COMP4801 – FYP Detailed Project Plan

Project Title: Quantitative Performance and Security Evaluation of Federated Learning on open-sourced platforms (Industry-based Project)

Student: Fong Cheuk Ting (3035688784)

Supervisors: Professor Yiu, Siu Ming

Industrial partner: Dr. Rocky Zhang

Project Background

Along with decades of AI application development, Federated Learning (FL), firstly proposed by Google [1] in 2016 for their Gboard text suggestion application, is recently becoming a popular AI technique due to the increasing concerns about data privacy. By definition, FL is an encrypted distributed machine learning. The high-level idea of FL [2] is that rather than centralizing all the data in a central server to train the Machine Learning (ML) or Deep Learning (DL) model, each data owner trains their local model on their local nodes and sends the aggregated local gradients in an encrypted manner to the central node for training the global model. The updated global model will then distribute to all the local nodes. The communication round repeats to improve the model iteratively. Thus, the contradiction between data privacy and data sharing can be resolved. For the case of B2C (business-to-consumer), FL is useful when clients' data is very sensitive or under legal protection, such as in financial and medical applications. For the case of B2B (business-to-business), FL allows industrial entities to bridge their data silo without disclosing their own datasets such that more training data are available to improve the training performance. This is useful when valid data is scarce or mining and labeling new data are costly. It is attractive to startup companies that don't have enough historical data for their AI applications.

However, industries still haven't fully embraced FL technologies because it is difficult for them to trust the "black box" as security vulnerabilities are a real concern for FL [2]. Typical security vulnerabilities are data leakage. The experiment conducted by Wei et al. [3] shows that a malicious server can partially recover clients’ data by back deduction of the transmitted gradients, which leads to potential leakage of sensitive data. FL is also vulnerable to model poisoning attacks. It is discovered by Bagdasaryan et al. [4] that only a very small portion of malicious participants is sufficient to alter the output of the global model against specific inputs. It is done by taking the weakness of FL that a malicious participant can have full control over the local model and the server node has no visibility over the client nodes, thus cannot detect abnormal updates. Data owners and model users may worry about such security risks. Currently, there is a lack of technical references and a standard prototype to convince industry users to embrace FL. Therefore, a trustworthy official assessment and endorsement system is significant for FL implementation. Although there are already two governmental FL accreditation bodies (CAICT and BCTC) available, their influence is limited to Mainland China. And their process of evaluation is done in a labor-intensive manner and does not provide a reference API to solve the problem of platform silo.
**Project Objective**

This project is an industry-based project proposed by ASTRI (Hong Kong Applied Science and Technology Research Institute). The goal of their original project is to design an authoritative, automatic, and efficient FL evaluation platform that validates commercial FL software via returning evaluation reports and issuing certificates based on the degree of computation efficiency and security the software achieved. The evaluation system consists of 3 different levels (Reference Technical Questionnaire, Automated Remote Evaluation System, Inbound/Outbound Monitoring Sandbox) of validation processes. Each level consists of multiple software modules to qualitatively and quantitatively evaluate different aspects of the FL platform performance. Since their project is relatively big, the scope of this project will only focus on quantitative performance and security evaluation of open-sourced FL platforms (e.g. FATE, Federated Scope, Tensorflow Federated). Dr. Rocky Zhang, assigned two subtasks of their project for me to help them out:

1. **Two evaluation reports of the open-source FL platforms** (such as FATE, Federated Scope, and Tensorflow Federated). Based on the same benchmark datasets, the reports should give a set of basic investigation results on the model performance and security evaluation on the selected two FL platforms. The investigated items may include, but not limited to the model performance, computation efficiency, and basic security evaluation.

2. **A visual web portal demo to show the process of at least one kind of back-door attacks and related defenses on the FL.** The related test that whether the above three FL opensource platforms can defend the back-door attack is a plus.

These deliverables serve as the trial demo of their evaluation system.

**Project Methodology**

**1st deliverables**

We need to use the same dataset to benchmark the quantitative performance of each open-sourced FL platform. Related researches [3, 4, 5, 6] commonly use CIFAR-10 and CIFAR-100 to evaluate FL models for image classification tasks. Each contains 60,000 images with 10 classes and 100 classes respectively. Dirichlet distribution can be used to divide the images amongst clients to simulate non-i.i.d. data [5], which is a common situation for real-world FL applications [1]. However, the scalability of the platforms may not be well reflected with a small dataset and perhaps only tens or hundreds of virtual client nodes. Computational resources are not available for simulating industrial-scale FL.

The evaluation metric should follow the standard according to the "IEEE Guide for Architectural Framework and Application of Federated Machine Learning" [7]:

**Performance discrepancy**

We should compare the accuracy of the classification between the FL model, the traditional ML/DL model that centralizes all data in one place, and the traditional ML/DL model with only a "single node" amount of data. The discrepancy can be visualized via plotting a graph of accuracy against epoch [7, Fig. 1]. If the accuracy gap between the FL case and the
centralized case is reasonably small, industry users can be more convinced to adopt FL.

![FL model accuracy gap](image)

**Figure 1- FL model accuracy gap**

**Computation efficiency**
This includes training time, testing time, intrinsic memory usage, and auxiliary memory usage. Different real-world FL applications require different degrees of computational efficiency in the above four items [7]. Unlike traditional ML, FL is a distributed system. It has communication overhead that hinders the training and testing time. Communication costs can be reduced by decreasing the communication rounds between local nodes and central node and decreasing the model update time of the local node [1]. FL also has the issue of structural heterogeneity. Hardware resources (e.g. CPU, memory, download and upload speed) of each client node can vary a lot [1]. For example, if it is a B2C application where the client nodes are edge devices like a smartphone, only a light-weighted local model is feasible. But if it is a B2B application, it would be more tolerable to computation efficiency. In conclusion, the feasibilities of FL applications are very sensitive to computation efficiency.

**Security evaluation**
FL applications, by incentives, generally require a very high degree of security. For testing whether the FL platforms are effective against gradient leakage attacks. For image classification, the extent of data leakage can be measured using Mean Square Error (MSE) or Structural Similarity Index (SSIM), by comparing the similarities between the ground truth image and the leaked image [7]. Gradient leakage attacks can be simulated as we will have direct access to all the local gradients sent by the local nodes during the experiment.

For model poisoning attack, we can simulate malicious participants by feeding the local model with images containing the same specific features or pixel patterns and change the images' label to a wrong class we desire [4]. The robustness of the FL platforms against model poisoning attacks can be quantitatively evaluated by experimenting how many training rounds and how much percentage of malicious participants out of all participants are required to corrupt the global model such that the global model will consistently misclassify that particular type of image.

**The 2nd deliverables**

**Implementation of the web portal demo**
The 2nd deliverables are pretty much about visualization of the security evaluation parts of 1st deliverables. Conveniently, the web engine WordPress, where my FYP webpage has already been located, provides a plugin called All-In-One Intranet that allows the creation of web portal. This would be an ideal candidate for deploying the web portal demo.
There are two approaches to visualize the attacks and defenses of FL. The first one is to virtually animate the approximated behaviors of the FL platform using the captured data during the experimental run. The second approach is to allow the web portal to run the related FL code and visualize the data in real-time. This allows visualization of real experimental data. It also provides a higher degree of interaction as it would allow end users to freely adjust any hyperparameters before running the demo. However, both approaches are mentioned because the feasibility of embedding computationally intensive and complex python code into a web portal demo is yet to be explored. Depending on the challenges and limitations found while working on the project, both approaches or a hybrid approach are possible.

**Related defenses**

Below are different approaches to defend against data leakage attack and model poisoning attack. We should at least showcase some of those defenses in the web portal demo for demonstration and also study which defense techniques are used by the selected FL platform and effective are they against the backdoor attacks.

According to [1], conventional methods to defend against data leakage attack includes differential privacy, homomorphic encryption, Locality-sensitive hashing (LSH), and secure aggregation. Differential privacy adds noise to blur raw data, but it will decrease model accuracy. Homomorphic encryption allows computation of encrypted data, but introduces new communication overhead for transmitting private key. LSH is another encryption mode that uses hashing to encrypt features. Similar inputs will generate similar outputs such that it can be used for FL. It doesn't cause accuracy decrease and communication overhead. Secure aggregation is a protocol that makes use of a cryptographic techniques so that server can only decrypt the average update if 100s or 1000s of users have participated [8]

For defending against data poisoning attack, the server can uses Byzantine-robust FL algorithms where some robust aggregation algorithm (AGR) are used to reduce the influence of malicious participants [1, 9]. However, it is proved to be weak against data poisoning attack. Shejwalkar et al. [9] presented a new divide-and-conquer (DnC) algorithm that outperforms all existing Byzantine-robust FL algorithms. The high-level idea of this approach is to detect and remove malicious gradient updates before aggregation. This is done by first randomly subsample dimensions of the gradient updates. Then compute the projection of their principal component. And finally filtering out all the outlier.
Project Schedule and Milestones

1st Project Milestone (30/11/2022)
- Complete the main tasks of deliverable 1, running these selected two FL platforms on the benchmark datasets.
- Test these FL platforms successfully and be able to give the evaluation results in terms of charts.

2nd Project Milestone: (28/02/2022)
- Complete the main tasks of deliverable 2, make it clear that the backdoor attack of FL and how to defense it.
- Can simulate the whole process of the backdoor attack and defense successfully and give the conclusion that whether these open-source FL platforms can defend this backdoor attack.

3rd Project Milestone: (28/04/2022)
- Finish the code work of visual web portal demo.
- Finish the two evaluation reports for the selected open-source FL platforms.

References


