Harmonia: Accurate Federated Learning with All-Inclusive Dataset

Wonmi Choi*, Juyoung Ahn*, Yeonho Yoo, Chuck Yoo, Gyeongsik Yang
Department of Computer Science and Engineering, Korea University

***Abstract*—Federated learning (FL) is an appealing model training technique that utilizes heterogeneous datasets and user devices, ensuring user data privacy. Existing FL research proposed device selection schemes to balance the computing speeds of devices. However, we observe that these schemes compromise prediction accuracy by ∼57.7%. To solve this problem, we present *Harmonia* that enhances prediction accuracy, while also balancing the diverse computing speeds of devices. Our evaluation shows that *Harmonia* improves prediction accuracy by ∼1.7× over existing schemes.**

***Index Terms*—Federated Learning, Data Privacy, Distributed Machine Learning, Collaborative Learning, Client Selection**

I. Introduction

The use of internet of things (IoT) devices generates vast amounts of data for training machine learning (ML) models [1]. To train an ML model with the data, federated learning (FL) is gaining widespread attention [2]. In FL, a central server selects devices to participate in each training round and sends the model structure for training on each device. Then, each device trains the model with its own data and sends the trained model's parameters back to the central server. The central server aggregates the received parameters from individual devices into a global model. Because the vast amount of data for model training is not collected centrally, the communication overhead between devices and the server is reduced [3]. In addition, as the training data exists only on devices, privacy is preserved [4].

Previous studies proposed various FL schemes, and their major concern was how to select devices for participation. Some schemes randomly selected devices for model training (for example, selecting 10% of total devices per round [2]). Also, several studies considered the heterogeneity in computing resources among devices. The range of devices participating in FL training spanned from resource-constrained IoT devices to high-performance servers [4], leading to differences in computing and communication power. As poorer devices could significantly reduce the overall training speed, some studies excluded devices with low computing resources and selected devices with higher resources [5], [6].

However, we observe that existing FL schemes significantly compromise prediction accuracy, the paramount metric in ML model performance. We evaluate existing FL schemes and compare their prediction accuracy to central learning, in which a single machine collects data from devices and trains a model. Our evaluation exhibits that prediction accuracy in FL is ∼57.7% worse than central learning. This reduction in accuracy stems from the selective participation of devices in FL; thus, data from devices that are excluded from FL training is not utilized, which leads to training on a more scarce and limited dataset.

To this end, we propose a new FL scheme, called *Harmonia*, that improves the prediction accuracy of FL by making full use of dataset on devices. The main observation behind *Harmonia* is that, nowadays, each user owns multiple devices, such as smartphones, tablets, smartwatches, and laptops, and within these devices, data sharing is allowable [7]. Because the data is transferred only between devices belonging to the same user and is never delivered to the central server, privacy is ensured.

Based on this observation, *Harmonia* first groups devices according to their owner. Within each group, *Harmonia* selects a device to participate in FL training, termed a "leader," based on the computing power of the devices to enhance training speed. Then, *Harmonia* orchestrates the devices in the group to send their data to the leader, enabling the leader to use all the group's data for training. Our evaluation demonstrates that *Harmonia* improves the prediction accuracy by ∼1.71× compared to existing FL schemes. *Harmonia* further improves the time-to-accuracy by ∼34.6× (details in § IV).

II. Backgrounds and Motivation

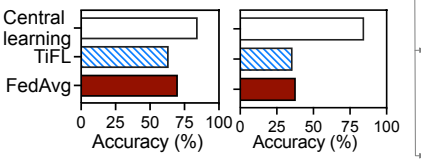
A. Background and Existing Schemes

1) FL: In comparison to central learning, which collects all the training data on a central server for the server to perform the ML model training [8], [9], FL enables collaborative model training across heterogeneous devices. A single trial of model training in FL is called a round. The devices in each FL round have different and imbalanced datasets for training.

FL works with two different scenarios of dataset existence on devices: 1) distribution difference and 2) quantity difference [10]. Distribution difference means that devices have imbalanced numbers of datasets for each prediction label. Assume two devices—device 1 and device 2— and the model predicts

*Wonmi Choi and Juyoung Ahn are co-first authors.

This work was supported by the National Research Foundation of Korea (NRF) (NRF-2023R1A2C3004145, RS-2024-00336564), and by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP-2024-2020-0-01819), all funded by the Korea government (MSIT). This research was also supported by Basic Science Research Program through the NRF funded by the Ministry of Education (NRF-2021R1A6A1A13044830), and by the Google Cloud Research Credits. Corresponding authors: Chuck Yoo and Gyeongsik Yang. E-mail: {ymcui, jyahn, yhyoo, chuckyoo}@os.korea.ac.kr, g_yang@korea.ac.kr.



(a) Distribution difference. (b) Quantity difference.

Fig. 1: Motivating experiments –poor accuracy.

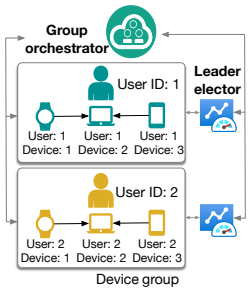


Fig. 2: *Harmonia* structure.

on two labels, "cat" and "dog." Then, in distribution difference, device 1 might have seven data records for "cat" and four for "dog," and device 2 has four for "cat" and nine for "dog." Quantity difference means that devices lack data for a certain prediction label. For example, device 1 has data records only for "cat," and device 2 has records only for predicting "dog." Generally, FL schemes are designed and evaluated based on both dataset existence scenarios.

2) Existing schemes: FedAvg [2] is the representative scheme. The central server of FedAvg randomly selects devices to join in each training round, and these selected devices start the training with each local model that is stored on the central server. The training is done by the local data of each device. After training, each device sends the trained model parameters back to the server. Finally, the server aggregates the received parameters to update the global model.

The heterogeneous computing resources in FL devices cause delays in training for every round, a phenomenon called the straggler effect [4]. So, several studies attempted to address device heterogeneity by selecting devices based on differences in computing resources. For example, TiFL [5] profiles the training speeds of devices and prioritizes the devices with fast speed to participate in each round. Another scheme, Oort [6], sets a threshold for training speed. It then profiles the training speeds of each device and selects devices whose speeds are faster than the threshold to participate in the training.

B. Motivating Experiment

Here, we examine the problem of existing FL schemes through the following experiments.

1) Setup: For experiments, we evaluate three different schemes: 1) central learning, 2) FedAvg, and 3) TiFL. For central learning, we use a machine equipped with an Intel i7-14700KF CPU and an RTX 3080 GPU. The central training is performed by the GPU. For FedAvg and TiFL, we set up a cluster that constitutes one central server with an Intel i5-2300 CPU and five user devices: three Raspberry Pi 4Bs and two NVIDIA Jetson Nanos. The user devices perform the training.

We train ResNet-18 model with CIFAR-10 dataset. We also train other models and datasets, but due to the page limit, we present the representative results here. In central learning, the machine has the entire dataset. For FL schemes, i.e., FedAvg and TiFL, we use identical models and datasets, but each user device holds a portion of the dataset and trains the ML model with that portion. We evaluate FedAvg and TiFL for the two dataset existence scenarios on devices: distribution and quantity differences. We use NIID-Bench [10] to evaluate the two scenarios.

2) Poor accuracy: Fig. 1 presents the prediction accuracy for two different dataset existence scenarios. We do not limit the number of FL training rounds for each FL scheme until the trained model converges. The bars in the figure represent the accuracy once the model has converged during the training process. In the distribution difference scenario (Fig. 1a), both FL schemes, FedAvg and TiFL, exhibit worsened prediction accuracy compared to central learning. Specifically, their decreases in accuracy are 16.9% and 24.8%, respectively.

For the quantity difference scenario (Fig. 1b), both FedAvg and TiFL also exhibit decreased prediction accuracy compared to central learning, by 55.1% and 57.7%, respectively. In the quantity difference scenario, we observe that the accuracy decrease is 2.8× greater on average than that of the distribution difference scenario. This is because, in the quantity difference scenario, each device does not possess data records of all prediction labels but contains only a few, leading to a more imbalanced situation than in the distribution difference scenario (Fig. 1a). In summary, for both dataset distribution scenarios, the existing representative FL schemes experience significant drops in accuracy.

III. Design

Here, we explain *Harmonia*. Fig. 2 shows the structure of *Harmonia*, which consists of two main components: group orchestrator and leader elector. The group orchestrator creates device groups from the devices participating in FL training and orchestrates data movements between devices within a group. The leader elector exists per device group and selects a leader from each group. We explain each component in detail.

Group orchestrator. We first explain the group orchestrator. The group orchestrator assigns a unique user ID to each user. For every new device, each device is identified by the user ID of the device owner and the unique index of the device (device ID). The group orchestrator then creates a group of devices having an identical user ID.

Also, the group orchestrator manages the data transfer between devices within a group. For each group, the leader elector (to be explained in the next subsection) selects leaders to participate in FL training rounds. Before starting an FL training round, the group orchestrator initiates data transfer from each device to the leaders. The data transferred to the leader is used for model training. If the data transferred to the leader in the previous FL training round has not changed, the group orchestrator omits the data transfer in the subsequent round. We implement the data transfer between devices using gRPC for compatibility between devices.

After the data transfer, the group orchestrator initiates the training round by sending a global model to leader devices. The leader device then starts the model training and, once the training is finished, sends the model parameters to the group orchestrator. Once all leader devices send the model parameters, the group orchestrator aggregates the results and

updates the global model. We use the model aggregation algorithm proposed in FedAvg [2]; however, it can be replaced with any other algorithm, such as FedProx [11].

Leader elector. The leader elector exists per device group and manages devices within that group. For example, whenever a new device joins a group, the leader elector runs. Also, the leader elector profiles the training speed of the newly joined device. Also, after every five FL rounds that each device has participated in, we reperform the profiling, because the available computing resources (e.g., CPUs, GPUs, or memory) can vary over time. The profiling is conducted as follows.

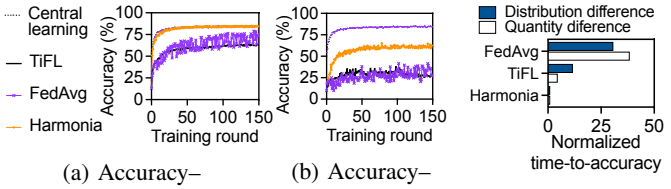
The profiling is conducted using the same model structure that is used in the FL training rounds. For each device, the leader elector sends the model structure and a dataset for profiling. Note that we use the same dataset across devices to fairly compare the training speeds between devices. The leader elector then initiates a single iteration of training on each device as profiling. The outcomes from each device can be as follows. First, a device can fail to execute profiling due to insufficient memory (out-of-memory) or CPU resources. Second, if a device has sufficient resources, it successfully completes the profiling. Each device reports its outcome, whether a failure or a success, to the leader elector. The leader elector records the device (device ID) that reports success, along with the training time calculated from the start of profiling to the report of success outcome.

Based on the profiled training times of the devices, the leader elector selects the leader of each group before every FL round begins. If the group has no newly joined devices and no updates in profiling exist, the leader remains unchanged. If there are updates, it selects the device that shows the shortest training time. The leader elector notifies the group orchestrator of the decision to facilitate further data transfer between the devices and the leader and to begin the FL training round.

IV. Evaluation

We implement *Harmonia* with Python of 855 lines of code. For evaluation, we measure the prediction accuracy and training time in the same setup as explained in §II-B1. Also, we configure five devices to belong to two users: one user has two Raspberry Pis and one Jetson Nano, and the other user owns one Raspberry Pi and one Jetson Nano. Also, we compare *Harmonia* with three different schemes, central learning, TiFL, and FedAvg (explained in §II-B1).

Accuracy. Figs. 3a and 3b show the prediction accuracy of training schemes for distribution difference and quantity difference scenarios, respectively. We first explain the results of the distribution difference scenario in Fig. 3a. Compared to existing FL schemes (TiFL and FedAvg), *Harmonia* improves the converged prediction accuracy by 1.32× and 1.19×, respectively. Furthermore, *Harmonia*'s converged accuracy is quite similar to central learning, with only a 0.78% difference. For the quantity difference scenario (Fig. 3b), *Harmonia* also improves the converged prediction accuracy compared with TiFL and FedAvg by 1.71× and 1.61×, respectively. Compared to central learning, it shows a 27.6% difference.



(a) Accuracy–distribution difference. (b) Accuracy–quantity difference. (c) Time.

Fig. 3: *Harmonia* evaluation.

Time. Fig. 3c presents the time-to-accuracy which means the training time required to achieve a specific target accuracy. As FL devices have highly different datasets and reveal different converged accuracies, existing studies measure the time to achieve the same target accuracy for fair comparison [5], [6]. Similar to these studies, we select the lowest converged accuracies from experiments as the target accuracies. In our case, these are 63.6% and 35.8% for distribution and quantity difference scenarios, respectively (as in Figs. 3a and 3b). We normalize the measured time-to-accuracy with the value of *Harmonia*; so, all *Harmonia* values are 1. Also, as central learning trains models on a different hardware (GPU server) without FL, we exclude it from the comparison. In both dataset distribution scenarios, *Harmonia* significantly improves the time-to-accuracy. On average, *Harmonia* reduces the time by 8.1× and 34.6× compared to TiFL and FedAvg, respectively. These results demonstrate that by utilizing more datasets from the devices, *Harmonia* achieves higher accuracy in less time.

V. Conclusion and Future Work

We introduce *Harmonia*, a new FL scheme for inclusive dataset usage. Our evaluation shows that *Harmonia* improves the prediction accuracy and time-to-accuracy ∼1.71× and ∼34.6×, each. In future work, we plan to further enhance the accuracy of *Harmonia* in the quantity difference scenario. Also, we will consider networking costs to further enhance the efficiency of the *Harmonia* framework.

References

[1] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," *Proceedings of machine learning and systems*, vol. 1, pp. 374–388, 2019.

[2] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017, pp. 1273–1282.

[3] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, pp. 1–210, 2021.

[4] A. Imteaj *et al.*, "A survey on federated learning for resource-constrained IoT devices," *IEEE Internet of Things Journal*, pp. 1–24, 2021.

[5] Z. Chai *et al.*, "TiFL: A tier-based federated learning system," in *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, 2020, pp. 125–136.

[6] F. Lai *et al.*, "Oort: Efficient federated learning via guided participant selection," in *15th USENIX OSDI*, 2021, pp. 19–35.

[7] "Principles relating to processing of personal data," accessed: 24-03-2024. [Online]. Available: https://gdpr-info.eu/art-5-gdpr

[8] C. Shin *et al.*, "Xonar: Profiling-based job orderer for distributed deep learning," in *IEEE 15th International Conference on Cloud Computing*, 2022, pp. 112–114.

[9] Y. Yoo *et al.*, "Machine learning-based prediction models for control traffic in SDN systems," *IEEE Transactions on Services Computing*, vol. 16, no. 6, pp. 4389–4403, 2023.

[10] Q. Li *et al.*, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE ICDE*. IEEE, 2022, pp. 965–978.

[11] T. Li *et al.*, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.