CrowdFL: A Marketplace for Crowdsourced Federated Learning

Daifei Feng[1], Cicilia Helena[1], Wei Yang Bryan Lim[2], Jer Shyuan Ng[2], Hongchao Jiang[2], Zehui Xiong[3], Jiawen Kang[1], Han Yu[1], Dusit Niyato[1], Chunyan Miao[1]

[1]School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore
[2]Alibaba-NTU Singapore Joint Research Institute (JRI), Singapore
[3]Singapore University of Technology and Design (SUTD), Singapore
{dfeng003, chelena001, limw0201, s190068, hongchao001}@e.ntu.edu.sg,
zehui_ xiong@sutd.edu.sg, {kavinkang, han.yu, dniyato, ascymiao}@ntu.edu.sg

Abstract

Amid data privacy concerns, Federated Learning (FL) has emerged as a promising machine learning paradigm that enables privacy-preserving collaborative model training. However, there exists a need for a platform that matches data owners (supply) with model requesters (demand). In this paper, we present *CrowdFL*, a platform to facilitate the crowdsourcing of FL model training. It coordinates client selection, model training, and reputation management, which are essential steps for the FL crowdsourcing operations. By implementing model training on actual mobile devices, we demonstrate that the platform improves model performance and training efficiency. To the best of our knowledge, it is the first platform to support crowdsourcing-based FL on edge devices.

Introduction

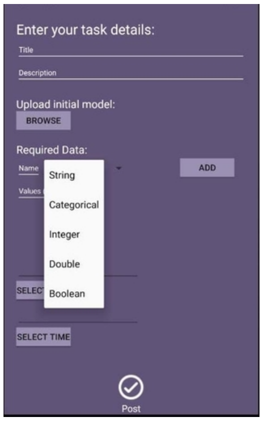
Federated Learning (FL) (McMahan et al. 2017; Kairouz et al. 2021) is a privacy-preserving machine learning paradigm where a shared model is trained by multiple data owners (i.e., clients) without exposing their data. FL is useful for organisations that require training data that are sensitive in nature (e.g., healthcare). However, there exists a gap between model demand and data supply. We present *CrowdFL*, a platform for crowdsourced FL for task listing, client recruitment, and efficient FL model training. To increase the likelihood of successful training completion amid potential straggling devices, we incorporate a client selection scheme to choose suitable clients for each training task. Upon task completion, incentives are distributed to the clients to encourage participation. Reputation scores are used to track desirable/malicious behaviours of clients in order to ensure quality of data provided. To evaluate the efficiency of *CrowdFL*, we further test the platform using actual mobile devices and real-world datasets.

System Functions and Architecture

The main features of *CrowdFL* are listed as follows:

1. *Training task listing to recruit clients*
   Model requesters publicize training tasks and the specific training requirements (e.g., task description, training start time, and data attributes required). This is to match the model requesters with the clients (Figure 1). The model requester also uploads an initial model and a test dataset to be used for model training and testing respectively.
2. *Participation in training tasks*
   A client can view the tasks available (Figure 2) and choose to participate in the training tasks for which they meet the requirements (Figure 3). Clients that have registered their interests will each receive a unique training ID that serves as a token identity for the task.
3. *Client selection*
   To mitigate potential client misbehaviours and improve training performance, *CrowdFL* selects the top K clients based on a weighted client score considering the CPU, RAM, storage and reputation score of the client device.
4. *Model training via FL*
   Using their local data, clients collaboratively train the shared model while keeping the training data on devices. This is done by sending parameter updates rather than data to the server. The server then applies the Federated Averaging (*FedAvg*) (McMahan et al. 2017) algorithm to update the global model. In case of a broken connection, training is resumed using a unique identifier present in each communication message. If the client is unreachable, the platform reaches out to backup clients instead to complete the training. After a specified number of rounds, the model requester can download the final model and access its performance statistics (Figure 5).
5. *Incentive and reputation scheme*
   After the training, the clients receive points for participating in the task. These points may be exchanged for monetary rewards. The clients' reputation scores are also updated (Figure 4) to penalize client device drop-outs or uploading of anomalous model parameters.

The system (Figure 6) consists of three main components. The mobile application is where the data and model requesters are matched, and where FL training occurs. The application downloads the FL model from the server and reads data from on-device storage for local training. For the back-end server, Spring Boot framework is used to implement REST API for convenient data management. *CrowdFL* uses WebSocket communication for low latency transfer of model weights between the server and clients.



Enter your task details:

Title

Description

Upload initial model:

BROWSE

Required Data:

Name

Values

String

Categorical

Integer

Double

Boolean

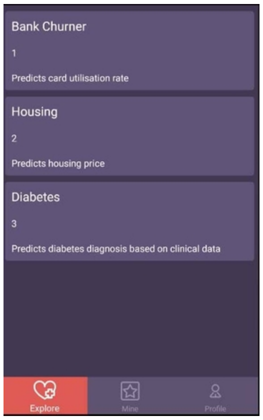
SELEC

SELECT TIME

ADD

Post

Figure 1: List a task



Bank Churner

1

Predicts card utilisation rate

Housing

2

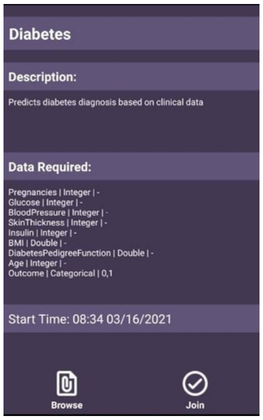
Predicts housing price

Diabetes

3

Predicts diabetes diagnosis based on clinical data

Figure 2: View listings



Diabetes

Description:

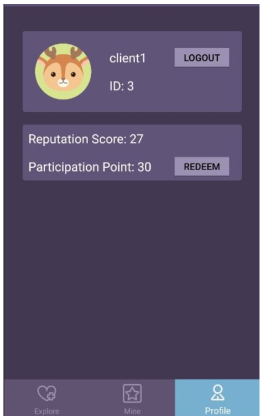
Predicts diabetes diagnosis based on clinical data

Data Required:

Pregnancies | Integer | -
Glucose | Integer | -
BloodPressure | Integer | -
SkinThickness | Integer | -
Insulin | Integer | -
BMI | Double | -
DiabetesPedigreeFunction | Double | -
Age | Integer | -
Outcome | Categorical | 0,1

Start Time: 08:34 03/16/2021

Figure 3: Requirements



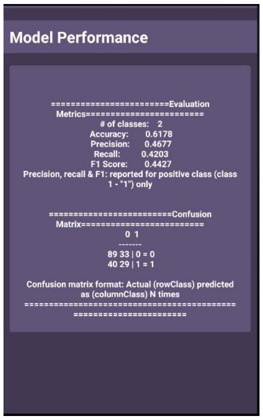
client1

ID: 3

Reputation Score: 27

Participation Point: 30

Figure 4: User profile



Model Performance

```
========================Evaluation
Metrics========================
# of classes:    2
Accuracy:      0.6178
Precision:     0.4677
Recall:        0.4203
F1 Score:      0.4427
Precision, recall & F1: reported for positive class (class
1 - "1") only

==========================Confusion
Matrix=========================
 0  1
-------
89 33 | 0 = 0
40 29 | 1 = 1

Confusion matrix format: Actual (rowClass) predicted
as (columnClass) N times
============================================
========================
```

Figure 5: Statistics

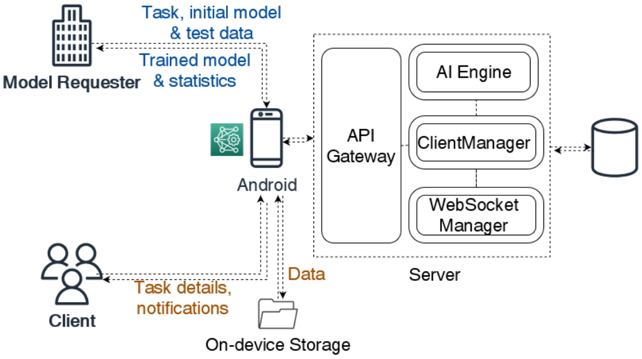


Figure 6: System Architecture

Experimental Evaluation

To evaluate *CrowdFL*, we utilize four mobile phones (specified in Table 2) and training data from the Pima Indians Diabetes Database (Smith et al. 1988) and Bank Customer Churn Dataset (kaggle.com/kmalit/bank-customer-churn-prediction).

We compare the performance of FL with the benchmark centralised approach where all training data reside on the server. As shown in Table 1, *CrowdFL* achieves comparable performance as the benchmark on both the classification and regression tasks while preserving client data privacy.

Dataset	Criterion	Benchmark	CrowdFL
Diabetes	Accuracy	0.6545	0.6492
BankChurner	MSE	0.5705	0.6085

Table 1: Model performance of CrowdFL

To test the effectiveness of our client selection mechanism, we carry out experiments with and without the use of reputation mechanism based client selection. Of the three clients, we initialize one to have lower reputation score (i.e., due to its missing data values from previous training iterations). When the reputation score based client selection is not adopted, those with missing data values are allowed to participate in training. This results in a significant decrease in model accuracy from 0.65 to 0.58.

For time sensitive tasks, mitigating straggling clients is important. When *CrowdFL* does not use device scores to select clients, all clients participate and the training is completed within 147 seconds, as the client with the lowest computation capability is included. With the use of the device score, the Device 1 (Table 2) was dropped and the training time decreased to 127 seconds.

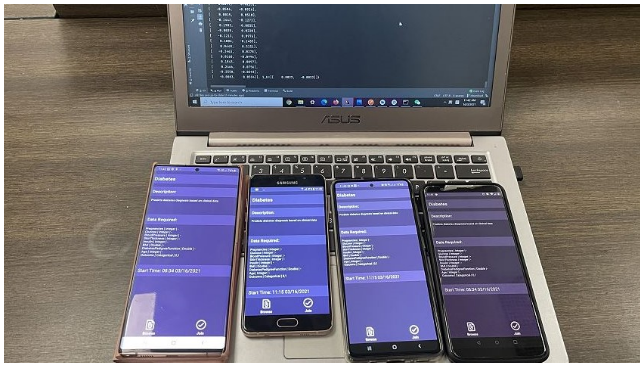


Figure 7: Experiment setup

ID	Free RAM (MB)	Free Memory (MB)	Model
1	113	109	Galaxy A5
2	254	193	OnePlus 5t
3	186	21	Galaxy A51
4	251	23	Note20

Table 2: Specifications of devices used in the training task

Conclusions and Future Work

CrowdFL is a unique platform to match data owners with model requesters in FL, while providing mechanisms to ensure training efficiency. To the best of our knowledge, it is the first platform to support crowdsourcing-based FL on edge devices. In the future, we will develop a comprehensive data preprocessing pipeline for complex datasets.

Acknowledgments

This research is supported, in part, by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-019); the NRF Investigatorship Programme (NRFI Award No: NRF-NRFI05-2019-0002); Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI) (Alibaba-NTU-AIR2019B1), Nanyang Technological University, Singapore; the Nanyang Assistant Professorship (NAP); NTU-SDU-CFAIR (NSC-2019-011); the RIE 2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund (No. A20G8b0102), Singapore, and SUTD SRG-ISTD-2021-165. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the funding agencies.

References

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G. L.; Eichner, H.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konečný, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Raykova, M.; Qi, H.; Ramage, D.; Raskar, R.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning*, 14(1-2): 1–210.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 1273–1282.

Smith, J. W.; Everhart, J. E.; Dickson, W.; Knowler, W. C.; and Johannes, R. S. 1988. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 261.