INDIANA UNIVERSITY
BLOOMINGTON

ENGR-E516 Engineering Cloud Computing

Project Proposal

by

Anirudh Penmatcha
Dilip Nikhil Francies
Subhadra Mishra

Luddy School of Informatics, Computing, and Engineering
Indiana University Bloomington

1 Project Overview

Build a medical data assortment application by optimizing a distributed training approach for a superior Machine Learning model while preserving patients' data privacy. Some key features will be utilizing the right kind of encryption method, exploring various open source federated learning frameworks, creating efficient data and training pipelines, and finally a comparison study between a vanilla ML training approach against the decentralized federated learning method.

2 Project Introduction

Privacy is a major concern in today's world and with the advent of better computing and storage capabilities, data is being collected more than ever. Many industries are working towards improving the handling of users' data while protecting all of their sensitive information. One such industry is the medical field. We aim to build a system where both privacy and functionality are delivered while minimizing the risk of a breach in a patient's data and maximizing the advantages of technological advancements in the industry. Often this is an inversely proportional relationship. Therefore, it becomes imperative to find a method to address this.

3 Related work

Google introduced Federated Learning in 2016, initially implementing it in Google Keyboards to collectively learn from multiple Android phones [1]. With its potential to extend to various edge devices, Federated Learning holds promise in transforming crucial sectors like healthcare, transportation, finance, and smart homes. A notable instance is the collaborative development of an AI pandemic engine for COVID-19 diagnosis from chest scans by researchers and medical professionals worldwide [1]. Another compelling application lies in transportation networks, where Federated Learning can train vehicles for autonomous driving and city route planning. Similarly, in smart-home setups, edge devices across different residences can jointly learn context-aware policies using a Federated Learning framework [2]. The AWS Machine Learning Blog series explores Federated Learning on AWS, emphasizing health analytics while preserving data privacy. [4]Part 1 by Olivia Choudhury et al. established foundational concepts, while Part 2 by Vidya Sagar Ravipati et al. highlighted advancements in privacy techniques and scalability.

3.1 Federated Learning Frameworks

3.0.1 NVFlare

NVFlare, developed by Nvidia,[3] is a robust federated learning (FL) framework designed for business applications. Supporting various models such as neural networks and tree-based models, NVFlare offers framework-agnostic capabilities, enabling easy migration of machine learning (ML) models into a federated setting. Users can manage FL jobs, perform training or evaluation, and monitor operations through a command-line interface or the NVFlare Dashboard. Jobs can be scheduled or run in parallel, with orchestration facilitated through docker-compose. Security features include server-client authentication and a filtering layer for privacy methods like differential privacy or homomorphic encryption.

3.0.2 FATE

Federated AI Technology Enabler Framework (FATE) [4] is a business-ready federated learning (FL) framework released by WeBank, a Tencent subsidiary, in February 2019. It provides numerous modules for preprocessing, machine learning algorithms, and privacy methods. FATE integrates with PyTorch and TensorFlow but lacks framework agnosticism. Notable features include live visualization with FATEBoard, model serving with FATEServing, and compatibility with Spark clusters and Kubernetes.[5] However, navigation of its extensive documentation, especially in English, can be challenging. Overall, FATE is a robust FL framework with pre-built modules and additional functionality but requires effort for custom module development and may benefit from improved usability.

3.0.3 Tensorflow Federated

TensorFlow Federated (TFF) is an open-source framework designed for machine learning and computations on decentralized data. It facilitates research and experimentation with Federated Learning.TFF allows developers to simulate FL algorithms on their models and data, as well as experiment with new algorithms. It provides high-level interfaces for applying federated training and evaluation to existing TensorFlow models, along with lower-level interfaces for expressing novel federated algorithms.

3.2 Gap Analysis

We plan to develop a cloud computing service aimed at simplifying the selection of the best method and parameter combinations for training federated learning models, especially on medical imaging datasets. This service will leverage various open-source frameworks such as NVFlare and TensorFlow Federated, diverse federated learning algorithms, encryption methods, and performance metrics to provide efficient and effective model training solutions.

4 Proposed Tasks

Firstly, we'll start by constructing the storage system using open-source software. Then, our plan involves setting up the essential data pipelines to train federated learning models, using NIH- Chest X-ray datasets as our benchmark. Next, we'll fine-tune the architecture by investigating various learning algorithms, encryption methods, and hyperparameters,[6] and we'll compare the outcomes with a centralized training approach. Following that, we'll conduct a thorough examination of creating the cloud-based computing service to ensure it's ready for production deployment. Lastly, we'll focus on developing the front end using Streamlit and integrating it with our cloud services.

4.1 Next Cloud

We intend to first use an open-source tool called Next Cloud as the backbone for storage. It is a self-hosted, file sharing and collaboration platform that allows users to store, access, and share their data from any device or location. One of the primary concerns is where the data will be stored. If we have our storage system, the cloud hosting site will not have access to it.

4.2 Federated learning model

We plan to develop a custom federated learning model trained using pre-trained weights from open-source NIH Chest X-ray images. Our exploration will encompass different learning algorithms like FedAvg and FedSGD,[[7]] [8] along with encryption methods such as Homomorphic encryption and differential privacy applied to the payload. Additionally, we'll examine the impact of factors like the number of clients and whether the dataset is IID or Non-IID on the training process.

4.3 Front end

We'll create several web pages using Streamlit for the front end as required. One page will provide a user-friendly interface for uploading a small set of images to initiate the training process. Another page will enable running inferences on the uploaded data. We intend to display the results of various learning algorithms and encryption methods as graphs, allowing users to select the appropriate framework for full deployment and scalability.

4.4 Running inferences

The weights of the model are available locally for the end user to run inferences on their side. Therefore, the data can be viewed and analyzed only by the end user and no one else. We will be working on this discretely to make it work effortlessly so an end user such as a hospital administrator does not have to worry about training the lab technician to use very sophisticated software or cloud service.

4.5 Comparison of models

In conclusion, we will conduct comparisons between federated learning models and centralized training approaches. Additionally, we will analyze the proposed system for research purposes to illustrate the value of finding an approach that satisfies all requirements. This comprehensive analysis will help showcase the effectiveness and benefits of our approach across various aspects, contributing to the advancement of federated learning methodologies.

5 Team members and workload allocation

Each team member will play an equal role in configuring the experimental setup, constructing the pipelines, and setting up the required cloud services. To ensure a well-rounded understanding, every member will participate in all aspects of the project, gaining practical experience with the technologies and methodologies involved. Ultimately, we will collaborate to analyze the results and interpret them effectively, culminating in a comprehensive report that showcases our collective insights and findings.

6 Planned timeline

Week 1: Setup next cloud
Week 2: Create the federated learning architecture on the local machine and jetstream2
Week 3: Optimize the architecture and create the vanilla model >> mid-term project report
Week 4: Build front end using streamlit
Week 5: Integrating the cloud services with the front end
Week 6: Perform an analysis, design, and evaluate the proposed model and system
Week 7-8: Summarize contributions and write the project paper

References

[1] Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data. *Google Research Blog*, 3, 2017.

[2] Yongchao Xu, Liya Ma, Fan Yang, Yanyan Chen, Ke Ma, Jiehua Yang, Xian Yang, Yaobing Chen, Chang Shu, Ziwei Fan, et al. A collaborative online ai engine for ct-based covid-19 diagnosis. *MedRxiv*, 2020.

[3] Tianlong Yu, Tian Li, Yuqiong Sun, Susanta Nanda, Virginia Smith, Vyas Sekar, and Srinivasan Seshan. Learning context-aware policies from multiple smart homes via federated multi-task learning. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 104–115. IEEE, 2020.

[4] Holger R Roth, Yan Cheng, Yuhong Wen, Isaac Yang, Ziyue Xu, Yuan-Ting Hsieh, Kristopher Kersten, Ahmed Harouni, Can Zhao, Kevin Lu, et al. Nvidia flare: Federated learning from simulation to real-world. *arXiv preprint arXiv:2210.13291*, 2022.

[5] Ivan Kholod, Evgeny Yanaki, Dmitry Fomichev, Evgeniy Shalugin, Evgenia Novikova, Evgeny Filippov, and Mats Nordlund. Open-source federated learning frameworks for iot: A comparative review and analysis. *Sensors*, 21(1):167, 2020.

[6] Lutho Ntantiso, Antoine Bagula, Olasupo Ajayi, and Ferdinand Kahenga-Ngongo. A review of federated learning: Algorithms, frameworks and applications. In *International Conference on e-Infrastructure and e-Services for Developing Countries*, pages 341–357. Springer, 2022.

[7] Priyanka Mary Mammen. Federated learning: Opportunities and challenges. *arXiv preprint arXiv:2101.05428*, 2021.

[8] Ajinkya Mulay, Baye Gaspard, Rakshit Naidu, Santiago Gonzalez-Toral, S Vineeth, Tushar Semwal, and Ayush Manish Agrawal. Fedperf: A practitioners' guide to performance of federated learning algorithms. In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, pages 302–324. PMLR, 2021.