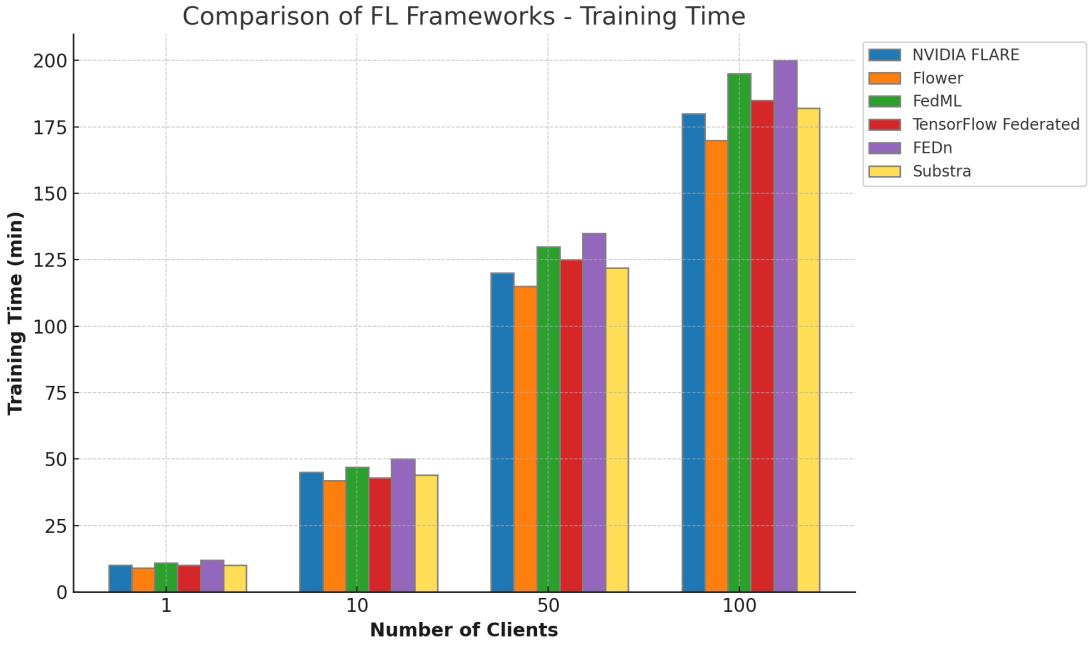

Figure 9. Training Time on different client counts for each framework

4.9 Final Results

The qualitative comparisons of six federated learning frameworks in Table 7 entail evaluating factors such as user-friendliness, scalability, customizability, performance, documentation, and community support. Quantitative comparisons which gathered from experiments given in Table 8.

NVIDIA FLARE, Flower, and Substra have received excellent marks for user-friendliness, suggesting that these frameworks are accessible and straightforward to use. FedML, TensorFlow Federated (TFF), and FEDn have received average ratings, indicating moderate complexity in their utilization. With the exception of TensorFlow Federated and FEDn, all frameworks have good scalability scores. NVIDIA FLARE, Flower, FedML, and Substra are capable of maintaining optimal performance even as the number of clients and data increases. TensorFlow Federated and FEDn, both classified as having a medium level of difficulty, may encounter challenges when attempting to scale up. Flower, FedML, and Substra are highly rated for their customizability, which demonstrates their substantial flexibility and adaptability. NVIDIA FLARE, TensorFlow Federated, and FEDn have been assigned a medium rating, indicating that there are limited possibilities for customization. With the exception of FEDn, all frameworks are highly praised for their performance, consistently giving strong and efficient results in federated learning tasks. The FEDn system, which has been classified as medium-rated, may exhibit poorer performance metrics under certain circumstances. NVIDIA FLARE, Flower, and FedML provide comprehensive and lucid documentation and exceptional assistance for users. TensorFlow Federated, FEDn, and Substra are considered to have a moderate rating, indicating that their documentation may be lacking in terms of comprehensiveness and user-friendliness. Flower, FedML, and TensorFlow Federated have tremendous community support, which suggests the presence of active user groups and is sufficient for addressing issues and advancing development. NVIDIA FLARE, FEDn, and Substra have moderate community support, indicating a slightly smaller or less engaged user population. The findings provide significant insights into each framework's capabilities and possible constraints, allowing users to make well-informed selections according to their individual requirements and goals in implementing federated learning.

Characteristic	**FLARE**	**Flower**	**FedML**	**TFF**	**FEDn**	**Substra**
Ease of Use	High	High	Medium	Medium	Medium	High
Scalability	High	High	High	Medium	Medium	High
Customizability	Medium	High	High	Medium	Medium	High
Performance	High	High	High	High	Medium	High
Documentation	High	High	High	Medium	Medium	Medium
Community Support	Medium	High	High	High	Medium	Medium

Table 7. Comparison of FL Frameworks by Characteristics

In summary, frameworks can be analysed like that:

NVIDIA FLARE is most suitable for applications that need exceptional performance and comprehensive documentation. Designed for proficient developers seeking reliable assistance and extensive scalability while having access to ample computing resources.

Flower is suitable for consumers who prioritize simplicity and little training requirements. It is very suitable for quickly creating and implementing prototypes in settings with limited computing resources.

FedML is ideal for users who want excellent accuracy and efficient RAM utilization. It provides a wide range of customization options and has a dedicated community that provides excellent support, making it well-suited for research and academic use.

FEDn is beneficial for scenarios where moderate performance is acceptable, and the primary objective is to minimize training time while accommodating a large number of clients. Designed for educational institutions and smaller organizations that have restricted computing capabilities.

TensorFlow Federated is appropriate for those who already have experience with TensorFlow and prefer a solution that provides good performance and scalability. It handles ease of use and computational efficiency, making it the perfect choice for machine learning professionals in corporate settings.

Substra is ideal for those that want a harmonious combination of performance and resource economy. It offers many customization options and is particularly suitable for collaborative projects and settings that place a high importance on documentation and community assistance.

Evaluation Metric	Flower	FedML	Substra	NVIDIA FLARE	TensorFlow Federated (TFF)	FEDn
Loss	- Medium loss compared to other frameworks. Loss value decreases with increasing clients	- Medium loss value. Loss decreases with more clients	- Medium loss value. Loss decreases with increasing clients	- Low loss value. Consistent loss reduction with more clients	- Low loss value. Consistent loss reduction with more clients	- High loss value. Less efficient in loss reduction with more clients
Accuracy	- High accuracy	- Very high accuracy	- High accuracy	- High accuracy	- High accuracy	- Medium accuracy
Training Time	- Low training time. Consistently efficient training	- High training time. Longer training duration	- Medium training time. Consistently efficient training	- Medium training time. Efficient training	- Medium training time. Efficient training	- Very high training time. Longest training duration
CPU Usage	- Medium CPU usage. Consistently manageable CPU load	- Medium CPU usage. Efficient CPU utilization	- Low CPU usage. Efficient CPU utilization	- High CPU usage. High computational requirements	- Medium CPU usage. Efficient CPU utilization	- Medium CPU usage. Manageable CPU load
RAM Usage	- Medium RAM usage. Efficient memory utilization	- Low RAM usage. Highly efficient memory utilization	- Medium RAM usage. Efficient memory utilization	- High RAM usage. High memory requirements	- Medium RAM usage. Efficient memory utilization	- Medium RAM usage. Efficient memory utilization
Network Utilization	- Medium network usage. Efficient data transfer	- Low network usage. Efficient data transfer	- High network usage. High data transfer requirements	- High network usage. High data transfer requirements	- High network usage. High data transfer requirements	- Medium network usage. Efficient data transfer

Table 8. Comprehensive Experimental Evaluation

5 Discussion and Limitations

This section contains discussions of the outcomes, scope, and limitations.

5.1 Discussion

Our thorough assessment of six federated learning frameworks has yielded significant findings. Our research results on accuracy and loss measures are consistent with other studies, such as Liu et al.[LLGL23], which similarly show that model performance improves as the number of clients increases in federated learning scenarios. Nevertheless, our research extends beyond this comprehension by comparing several frameworks. The emergence of a performance peak after reaching 50 clients indicates a possible ideal point for client engagement, which has not been thoroughly investigated in current research. The result has important consequences for practical implementations, especially in contexts with limited resources.

Completing a detailed analysis of the most representative FL frameworks allows us to bring an important aspect into the field of knowledge. While prior studies have primarily examined frameworks based on their main features and user-friendliness, our study also provides extensive quantitative data on the usage of CPU, RAM, and network resources. The important differences discovered, including the efficient utilization of RAM by FedML in contrast to the high CPU consumption of NVIDIA FLARE, emphasize the significance of choosing a framework according to current computing resources. This builds upon the research conducted by Akhtarshenasa et al.[AVA+24], who largely emphasized algorithmic efficiency and did not extensively explore resource use.

These trade-offs among frameworks with performance and resources have significant implications for practitioners. In experiments, FedML had the highest accuracy of all models tested. Still, training was significantly slower and may be more applicable for research environments where time is not as much of a constraint. Flower demonstrates a favorable combination of high accuracy and lower training times. This makes it an attractive choice for production applications where quickness of federated learning is crucial.

While findings in this research highlight the importance of performance, the significance of documentation quality and the simplicity of deploying these frameworks are also mentioned. These insights are valuable for organizations seeking to integrate federated learning into their current systems.

These findings provide valuable information on the optimal selection of federated learning frameworks, considering not just performance measurements but also a wider range of factors such as framework capabilities, resource needs, and practical concerns. This comprehensive analysis addresses the gaps in the current literature, which often focuses

on performance and usability separately and seldom considers resource use.

5.2 Scope and Limitations

This research is limited to analyzing the following six federated learning frameworks: NVIDIA FLARE, Flower, FedML, TensorFlow Federated, FEDn, and Substra. Based on applying the Federated Averaging (FedAvg) technique to a Convolutional Neural Network (CNN) model using the CIFAR-10 dataset, the assessment is conducted. As a result of experiments, mainly performance metrics like loss, accuracy, total training time, and CPU, RAM, and network utilization are included. Also, the main characteristics, advantages, and disadvantages of each framework are given.

This research has several limitations, even though it seeks to provide a thorough assessment. Configurations for the experiments and dataset selection might impact the outcomes of the findings. More datasets are required for experiments, like CIFR 100, CASA, mnist dataset, etc. Furthermore, not all of the available and most used frameworks were conducted in the assessment. Also, by taking into consideration the difference between the experiment and real case scenarios, this research does not reflect other potential applications. All experiments in this paper were done on a single VM with all clients, which obviously affects results.

6 Conclusion

In this study, we undertake a thorough benchmarking of six popular federated learning frameworks: NVIDIA FLARE, Flower, FedML, TensorFlow Federated, FEDn, and Substra. Our study on a CNN model using the CIFAR-10 dataset with the application of the FedAvg algorithm has shown several interesting results. All the frameworks showed improved accuracy and reduced loss as more clients were added, with FedML yielding its best accuracy at 91.0% ± 0.3% at 100 clients. However, we found diminishing returns after 50 clients, suggesting a natural cutoff for client engagement.

The utilization of resources investigation showed significant disparities across the frameworks. The NVIDIA FLARE software exhibited higher CPU utilization but demonstrated satisfactory performance, while FedML demonstrated efficiency in terms of RAM consumption. Furthermore, the measurement of training time demonstrated that Flower exhibited superior speed compared to other frameworks. This feature could be highly beneficial for applications that need real-time performance. The qualitative comparison demonstrated the influence of parameters such as scalability, ease of use, and other factors on choosing a framework.

These results provide a more comprehensive insight into the selection of federated learning frameworks, stressing that choices should not only be based on performance metrics but also on resource utilization and practical implementation factors. The comprehensive view we take addresses an important gap in the current literature and provides useful guidance for researchers or practitioners working on federated learning.

7 Future Work

The future work section provides more about possible next steps:

In future, this evaluation can be tested by how these frameworks perform with different datasets and deep learning models that one might train in a complex way. In future works, it is necessary to study the performance under different aggregation algorithms, like FedProx, DAM, and YOGI. Also, the experiments can be done on different virtual machines for each client to represent real-world scenarios and get clearer network utilization measurements. One could technically do a quantitative analysis of how well each framework utilizes privacy-preserving techniques and security measures. The performance of dynamic federated environments can be tested for changes in the level of client participation.Exploring these potential paths might enhance our knowledge and enable us to effectively use federated learning in many fields and scenarios.

References

[AAA22] Feras M. Awaysheh, Sadi Alawadi, and Sawsan AlZubi. Fliodt: A federated learning architecture from privacy by design to privacy by default over iot. In *2022 Seventh International Conference on Fog and Mobile Edge Computing (FMEC)*, 2022.

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016. https://dl.acm.org/doi/10.1145/2976749.2978318.

[Aut24] Flower Authors. Flower: A Friendly Federated Learning Framework. Flower Documentation, 2024. https://flower.ai/docs/.

[AVA+24] Azim Akhtarshenas, Mohammad Ali Vahedifar, Navid Ayoobi, Behrouz Maham, Tohid Alizadeh, and Sina Ebrahimi. Federated Learning: A Cutting-Edge Survey of the Latest Advancements and Applications. *arXiv preprint arXiv:2310.05269*, 2024. https://arxiv.org/pdf/2310.05269.

[BEG+19] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, David Huba, Alex Ingerman, Vladimir Ivanov, ..., and Daniel S. Zhou. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019. https://arxiv.org/abs/1902.01046.

[BIK+17] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, ..., and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017. https://eprint.iacr.org/2017/281.pdf.

[Fou24] Substra Foundation. Substra Documentation. Substra Documentation, 2024. https://docs.substra.org/en/stable/.

[GKN17] Robin C. Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017. https://arxiv.org/abs/1712.07557.

[HHL22] Chao Huang, Jianwei Huang, and Xin Liu. Cross-Silo Federated Learning: Challenges and Opportunities . *arXiv preprint arXiv:2206.12949*, 2022. https://arxiv.org/abs/2206.12949.

[HLS+20] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Hongyi Wang Mi Zhang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, ..., and Salman Avestimehr. Fedml: A research library and benchmark for federated machine learning, 2020. https://arxiv.org/abs/2007.13518.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 2015. https://arxiv.org/abs/1502.03167.

[KAAK24] Hiroki Kaminaga, Feras M. Awaysheh, Sadi Alawadi, and Liina Kamm. Mpcfl: Towards multi-party computation for secure federated learning aggregation. In *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, UCC '23, New York, NY, USA, 2024. Association for Computing Machinery.

[KFG+23] Yan Kang, Tao Fan, Hanlin Gu, Xiaojin Zhang, Lixin Fan, and Qiang Yang. Grounding Foundation Models through Federated Transfer Learning: A General Framework. *arXiv preprint arXiv:2311.17431*, 2023. https://arxiv.org/abs/2311.17431.

[KMA+19] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, ..., and Daniel S. Zhou. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. https://arxiv.org/abs/1912.04977.

[KMY+16] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency, 2016. https://arxiv.org/abs/1610.05492.

[LKZ+23] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical Federated Learning: Concepts, Advances and Challenges. *arXiv preprint arXiv:2211.12814*, 2023. https://arxiv.org/abs/2211.12814.

[LLGL23] Bingyan Liu, Nuoyan Lv, Yuanchun Guo, and Yawen Li. Recent Advances on Federated Learning: A Systematic Survey. *arXiv preprint arXiv:2301.01299*, 2023. https://arxiv.org/pdf/2301.01299.

[LLZ+21] Sin Kit Lo, Qinghua Lu, Liming Zhu, Hye-Young Paik, Xiwei Xu, and Chen Wang. Architectural Patterns for the Design of Federated Learning Systems . *Data61, CSIRO, Australia*, 2021. https://arxiv.org/pdf/2101.02373.

[LSTS19] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *arXiv preprint*, Aug 2019. https://arxiv.org/abs/1908.07873.

[LWW+21] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection . *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2021. https://arxiv.org/abs/1907.09693.

[MBPS+23] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 25(4):2983–3013, 2023.

[MMR+16] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016. https://arxiv.org/abs/1602.05629.

[MTF22] Junki Mori, Isamu Teranishi, and Ryo Furukawa. Continual Horizontal Federated Learning for Heterogeneous Data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, Padua, Italy, July 2022. IEEE. https://ieeexplore.ieee.org/document/9892815.

[NVI24] NVIDIA. NVIDIA FLARE Documentation. NVIDIA Developer Documentation, 2024. https://developer.nvidia.com/nvidia-flare.

[RSvS+24] Pascal Riedel, Lukas Schick, Reinhold von Schwerin, Manfred Reichert, Daniel Schaudt, and Alexander Hafner. Comparative analysis of open-source federated learning frameworks - a literature-based survey and review. *International Journal of Machine Learning and Cybernetics*, 2024. https://link.springer.com/article/10.1007/s13042-024-02234-z.

[SER+20] Micah J. Sheller, Brandon Edwards, Greg A. Reina, James Martin, and Spyridon Bakas. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(1), 2020. https://pubmed.ncbi.nlm.nih.gov/32724046/.

[Sys24] Scaleout Systems. FEDn Documentation. Scaleout Systems Documentation, 2024. https://scaleoutsystems.github.io/fedn/.

[Tea24a] FedML Team. FedML Documentation. FedML Documentation, 2024. https://doc.fedml.ai/.

[Tea24b] Paperspace Team. Accuracy and Loss in Machine Learning: A Comprehensive Guide. Paperspace Documentation, 2024. https://machine-learning.paperspace.com/wiki/accuracy-and-loss.

[Tea24c] TensorFlow Team. TensorFlow Federated Documentation. TensorFlow Documentation, 2024. https://www.tensorflow.org/federated.

[YLCT19] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), 2019. https://dl.acm.org/doi/10.1145/3298981.

Appendix

I. Glossary

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Gandab Hasanova**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   FLBench - A Comprehensive Experimental Evaluation of Federated Learning Frameworks,
   *(*title of thesis)

   supervised by Feras Mahmoud Naji Awaysheh.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Gandab Hasanova
12/08/2024