FinTech Industrial Project Progress Check Form (Final)

Student Name & ID: DU Jiaxing, 1155204280

Project title: Graph Neural Network Implementation

Sponsoring Organisation: CEFAR

Industrial Supervisor: Cobi XU, A.I. Phoenix Technology Co

Instructions: This form is to report the student's progress up to the final. Please complete the form using Microsoft Word and add rows if necessary. Completion percentage refers to the percentage of the original plan completed. If the percentage is less than 100%, explanation must be provided. Please report your progress accurately and evidence should be provided to the supervisors for them to verify.

Task	Actual achievements	Completion Percentage	Explanation
Federated learning environment setup	1. Construct a simple framework to realize federated learning aggregation function locally. 2. Set up *FedScale* environment on MacOS and CentOS. 3. Set up *FATE* environment on CentOS VM.	100%	
Federated learning on computer vision (CV) datasets	1. Train CIFAR10 and FEMNIST dataset using ResNet18, under IID and non-IID settings. 2. Use *WandB* to visualize gradients and parameters of trained model to ensure the model convergence. 3. Compare the federated learning result with centralized training. In both IID and non-IID setting, train accuracy is almost the same as centralized training.	100%	

Federated learning on graph neural network (GNN) datasets	1. Understand three kinds of Federated GNN tasks. 2. Implement Federated GCN on graph classification tasks like MoleculeNet, under uniform and non-uniform partition. 3. Compared with Federated CNN, Federated GNN performs a little worse relative to centralized training. This may be due to the high heterogeneity and complex structure of samples.	80%	Mainly focus on GNN implementation in graph-level classification that is easy to handle, since its structure is similar to CNN. The set-up process for deploying *FedScale* and *FATE* finally succeeds after many attemps, due to incompatibility of arm64 architecture. Besides, training ResNet18 locally takes up a lot of memory, which crasher the computer frequently.