Global Model Privacy Protection Mechanism in Federated Learning

Ajit Kumar and Bong Jun Choi
School of Computer Science and Engineering
Soongsil University, Seoul, Korea
ajitkumar.pu@gmail.com, davidchoi@soongsil.ac.kr

***Abstract*—Data scarcity is a crucial concern in the traditional approach of training deep learning and one of the bottlenecks that limit its growth. Recently, Federated Learning (FL) has become a suitable approach for providing data privacy and is emerging as a solution for data scarcity. However, FL has opened up a new issue, i.e., model privacy and security. In vanilla FL, each participant receives the updated global model in every training round. Hence, if a model trainer wants to keep the updated global model private from participants, there is limited scope to protect the model access. There needs to be more literature on preserving the global model, and possible solutions like differential privacy, cryptography, or subnetworks are insufficient. In the proposed work, we have introduced the privacy issues in the global model and provided experimental results to demonstrate global model leaks, i.e., each participant has a model with equivalent accuracy to the global model in the subnetwork-based FL approach.**

***Index Terms*—federated learning, model privacy, model sampling, subnetwork training, global model protection, global model access**

I. Introduction

Today, Deep Learning (DL), under the umbrella of Artificial Intelligence (AI), is leading the digital growth. However, the paradox between the *high requirement* versus the *access limitation* on data has become the bottleneck for DL. Recently, various privacy concerns from users, governments, and enterprises have resulted in different *data privacy* laws (for example, GDPR) that have further intensified the *data scarcity*. The restriction on *data access* and *data privacy* has motivated the development of Privacy-Preserving Machine Learning (PPML). Under PPML, it is advocated that the model trainer or model owner must not directly access the user's raw data for model training [1]. Some popular privacy-preserving methods, such as Homomorphic Encryption (HE) and Differential Privacy (DP) are used for centralized training. These are also used in distributed training, and specialized approaches, such as Federated Learning (FL) [2], and Split Learning (SL) [3] for providing privacy to user data and enabling model training.

In traditional or vanilla FL, the focus is to protect "data privacy," so each participant receives the updated global model in each training round and has access to the complete model [4]. Nevertheless, such architecture overlooked the privacy and access restrictions of the global model that trained on different client's data. Protecting access to the global model became crucial in scenarios where differentiating between participation is essential, such as providing incentives for contributing to the global model training. In addition, use cases, such as using crowdsourcing for FL (e.g., CrowdFL [5]) have a high requirement to protect full access (i.e., any participant will have a model with equivalent accuracy to the updated global model) to the global model during training.

Currently, there is minimal literature on privacy and access limitations of the global model in FL [4]. SL aims to provide the privacy of the global model during training. However, SL training is done sequentially, and the splitting model has its own limitations. Due to FL's widespread acceptance and appeal as a centralized deep learning training alternative, the proposed work has leveraged it to explore the issues regarding access to the global model. In FL, global model privacy is not discussed, and model splitting is done as subnetworks. The subnetworks-based training aims to achieve computational suitability on edge devices or reduce the computation and communication cost of model training [6], [7].

In the proposed work, we have introduced and demonstrated the issues of protecting access to the global model. The model access pitfalls are apparent in vanilla FL, so we have experimented with subnetwork-based FL approaches, which may be misleading the possibility of model access protection. To the best of our knowledge and available literature, we are the first to present literature on the access issue of the global model in FL.

We have organized the rest of the paper as follows. Section II provides the required background and related works. Section III provides details about subnetwork-based model training in FL and the issue with the access of the global model. The experimental setup and results are presented in Section IV, while Section V concludes the proposed work and provides future research direction.

II. Background and Related Works

The federated learning [2] started to enable decentralized model training in a data privacy-preserving manner by removing the centralized collection of raw data from sources. With this motivation, earlier literature mainly focus on aggregation approaches and communication and computational efficiency. Later, methods were proposed to defend training from various attacks, such as data and model poisoning and privacy of communication via differential privacy, homomorphic encryption, and multi-parties computation. FL also makes it possible to use new sources, such as data from individual users' devices

979-8-3503-3094-6/24/$31.00 ©2024 IEEE

TABLE I
LIST OF SYMBOLS AND ABBREVIATIONS USED

Symbol	Description
α	Subnetwork cardinality
θ	Local training update
C_i	i-th client
SW_i	Subnetwork for i-th client
X_i,Y_i	Labeled Dataset of i-th client
Acc	Accuracy of global model
AI	Artificial Intelligence
Avr	Average accuracy of subnetworks
D	Neurons dropping percentage
DL	Deep Learning
DP	Differential Privacy
FC	Fully Connected
FL	Federated Learning
GDPR	General Data Protection Regulation
HE	Homomorphic Encryption
IID	Identically Distributed
PFL	Personalized Federated Learning
PPML	Privacy-Preserving Machine Learning
PVT	Partial Variable Training
SL	Split Learning
SW	Set of subnetworks
W	Global model

or storage for training. Considering such a crowdsourcing training paradigm, various research on incentivizing the participants in FL emerged. For example, CrowdFL [5] outlines the methods and process of FL using crowdsourcing. However, the existing literature should address global model access protection [4], especially in cases where the model trainer incentivizes participants (monetary or in other modes instead of access to the global model) and desires exclusive access to the final global model, limiting participants' access.

A few works related to incentive mechanism in FL have advocated for sharing varying levels of the global model as per the quality or quantity of updates received from the participants [8]–[10]. However, the proposed work differs from incentive approaches because it highlights the drawback of sharing the global model with each participant in every training round. Similarly, it also differs from Personalized FL (PFL) in that instead of sharing a complete global model, globally learned features (global part) are shared with all participants, and each participant can have their local head (output layer) and trains the whole model on its data [11], [12].

The model privacy, i.e., to protect access to the global model from each participant, splitting and sharing only part of the model in each training round, seems a feasible solution. Split Learning (SL) adopts this approach and splits the global model horizontally from a cut layer and only shares a part of the model with participants for training [3]. However, it has two main bottlenecks: 1) finding suitable cut layers that limit the use of possible deep learning architecture, and 2) It requires training to be sequential (relay-based), i.e., one after another client [4]. The proposed work is based on federated learning, and hence, further discussion on split learning is out of scope. In federated learning, global model splitting is done vertically, keeping the original architecture intact, and many splitting is possible. Each part is known as a subnetwork. Each subnetwork is shared with individual participants for each training round.

Yang et al. [6] have proposed Partial Variable Training (PVT) using only a small subset of model weights (variables) in each training round by dividing variables into freezable vs. non-freezable. The authors aim to minimize on-device training memory usage and communication costs without altering model architectures or needing network-specific knowledge for training. However, the authors have not presented any discussion about the privacy of the global model (only have non-freezable variables) during the training round. FedDrop [7] uses dropout to create several subnetworks (limited by total clients participating in each round) at the server. Then, each unique subnetwork is shared with individual clients for every training round. Authors claim to achieve communication and computational efficiency to enable FL on resource-constrained devices and also be able to handle model overfitting issues due to dropout. Dropout is only applied to the fully connected layers because other layers have fewer parameters.

Based on the literature available on dividing the global model for training, we have observed two main streams of work: 1) global model splitting (subnetwork, dropout, etc.) is mainly carried out to achieve efficient training by reducing communication and computation cost, 2) there is minimal discussion on the privacy of the global model. Table I lists the symbols and abbreviations used in the paper.

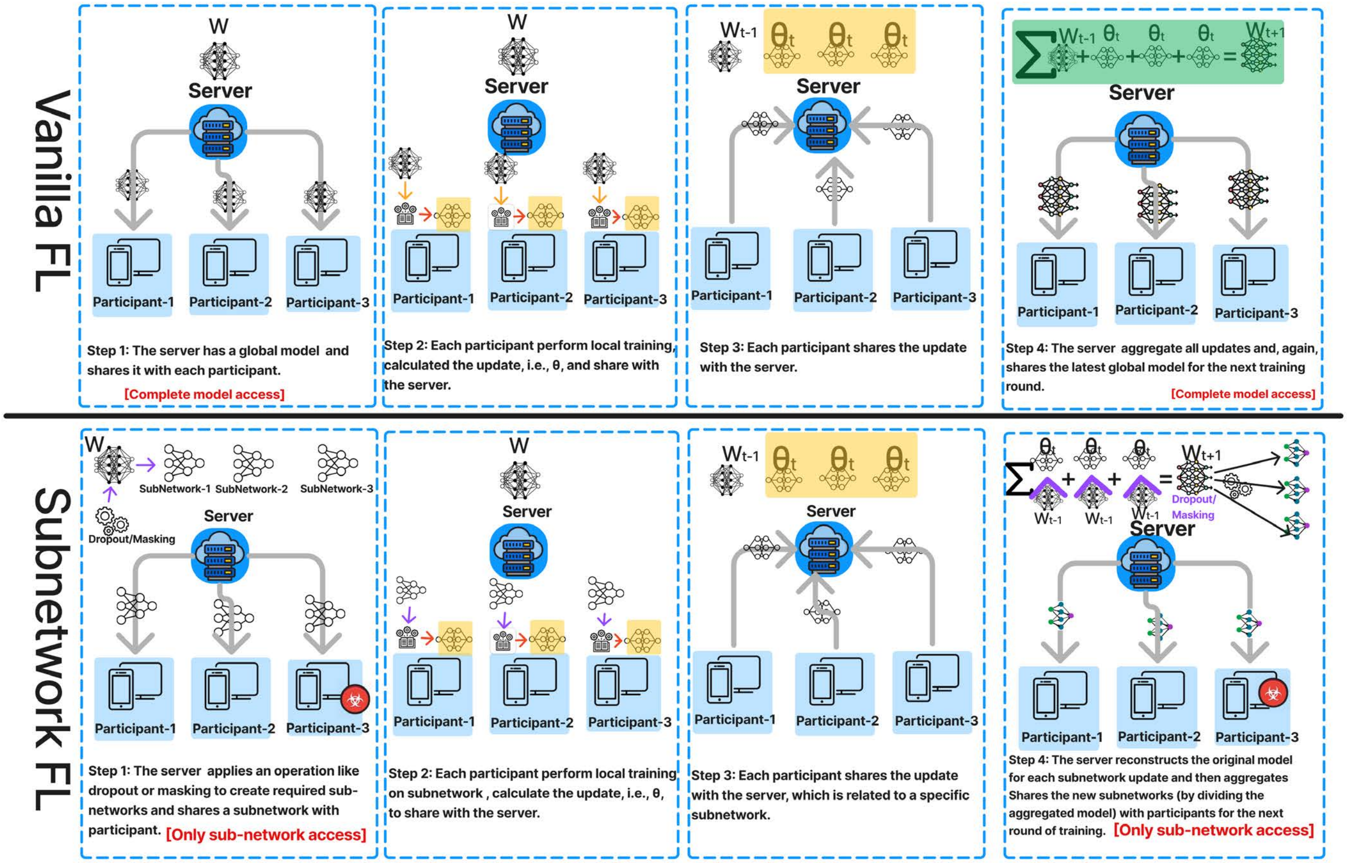
III. Proposed Global Model Protection Mechanism in Federated Learning

A. Overview

Federated learning provides data privacy to the clients/participants. However, traditional FL (hereinafter referred to as Vanilla FL) architecture does not consider the model privacy (access restriction of the final global model). In the upper part of Fig. 1, steps of vanilla FL are shown, and it can be observed that each participant has access to the updated global model in every training round. Recently, many new FL architectures have been proposed to address communication and computation efficiency. The lower part of Fig. 1 shows the subnetwork-based FL, in which the parameter server divides the global model into required numbers of subnetworks and then shares subnetworks with participants for training. Each participant performs local training on a subnetwork and shares the update with the server for aggregation. Considering the local training on separate network architecture, it is possible to gain global model privacy. However, there is limited number of literature to validate the privacy of the global model. In the proposed work, we aim to study the privacy concern of the global model in subnetwork FL.

If we want to protect the privacy of the global model using model splitting approaches, then there needs to be a few key considerations, such as:

- The global model should be divided in such a way that local training on the subnetwork and then their aggrega-



Step 1: The server has a global model and shares it with each participant.

Step 2: Each participant perform local training, calculated the update, i.e., θ, and share with the server.

Step 3: Each participant shares the update with the server.

Step 4: The server aggregate all updates and, again, shares the latest global model for the next training round.

Step 1: The server applies an operation like dropout or masking to create required sub-networks and shares a subnetwork with participant.

Step 2: Each participant perform local training on subnetwork , calculate the update, i.e., θ, to share with the server.

Step 3: Each participant shares the update with the server, which is related to a specific subnetwork.

Step 4: The server reconstructs the original model for each subnetwork update and then aggregates Shares the new subnetworks (by dividing the aggregated model) with participants for the next round of training.

Fig. 1. The federated learning process and the possible global model privacy and access issues in Vanilla FL (upper) and Subnetwork-based FL (lower). In the case of Vanilla FL, there is no protection for global model access by a participant while in subnetwork-based FL, a participant has only access to subnetworks through various rounds. However, a malicious participant (e.g., Participant-3) can store these subnetworks and recover an equivalent global model by aggregating these subnetworks. There can be many methods to create a subnetwork. For example, Yang et al. [6] have created a subnetwork by freezing a layer, and a participant only trains the unfreeze layers and shares the update.

tion at the server will not impact the model performance (accuracy), and the overall process should require less computation.

- The clients should not be able to collaborate to reconstruct the original global model by sharing its subnetworks (e.g., Byzantine attacks).
- The main challenge is deciding the level of shallowness of the subnetwork so that the accuracy difference should be significant between the aggregated model and individual subnetwork in every training round.

Algorithm 1 presents the steps and process of subnetworks creation, federated training, and calculating the accuracy differences between the global model and each subnetwork. The value of P is the accuracy difference between the global model and the client's subnetwork. A higher value of P indicates a large accuracy gap between the global and local subnetworks that is suitable to protect the privacy of the global model. In contrast, a lower value of P indicates a lower accuracy difference; hence, access to the global model is not protected. In Table II, the difference is shown as the average of all subnetworks of the last training round; however, a similar trend is observed in all the training rounds. In Algorithm 1, the method $CreateSubnetworks()$ is the main difference from Vanilla FL, and to achieve the privacy of the global model or protect the access of the global model from participants during the training round, i.e., we need to enhance the method in such a way that we achieve a lower value of P in each training round.

B. Steps for Global Model Privacy Protection

- **Step 1:** Splitting the global model (pre-trained dense versus randomly assigned weight or initialization): one common approach would be to turn odd /even or random percentages of neurons on and off to create a subnetwork.

Algorithm 1: Subnetworks-based Federated Learning and Global Model Privacy Measurement

```
Input: $W$, $\alpha$, $D$
Output: $P$ (Privacy Level)
Data: Client Dataset $X_i, Y_i$
for round do
    /* At Server: */
    $\mathbb{SW} \leftarrow$ CreateSubnetworks($W$, $\alpha$, $D$)
    for client do
        $\mathbf{C_i} \leftarrow SW_i$
        /* At Client: */
        for epoch do
            $\theta \leftarrow LocalTraining(SW_i, X_i, Y_i)$
        SendUpdate($\theta$)
    $W^r \leftarrow W^{r-1} + \sum_{i=0}^{N} \theta_i$
    $P \leftarrow CalculateAccuracyDiff(W^r, {SW_i}^{r-1})$
Function CreateSubnetworks($W$, $D$):
    FC $\leftarrow$ SelectFullyConnectedLayer($W$)
    Layers $\leftarrow$ Count($FC$)
    for $i \in 1 \ldots$ Layers do
        for $fc \in FC$ do
            masks $\leftarrow$ CreateMask(fc)
            neuorns $\leftarrow$ Choose$_{n}eurons(fc$,D)
            SW $\leftarrow$ ApplyMask(fc,neurons,masks)
        $SW_i \leftarrow SW$
    return $SW$
```

Similarly, creating a variable α will decide how many subnetworks need to be created for a particular round of training. Based on α, random activation of links among the nodes can be used to create the required subnetwork.

- **Step 2:** Sharing the subnetwork to the client (repetition vs. unique subnetwork): The rule for sharing the subnetwork will be crucial for the privacy protection of the global model. Some of the possible rules could be sharing a unique subnetwork with each client (a vast number of submodels need to be created, i.e., a large value of α) versus repetition of submodel, i.e., a single submodel can be shared among multiple clients (it will reduce the value of α, however, will pose threats to privacy, in case of collaborating clients).
- **Step 3:** Local training and update sharing: It will be the same as vanilla FL and needs no changes.
- **Step 4:** Aggregation of update and recovering the dense global model. Aggregation of updates will vary slightly, and the server needs to aggregate various subnetworks instead of updates. For the privacy of the global model, there should be a significant difference between the accuracy of the subnetwork and the global model in each training round.

C. *Usecase: Binary Split and Aggregation*

A simple approach could be a simple binary split of the global model by enabling and disabling alternative activation links and sharing both sub-models with two disjoint sets of clients. Each set of clients will train a part of the sub-model and share updates with the server. Once the updates are received from each set of clients, the server can aggregate each sub-model separately and merge the model to get a better global one. The process can be repeated for a fixed number of training rounds or up to required performance requirements. However, this approach will not be sufficient if each subnetwork has good individual accuracy and there is less improvement after aggregation.

IV. Experiments and Results

We have performed all the experiments on a server computer having Intel(R) Core(TM) i9-10980XE CPU @ 3.00GHz processor with Ubuntu 18.04 64-bit operating system and Python 3.9 with PyTorch deep learning framework federated learning. We experimented with the MNIST [13] having a total $70K$ data sample divided into training $60K$ and test dataset $10K$.

We divided the training dataset into $1,000$ clients using an independent and identically distributed (IID) approach, i.e., each client has similar data and class distribution. For training, each client got 60 samples for all ten classes, and the testing data has $10,000$ samples. Our model architecture has three fully connected (FC) layers, and a ReLu activation layer follows the first two FC layers. In each training round, ten clients get selected randomly, and so the server creates ten subnetworks, one for each client. The input image in the MNIST dataset has 28×28 pixels, so the first FC has 728 neurons. Local training is done with 10 epochs with a batch size of 32.

Yang et al. [6] have suggested three choices for freezing the variables: 1) be fixed (fixed), 2) vary per round (PR), and 3) vary per client per round (PCPR). The authors stated that PCPR provides the highest accuracy than others with the same training rounds, and it also works well for training that starts with a scratch model. Our privacy study is similar to PCPR; we choose a fixed percentage of neurons to deactivate in each training round and create subnetworks to share with participants. The number of subnetworks equals the number of participants (10 in our study).

FedDrop [7] proposes an adaptive dropout rate as per the device capacity, i.e., the device with a large capacity will have a lower dropout rate, which means a larger model size. In the case of a malicious participant (for example, participant 3 shown in Fig. 1) with more capacity will get a subnetwork having a large number of enable weights, and that will help to recover the global model faster with higher accuracy. The uniform dropout is similar to vanilla FL, except it has a smaller size model (due to dropout), but all clients trained and shared updates using the same subnetwork in a training round.

For our experiments, first, we excluded the adaptive dropout and applied the same dropout rate for each device. Later,

TABLE II
ACCURACY (ACC) OF AGGREGATED GLOBAL MODEL, AVERAGE ACCURACY (AVR) AND MAXIMUM ACCURACY (MAX) OF THE LAST SET OF SUBNETWORKS AFTER FINAL TRAINING ROUND (100 ROUNDS) IN IID FOR MNIST, D: NEURONS DROPOUT PERCENTAGE

D%	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Avr	81.57	74.18	62.57	58.73	60.03	44.61	41.24	27.10
Acc	86.86	82.91	71.60	66.60	64.72	48.34	46.42	31.88
Diff.	5.29	8.73	8.93	7.87	4.69	3.73	5.18	4.78
Max	84.04	79.51	65.17	62.57	68.59	49.80	49.28	34.37

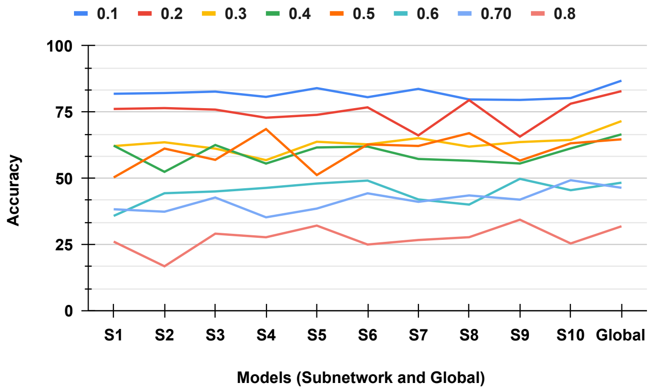


Fig. 2. Accuracy of subnetworks and global model with different value of neurons dropping percentage (D) in the last training round

we assume that a malicious participant has more resources and gets a subnetwork created with a lower dropout rate. Table II shows the accuracy of the aggregated model and the average accuracy of all ten subnetworks after the 100-th round of training on MNIST. Based on the result, we can observe the accuracy difference between average accuracy (Avr) and aggregated accuracy (Acc) of subnetworks range between $3 - 8\%$; such a narrow gap highlights the privacy concerns of the global model. In addition, the difference between the highest subnetwork accuracy and the aggregated model accuracy is dependent on the level of dropout or deactivation of neurons. For example, the aggregated model has higher accuracy up to 0.5, while individual subnetwork accuracy is higher if the neuron deactivation crosses 0.5.

Fig. 2 shows the accuracy of all ten subnetworks and global models with different values of D i.e., $(0.1 \ldots 0.8)$ in the last training round. From the line chart, we can observe that in each dropout case, there are 1-3 subnetworks with equivalent accuracy as the global model, which indicates the privacy breach of the global model. Further, if the participants with high-accuracy models can collaborate, they can achieve a model equivalent to the global one. So, such subnetwork-based federated learning cannot protect the access to the global model from the participants.

V. CONCLUSION AND FUTURE DIRECTIONS

Federated Learning is a data-centric approach and aims to protect the privacy of client's data. However, model privacy is also essential and has become critical for use cases like incentive-based or crowdsourcing-based FL. The proposed work introduces model privacy concerning the requirement of restricted access to the global model in vanilla FL and subnetwork-based FL. We also studied the possibility of model privacy in subnetwork-based FL. We provided experimental results to demonstrate that individual participants can have a model with accuracy close to the global model. Further, we can extend our study to observe model privacy in the case of participants collaboration and different training scenarios, such as the impact of resource-based adaptive deactivation of neurons and malicious participants with extensive resources.

ACKNOWLEDGMENT

This research was supported by the MSIT Korea under the NRF Korea (NRF-2022R1A2C4001270) and the Information Technology Research Center (ITRC) support program (IITP-2022-2020-0-01602) supervised by the IITP.

REFERENCES

[1] M. Barni, C. Orlandi, and A. Piva, "A privacy-preserving protocol for neural-network-based computation," in *Proceedings of the 8th Workshop on Multimedia and Security*, MM&Sec '06, (New York, NY, USA), p. 146–151, Association for Computing Machinery, 2006.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[3] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *arXiv preprint arXiv:1812.00564*, 2018.

[4] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 8485–8493, 2022.

[5] D. Feng, C. Helena, W. Y. B. Lim, J. S. Ng, H. Jiang, Z. Xiong, J. Kang, H. Yu, D. Niyato, and C. Miao, "Crowdfl: A marketplace for crowdsourced federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 13164–13166, 2022.

[6] T.-J. Yang, D. Guliani, F. Beaufays, and G. Motta, "Partial variable training for efficient on-device federated learning," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4348–4352, IEEE, 2022.

[7] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout—a simple approach for enabling federated learning on resource constrained devices," *IEEE wireless communications letters*, vol. 11, no. 5, pp. 923–927, 2022.

[8] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10700–10714, 2019.

[9] J. Zhang, C. Li, A. Robles-Kelly, and M. Kankanhalli, "Hierarchically fair federated learning," *arXiv preprint arXiv:2004.10386*, 2020.

[10] A. Ahmed and B. J. Choi, "Frimfl: A fair and reliable incentive mechanism in federated learning," *Electronics*, vol. 12, no. 15, p. 3259, 2023.

[11] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[12] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.

[13] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.