

# Centralized Machine Learning Versus Federated Averaging: A Comparison using MNIST Dataset

Sony Peng<sup>1</sup>, Yixuan Yang<sup>1</sup>, Makara Mao<sup>1</sup>, and Doo-Soon Park<sup>2\*</sup>

<sup>1</sup> Department of Software Convergence, Soonchunhyang University, South Korea  
[e-mail: peng.sony61@gmail.com]

<sup>2</sup> Department of Computer Software Engineering, Soonchunhyang University, South Korea  
[e-mail: parkds@sch.ac.kr]

\*Corresponding author: Doo-Soon Park

*Received September 15, 2021; revised December 27, 2021; accepted January 20, 2022;  
published February 28, 2022*

---

## Abstract

A flood of information has occurred with the rise of the internet and digital devices in the fourth industrial revolution era. Every millisecond, massive amounts of structured and unstructured data are generated; smartphones, wearable devices, sensors, and self-driving cars are just a few examples of devices that currently generate massive amounts of data in our daily. Machine learning has been considered an approach to support and recognize patterns in data in many areas to provide a convenient way to other sectors, including the healthcare sector, government sector, banks, military sector, and more. However, the conventional machine learning model requires the data owner to upload their information to train the model in one central location to perform the model training. This classical model has caused data owners to worry about the risks of transferring private information because traditional machine learning is required to push their data to the cloud to process the model training.

Furthermore, the training of machine learning and deep learning models requires massive computing resources. Thus, many researchers have jumped to a new model known as "Federated Learning". Federated learning is emerging to train Artificial Intelligence models over distributed clients, and it provides secure privacy information to the data owner. Hence, this paper implements Federated Averaging with a Deep Neural Network to classify the handwriting image and protect the sensitive data. Moreover, we compare the centralized machine learning model with federated averaging. The result shows the centralized machine learning model outperforms federated learning in terms of accuracy, but this classical model produces another risk, like privacy concern, due to the data being stored in the data center. The MNIST dataset was used in this experiment.

---

**Keywords:** Artificial Intelligence, Conventional machine learning, Deep neural network, Federated learning, and Federated averaging.

---

A preliminary version of this paper was presented at APIC-IST 2021, and was selected as an outstanding paper. This research was supported by the National Research Foundation of Korea (No. NRF-2020R1A2B5B01002134) and BK21 FOUR (Fostering Outstanding Universities for Research) (No.5199990914048).

## 1. Introduction

With the emergence of Big Data and the era of the 4th Industrial Revolution [1], smart devices have come to be widely used worldwide to respond to the demands of the internet era. The Internet of Things (IoT), Artificial Intelligence, Autonomous Robots, Blockchain, Autonomous Vehicles, Smart Devices, and others will improve the ecosystem and stimulate business development in innovative ways [2]. Therefore, increasing innovative digital technologies can make accessing the entire world easier, better, more portable, and more fun. The linkage between smart technology and the internet era could create massive amounts of structured and unstructured data. Indeed, a significant increase in big data and its characteristics can make it difficult to manage complex data in the whole process.

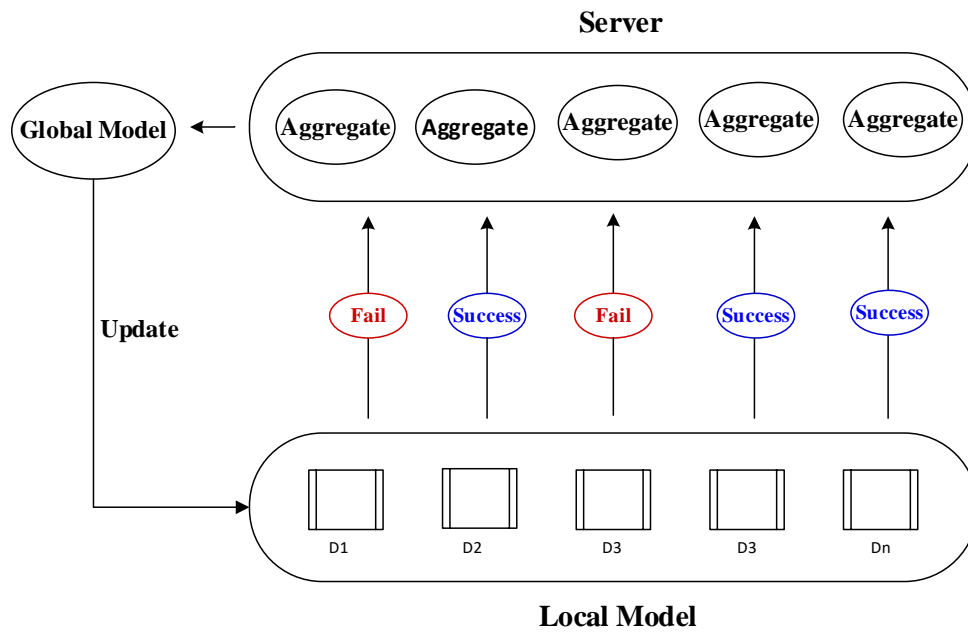
In traditional machine learning, massive amounts of data have been stored and analyzed in one centralized location; data owners must transfer their data to keep it in cloud-based or centralized areas. A conventional centralized data analytics and storage mechanism could have privacy risks for the data owners [3]. However, many important data types are rigorously shared using centralized storage, including hospital data (patient information), government data, bank data, and more. Therefore, federated learning has attracted attention in fields that must prioritize the preservation of privacy such as those involving the 5G network and the Internet of Things (IoT) and in the communication and networking fields, in particular, to optimize resource allocation in communication rounds [4]. Further, federated learning has attracted increasing attention in research fields because its algorithm can be trained without the data owners pushing their data to the cloud server. Therefore, federated learning is a great solution that enables proper on-device machine learning without requiring owner data transfer to the central cloud.

Federated learning has found its most effective use in achieving personal credentials with privacy protection; for example, it has improved device performance in IoT applications. Federated learning was first utilized to enhance Google's Android keyboard prediction [5] without submitting the user's credentials data to the cloud in IoT applications. Apple also employs federated learning to improve Siri's voice recognition [6]. Undoubtedly, blockchain technology has used federated learning to adjust the model and preserve the organization's privacy and data. Federated learning is also essential in the field of cyber-security; it keeps the information on the device safe, and it merely distributes model updates across connected networks. As a potential technique, federated learning has developed to the point where it has been proven to protect privacy concerns for medical data [7]. Moreover, it can handle the vast volumes of data coming from modern healthcare systems.

Federated learning (alternatively referred to as collaborative learning) comprises multiple decentralized edge devices or servers storing local data samples. Federated learning trains local data without sharing the data [7-8]. By contrast, conventional centralized machine learning methods upload all local datasets to one server, whereas traditional decentralized approaches generally presume that local data samples are equally shared throughout the datasets [8]. Fig. 1, shows the general federated learning model, which contains three essential processes: local model, central server aggregation, and global model averaging, the sum output from aggregating in the server. In detail, the user first provides local data for training the local model, which then adjusts the global model at the access point. The aggregated global model is then used to train the local models. These procedures continue until convergence of the global model is achieved [9].

The collaborative learning approach provides three data partitions: vertical federated learning, horizontal federated learning, and federated transfer learning. Vertical federated

learning enables multiple owners from the several parties that possess various characteristics (e.g., features and labels) of the same data point (e.g., person) to train a model collaboratively. In vertical federated learning, the data entities shared by all parties must be determined [10]. For example, consider Client A (Amazon), which consists of user information about purchasing books on Amazon, and Client B (Goodreads dataset [11]), which contains the customer's book reviews. Utilizing these two distinct datasets allows one to offer better customer service by using book review data to provide excellent book recommendations. Horizontal federated learning uses the same character data across many devices, thereby allowing Client A and Client B to do the same task on a device with the same characteristics, as shown in Fig. 2, Federated transfer learning refers to a type of vertical federated learning that uses a pre-trained model learned on an identical dataset to solve a new issue [12]. For instance, for transfer learning, making a book recommendation based on the user's previous browsing history requires training a personalized model. Once again, Fig. 2, shows federated learning types, where (a) represents horizontal federated learning, (b) refers to vertical federated learning, and (c) is federated transfer learning.

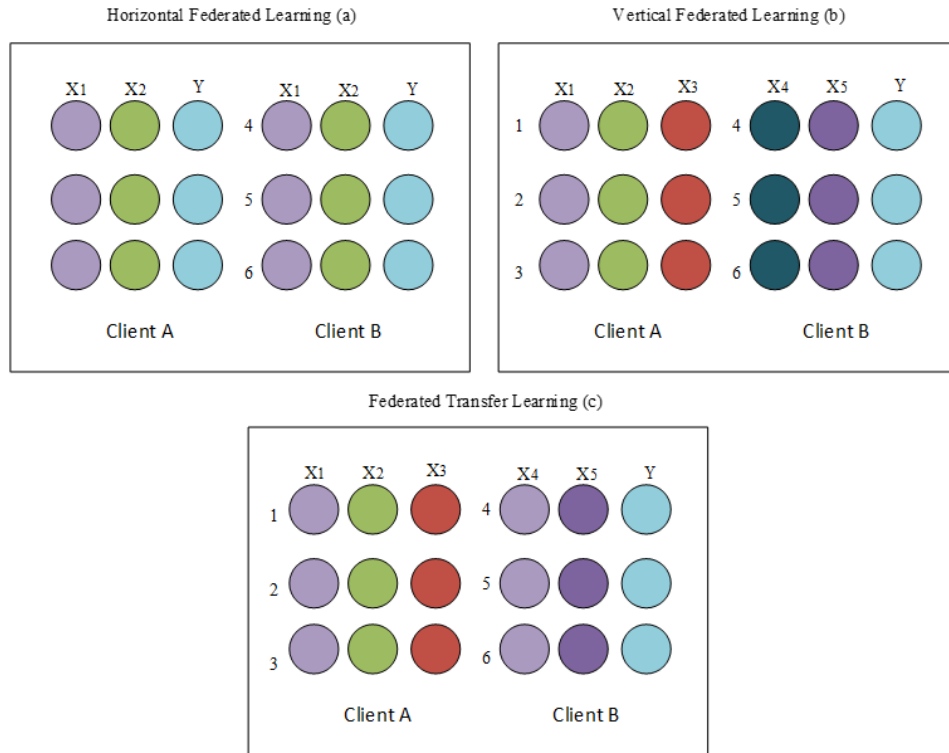


**Fig. 1.** Federated Learning Overview.

This paper implements image classification using the MNIST dataset to compare centralized machine learning and federated averaging. The centralized model can achieve the best accuracy in a classified stage. However, as it cannot secure user-sensitive data, we combine it with federated averaging to protect the user credentials. Finally, we compare the centralized model to the federated averaging model with different model settings and domains (centralized and decentralized data). This implementation allows the deep neural network to train in centralized and decentralized settings using federated averaging [13].

This article is organized as follows: Section 1 briefly introduces the federated learning algorithm. Next, Section 2 mentions the related work that other researchers in the corresponding domain have conducted. The model overview and implementation detail are then given in Section 3. Finally, Section 4 describes the Experiment and its Results, while

Section 5 describes the Conclusions and Future Work in federated learning.



**Fig. 2.** Federated Learning Types.

## 2. Related work

The federated learning algorithm has been researched from a variety of different perspectives. For example, Zhu et al. [14] surveyed federated learning on non-IID data, which mentioned the impact of non-IID data on parametric and non-parametric machine learning models in both horizontal and vertical federated learning. Sun et al. [15] investigated FedAvg with momentum (DFedAvgM) decentralized. It was implemented to connect clients with the undirected graph. All clients in the algorithms executed the stochastic gradient descent with the momentum, and they only interacted with their neighbors. Moreover, the author explored the potential ability of factional DFedAvgM to decrease communication costs. Given the simplified assumptions, the author confirmed that the convergence rate could be enhanced when the loss function meets the PL property.

McMahan et al. [16] presented a practical technique using iterative model averaging for federated learning deep learning. A thorough empirical assessment involving five distinct model architectures and four datasets was used to demonstrate the strategy's effectiveness. These tests showed that the method is impervious to the imbalanced and non-IID data distributions that define this environment's features. However, the author noted that communication costs are the primary constraint. The articles proved that the number of communication rounds might be reduced by 10–100 times compared to that involved in synchronized stochastic gradient descent.

In another study, Onoszko et al. [17] addressed the non-convex issues involved in developing a decentralized setting of the personalized deep learning model. The author studied decentralized, federated learning wherein data is shared across multiple clients and no single server orchestrates the training. In this case, data dispersion across clients is frequently diverse. Therefore, they investigate how to effectively train a model in a peer-to-peer system. Performance-based neighbor selection (PENS) is a technique wherein clients with equal data distributions identify and collaborate to build a model appropriate for the local data distribution. They tested benchmark datasets to indicate that their suggested approach has higher accuracy than strong baselines.

Yu et al. [18] offered a new generic framework called FedAwS for a training method that exclusively uses positive labels. The proposal demonstrated that federated averaging with Spreadout (FedAwS) could deliver performance comparable to traditional learning when clients have access to pejoratives. Nowadays, due to the popularity of federated learning, many researchers have surveyed the challenges and opportunities associated with federated learning in their fields.

Jiang et al. [19] surveyed smart city sensing in general and its current difficulties, along with the role of federated learning; moreover, the author offered clear perspectives on the existing challenges to guide researchers interested in this type of empirical research. Moreover, Hard et al. [20] applied federated learning in mobile keyboard prediction in mobile applications. The idea involved training language models on client devices without exchanging the user credentials to the server. This work proved this to be a feasible method for use and beneficial for the data owner. Meanwhile, other authors such as Zhang et al. [21] provided a survey review of the five main components of federated learning: data segmentation, machine learning model, privacy approach, communication scheme, and systems heterogeneity.

Moreover, several survey papers have discussed federated learning issues and other research directions. Other authors have also introduced federated learning in digital health; for example, Rieke, Nicola, et al. [22] described a federated model in the healthcare system. Beyond the healthcare system, federated learning has been used with blockchain to protect the owner's privacy using IoT devices [23].

### 3. Materials and Methods

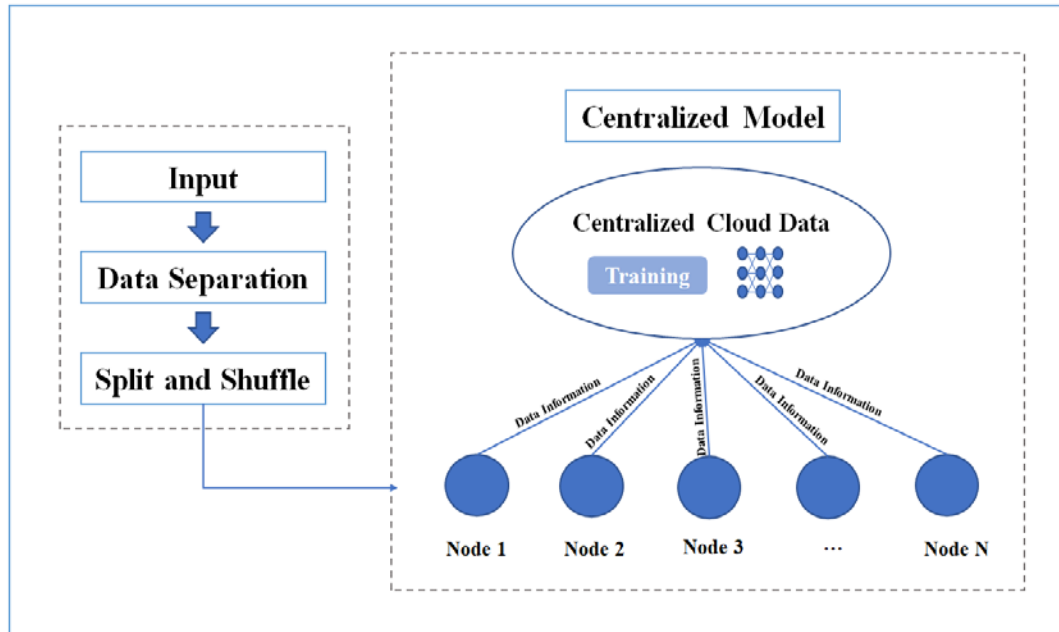
In this session, we explain a model's Classifier and describe the details of its implementation.

#### 3.1 Data Preprocessing

Data Preprocessing refers to transforming or encoding the raw data to understandable data (Knowledge). During the preprocessing, data will be cleaned, removed, dropped if they consist of outlier, missing value, or unused variable. It is a vital step in machine learning algorithms to enhance the model's performance.

## 3.2 Model Classifiers

### 3.2.1 Centralized Machine Learning



**Fig. 3.** Centralized Learning Model.

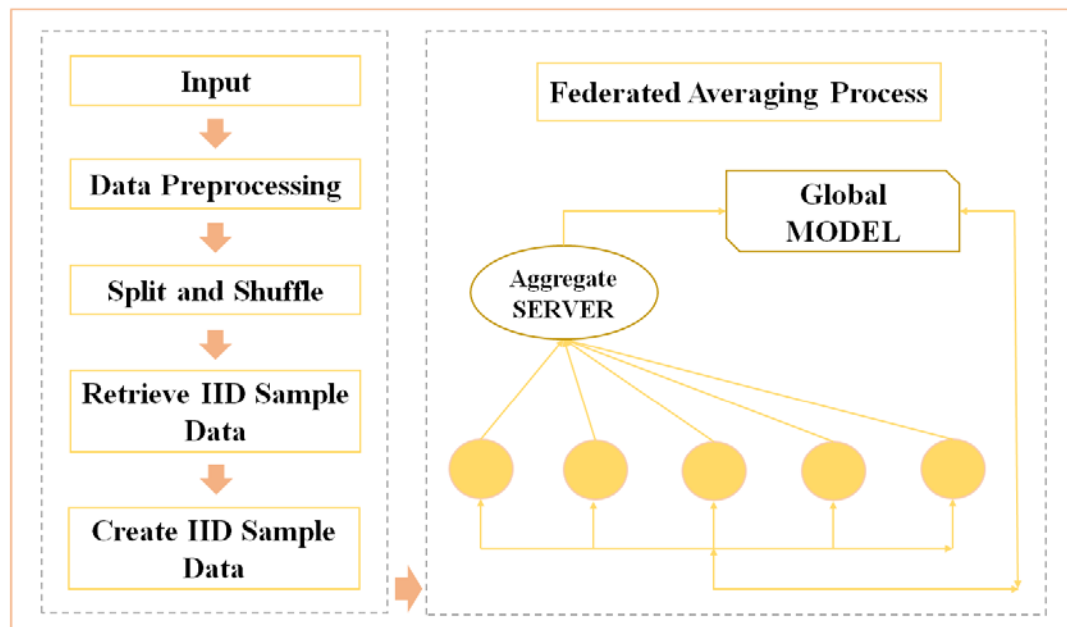
Centralized Machine Learning was achieved by uploading all data from each connected device to one machine (data center) to construct a generic model that can be disseminated and applied to all devices. The model identifies a specific issue related to data in generalized ways. However, we provide centralized machine learning using the MNIST dataset for training in this session. **Fig. 3**, illustrates the flow of centralized model while each node refers to local devices (Smart Phone, Computer, Digital Device) which those devices are transferred their data information to the Cloud to build the generic model, and then the cloud servers execute all computational tasks to train the data then respond the result back to each device(note). Certainly, centralized training is computationally efficient for each device because the devices are not responsible for the highly computational work requiring high computational resources. Moreover, a cloud server can be hostile or infer through attackers, putting clients' sensitive data in danger. Furthermore, uploading large amounts of data might slow down communication between clients and the cloud server.

The implementation process consists of several vital parts in the following:

1. First, the MNIST dataset is used to perform image classification. Next, the data requires preprocessing to remove some noises to fit the model.  
This dataset is separated into a training set and a testing set. Moreover, each training and testing data point has been split and shuffled in a parallel way.
2. Second, each node passes its sensitive data to the centralized cloud to fit the model training.
3. Finally, a deep neural network is used to train the model and do the classification; the result will be provided in Section 4.

### 3.2.2 Federated Averaging

Nowadays, the availability of technology from smart devices or sensors has produced an overload of information while the information is composed of structured and unstructured from anonymized sources. Therefore, some information is required to keep on a personal device without uploading it to the data center. Due to privacy concerns, many researchers have found a way to prevent data from uploading to the cloud. The Federated Learning method is the mechanism that allows the devices to develop a shared prediction model cooperatively while maintaining all training data on the device, isolating machine learning from the desire to retain data on the cloud. Instead of relying on the local models, this approach brings model training to the local devices without sharing the data. The following figures mention the detailed flow of the federated averaging to predict the image classification.



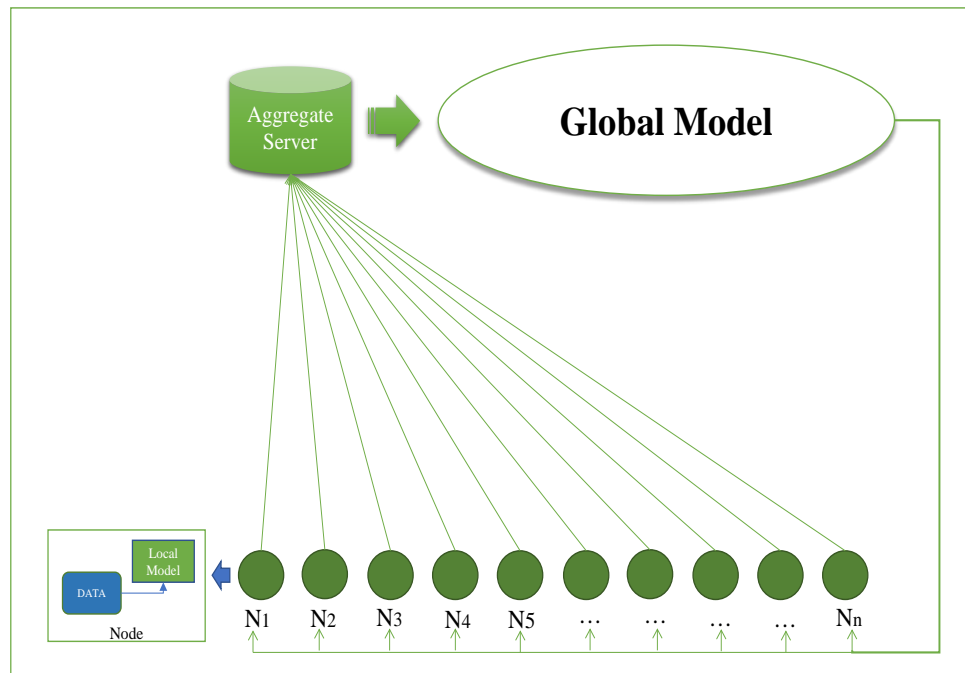
**Fig. 4.** The Overall Flow of Federated Averaging with Data Preprocessing.

**Fig. 4**, and **Fig. 5**, illustrate the preprocessing steps used to work with federated averaging and deep learning models. Some important preprocessing parts are briefly described here:

1. First, the MNIST dataset is used as input to perform image classification. Then, the data needs to undergo preprocessing to remove some noise to fit the model.  
This dataset has been separated into a training set and a testing set. Moreover, each training and testing data point has been split and shuffled in a parallel manner.
2. After the preprocessing process, we retrieve IID sample data from training with the limited local sample; however, the IID data sample has been created for application in the federated learning model.
3. Each node passes all its sensitive data to the aggregate server to fit the model training.
4. In this step, we present the implementation of federated averaging as follows:
  - A. The server selects K random node (client) during the communication round to train.
  - B. After each selected client obtains the current model from the aggregate server, they will perform the local update in their current local training node. A single epoch of batch stochastic gradient descent (SGD) could be achieved in this case.



- C. When the client submits their update on the model, the server integrates the weight data, the disparity between the final settings after training, and the initial values.
  - D. The server then calculates their contributions to create the global model.
5. Finally, after the communication round, the training and testing result will be released and provided in Section 4.



**Fig. 5.** Federated Averaging Flow in our Actual Implementation.

## 4. Experiments and Results

This section describes experiments through simulation and provides the classification results.

### 4.1 Environment setup

This experiment uses Window 11 Pro operating system, AMD Ryzen 9 5900HX with Radeon Graphics, CPU @ 3.30 GHz for a processor, and 16GB Ram. In addition, the Python language and PyTorch framework have been used to form a deep learning model with federated learning. The experimental environment is presented in [Table 1](#).

**Table 1.** Experimental environment

Requirements	Tools
Operating System	Window 11 Pro
Processor	AMD Ryzen 9 5900HX, CPU @3.30GHz
Ram	16 GB
Programming Language	Python Version 3.8.10



## 4.2 The Handwritten Digits – The MNIST Dataset

The MNIST dataset refers to the Modified National Institute of Standards and Technology dataset [24]. It is considered an extensive database of handwritten digits widely utilized to train various image processing systems. Moreover, it contains a vast database of small grayscale images of handwritten single digits between 0 and 9, and it consists of a square of 28\*28 pixels. In total, 70,000 handwritten images were divided into training and testing datasets. Fig. 6, shows the plotting result of training the MNIST Data.

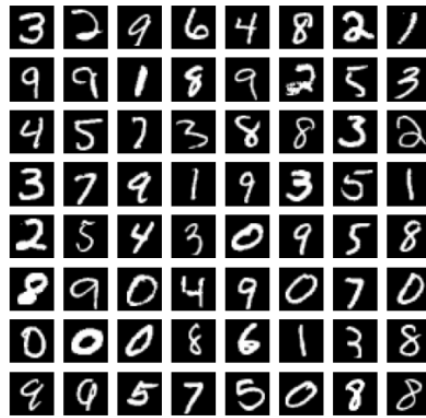


Fig. 6. MNIST Dataset.

## 4.3 Model Setting

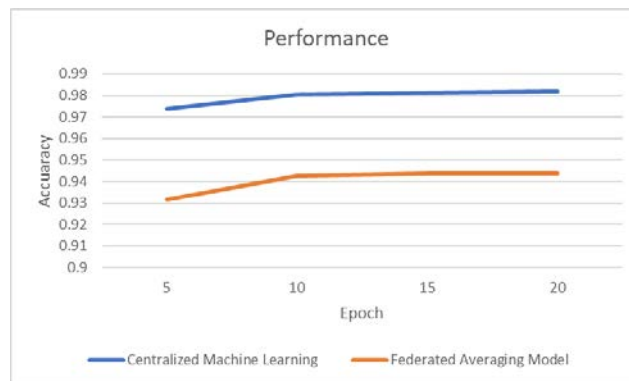
Table 2, lists the deep learning parameters that we used in the training process.

Table 2. Model parameters used in the experimental process

Parameters	Size
Number of Samples	150
Learning Rate	0.01
Epoch	5,10, 15 ,20
Batch Size	32
Momentum	0.9

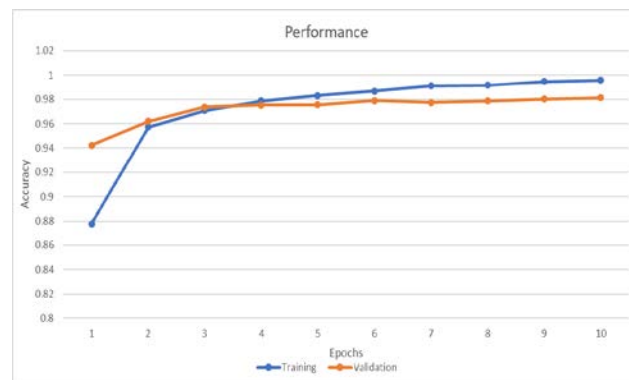
## 4.4 Results

This session explains the Centralized model and Federated averaging performance. We conduct the experiment 4 times to get different results from various Epochs. We follow the parameter in Table 2. Fig. 7, shows the performance result of the Centralized machine learning model and Federated averaging Model. The result of the Federated Averaging model is approximately about 0.93 to 0.94 in various epochs. And the result of Centralized Machine learning is about 0.97 to 0.98. Moreover, each training block applies with the same parameters except the epoch. Due to the training, we choose epoch 10 to briefly explain the detail of the model which are shown in the following figures.



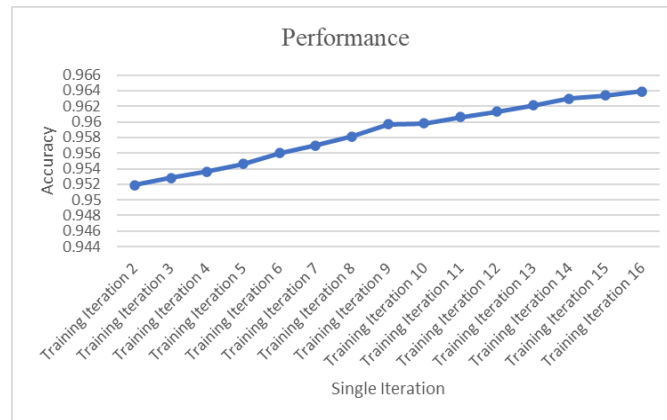
**Fig. 7.** Centralized Machine Learning and Federated Average Accuracy.

**Fig. 8,** shows the performances of the centralized model between the training and testing data (validation). However, in the first round of epochs between the training and testing data, we can see that when the first epoch was trained, the accuracy started from 0.88, and it kept increasing during each training epoch. Thus, in the experiment, we have only initialized 10 epochs in each round of communication. The testing accuracy also begins low in the first epoch and keeps growing in each iteration.



**Fig. 8.** Centralized Model Performance.

**Fig. 9,** shows the accuracy of the federated averaging model in each iteration. The model parameters have been initialized similarly to the centralized model. Each round begins from iteration 2, as shown in the figure. First, the federated averaging result, which starts at approximately 0.95 to 0.96 percent in iteration 16, is provided. However, the accuracy increases from iterations 1-9, then decreases in iteration 10. Finally, the precision keeps improving in iteration 11 and beyond. We start with a simple deep learning model using federated averaging in the experiment. The federated averaging had good performance accuracy even with a basic deep learning model. This is promising because federated learning is trained without exchanging any data to the cloud. Moreover, as it can protect user-sensitive data, its performance cannot be overlooked. The performance of the individual local client in each training process is shown in **Fig. 10**, **Fig. 11**, **Fig. 12**, **Fig. 13**, and **Fig. 14**.



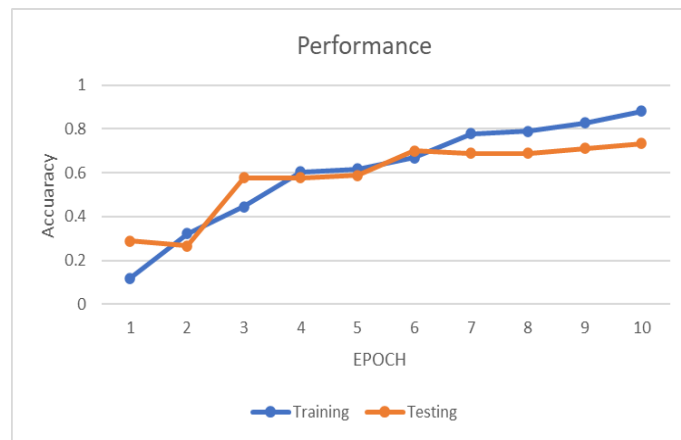
**Fig. 9.** Federated Averaging Model Performance.



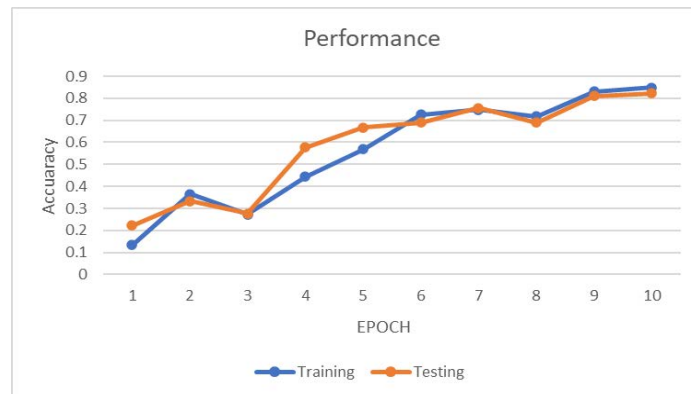
**Fig. 10.** Individual Local Client in Subset 0.



**Fig. 11.** Individual Local Client in Subset 1.



**Fig. 12.** Individual Local Client in Subset 2.



**Fig. 13.** Individual Local Client in Subset 3.



**Fig. 14.** Individual Local Client in Subset 4.

## 5. Conclusions and Future Work

Federated learning has recently attracted attention as an important area of investigation. However, the learning gap between centralized machine learning and decentralized machine learning is a critical issue, and each model has its advantages and disadvantages. For example, Federated learning can be beneficial because this model can protect the privacy and improve communication efficiency [25].

This study compares a federated averaging algorithm and centralized machine learning model. However, we conduct an experiment with different models—between centralized and decentralized models—using the same hyperparameters. The centralized model obtained better performance than the federated averaging model. However, the centralized model still has significant privacy concerns because users need to upload their data to the central cloud to perform the model training. Therefore, the federated learning model could be wisely applied to every domain since it helps protect user-sensitive data and does not have to be sent to the cloud. We only use a simple federated averaging model in this work, so we hope to try a more complex model and change its learning hyperparameters to obtain even better performance.

Although it alleviates privacy concerns, the proposed model still has many challenges that should be researched, including its slow and unbalanced network connection in IoT and the 5G network, a synchronized optimization method, solving a high number of nodes (clients), finding a way to optimize the resource allocation, and more.

## References

- [1] H. Yoo, R.C. Park, and K. Chung, "IoT-Based Health Big-Data Process Technologies: A Survey," *KSI Transactions on Internet and Information Systems*, vol. 15, no. 3, pp. 974-992, 2021. [Article \(CrossRef Link\)](#)
- [2] Z. Du, C. W. T. Yoshinaga, K.A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular internet of things: Recent advances and open issues," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 45-61, 2020. [Article \(CrossRef Link\)](#)
- [3] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M.A. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476-5497, 2021. [Article \(CrossRef Link\)](#)
- [4] L. Wang, and D. Xu, "Resource allocation in downlink SWIPT-based cooperative NOMA systems," *KSI Transactions on Internet and Information Systems*, vol. 14, no. 1, pp. 20-39, 2020. [Article \(CrossRef Link\)](#)
- [5] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018. [Article \(CrossRef Link\)](#)
- [6] D. Guliani, F. Beaufays, and G. Mott, "Training speech recognition models with federated learning: A quality/cost framework," in *Proc. of 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3080-3084, 2021. [Article \(CrossRef Link\)](#)
- [7] C.C. Ma, K.M. Kuo, and J.W. Alexander, "A survey-based study of factors that motivate nurses to protect the privacy of electronic medical records," *BMC medical informatics and decision making*, vol. 16, no. 1, pp. 1-11, 2015. [Article \(CrossRef Link\)](#)
- [8] D. Lia, and M. Togan, "Privacy-Preserving Machine Learning Using Federated Learning and Secure Aggregation," in *Proc. of 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1-6, 2020. [Article \(CrossRef Link\)](#)
- [9] X. Li, L. Zhang, A. You, M. Yang, K. Yang, and Y. Tong, "Global aggregation then local distribution in fully convolutional networks," *arXiv preprint arXiv:1909.07229*, 2019. [Article \(CrossRef Link\)](#)

- [10] Liu, Y., Kang, Y., Li, L., Zhang, X., Cheng, Y., Chen, T., ... & Yang, Q, "A Communication-Efficient Collaborative Learning Framework for Distributed Features," *ArXiv abs/1912.11187*, 2019. [Article \(CrossRef Link\)](#)
- [11] S.K. Maity, A. Panigrahi, and A. Mukherjee, "Book reading behavior on good reads can predict the amazon best sellers," in *Proc. of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pp. 451-454, 2017. [Article \(CrossRef Link\)](#)
- [12] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70-82, 2020. [Article \(CrossRef Link\)](#)
- [13] S. Ji, W. Jiang, A. Walid, and X. Li, "Dynamic sampling and selective masking for communication-efficient federated learning," *arXiv preprint arXiv:2003.09603*, 2020. [Article \(CrossRef Link\)](#)
- [14] H. Zhu, H. Zhang, and Y. Jin, "From federated learning to federated neural architecture search: a survey," *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 639-657, 2020. [Article \(CrossRef Link\)](#)
- [15] T. Sun, D. Li, and B. Wang, "Decentralized Federated Averaging," *arXiv preprint arXiv:2104.11375*, 2021. [Article \(CrossRef Link\)](#)
- [16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of the 20th International Conference on Artificial intelligence and statistics*, pp. 1273-1282, 2017. [Article \(CrossRef Link\)](#)
- [17] N. Onoszko, G. Karlsson, O. Mogren, and E.L. Zec, "Decentralized federated learning of deep neural networks on non-iid data," *arXiv preprint arXiv:2107.08517*, 2021. [Article \(CrossRef Link\)](#)
- [18] F. Yu, A.S. Rawat, A. Menon, and S. Kumar, "Federated learning with only positive labels," in *Proc. of International Conference on Machine Learning*, pp. 10946-10956, 2020. [Article \(CrossRef Link\)](#)
- [19] J.C. Jiang, B. Kantarci, S. Oktug, and T. Soyata, "Federated learning in smart city sensing: Challenges and opportunities," *Sensors*, vol. 20, no. 21, pp. 6230, 2020. [Article \(CrossRef Link\)](#)
- [20] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018. [Article \(CrossRef Link\)](#)
- [21] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, pp. 106775, 2021. [Article \(CrossRef Link\)](#)
- [22] N. Rieke, J. Hancox, W. Li, et al., "The future of digital health with federated learning," *npj digital medicine*, vol. 3, no. 119, 2020. [Article \(CrossRef Link\)](#)
- [23] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177-4186, 2020. [Article \(CrossRef Link\)](#)
- [24] MNIST database, "THE MNIST DATABASE of handwritten digits," 2021. [Article \(CrossRef Link\)](#)
- [25] O. Shahid, S. Pouriye, R.M. Parizi, Q.Z. Sheng, G. Srivastava, L. Zhao, "Communication Efficiency in Federated Learning: Achievements and Challenges," *arXiv preprint arXiv:2107.10996*, 2021. [Article \(CrossRef Link\)](#)



**Sony Peng** received B.S. degree in Information Technology Engineering from the Royal University of Phnom Penh (2018), Cambodia. She is currently a Combined candidate (Master and PhD) at Soonchunhyang University, South Korea. Her research interests include Mobile AI, Data mining, Recommendation System, and Cloud Computing. Contact her at peng.sony61@gmail.com



**Yixuan Yang** received the B.Sc. degree in Software Engineering from Taiyuan University of Technology, China, in 2017 and the M.Sc. degree in Software Engineering from Shaanxi Normal University, China, in 2020. She is currently pursuing her Ph.D. degree at Soonchunhyang University, South Korea. Her research interests include social computing, community detection and formal concept analysis. Contact her at yangyixuan621@gmail.com.



**Makara Mao** received B.S. degree in computer science from Royal University of Phnom Penh, Cambodia in 2016. He is currently studying (Combined Master's and Ph.D. Degree) at Soonchunhyang University, South Korea. His current research includes Natural Language Processing (NLP), deep learning and data mining. Contact him at makaramao07@gmail.com.



**Doo-Soon Park** received his PhD in Computer Science from Korea University in 1988. Currently, he is a professor in the Department of Computer Software Engineering at Soonchunhyang University, South Korea. He is Dean of Graduate School at Soonchunhyang University, Director of BK21 FOUR Well-Life Big Data at Soonchunhyang University. He was President of Korea Information Processing Society from 2015 to 2015. He was editor in chief of Journal of Information Processing Systems from 2009 to 2012. He has served as an organizing committee member of international conferences including, BIC 2021, MUE 2021, WorldIT 2021, CSA 2020, BIC 2019, FutureTech 2019, WorldIT 2019, FutureTech 2018, BIC 2018. His research interests include social computing, big data analysis. He is a member of ACM, KIPS, and KMS. Contact him at parkds777@gmail.com.