A Model Personalization-based Federated Learning Approach for Heterogeneous Participants with Variability in the Dataset

RAHUL MISHRA and HARI PRABHAT GUPTA, IIT (BHU) Varanasi, India

Federated learning is an emerging paradigm that provides privacy-preserving collaboration among multiple participants for model training without sharing private data. The participants with heterogeneous devices and networking resources decelerate the training and aggregation. The dataset of the participant also possesses a high level of variability, which means the characteristics of the dataset change over time. Moreover, it is a prerequisite to preserve the personalized characteristics of the local dataset on each participant device to achieve better performance. This article proposes a model personalization-based federated learning approach in the presence of variability in the local datasets. The approach involves participants with heterogeneous devices and networking resources. The central server initiates the approach and constructs a base model that executes on most participants. The approach simultaneously learns the personalized model and handles the variability in the datasets. We propose a knowledge distillation-based early-halting approach for devices where the base model does not fit directly. The early halting speeds up the training of the model. We also propose an aperiodic global update approach that helps participants to share their updated parameters aperiodically with server. Finally, we perform a real-world study to evaluate the performance of the approach and compare with state-of-the-art techniques.

CCS Concepts: • **Computing methodologies** → **Machine learning**; • **Human-centered computing** → **Ubiquitous and mobile devices;**

Additional Key Words and Phrases: Dataset variability, early halting, federated learning, personalization

ACM Reference format:
Rahul Mishra and Hari Prabhat Gupta. 2023. A Model Personalization-based Federated Learning Approach for Heterogeneous Participants with Variability in the Dataset. *ACM Trans. Sensor Netw.* 20, 1, Article 22 (December 2023), 28 pages.
https://doi.org/10.1145/3629978

1 INTRODUCTION

Ubiquitous system has recently emerged as an attractive paradigm, which facilitates easier and convenient data collection using low-cost and small-size devices [36]. The system generates a large amount of valuable data that can be used to train deep learning models on the centralized server. In addition, the growth of the ubiquitous system also creates the possibility of collecting and processing personalized data. However, sharing personalized data to the central server has raised privacy

Authors' address: R. Mishra and H. P. Gupta, Indian Institute of Technology (Banaras Hindu University) Varanasi, BHU Campus, Varanasi, Uttar Pradesh, 221005, India; e-mails: {rahulmishra.rs.cse17, hariprabhat.cse}@iitbhu.ac.in.

1550-4859/2023/12-ART22 $15.00
https://doi.org/10.1145/3629978

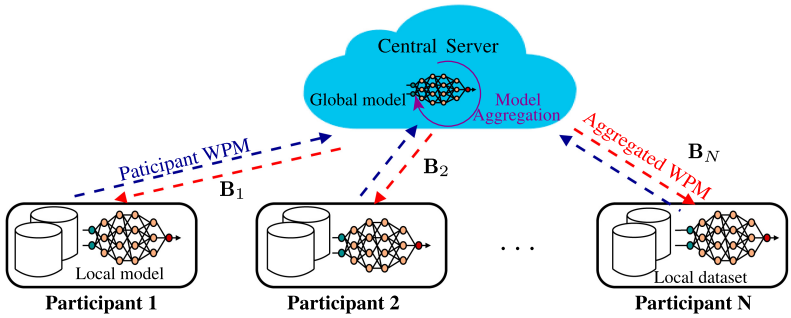


Fig. 1. An example scenario of FL with N participants (e.g., smartphones) with heterogeneous devices and networking resources. $B_1, \ldots, B_N$ denote unequal bandwidths among the participants and the central server.

concerns and security threats. Moreover, limited networking resources (e.g., bandwidth) between the devices and the central server incur communication delays and increase the training time of the model [17, 40]. **Federated Learning (FL)** is a collaborative technique to train the model without using local data of participant devices at the central server [33, 35, 44]. Each participant device in FL trains the model using the local dataset and sends the trained model's **Weight Parameter Matrices (WPM)** to the central server. In turn, the central server aggregates the WPM received from participant devices and sends them back the updated WPM. FL finds various applications in the ubiquitous system, including human activities recognition, healthcare system, vehicular edge computing, intelligent recommendations (e.g., Gboard), and so on [26, 34, 39].

A participant device in FL uses its resources, such as memory and processing power to load the model and train them locally. The availability of resources at the participant devices depends on their type and other installed services. Such heterogeneity in device resources requires unequal time to train the model using local datasets. The heterogeneity in the devices and networking resources implies that all devices may not simultaneously transfer the WPM to the central server for the aggregation and hence slow down the FL. Therefore, it becomes challenging to handle heterogeneity among participant devices in FL. Figure 1 illustrates an example scenario of FL that trains a shared model on the participant devices using local datasets. The participant devices are the smartphones of different brands, and have unequal memory and processing power. Similarly, the participants are at diverse locations; thus, the bandwidth between the devices and the server may be non-identical.

FL involves the aggregation of WPM from multiple participant devices after each communication round. Such aggregation improves the generalization ability by utilizing the local dataset characteristics of multiple participant devices. However, the aggregation may vanish the local properties or personalized features of the participants' datasets. Personalization in FL refers to the process of preserving the dataset characteristics of the participants [7, 20, 25, 28]. For example, the participants can only send the personalized layers for aggregation and train the local model with personalized layers [20]. Dataset variability leads to performance diminution and frequently occurs in real-world [15, 47, 57]. Specifically, the available dataset of the participants in FL is sensitive to the time of data collection, environmental conditions, and even the emotions of the participants. Thus, it is tedious to simultaneously maintain personalization and handle dataset variability on the participant devices.

To address the heterogeneity in resources among the participants, prior studies proposed mechanisms that discard slow processing participants, called *stragglers*, from the federation [3, 22]. However, the removal of stragglers hampers effective utilization of the local datasets (on stragglers) and

prohibits performance improvement. Some existing work [27] have considered a fixed size model for all the devices with heterogeneous resources. The fixed size model may not fully utilize the colossal resources on the devices. The existing work [45, 56] used **Knowledge Distillation** (**KD**) to resize and train the model that fit on the devices. The KD is a student-teacher learning process. The training of the student model by the teacher model sometimes requires multiple epochs in KD; therefore, delays the aggregation process at the central server.

While considering the heterogeneity in devices and networking resources, we are the first to simultaneously personalize the model for the participants and handle the dataset variability. We present a model personalization-based federated learning approach in the presence of heterogeneity in devices and networking resources. This work investigates the following problem: *how does FL successfully train a model on the participant devices with heterogeneous resources while ensuring model personalization and considering dataset variability?* To this end, the major contributions and novelty of this work are as follows:

- We first construct a base model on the server using the resources information of the participants. We next design an approach to identify the personalized layers of the base model on participants in accordance with their datasets and resources. We consider the *dataset variability* on the participants, which changes personalized layers over time. Therefore, we preserve non-personalized layers and train with personalized layers for limited epochs to handle variability. We called this novel technique as *re-personalization*, which retains the performance despite dataset variability.
- We consider the scenarios where devices have sufficient, colossal, and insufficient resources to train the model. The approach uses *knowledge distillation* to train resized base models for insufficient and colossal resource devices. To speed up the model's training at insufficient and colossal resource devices, we propose *a novel early halting technique*, where training halts at halting epochs. We derive an *expression to determine the halting epoch* for a given accuracy. Furthermore, we propose *an aperiodic global update approach* where the server does not wait to receive the WPM from all the participants to estimate the aggregated WPM. The duration of two consecutive global updates divides into fixed time intervals. The server aggregates the received WPM in each interval and uses the aggregated WPM in the next interval. Such aperiodic global update speedups the training.
- Finally, we perform a *real-world study* to analyze the heterogeneity in devices and networking resources among the participants. The task of the real-world study was to recognize the locomotion modes of the users. We also consider the task of handwritten digit recognition, image classification, and human activity recognition. We verify the proposed approach on the existing baseline techniques (HetroFL [5], FedProx [23], FedAvg [31], and Hermes [20]), collected and existing datasets [1, 16, 19, 41], and validation metrics.

The rest of the article is organized as follows. In the next section, we briefly discuss the prior studies and motivation. Section 3 presents the approach to train a model on the participant devices with heterogeneous resources, followed by the theoretical analysis in Section 4. The real-world study and performance evaluation are discussed in Sections 5 and 6, respectively. Finally, the article concludes in Section 7.

2 BACKGROUND AND MOTIVATION

This section first presents the existing FL work that considered the heterogeneity in devices and communication resources. We next discuss the prior studies that considered KD, personalization of models in FL and heterogeneous dataset in FL.

2.1 Heterogeneity in Device Resources

The authors in [12] highlighted the problem of heterogeneous devices in FL, which limits the size of the global model to accommodate low resource or slow participants. They proposed an ordered dropout approach, FjORD, which dynamically adapts the model's size for heterogeneous devices. FjORD identified the candidate values of dropout and determined multiple models. The participant selects an appropriate model. The authors in [5] introduced the technique of handling variation in computational and communication resources. They named the technique as HetroFL, which produces a single global model apart from multiple size global models in [12]. The server sent the portion of the global model to the participants.

2.2 Heterogeneity in Communication Resources

The authors in [44] recognized the issue of limited and dynamic communication resources among participants in FL. They proposed a mechanism to control the number of global aggregations at the server and reduce the learning loss to minimize the communication budget. The authors in [18] presented a specialized technique, namely, Oort, which prioritizes the selection of participants in FL. The technique selected those participants that offered the highest utility and minimizes the communication delay. The authors in [23] introduced a framework called FedProx to handle the issue of heterogeneity in FL. FedProx used a proximal term to minimize the impact of updates and restricted it close to the server's model.

2.3 Knowledge Distillation in FL

The authors in [9] presented a group knowledge transfer training algorithm, abbreviated as FedGKT. The target is to train a large-size CNN model on the server using the WPM from the different and small-size models on heterogeneous participants. Authors in [27] utilized KD in FL for transforming different size models of participants into equal sizes. The participant devices in [45] trained a large-size model, converted to the lightweight model using KD, and communicated to the server. The devices converted the received lightweight model from the server to a large-size using reverse-KD. It helped to reduce communication overhead.

2.4 Personalization in FL

Hermes [20] is a computational and communication efficient framework to handle the critical bottleneck of communication cost and data heterogeneity. Hermes used a structured pruning technique to develop the personalized model for each participant. Authors in [6] proposed an algorithm that performed global aggregation in FL, namely, FedDist, where the aggregation depends upon the similarity among the WPM from the participants. Furthermore, the authors in [25] proposed an approach that addressed the data heterogeneity and provided communication efficiency using two-step learning to achieve personalization.

2.5 Heterogeneous Dataset in FL

The authors in [46] introduced a federated learning approach, HiFlash, which ensured communication efficiency using adaptive staleness control and also caters heterogeneity in participants-edge association. HiFlash proved to be an important approach for addressing heterogeneity in federated learning through its hierarchical structure and adaptive staleness control. The approach optimized communication and enhanced its efficiency. To handle heterogeneity in dataset of federated learning, the authors in [24] proposed a blockchain-based decentralized federated learning framework. The framework tends to be a novel and intriguing for heterogeneity in a distributed setting. The authors have utilized committee consensus method to potentially facilitate the learning process

while handling disparate datasets. The proposed work did not involve blockchain applications, however, we acknowledge the importance of exploring diverse methods to handle heterogeneity. Fedmd (heterogeneous federated learning via model distillation) [21] is introduced to handle heterogeneity of dataset in federated learning. It leveraged the knowledge of a global model to distill the local models' information, this method can potentially mitigate the effects of varying datasets.

Furthermore, Zheng et al. in [4] addressed the critical challenge of handling resource and data heterogeneity in the context of federated learning. The authors proposed strategies to mitigate the impact of varying computational resources and diverse data distributions across clients in a federated learning. Similarly, the authors in [13] addressed the challenge of non-IID (Non-Independently and Identically Distributed) data in federated learning, especially when dealing with heterogeneous datasets. They employed cross-silo federated learning, where data from different sources exhibit significant differences. In addition, the approach utilized data transformation and adaptation strategies to enable effective knowledge sharing across diverse data distributions. Finally, the authors in [38] proposed a federated learning framework, FedBoost, which leverages the diversity of participants' datasets to improve overall performance. FedBoost strategically assigning more weight to participants with better performance during aggregation, which helps in handling heterogeneity of participants' dataset.

2.6 Motivation

This work is motivated from the following limitations in prior studies. The local model may not achieve adequate accuracy if its weights are discarded during aggregation [3]. Reducing the processing power of the device during training of the model slows down the aggregation process [49, 51]. Additionally, estimating the exact complexity of the model supported by a participant is tedious [5]. Suppressing the communication round for aggregation [14, 44, 48] also increases the stale models. The parallel training and communication come with the cost of gradient-staleness [55]. Considering a fixed size of lightweight models is not suitable for unequal resources participant devices [9]. Sending WPM of the lightweight model to the central server increases the number of the round for aggregation [45]. Moreover, using KD in FL slows down the training of models [27, 45, 56]. The existing approaches have considered model personalization [7, 20, 25, 28]; however, non-of-the existing approaches have considered variability in the datasets. *In summary, the existing FL in the presence of heterogeneous resources avoids the straggler devices during aggression, delays the aggression process, and/or reduces the number of aggregation round. Moreover, none-of-the existing work handled the issue of variability in dataset of participants.*

3 MODEL PERSONALIZATION-BASED FEDERATED LEARNING APPROACH

This section proposes a model personalization-based FL approach for heterogeneous participants with variability in their local datasets. Figure 2 shows the framework of the proposed approach. The server initiates the approach by collecting the available resource information from K out of N participants then constructs and randomly initializes a base model, denoted as M_o (①). The server transfers M_o to all the devices (②). M_o is successfully trained on sufficient resources devices, where personalized models are obtained by analyzing the WPM of trained M_o. We next propose a KD-based early halting approach for insufficient or colossal resources devices to train M_o. The halting approach speeds up the training process and improves the performance of the model within available resources (③). Afterwards, participants share their updated WPM of the personalized and trained model (④). Furthermore, we propose an aperiodic global update approach that helps the participant devices to share their updated WPM aperiodically with server (⑤). Finally, the server sends the aggregated and personalized WPM to the participants (⑥). Steps ③ – ⑥ are repeated for sufficient communication rounds to achieve desired performance. Algorithm 1 shows the steps

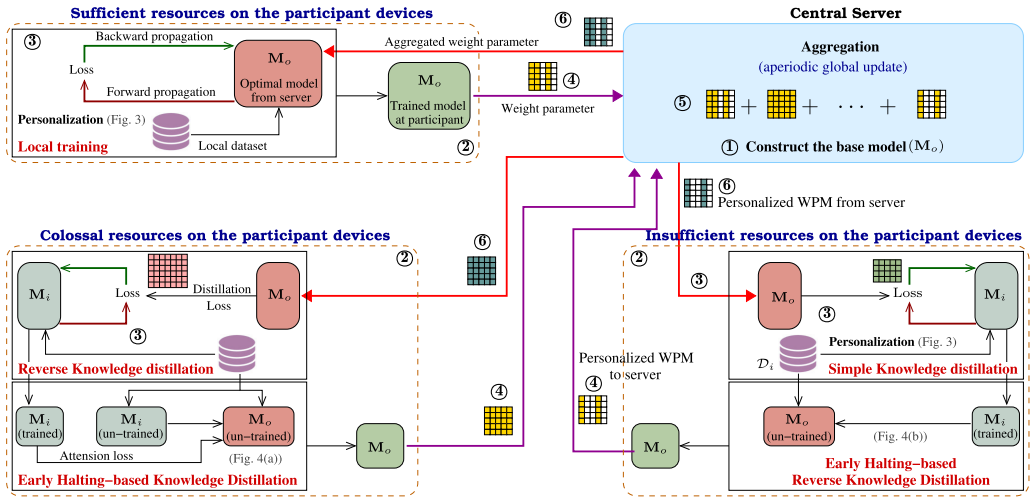


Fig. 2. Illustration of the framework for the model personalization-based FL approach.

of the proposed approach. In this work, the participants do not share data (or its characteristics) with each other and the server. Therefore, the approach did not violate the dataset privacy of the participants. However, we do not consider the mechanism to encrypt or protect WPM from attackers.

3.1 Construction of Model at Central Server

This work assumes a scenario comprising a set $\mathcal{P}$ of N participants, $\mathcal{P} = \{p_1, \ldots, p_N\}$ and a central server. The server randomly selects K ($K < N$) out of N participants for training. The server constructs a model prior to the FL-based training on K participants. However, constructing a one-fits-all model for K participants is tedious. If the model is too large, some participants may not accommodate and delay the aggregation. We introduce two simple and sequential steps to circumvent the difficulties to construct a base model. Firstly, the K participants send available resource information to the server. The server extracts the information of the available resources from participants and then develops an initial model ("base model in this work") suitable for most of the participants. Specifically, the server constructs a one-fits-all model, which may be too small to achieve adequate performance.

In the second step, the server enhances the model size using the technique discussed in [30]. This step makes the model large enough to achieve adequate performance with the condition that the majority of participants can accommodate it. By majority, we mean that more than 50% participants can directly run the base model M_o. Initially, determining the one-fit-all lightweight model and imposing a 50% participants cap differentiate the proposed approach from [30]. Moreover, the technique in [30] is tested in centralized training, and we utilize and test it in federated learning-based model training.

Let $\mathcal{T}$ be the maximum allowable time between two communication rounds. The server sets the local epochs, denoted as E, to train the base model M_o on all the participants. Later, the server fetches the information about the number of instances n_i in the local dataset d_i of p_i, $\forall i \in \{1 \leq i \leq K\}$. The server estimates the local training time by determining the floating-point operations of M_o for one epoch using n_i. Afterward, the time to execute the floating-point operations is estimated for one epoch using the available processing power of the participants. Further,

the server evaluates the time to execute E epochs of training, i.e., local training time. The server enhances model M_o such that majority of the participants can train the base model within $\mathcal{T}$. The remaining participants that are incompetent in training M_o, reduce the size of M_o using pruning and KD techniques, discussed later (participants with insufficient resources). The model selection steps are shown in Procedure 1.

Procedure 1: Construction of model at central server

Input: Resource information of $\mathcal{P}$ participants, n_i of p_i $(1 \leq i \leq N)$, local epoch E, and threshold $\mathcal{T}$;
Server randomly selects K out of N participants;
/*__Step 1:__ Constructing one-fits-all small size base model*/
for *each participant* $p_i \in \mathcal{P}$, *where* $1 \leq i \leq K$ **do**
    Server extracts resources information of p_i;
    $A \leftarrow A.append$ (available resource on p_i); $B \leftarrow B.append$ (n_i on p_i);
$a \leftarrow \min(A)$; Construct a one-fits-all model M_o satisfying a;
/*__Step 2:__ Enhancing the model size M_o using [30]*/
while $P_{count} \geq K/2$ **do**
    $P_{count} \leftarrow 0$;
    for *each participant* $p_i \in \mathcal{P}$, *where* $1 \leq i \leq K$ **do**
        $e \leftarrow$ Estimate training time of M_o on p_i using n_i and E; /*Using list B obtain n_i on p_i*/
        if $e \leq \mathcal{T}$ **then**
            $P_{count} \leftarrow P_{count} + 1$;
    $M_o \leftarrow$ enhance size of M_o;
return base model M_o;

3.2 Model Personalization and Training on Participant Devices

This section describes the model personalization and training of the base model M_o, obtained previously. Procedure 2 illustrates the different steps for model personalization and training in presence of dataset variability.

3.2.1 Model Personalization. This work designs a personalized model for each participant depending upon the current state of the dataset and M_o from the server. The authors in [20] performed structured pruning of layers to obtain personalized model, where few unimportant layers of the models are pruned permanently. Since we consider the high-level of variability in the local dataset; thus, personalized layer changes overtime. Thus, we can not directly employ such pruning technique. In the proposed approach, each p_i learns a personalized model using the structured pruning technique, as discussed in [20], and also retains the pruned connection to handle the variability in the dataset. We introduce a mechanism to perform efficient training that stops training of non-personalized layers of model at the participant after pre-specified epochs $< E$, e.g., 25% epochs considered in the experiment. However, the training of personalized layers is continued for E. This stopping preserves the training resources without performance compromise and makes pruned layers usable on changing the dataset status.

Each participant performs channel-wise and filter-wise pruning for CNN and row-wise and column-wise pruning for fully connected layer. The participant generates a binary mask of the WPM, where 0 and 1 indicate pruned and un-pruned values, respectively. After training, the participants send WPM along with the binary mask, which facilitates the server to identify the pruned connections. Here, the transmission of the binary mask produces ignorable communication overhead compare to floating-point parameters. Figure 3 illustrates the model personalization and re-personalization for handling dataset variability on p_i, where $i \in \{1 \leq i \leq N\}$. ① server initially

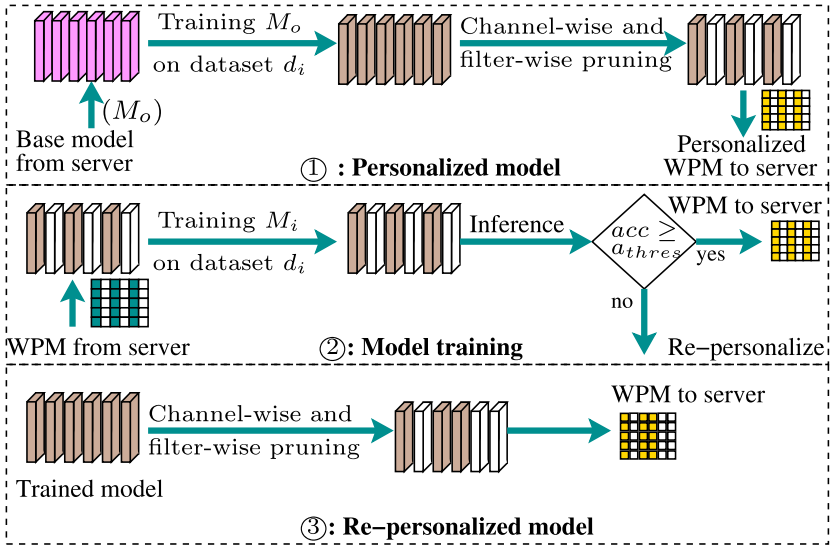


Fig. 3. Illustration of model personalization and re-personalization to handle variability in the local dataset of the participant p_i, where $i \in \{1 \leq i \leq N\}$.

sends base model M_o to the participant that trains and personalizes the model. ② participant p_i train personalized model. ③ re-personalization of the model on the participant to handle variability in the dataset.

3.2.2 Model Training. Each participant $p_i \in \mathcal{P}$ trains the received model then personalize M_o using local dataset d_i. The duration of the training, personalization and inference on the participant devices depend on their processing power. Similarly, the storage requirement relies upon the size of the model. We consider three scenarios based on the heterogeneity in available resources on the participant devices. In the first scenario, the available resources are sufficient to train and obtain a personalize model from M_o. The other scenarios are possible when the participant device's resources are colossal or insufficient to train, personalized and perform inference on model M_o.

(1) Participant devices with sufficient resources: In this scenario, the available resources of participant devices match the requirement of resources to train, personalize and perform inference on model M_o. This scenario is illustrated in Figure 2. The training of M_o incorporates forward and backward propagation for E local iterations. WPM of the trained model is analyzed to determine the personalized and non-personalized layers. Later, the binary mask is generated and sent with the WPM of the personalized layer to the server for aggregation.

The server sends the updated WPM of the personalized layer to the participants. Furthermore, the participants replace their WPM of personalized layer and train the local model with personalized and non-personalized layers. The training of the non-personalized layers is stopped after certain epochs prior to E, whereas personalized layers are trained for E epochs. This training of the local model and aggregation is continued until the performance is greater than a threshold, denoted as a_{thres}. When the model's performance deteriorates due to variability, the personalized layers are re-identified from the trained model, and the above steps are repeated, as shown in Figure 3. We also incorporate loss function, denoted as $\mathcal{L}_a(\cdot)$, during training of M_o on p_i. $\mathcal{L}_a(\cdot)$ estimates the discrepancy between predicted and actual labels of d_i for classification tasks. $\mathcal{L}_a(\cdot)$

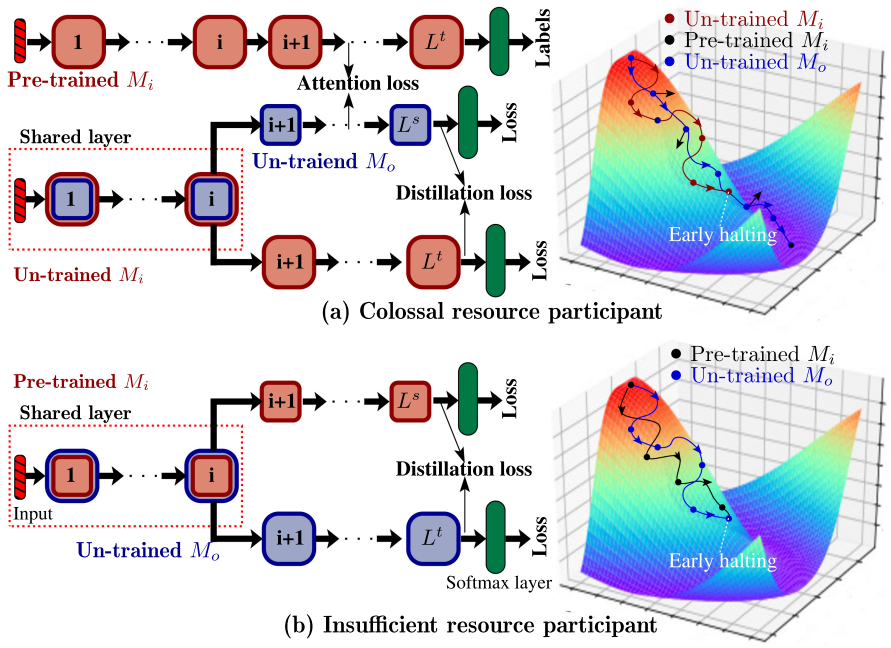


Fig. 4. Illustration of early halting and layer sharing among student and teacher, while training the base model.

can be probabilistic (i.e., cross-entropy), regression (i.e., mean square error), or hinge (i.e., hinge function) depending upon the scenarios.

(2) Participant devices with colossal resources: A participant p_i with colossal resources can run a sophisticated model M_i that achieves better inference performance than model M_o. We use the deep network growing concept discussed in [43] to generate M_i from M_o. We also employ reverse KD technique with the loss $\mathcal{L}_a(\cdot)$ and distillation loss $\mathcal{L}_{DL}(\cdot)$ [50] to utilize features of M_o while training M_i, as shown in Figure 2. The training of M_i continued for E local epochs on d_i. Later, the participant p_i uses trained M_i for local inference. p_i regenerates M_o from trained M_i to share the updated WPM to the server for the next round of aggregation. We use the KD technique [11] to transfer the knowledge from the trained M_i to M_o. The logits of trained M_i become a hard target for M_o; therefore, the comparisons of their logits don't provide satisfactory performance [52]. Furthermore, using untrained M_i and trained M_i teachers for training student M_o guides M_o with appropriate initialization and provides smooth logits for comparison. This process requires huge training parameters [54]. To overcome these issues, the approach considers M_i as untrained and trained teacher models and M_o as student model with layers sharing, as shown in Figure 4(a). However, the training of M_o using trained and untrained M_i requires enormous resources.

The proposed approach *early halts* the training of the untrained M_i model after epochs h_1 to overcome the requirement of enormous resources, where $h_1 < E$. The early halting saves the device's resources during training of M_o with no performance compromise. Hereafter, the training of M_o will continue under the guidance of trained M_i. Theorem 1 proves that the number of epochs h_1 to halt the training of untrained M_i is sufficient to achieve the desired accuracy. The early halting technique uses loss $\mathcal{L}_a(\cdot)$, attention loss $\mathcal{L}_{AL}(\cdot)$ [52], and distillation loss $\mathcal{L}_{DL}(\cdot)$ [11], as shown in Figure 4(a). The performance of M_o can be improved under the supervision of trained M_i that

compares output at each epoch. The comparison is carried out using attention loss between M_o and M_i. The combined loss ($\mathcal{L}_{comb}(\cdot)$), which operates during simultaneous training of M_o and untrained M_i, is given by

$$\mathcal{L}_{comb}(\cdot) = \begin{cases} \lambda_1\mathcal{L}_a^o(\cdot) + \lambda_2\mathcal{L}_{AL}(\cdot) + \lambda_3\mathcal{L}_{DL}(\cdot) + \lambda_4\mathcal{L}_a^i(\cdot), \text{ till training of untrained } M_i, \\ \lambda_1\mathcal{L}_a^o(\cdot) + \lambda_2\mathcal{L}_{AL}(\cdot) + \lambda_3\mathcal{L}_{DL}(\cdot), \end{cases} \tag{1}$$

where λ_1, λ_2, λ_3, and λ_4 are the fractional contribution of different loss functions, $0 < \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} < 1$. We only optimize the combined loss associated with M_o, as the contribution of the loss of untrained M_i is uniform throughout the training of M_o. The early halting optimizes the following problem

$$\min \ \mathcal{L}_{comb}^o(\cdot) \quad s.t., \lambda_1 + \lambda_2 + \lambda_3 = 1, \ 0 < \{\lambda_1, \lambda_2, \lambda_3\} < 1. \tag{2}$$

The participants with colossal resources do not identify personalized layers. Such participants with abundant resources can effectively train and communicate WPM of the model incorporating all the layers without pruning.

(3) Participant devices with insufficient resources: The participants with insufficient resources can not perform training and personalization, simultaneously. In other words, a participant p_i with insufficient resources can run a less complex model M_i, which provides inferior performance than M_o. We use the pruning technique discussed in [8] to generate a small size model M_i from M_o. We also employ KD technique [11] while generating small model using $\mathcal{L}_a(\cdot)$ and $\mathcal{L}_{DL}(\cdot)$, as shown in Figure 2. The participant uses trained M_i for inference.

The participant p_i regenerates M_o using trained M_i to obtain a personalized model. We use KD to transfer the knowledge from the trained teacher model M_i to the student model M_o. Due to insufficient resources on participant p_i, it is tedious to train M_o for E epochs in a limited time. We use the proposed early halting approach as shown in Figure 4(b). During the training of the M_o model, we do not use knowledge from the untrained model M_i due to limited resources. The knowledge from trained M_i is used to guide the training of M_o, where training of M_o is halted at epoch h_2 ($h_2 < E$). We can obtain h_2 using Theorem 1. Similar to the sufficient resources participants, we determine the personalized layers in trained M_o on the participant by analyzing the WPM. The training is continued until the performance $\geq acc_{thres}$. When the model's performance deteriorates due to dataset variability, the personalized layers are re-identified from the trained local model, and the above steps are repeated.

3.3 Aperiodic Global Update

Each participant sends the WPM of the trained model to the central server for aggregation. The server may not receive the updated WPM simultaneously from all the participants if participant devices have unequal network bandwidth and processing power. The waiting for the updated WPM from all the participants at the server introduces unavoidable delays during aggregation. The proposed approach introduces aperiodic global updates at the server inspired from [53] to overcome the above problem. It allows each participant p_i to aperiodically transfers its updated WPM $W_i^{[t]}$ to the server at global iteration t of T time interval, where $t \leq R$ and R is the number of global iteration. The following steps are executed at iteration t:

Step 1: Let none of the participants has sent the updated WPM before t, and the server has WPM $W^{[t-1]}$. Let k participants of set $\mathcal{P}$ have to send their WPM in threshold time $\mathcal{T}$. Let η and Q_i denote the learning rate and the number of instances in d_i, respectively. The server performs

Procedure 2: Model personalization and training on participant devices

Input: Base model M_o, h_1 h_2, local epochs E, accuracy threshold acc_{th}, R communication rounds, E^1_{np}, and E^2_{np};

for *each participant* $p_i \in \mathcal{P}$ **do**
  if *(p_i has sufficient resources)* **then**
    if $R == 1$ ***and*** *accuracy of* $M_o < acc_{th}$ **then**
      Train M_o on d_i; lp_i, lnp_i;
      $B_i \leftarrow$*Personalization* (M_o, d_i);
    Train M_o with lp_i and lnp_i layers for E^1_{np} epochs;
    Train M_o with lp_i layers for remaining $(E - E^1_{np})$ epochs;
    return WPM for lp_i layers and B_i to the central server;
  else if *(p_i has colossal resources)* **then**
    Train M_i from M_o using reverse KD and d_i;
    for *epoch* $e \leq E$ **do**
      if $e \leq h_1$ **then**
        Train M_o using pre-trained and untrained M_i;
      else
        Train M_o using pre-trained M_i;
    return WPM for M_o to the central server;
  else
    if $R == 1$ ***and*** *accuracy of* $M_o < acc_{th}$ **then**
      i). Train M_i from M_o using KD and d_i;
      ii). Train M_o using pre-trained M_i for h_2 epochs;
      lp_i, lnp_i, $B_i \leftarrow$*Personalization* (M_o, d_i);
    Train M_o with lp_i and lnp_i layers for E^2_{np} epochs;
    Train M_o with lp_i layers for remaining $(E - E^2_{np})$ epochs;
    return WPM for lp_i layers and B_i to the central server;
return M_i for p_i and WPM of M_o for central server;

Function *Personalization* (Model M, Dataset d)
  Use pruning to obtain personalized layers L_p of M and determine binary mask B_M ;
  Preserve the pruned components to obtain non-personalized layers L_{np} of M;
  return L_p, L_{np}, and B_M;

aggregation at $t + \mathcal{T}$ and updated WPM is given as

$$W^{[t+\mathcal{T}]} = W^{[t-1]} - \eta\nabla(W'), \text{where}, W' = \sum_{i=1}^{k} \left(\frac{Q_i}{Q_1 + Q_2 + \cdots + Q_k}\right) W_i^{[t]}. \tag{3}$$

The sever sends back the updated WPM $W^{[t+\mathcal{T}]}$ to all k participants.

Step 2: Let l ($l \in \{1, 2, \ldots, N - k\}$) denotes the number of participants, which have send their WPM in the interval $t + \mathcal{T}$ and $t + 2\mathcal{T}$. Similar as Step 1, the server performs aggregation at $t + 2\mathcal{T}$ to obtain WPM $W^{[t+2\mathcal{T}]}$ as

$$W^{[t+2\mathcal{T}]} = W^{[t+\mathcal{T}]} - \eta\Big(\nabla(W') + \delta\nabla(W') \odot \nabla(W') \odot (W^{[t+\mathcal{T}]} - W^{[t-1]})\Big), \tag{4}$$

where, W' is evaluated using Equation (3) for l participants. δ is a variable that lies in range $[0, 1]$ and symbol $\odot$ represents an element-wise product. This step repeats until all the participants not send their WMP to the server. Aperiodic global update steps are shown in Procedure 3.

The global iteration time interval is estimated using the processing capacity of the slowest participant, where the server waits for the WPM from the slowest participant. Moreover, the approach provides communication efficiency; thus, the bottleneck due to the slowest participant is low. Apart from [53], the aperiodic global update mechanism allows all the participants to take part in the aggregation after each communication round, similar to the synchronous global update. It avoids the possibility of achieving desired global performance using only a subset of participants, leaving some participants partially trained or untrained [42, 53]. We deduce the mechanism of multiple intermediate updates between two global updates. It provides faster participants to perform multiple local training in-between two rounds, which aggregately results in performance improvement of all the participants.

Procedure 3: Aperiodic global update

Input: Global aggregation interval T, time threshold $\mathcal{T}$;
Initialize: $q \leftarrow 1$;
for $j \leftarrow 1$ *to* T **do**
  if $j \leq \mathcal{T}$ **then**
    Collect WPM from the participants;
    /*Let k participants send WPM in first $\mathcal{T}$ interval*/
  Aggregate WPM from k participants using Equation (3);
  while $q < \frac{T}{\mathcal{T}}$ **do**
    if $q.\mathcal{T} < j \leq (q+1)\mathcal{T}$ **then**
      Collect WPM from the participants;
    Aggregate WPM using Equation (4);
    $q \leftarrow q + 1$;
return Aggregated WPM of M_o at central server;

ALGORITHM 1: Model personalization-based federated learning approach.

Input: Set $\mathcal{P}$ of N participants with their local dataset, global iteration R;
Output: Trained model on each participant p_i $(1 \leq i \leq N)$;
Call **Procedure 1** to select base model at central server;
Central server shares model to the participants $\mathcal{P}$;
for *R global iterations* **do**
  Call **Procedure 2** to train base model using local dataset at each device;
  Call **Procedure 3** to send updated WPM from participant devices to the server;
Return Trained model at each participant device;

4 THEORETICAL ANALYSIS

This section deduces the expression for halting epochs and analyzes the convergence of the proposed approach.

4.1 Deriving Expression for Halting Epoch

We derive the expression of halting epoch h (h_1 and h_2) in terms of allowable loss variance ϵ. Let $W_H(e)$ and $W_E(e)$ denote the WPM at epoch e ($e \leq E$) when the training of model at participant incorporates halting or non-halting mechanism, respectively. Similarly, $\mathcal{L}_{comb}(W_H(e))$ and

$\mathcal{L}_{comb}(W_E(e))$ represent the combine loss (Equation (1)) at participant incorporate halting or non-halting mechanism, respectively. To derive the expression for threshold ϵ, we use the assumptions as given in [44]:

Assumption 1. *$\mathcal{L}_{comb}(\cdot)$ is ρ-Lipschitz, i.e., $\|\mathcal{L}_{comb}(W) - \mathcal{L}_{comb}(W')\| \leq \rho\|W - W'\|$ for random W and W'.*

Assumption 2. *$\mathcal{L}_{comb}(\cdot)$ is β-smooth, i.e., $\|\nabla\mathcal{L}_{comb}(W) - \nabla\mathcal{L}_{comb}(W')\| \leq \beta\|W - W'\|$.*

Definition 1 (Gradient Discrepancy). For epoch e and WPM W, upper bound of $\|\nabla W_H(e) - \nabla W_E(e)\|$ is given as:

$$\begin{array}{ll} \|\nabla\mathcal{L}_{comb}(W_H(e)) - \nabla\mathcal{L}_{comb}(W_E(e))\| = 0; & \textbf{if } e \leq h, \\ \|\nabla\mathcal{L}_{comb}(W_H(e)) - \nabla\mathcal{L}_{comb}(W_E(e))\| \leq \phi(e); & \text{otherwise} \end{array}. \tag{5}$$

$$\implies \phi = \frac{\sum_{e=h}^{E} \|\nabla\mathcal{L}_{comb}(W_H(e)) - \nabla\mathcal{L}_{comb}(W_E(e))\|}{E - h}.$$

Lemma 1. *For epoch e, where $e \in (h \leq e \leq E)$, we have: $\mathcal{V}(\|W_H(e) - W_E(e)\|_{e=h}^{E}) \leq q(e)$, where $q(e) = (1-\eta\beta)\frac{\phi}{\beta}((1-\eta\beta)^e + (1-\eta\beta))$, $\eta < \frac{1}{\beta}$, $\eta > 0$, and $\beta > 0$. $\mathcal{V}(\cdot)$ denotes the variance.*

Proof. To prove the lemma, we consider the following induction $\mathcal{V}(\|W_H(e) - W_E(e)\|_{e=h}^{E}) \leq q(e)$ and using the rule of gradient update, $W_H(e+1) = W_H(e) - \eta\nabla\mathcal{L}_{comb}(W_H(e))$, we obtain the following expression

$$\mathcal{V}(\|W_H(e+1) - W_E(e+1)\|_{e=h}^{E-1}) = \mathcal{V}(\|W_H(e) - \eta\nabla\mathcal{L}_{comb}(W_H(e)) - (W_E(e) - \eta\nabla\mathcal{L}_{comb}(W_E(e)))\|_{e=h}^{E}).$$

Using triangle inequality [44] and property of variance: $\mathcal{V}[aX + b] = a^2\mathcal{V}(X)$ for constant a and b, we obtain

$$\mathcal{V}(\|W_H(e+1) - W_E(e+1)\|_{e=h}^{E-1}) \leq (1-\eta\beta)^2\mathcal{V}(\|W_H(e) - W_E(e))\|_{e=h}^{E}). \tag{6}$$

Using considered induction, we reach to following expression

$$\mathcal{V}(\|W_H(e+1) - W_E(e+1)\|_{e=h}^{E-1}) \leq (1-\eta\beta)^2 q(e) \implies \mathcal{V}(\|W_H(e+1) - W_E(e+1)\|_{e=h}^{E-1}) = q(e+1). \tag{7}$$

Using Equation (7), we obtain, $\mathcal{V}(\|W_H(e) - W_E(e)\|_{e=h}^{E}) \leq q(e)$. Hence proved. □

Theorem 1. *If $\mathcal{V}(\|\mathcal{L}_{comb}(W_H(e)) - \mathcal{L}_{comb}(W_E(e))\|_{e=h}^{E}) \leq \epsilon$, then relation between ϵ and e $(e = h)$ is defined as: $\epsilon = \sqrt{\frac{2\rho\phi(1-\eta\beta)^{e+1}}{\upsilon\eta(2-\beta\eta)}}$, where $\eta < \frac{1}{\beta}$, $\eta > 0$, $\beta > 0$ and $\upsilon = \frac{1}{\mathcal{V}(\|W_H(e) - W_E(e)\|_{e=h}^{E})^2}$.*

Proof. For an epoch $e \in (h \leq e \leq E)$, we assume $\Psi(e)$ as

$$\Psi(e) = \mathcal{V}(\|\mathcal{L}_{comb}(W_H(e)) - \mathcal{L}_{comb}(W_E(e))\|_{e=h}^{E}). \tag{8}$$

Using β-smoothness of the loss function and property discussed in [2]:
$\mathcal{L}_{comb}(W) \leq \mathcal{L}_{comb}(W') + \nabla\mathcal{L}_{comb}(W')^T(W - W') + \frac{\beta}{2}\|W - W'\|^2$, we get the following

$$\mathcal{V}(\|\mathcal{L}_{comb}(W_H(e+1)) - \mathcal{L}_{comb}(W_H(e))\|_{e=h}^{E-1}) \leq \mathcal{V}(\|\nabla\mathcal{L}_{comb}(W_H(e))^T(W_H(e+1) - W_H(e))\|_{e=h}^{E}), \tag{9}$$

$$\leq -\eta^2\left(1 - \frac{\beta\eta}{2}\right)\mathcal{V}(\|\nabla\mathcal{L}_{comb}(W(e))\|_{e=h}^{E})^2. \tag{10}$$

From Equation (8), we have following expressions:

$$\Psi(e) = \mathcal{V}(\|\mathcal{L}_{comb}(W_H(e)) - \mathcal{L}_{comb}(W_E(e))\|_{e=h}^{E}), \Psi(e+1) = \mathcal{V}(\|\mathcal{L}_{comb}(W_H(e+1)) - \mathcal{L}_{comb}(W_E(e+1))\|_{e=h}^{E-1}),$$

Further, using the expression derived in Equation (10), we get

$$\Psi(e+1) \leq \Psi(e) - \eta^2\left(1 - \frac{\beta\eta}{2}\right)\mathcal{V}(\|\nabla\mathcal{L}_{comb}(W(e))\|_{e=h}^{E})^2. \tag{11}$$

Assuming independent $W_H(\cdot)$ and $W_E(\cdot)$, we have

$$\Psi(e) = \mathcal{V}(\|\mathcal{L}_{comb}(W_H(e)) - \mathcal{L}_{comb}(W_E(e))\|_{e=h}^{E}) \leq \mathcal{V}(\|\nabla\mathcal{L}_{comb}(W_H(e))^T(W_H(e) - W_E(e))\|_{e=h}^{E}),$$

$$\frac{\Psi(e)}{\mathcal{V}(\|W_H(e) - W_E(e)\|_{e=h}^{E})} \leq \mathcal{V}(\|\nabla\mathcal{L}_{comb}(W_H(e))\|_{e=h}^{E}).$$

Using value of $\mathcal{V}(\|\nabla\mathcal{L}_{comb}(W_H(e))\|_{e=h}^{E})$ in Equation (11), we get

$$\Psi(e+1) \leq \Psi(e) - \upsilon\eta^2\left(1 - \frac{\beta\eta}{2}\right)\Psi(e)^2. \tag{12}$$

Since, $\Psi(e+1)\Psi(e) > 0$, thus, it would not harm the inequality of Equation (12) upon division on both side.

$$\frac{1}{\Psi(e+1)} - \frac{1}{\Psi(e)} \geq \frac{\upsilon\eta(1 - \frac{\beta\eta}{2})\Psi(e)}{\Psi(e+1)} \geq \upsilon\eta^2\left(1 - \frac{\beta\eta}{2}\right). \tag{13}$$

Using ρ-*Lipschitz* property and Lemma 1, we have,

$$\frac{\Psi(e) - \Psi(e+1)}{\Psi(e+1)\Psi(e)} \geq \frac{\rho\eta\phi(1-\eta\beta)^{e+1}}{\Psi(e+1)\Psi(e)}. \tag{14}$$

Using Equation (8) and assuming $\epsilon > 0$, we get

$$\mathcal{V}(\|\mathcal{L}_{comb}(W_H(e+1)) - \mathcal{L}_{comb}(W_E(e+1))\|_{e=h}^{E-1}).\mathcal{V}(\|\mathcal{L}_{comb}(W_H(e)) - \mathcal{L}_{comb}(W_E(e))\|_{e=h}^{E}) \leq \epsilon^2. \tag{15}$$

$$\frac{1}{\Psi(e+1)\Psi(e)} \geq \frac{1}{\epsilon^2}. \tag{16}$$

Using Equation (16) and Equation (14), we obtain

$$\frac{\Psi(e) - \Psi(e+1)}{\Psi(e+1)\Psi(e)} \geq \frac{\rho\eta\phi(1-\eta\beta)^{e+1}}{\epsilon^2}. \tag{17}$$

Using Equation (13) and Equation (17) and taking limiting condition

$$\upsilon\eta^2\left(1 - \frac{\beta\eta}{2}\right) = \frac{\rho\eta\phi(1-\eta\beta)^{e+1}}{\epsilon^2}.$$

Since, $\epsilon > 0$, taking positive value of square root, we obtain the desired expression:

$$\epsilon = \sqrt{\frac{\rho\phi(1-\eta\beta)^{e+1}}{\upsilon\eta(1 - \frac{\beta\eta}{2})}} = \sqrt{\frac{2\rho\phi(1-\eta\beta)^{e+1}}{\upsilon\eta(2 - \beta\eta)}}. \tag{18}$$

□

4.2 Convergence Analysis of the Proposed Approach

From Algorithm 1 of the proposed approach, we identify that the convergence depends upon Procedure 1, Procedure 2, and Procedure 3. Procedure 1 is a one-time initialization process; thus, it does not play a significant role in the convergence. It implies convergence depends upon Procedure 2 and Procedure 3, i.e., knowledge-distillation based training and aperiodic global update in communication, respectively. We first discuss the convergence of the approach considering simple (non-KD) training, i.e., convergence depends upon Procedure 3. We use Assumption 2 and other assumptions given in [48], as

Assumption 3. *The combined loss function $\mathcal{L}_{comb}(\cdot)$ at any participant ($\in \mathcal{P}$) is μ-strongly convex for all WPM W and W'; thus, following inequality holds: $\mathcal{L}_{comb}(W) \geq \mathcal{L}_{comb}(W') + (W - W')^T \nabla \mathcal{L}_{comb}(W') + \frac{\mu}{2}\|W - W'\|$. Given $\mathcal{L}_{comb}(\cdot) = \mathcal{L}_a(\cdot)$ for sufficient resource participants.*

Assumption 4. *Let κ_i^t represents the uniformly and randomly selected sample from the local dataset of any participant $p_i \in \mathcal{P}$ at global iteration t, where $1 \leq t \leq R$. Let $\nabla \mathcal{L}_{comb}(\kappa_i^t, W_i^t)$ and $\nabla \mathcal{L}_{comb}(W_i^t)$ represent the gradient of loss function $\mathcal{L}_{comb}(\cdot)$ on κ_i^t samples and entire samples of the local dataset, respectively. The variance of the gradient on each participant p_i is bounded as $\mathbb{E}\|\nabla \mathcal{L}_{comb}(\kappa_i^t, W_i^t) - \nabla \mathcal{L}_{comb}(W_i^t)\| \leq \sigma_i^2$.*

Assumption 5. *The expected squared norm of loss function gradient is uniformly bounded as $\mathbb{E}\|\nabla \mathcal{L}_{comb}(\kappa_i^t, W_i^t)\|^2 \leq G^2$, where $1 \leq t \leq R$ and $1 \leq i \leq N$.*

Using Assumptions 2, 3, 4, and 5, we can obtain the expression for desired precision (q_o) in terms of local epoch count E, and global iteration R. The desired precision is defined as: $q_o = \mathbb{E}[\mathcal{L}_{comb}(W^R)] - \mathcal{L}^*_{comb}$, where W^R is the aggregated weight at final global epoch R and $\mathcal{L}^*_{comb}$ is the minimum and unknown value of $\mathcal{L}_{comb}$ at the server. Let $\mathcal{L}^*_{comb_i}$ is the minimum value of $\mathcal{L}_{comb_i}$ at participant $p_i \in \mathcal{P}$ then degree of non-i.i.d datasets among different participants: $a_3 = \mathcal{L}^*_{comb} - \sum_{i=1}^{N} \mathcal{L}^*_{comb_i}$. Let $\mathcal{G}$ denotes the total number of SGD operations on the participants. FedAvg converges at the rate of $O(1/\mathcal{G})$ [48], if following condition is satisfied:

$$\mathbb{E}[\mathcal{L}_{comb}(W^R)] - \mathcal{L}^*_{comb} \leq \frac{\beta/2\mu^2}{a_1 + \mathcal{G} - 1}\Big(4a_2 + \mu^2 a_1 \mathbb{E}\|W^1 - W^*\|\Big),$$
$$a_1 = \max\{8\beta/\mu, E\}, \quad a_2 = \sum_{i=1}^{N} b_i^2 \sigma_i^2 + 6\beta a_3 + 8(E-1)^2 G^2, \tag{19}$$

where b_i is the fraction contribution of participant p_i, $\forall p_i \in \mathcal{P}$.

Procedure 3 proposes the aperiodic global update, which reduces the number of rounds for convergence, $R \leq R_{FedAvg}$. It is because the proposed approach allows the colossal resource participants to perform more updates in the given communication interval. It implies the number of SGD operations in the approach, denoted as $\mathcal{G}'$, is less than FedAvg, $\mathcal{G}' < \mathcal{G}$. We obtain the convergence rate of the proposed approach as $O(1/\mathcal{G}')$ using Equation (19) and considered assumptions, which is greater than FedAvg, i.e., $O(1/\mathcal{G}') > O(1/\mathcal{G})$. From this discussion, we observe that the proposed approach follows a similar convergence pattern as FedAvg in non-KD based training.

The work presented in [37] determined the faster convergence of the model training using KD. They observed that the classifiers constructed using the outputs of another classifier as soft labels, instead of ground truth data, converge much faster and are reliable. Similarly, we observed that the KD-based training requires lesser epochs for convergence during experimental evaluations. Thus, from our observations and the previous work [37], we conclude that when we incorporate KD to train the models on the participants, the optimization steps are generally well-behaved in contrast with non-KD based training. Therefore, it reduces the number of local epochs for convergence and the number of communication rounds.

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output spaces, respectively, with $P(\mathbf{X})$ being the probability distribution. The hypothesis function of teacher is $H^* : \mathcal{X} \rightarrow \mathcal{Y}$ and that of student is $H : \mathcal{X} \rightarrow \mathcal{Y}$. The target of KD-based training is to minimize the empirical risk, defined as the the probability of student output different from teacher:

$$Risk(H) = P_{\mathbf{x}\sim P(\mathbf{X})}[H^*(x) \neq H(x)]. \tag{20}$$

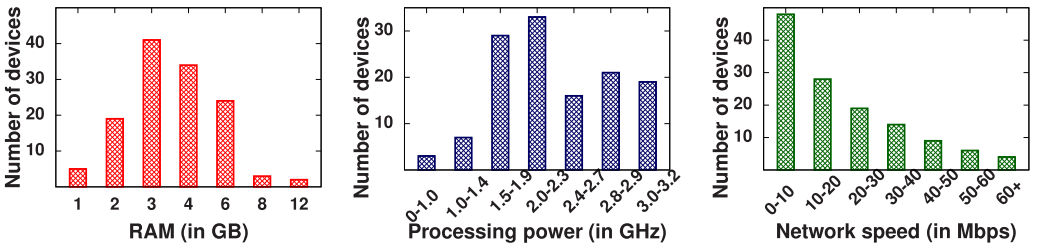
KD-based training of deep learning model with polynomial distribution at any participant $p_i \in \mathcal{P}$ is converged when the following condition is satisfied [37]: $\mathbb{E}_{x\sim P(\mathbf{X})^{n_i}}[Risk(\widehat{H(x)})] \leq C\frac{1+(\log n_i)^{\omega}}{n_i^{\omega}}$, where n_i number of instances in the training dataset d_i of participant $p_i \in \mathcal{P}$. $C > 0$ and $\omega \geq 0$ are the constants. $Risk(\widehat{H(x)})$ is the optimal or minimum value of the empirical risk. In ideal condition $Risk(\widehat{H(x)}) \rightarrow 0$; however, for general case it must be satisfied to ensure convergence of KD-based training. Moreover, we also introduce early-halting in KD-based training; thus, the convergence is achieved much earlier. From this discussion, we observe that the proposed approach follows a similar convergence pattern as FedAvg in non-KD based training. Additionally, the incorporation of KD for training local models at the participants accelerates the convergence.

Notably, we adopt different assumptions to determine the expression for the early halting epoch and analyze the convergence of the proposed approach. Such assumptions are suitable for convex loss functions like squared SVM, logistic regressions, and so on. However, the loss function in the deep neural networks is non-convex due to cascade linear and non-linear transforms. Thus, if we can establish the existence of solutions and convergence of the gradient-based method to the global minimizer then we can ensure the convergence of the proposed approach. It also ensures the derived early halting expression and convergence suit deep neural networks. The authors in [29] presented a general framework to study the non-convex landscapes and optimizers of over-parameterized system, i.e., deep neural networks, in terms of the **Polyak-Lojasiewicz** (**PL**) condition. They argued that PL holds on to most of the parameter space that is sufficient for the existence of the solution and convergence to a global minimizer. Therefore, using [29], we can conclude the proposed approach converges to an optimal solution, despite non-convex loss functions in the deep neural networks.

5 REAL-WORLD STUDY

The real-world study analyzes the resource characteristics of 128 smartphone users. The smartphones of the users have different specifications and brands; thus, the smartphones have heterogeneous resources, i.e., memory and processing power. RAM on these smartphones lie in the range of 1 GB to 12 GB, as illustrated in Figure 5(a). Similarly, the processing power or CPU clock speed lies in different ranges, as illustrated in Figure 5(b). We consider the processing power of a single core with the highest clock speed. The network bandwidth between users and the central server located at the institute possesses a high level of heterogeneity. Figure 5(c) illustrates the network bandwidth in the ranges. Besides, from Figure 5, we can observe that maximum volunteers have RAM of 3 GB (41 volunteers), CPU speed in the range 2.0–2.3 GHz (33 volunteers), and network speed $<$ 10 Mbps (48 volunteers).

Figure 6 illustrates the normalized values of resources of first 100 users. We used unit-based normalization, which brings all values into the range [0,1]. The devices are arranged in ascending order of their proceeding capacity then bandwidth and memory resources are arranged accordingly. Ellipses **e1**, **e2**, and **e3** in Figure 6 illustrate the normalized values of resources of devices, where **e1** and **e3** are the devices with least and colossal resources, respectively. This case study illustrates the high level of heterogeneity among the users.



(a) Memory. (b) Processing. (c) Bandwidth.

Fig. 5. Participants devices and central server in real-world study. (a) RAM , (b) Processing, and (c) Bandwidth.

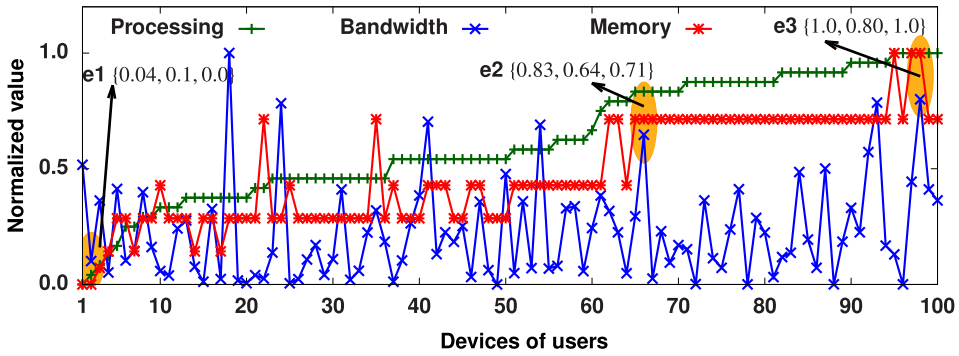


Fig. 6. Illustration of devices and networking resources (normalized values) of first 100 users.

5.1 Dataset Collection

We collected a dataset, called as the LMR dataset, to recognize six locomotion modes, including bicycle, bike, car, auto-rickshaw, bus, and train. We developed an android application that uses three onboard sensors of the smartphone (i.e., accelerometer, magnetometer, and gyroscope) for data collection. We selected 40 volunteers for collecting the data for a period of two months. The volunteers were in the age group of 18 to 55 years, where 20 volunteers were men and 20 were women. Each volunteer used an android based smartphone and installed the developed application. A volunteer selected a locomotion mode from the provided menu and recorded measurements for 60 seconds. We set the sampling rate of all the sensors to 100 Hz to collect 6, 000 data points per minute. In this experiment, the participants are directed to perform 200 repetitions for each mode. The dataset is collected for a period of two months; thus, distinct readings are recorded for each day. As a result, the created dataset consists of a total of 48, 000 (i.e., 40 participants $\times$6 modes $\times$200 repetitions) instances.

5.2 Challenges Observed During Study

The first challenge encountered during the study was the *inconsistency in the availability* of the devices. This inconsistency is a matter of the fact that the student volunteers are at distinct locations. Next, the limited mobility is another challenge of *data scarcity*. The volunteers collected only a few samples of locomotion modes data. In addition, some volunteers collected data for a sub-set of classes only. We also encountered the variation in the sensory data instances due to *different brands of smartphones*.

5.3 Criteria of Selecting Participants

We selected 100 participants for FL-based training from 128 users such that the observed challenges are minimized. For example, we prefer those users as participants that are *always available* and have *sufficient amount of data against all the classes*. The selection of the participants relies upon a high level of heterogeneity, i.e., the users having *different specifications of resources* are preferred over similar ones. Network bandwidth and residual energy also play a vital role in selecting the participants. The participants can use Wi-Fi or cellular connectivity. The participants using the *same Wi-Fi are less preferred over distinct*. The devices mainly were *plugged-in are preferred* over the ones that are operating over batteries during training.

6 PERFORMANCE EVALUATION

This section describes the existing datasets, baseline techniques, implementation details, and validation metrics used to evaluate the performance of the proposed approach.

6.1 Datasets and Baseline Techniques

This section describes the existing SHL [41], HAR [1], MNIST [19], CIFAR-10 [16] datasets used during experimental evaluations. SHL [41] dataset was collected from the onboard sensors of HUAWEI Mate 9 smartphones to recognize locomotion modes of the users. HAR [1] is a smartphone-based dataset used for recognizing six different activities. Furthermore, the broader applicability of MNIST and CIFAR-10 for validating different FL-based approaches has motivated its consideration in the experiment. MNIST is a handwritten digit recognition dataset comprising 60, 000 images of digits from $0 - 9$ in the training and 10, 000 images in the testing sub-datasets. CIFAR-10 dataset comprises 60, 000 images of 10 different classes, where each class has 6, 000 images.

We considered the existing techniques [5, 20, 23, 31] as baselines, noted as HetroFL [5], FedProx [23], FedAvg [31], and Hermes [20], to evaluate and compare the performance of the proposed approach. HetroFL [5] divided the heterogeneous participants into different clusters depending upon the various level of complexity. FedProx [23] handled the problem of heterogeneity by using a proximal term, which minimized the impact of local updates and restricted such updates closer to the server's model. FedAvg [31] is the benchmark and classical FL learning technique. Hermes [20] is a communication efficient framework to handle the critical bottleneck of communication cost and data heterogeneity.

6.2 Implementation Details

We implemented the proposed approach using Python with Tensorflow and Keras libraries. We selected the window size of 20 to reduce the length of dataset instances from 6000 to 300 during pre-processing in LMR and SHL datasets. We perform the random and disjoint partitioning of the datasets into training and testing in ratio 70 : 30 using *sklearn.model_selection.train_test_split*().

- *Data distribution among the participants:* We randomly split the datasets into 100 overlapping partitions using LMR, SHL, HAR, MNIST, and CIFAR-10 datasets. The overlapped partitioned datasets are allocated to the 100 participants. We divided both the testing and training datasets for all the participants.
- *Data variability:* To induce the variability in the collected LMR and existing SHL, HAR, MNIST, and CIFAR-10 datasets, we remove the training dataset assigned to each participant with the randomly re-partitioned training datasets. We perform this exercise at random communication rounds in increasing order. For example, during training for 100 communication rounds, the data re-partitioning is performed at rounds like 10, 20, 30, and so on.

— *Models:* Let $C(X)$ and $F(X)$ represent the convolutional and fully connected layers with X filters and neurons, respectively. For MNIST and HAR datasets, we used the configuration of the model as $C(128) - C(64) - C(128) - C(256) - C(512) - F(classes\_count)$. C is a one-dimensional convolutional layer (Conv1d) for sensory datasets, i.e., LMR, SHL, and HAR. For the image-based MNIST and CIFAR-10 dataset, C is a two-dimensional convolutional layer (Conv2d). The model for CIFAR-10 is ResNet-18 [10], whereas LMR and SHL used DeepZero model [32].

We observe an error range of −1.5 to +0.95 in the estimated results; thus, we repeat each experiment for 20 time and put the average values of the results in the entire experimental evaluations. The optimal a_thres for MNIST, HAR, LMR, and SHL datasets are 99%, 96%, 93.5%, and 93%, respectively. We determine the optimal a_thres using a survey of existing federated learning literature and rigorous experimental evaluations.

6.3 Validation Metrics

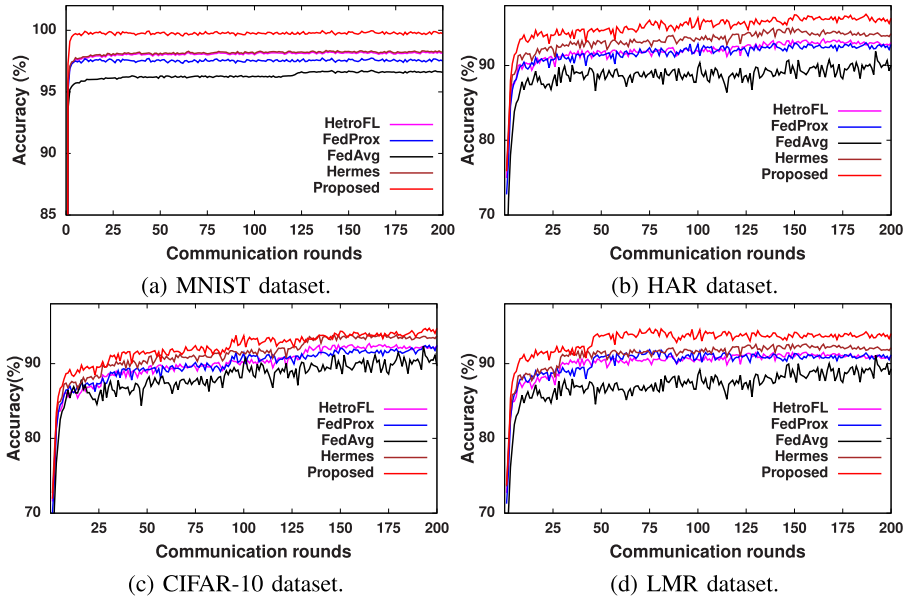
This work used the following standard classification metrics: F_1-score, accuracy, and leave-one-out test validation. Let a given dataset consists of a set of $\mathcal{A}$ classes, and $|\mathcal{A}|$ represents the number of classes. Let TP_i, TN_i FP_i, and FN_i are the true positive, true negative, false positive, and false negative counts of a class $i \in \mathcal{A}$, respectively. The *accuracy* and F_1*-score* metrics are computed as $\frac{1}{|\mathcal{A}|}\sum_{i=1}^{|\mathcal{A}|}\frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i}$ and $\frac{1}{|\mathcal{A}|}\sum_{i=1}^{|\mathcal{A}|}\frac{2\times TP_i}{2\times TP_i+FP_i+FN_i}$, respectively. We finally consider the *leave-one-out test* metric that trains the model for all class labels except for one randomly chosen label. However, the unseen class instance is also supplied for predicting the output during training.

6.4 Results

6.4.1 Impact of Datasets on the Convergence. This experiment aims to determine the impact of datasets on the convergence of the proposed approach and considered baselines, i.e., HetroFL [5], FedProx [23], FedAvg [31], and Hermes [20]. We considered 200 communication rounds with heterogeneous participants, where heterogeneity is introduced via randomly assigned local epochs. We set the proximal term in FedProx as 0.01 and compressed the hidden layers to determine the lightweight models. We induce data variability after 40 communication rounds.

Figure 7 illustrates the impact of datasets on existing (HetroFL, FedProx, FedAvg, Hermes) and proposed approaches, where the proposed approach achieved the best accuracy. It is because of using KD and appropriate handling of variability in the dataset. The learning curve presented in Figure 7 depicts two-step behavior and exhibits a classic shape. As the conventional learning mechanism, the learning starts with a steep increment in the performance until it reaches a monotonic plateaued value after 20^{th} communication rounds. Next, the accuracy grows with more communication rounds. The convergence of the approaches on the MNIST dataset is achieved at fewer communication rounds and marginal improvement afterward, as shown in Figure 7(a). It is due to the balanced and sufficient number of instances for all the classes in MNIST. FedAvg achieved slower convergence with minimal accuracy due to incompetence in handling heterogeneity among the participants and dataset variability. Hermes achieved comparable performance to the proposed approach due to a similar strategy for addressing the heterogeneity among the participants through personalization. However, it did not incorporate a mechanism to tackle variability and did not use KD; thus, it lags in the accuracy.

Observation: The first observation from the result is that the convergence curve of FL-based training follows a traditional pattern, where learning starts with steep steps, followed by monotonically plateaued performance. The next observation is the positive effect of using the technique to handle



(a) MNIST dataset. (b) HAR dataset.

(c) CIFAR-10 dataset. (d) LMR dataset.

Fig. 7. Illustration of impact of datasets on the convergence of proposed approach, HetroFL, FedProx, FedAvg, and Hermes.

heterogeneity among the participants and tackling dataset variability. The involvement of KD during training also plays a crucial role in improving performance.

6.4.2 Performance of the Approach. The objective of this experiment is to evaluate and compare the performance of the proposed approach with considered baselines. The experimental setup is similar to the previous result and the validation metrics as given in Section 6.3. We considered all datasets, where variability is induced at an interval of 40 communication rounds. We set the communication rounds to 200.

Table 1 illustrates the performance of different approaches on considered datasets, including HetroFL, FedProx, FedAvg, Hermes, and the proposed. FedAvg achieves the lowest performance among FL-based approaches due to the absence of a mechanism to handle heterogeneity among participants and dataset variability. The proposed approach outperforms the existing approaches in terms of accuracy and F_1-score on all the considered datasets. The approach involved a mechanism for handling heterogeneity among the participants simultaneously with effective management of dataset variability. Contrastively, HetroFL, FedProx, and Hermes proposed the approach to manage heterogeneity among the participants without considering the dataset variability. We also observed that the F_1-score is greater than the accuracy. The statistical heterogeneity among the participants' datasets makes false-negative and false-positive more crucial than true-negative and true-positive, which appeared in the form of higher F_1-score. The achieved performance on the MNIST dataset appears to be the highest among all the datasets due to the availability of sufficient data instances.

Observation: An interesting observation from the result is that effectively managing heterogeneity among the participants with the dataset variability is crucial for attaining high-order performance. Additionally, a large number of training data instances improves the performance.

6.4.3 Impact of Learning Rate. This experimental evaluation studies the impact of the learning rate on the performance of the proposed approach. We used MNIST, HAR, CIFAR-10, and LMR

Table 1. An Illustration of Global Performance Achieved by the Different Approaches, i.e., HetroFL, FedProx, FedAvg, Hermes, and Proposed, on Considered Datasets (MNIST, HAR, CIFAR-10, LMR, and SHL)

Datasets	**MNIST**		**HAR**		**CIFAR-10**		**LMR**		**SHL**	
Approach	**Acc. (%)**	**F_1 (%)**	**Acc. (%)**	**F_1 (%)**	**Acc. (%)**	**F_1 (%)**	**Acc. (%)**	**F_1 (%)**	**Acc. (%)**	**F_1 (%)**
HetroFL	98.12	98.53	93.27	93.79	91.79	92.23	90.81	91.73	89.71	90.45
FedProx	97.90	98.31	92.90	93.23	91.41	91.87	90.21	91.10	89.23	91.03
FedAvg	97.17	97.44	91.11	91.59	90.73	91.30	89.47	90.13	88.66	89.35
Hermes	98.23	98.71	94.19	94.62	92.31	92.84	90.07	91.14	89.83	90.67
Proposed	**99.14**	**99.37**	**96.97**	**97.41**	**94.71**	**95.19**	**93.83**	**94.21**	**93.21**	**93.91**

Acc.= Accuracy and F_1 = F_1-score.

datasets and restricted the communication round to 10. We bound the round as the approach converges for all the learning rates at the higher communication rounds. The parameters are the same as discussed in Section 6.4.1 and data variability induced after 40 communication rounds.

Figure 8 illustrates the impact of different learning rates on the performance of the proposed approach using considered datasets (MNIST, HAR, CIFAR-10, and LMR). The results depicted the efficacy of the approach on smaller values of the learning rate (e.g., 0.001). The model converged to sub-optimal weights, or the training became unstable for a higher learning rate. The approach converges faster for the MNIST dataset; thus, we achieved accuracy beyond 90% for all the datasets at different learning rates. However, we obtained the lowest accuracy for the learning rate = 0.010 due to faster convergence. The approach achieved accuracy following a linear curve for MNIST, whereas other datasets have shown a plateaued behavior due to slow convergence.

Observation: We observed from the result that the learning rate plays a crucial role in ensuring the performance of the approaches. If we can train the model for a large number of communication rounds, a small learning rate is beneficial. Contrarily, if limited communication rounds are available for training the model, it is helpful to use a higher learning rate. This work considered a smaller value of learning rate (i.e., 0.001*) as we are using* 200 *communication rounds.*

6.4.4 Impact of Dataset Variability. This experiment aims to highlight and compare the impact of dataset variability on the performance of Hermes and proposed approaches. We introduced dataset variability after $10 - 60$ communication rounds to study its impact. Let $\mathbf{V_1}, \ldots, \mathbf{V_6}$ denote the level of dataset variability introduced at communication rounds $10, \ldots, 60$, respectively.

Table 2 illustrates the impact of dataset variability on the achieved accuracy of the proposed approach. We diversify the dataset variability by shuffling the participants' dataset after communication rounds $10, 20, 30, 40, 50$, and 60, using the technique discussed in Section 6.2. The level of dataset variability is highest when the dataset of the participants are shuffled after 10 communication rounds ($\mathbf{V_1}$) and lowest for shuffling after 60 ($\mathbf{V_6}$). The result demonstrated the rapid increment in the accuracy upon decreasing the level of data variability. It is because when the model is trained for many communication rounds on a specific dataset, it learnt more refined features. The impact of lower dataset variability is minimal on the performance as the model gets sufficient time to learn well-refined features at such variability. The result in Table 2 demonstrated that the approach outperforms Hermes. It is because the proposed approach involved the pruned layers whenever dataset variability is observed during training. We can make similar observations for F_1-scores, as shown in Table 2. The achieved inference accuracy and F_1-score are lower for SHL than LMR due to the imbalanced and more number of classes in SHL.

Observation: We observed from the result that the dataset variability plays an adversarial role in achieving the desired performance. Therefore, effective handling of the dataset variability is requisite.

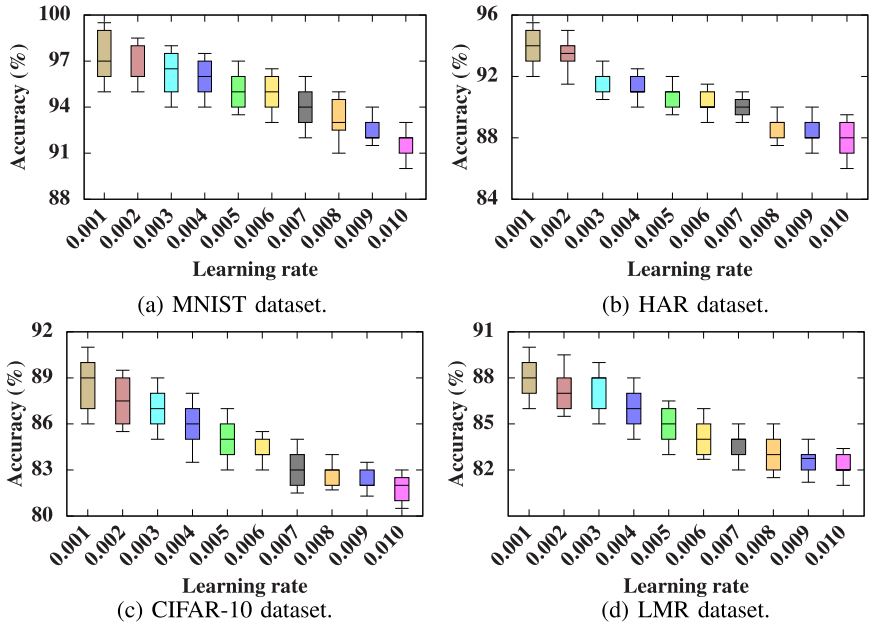


Fig. 8. Impact of learning rate on the performance of the approach on different datasets at communication rounds = **10**.

Table 2. Impact of Dataset (LMR and SHL) Variability on Accuracy and F_1-score on Different Approaches

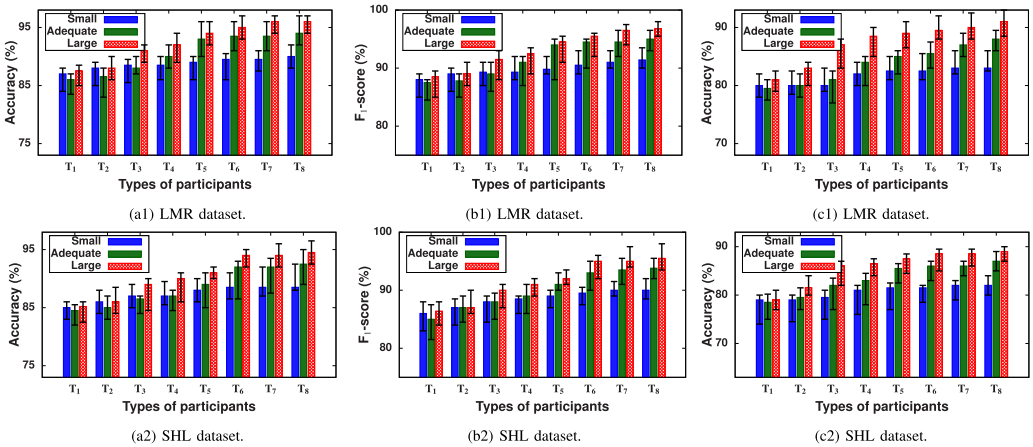
Metrics		Accuracy (%)		F_1-score (%)	
Approach → Variability ↓		Hermes	Proposed	Hermes	Proposed
LMR dataset	V_1	86.45	92.32	87.79	92.81
	V_2	88.71	92.68	90.06	93.21
	V_3	89.23	93.17	90.72	93.79
	V_4	**90.07**	**93.83**	**91.14**	**94.21**
	V_5	90.23	93.97	91.29	94.87
	V_6	91.07	94.07	91.85	95.17

Metrics		Accuracy (%)		F_1-score (%)	
Approach → Variability ↓		Hermes	Proposed	Hermes	Proposed
SHL dataset	V_1	85.71	91.42	86.93	92.17
	V_2	87.73	92.03	89.19	92.71
	V_3	88.76	92.43	89.81	93.29
	V_4	**89.84**	**93.21**	**90.67**	**93.95**
	V_5	89.91	93.41	91.21	94.31
	V_6	90.46	93.82	91.73	94.89

Additionally, the proposed approach maintains accuracy ≈ 90% for SHL and LMR datasets even at high-order variability ($\mathbf{V_1}$). Moreover, the achieved accuracy > 93% for the datasets at the medium level of the dataset variability ($\mathbf{V_4}$). Therefore, the experiments presented in this article used such a medium level of variability.

6.4.5 Impact of Model Sizes. This experiment studies the impact of different size models on the performance of heterogeneous participants. We divided the 100 available participants into eight different types, denoted as $\mathbf{T_1} - \mathbf{T_8}$. The processing capacity of participant type $\mathbf{T_1}$ is the lowest and highest for $\mathbf{T_8}$. We obtained three types of models from the DeepZero models [32], i.e., small model (with three layers of CNN), adequate model (with all five layers of CNN), and large model (with seven layers of CNN).

Figure 9(a1) illustrates the accuracy of different considered models on the LMR dataset. It shows the large-size model in the proposed approach gives higher accuracy than the small and adequate size models for all categories of devices. It is because the small model is a lightweight for all the devices that inefficiently utilize the available resources. Similarly, the adequate size model did not exploit the colossal resources of $\mathbf{T_6} - \mathbf{T_8}$. However, using the large-size model on



(a1) LMR dataset. (b1) LMR dataset. (c1) LMR dataset.

(a2) SHL dataset. (b2) SHL dataset. (c2) SHL dataset.

(a) Accuracy. (b) F_1-score. (c) Accuracy using leave-one-out test.

Fig. 9. Illustration of the accuracy, F_1-score, and leave-one-out test of the proposed approach on different model sizes.

all the participant devices requires extra communication costs and memory, which may not be available. The approach resizes the model based on the limited resources of $\mathbf{T_1}$ and directly uses the adequate-size model for $\mathbf{T_5}$. The higher category of devices have more resources and therefore achieves higher accuracy on large-size model. The results in Figure 9(a2) give less accuracy than Figure 9(a1) for all the models in the proposed approach and categories of participant devices because of the SHL dataset that has imbalanced classes. Figure 9(b) illustrates the F_1-score for validating the preference of different models in the proposed approach. The class distribution in the collected LMR dataset is similar; therefore, the true positives and true negatives are important and false negatives and false positives are not crucial. The results in Figure 9(b1) illustrate the similar behavior as shown in Figure 9(a1). However, it is not true for Figure 9(b2), where false negatives and false positives are crucial and give less accuracy. Finally, we performed experiments using a leave-one-out validation metric where instances of one class are not considered during training. Figure 9(c) illustrates the results of the leave-one-out validation metric where instances of one class are not considered during training. The results illustrate that the difference of the accuracy in Figure 9(a1) and Figure 9(c1) is less for colossal resources devices $\mathbf{T_8}$ than insufficient resource devices $\mathbf{T_1}$. It is because the colossal resource devices train more successfully than insufficient ones.

6.4.6 Impact of Early Halting. The objective of this experiment is to illustrate the impact of the early halting on the performance of participants with colossal and insufficient resources on LMR dataset. We also demonstrate the reduction in **floating-point operations** (**FLOPs**) using the early halting.

We considered the participants of category $\mathbf{T_7}$ to describe the impact of the early halting mechanism on colossal resources participants. Table 3 illustrates that significant improvement in accuracy and F_1-score of M_o is observed up to 20 epochs of simultaneous training with untrained large-size M_i (model on participant p_i) under the guidance of trained M_i. After that, we observe a minor improvement in the accuracy and F_1 score of M_o. However, the required resources for further training increases sharply. It indicates that the training of untrained M_i with M_o do not improve performance despite consuming enormous resources (FLOPs). Therefore, we can halt the training of untrained M_i at 20 epochs. Next, we considered participant type $\mathbf{T_3}$ to study the role of

Table 3. Impact of Early Halting on $\mathbf{T_7}$ Participants

Epochs	10	15	**20**	25	30	35	40
Accuracy (%)	86.47	90.02	**94.23**	94.41	94.76	94.81	95.23
F_1-score (%)	87.92	92.31	**95.19**	95.03	95.17	95.29	96.17
FLOPs per round ($\times 10^{13}$)	2.53	2.77	**2.93**	3.09	3.29	3.56	3.73

Table 4. Impact of Early Halting on $\mathbf{T_3}$ Participants

Epochs	10	15	20	**25**	30	35	40
Accuracy (%)	74.39	79.89	86.43	**90.72**	91.22	91.53	91.79
F_1-score (%)	76.07	81.24	87.19	**92.19**	92.37	91.83	93.11
FLOPs per rounds ($\times 10^{13}$)	1.17	1.23	1.41	**1.54**	1.71	1.82	1.91

Table 5. Impact of AGU and SGU on the Communication Rounds to Achieved Given Accuracy (x%)

Technique	**Communication rounds to reach x% accuracy on the dataset**				
	MNIST (98%)	**HAR (96%)**	**CIFAR-10 (92%)**	**LMR (91%)**	**SHL (91%)**
AGU ($\mathcal{T} = 0.8 \times T$)	8	41	70	57	79
AGU ($\mathcal{T} = 0.5 \times T$)	**6**	**37**	**67**	**53**	**73**
AGU ($\mathcal{T} = 0.3 \times T$)	9	39	69	55	77
SGU	11	54	84	66	91

T is time required for a communication round and $\mathcal{T}$ is the time threshold.

early halting on insufficient resources participants. Table 4 depicts the performance improvement up to 25 epochs during training of M_o under guidance of lightweight M_i is rapid. After that, the performance improvement is low, but resource consumption is high. Thus, we halt the training of M_o for type $\mathbf{T_3}$ participants at 25 epochs.

6.4.7 Ablation Studies: Aperiodic Global Update, Knowledge Distillation, and Pruning. This experiment aims to determine the impact of aperiodic global updates, knowledge distillation, and pruning on the communication rounds for convergence and performance of the proposed approach. Similar to previous experiments, we induce dataset variability after 40 communication rounds. During the experiment, we obtained time for one communication round T as 14, 27, 138, 117, and 128 minutes for MNIST, HAR, CIFAR-10, LMR, and SHL, respectively. We use the time threshold $\mathcal{T}$ as a fractional multiple of T, i.e., $0.3 \times T$, $0.5 \times T$, and $0.8 \times T$ and set dropout to 0.25, 0.50, and 0.75 for pruning personalized layers.

Table 5 illustrates the impact of **Aperiodic Global Updates** (**AGU**) and **Synchronous Global Updates** (**SGU**) on the communication rounds to reach the x% accuracy threshold. We used three variants of AGU, where we set $\mathcal{T} = 0.8 \times T, 0.5 \times T$, and $0.3 \times T$. We observed that the AGU requires fewer communication rounds to reach the accuracy threshold on any considered datasets. It is because AGU offers some faster participant devices to perform multiple local training in between a communication round. We also observed the lowest communication round is possible when $\mathcal{T} = 0.5 \times T$. It is because at a lower value of $\mathcal{T}$ only a few participants are involved in the intermediate aggregations, which results in inferior quality of aggregated WPM. On the other hand, the increment in $\mathcal{T}$ allowed more devices to participate in intermediate aggregation, which generated precise aggregated WPM. However, beyond the ratio of 0.5, the

Table 6. Impact of using KD on **Communication Rounds** (**CR**) for Convergence and **Accuracy** (**Acc**)

Datasets →		**MNIST**	**HAR**	**CIFAR-10**	**LMR**	**SHL**
CR	*Without KD*	17	72	129	117	137
	With KD	10	48	74	65	92
Acc. (in %)	*Without KD*	97.22	91.27	90.43	89.41	88.22
	With KD	99.14	96.97	94.71	93.83	93.21

Table 7. Impact of Repersonalization and Pruning (Dropout) on Accuracy

Accuracy (in %)	**Dropout (Pruning)**	**Repersonalization**	**Dataset**				
			MNIST	**HAR**	**CIFAR-10**	**LMR**	**SHL**
	0.25	*Without*	99.14	94.73	87.82	86.57	84.63
		With	99.14	96.97	94.71	93.83	93.21
	0.50	*Without*	92.37	84.41	77.35	72.23	64.51
		With	92.37	88.73	83.23	78.29	72.38
	0.75	*Without*	80.31	61.79	54.64	52.67	39.78
		With	80.31	70.26	62.24	56.39	44.31

number of participants involved in the intermediate aggregation increases, but the aggregation is performed only once between two global iterations. The participant devices with colossal resources get the updated WPMs quickly but wait for the next global iteration. Therefore, in this work, we select $\mathcal{T} = 0.5 \times T$ in all the experiments. Similarly, Table 6 depicts the impact of knowledge distillation on communication rounds and the accuracy of the proposed approach. We observed that knowledge distillation from the large-size model to the lightweight improves the performance and reduces the communication rounds.

Further, Table 7 illustrates the impact of pruning and repersonalization on different datasets. To perform personalized pruning on the participants, we adopted the dropout mechanism and set three different values, i.e., 0.25, 0.50, and 0.75. As expected, the highest accuracy is obtained at the dropout of 0.25. We used the dropout of 0.25 in the entire presented experiment. Moreover, we observed that repersonalization improves performance on HAR, CIFAR-10, LMR, and SHL datasets. It is because we induce data variability after 40 communication rounds, which hampers performance if not handled appropriately. We do not observe the impact of the repersonalization on the MNIST dataset at its converses before the 20 communication rounds, where data variability is induced.

7 CONCLUSION

This article proposed a model personalization-based federated learning approach for heterogeneous devices and networking resources. The approach also handled the variability in the local datasets of the participants. Unlike the existing work, the proposed approach trained the model on participant devices with resource heterogeneity and dataset variability. The central server initiated the approach and constructed a base model. Next, the proposed approach simultaneously learnt the personalized model and handled the dataset variability. We further proposed an early halting approach for faster training of the resized model, which fits on the insufficient and colossal resource devices. We finally proposed an aperiodic global update approach for aggregation of WPM at the server. We also did a real-world study to evaluate feasibility and performance of the proposed work.

We found the following conclusions from this work: federated learning work successfully only when the devices and networking resources are considered simultaneously with dataset variability; if the dataset is imbalanced then the selection of the base model must be favorable for insufficient resource devices. The proposed approach considers the heterogeneity in three resources (memory, processing power, and bandwidth). We will consider the heterogeneity in other parameters (such as data sources and sampling rate) in the future. We also plan to consider different types of challenges in datasets, such as imbalanced, noisy, and unseen classes.

REFERENCES

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra Perez, and Jorge Luis Reyes Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the ESANN*. 437–442.

[2] Sébastien Bubeck et al. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning* 8, 3-4 (2015), 231–357.

[3] Zheng Chai, Ahsan Ali, Syed Zawad, Stacey Truex, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Heiko Ludwig, Feng Yan, and Yue Cheng. 2020. TiFL: A tier-based federated learning system. In *Proceedings of the HPDC*. 125–136.

[4] Zheng Chai, Hannan Fayyaz, Zeshan Fayyaz, Ali Anwar, Yi Zhou, Nathalie Baracaldo, Heiko Ludwig, and Yue Cheng. 2019. Towards taming the resource and data heterogeneity in federated learning. In *Proceedings of the 2019 USENIX Conference on Operational Machine Learning (OpML 19)*. 19–21.

[5] Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. In *9th International Conference on Learning Representations*, ICLR. 1–24.

[6] Sannara EK, François PORTET, Philippe LALANDA, and German VEGA. 2021. A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison. In *Proceedings of the PerCom*. 1–10.

[7] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. *Personalized Federated Learning: A Meta-Learning Approach*. arXiv:2002.07948 [cs.LG].

[8] Yiwen Guo, Anbang Yao, and Yurong Chen. 2016. Dynamic network surgery for efficient DNNs. In *Proceedings of the NIPS*. 1–9.

[9] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group knowledge transfer: Federated learning of large cnns at the edge. *Proceedings of the NIPS*, Vol. 33. 14068–14080.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the CVPR*. 770–778.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. *Distilling the Knowledge in a Neural Network*. arXiv:1503.02531 [stat.ML].

[12] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. 2021. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Proceedings of the NIPS*, Vol. 34. 12876–12889.

[13] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial intelligence*. 7865–7873.

[14] Zhongming Ji, Li Chen, Nan Zhao, Yunfei Chen, Guo Wei, and F. Richard Yu. 2021. Computation offloading for edge-assisted federated learning. *IEEE Transactions on Vehicular Technology* 70, 9 (2021), 9330–9344.

[15] Woojin Kang, In-Taek Jung, DaeHo Lee, and Jin-Hyuk Hong. 2021. Styling words: A simple and natural way to increase variability in training data collection for gesture recognition. In *Proceedings of the CHI*. 1–12.

[16] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning Multiple Layers of Features from Tiny Images*. Technical Report 0. University of Toronto, Toronto, ON. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[17] Ramakant Kumar, Rahul Mishra, and Hari Prabhat Gupta. 2023. A federated learning approach with imperfect labels in LoRa-based transportation systems. *IEEE Transactions on Intelligent Transportation Systems* 24, 11 (2023), 1–9.

[18] Fan Lai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. 2021. Oort: Efficient federated learning via guided participant selection. In *Proceedings of the USENIX OSDI*. 19–35.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.

[20] Ang Li, Jingwei Sun, Pengcheng Li, Yu Pu, Hai Li, and Yiran Chen. 2021. Hermes: An efficient federated learning framework for heterogeneous mobile clients. In *Proceedings of the ACM Mobicom*. 420–437.

[21] Daliang Li and Junpu Wang. 2019. *FedMD: Heterogenous Federated Learning via Model Distillation*. arXiv:1910.03581 [cs.LG].

[22] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.
[23] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of the MLSys* 2 (2020), 429–450.
[24] Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, Zibin Zheng, and Qiang Yan. 2021. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network* 35, 1 (2021), 234–241.
[25] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B. Allen, Randy P. Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. *Think Locally, Act Globally: Federated Learning with Local and Global Representations.* arXiv:2001.01523 [cs.LG].
[26] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. 2020. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys and Tutorials* 22, 3 (2020), 2031–2063.
[27] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. In *Proceedings of the NeurIPS*. 2351–2363.
[28] Bingyan Liu, Yifeng Cai, Ziqi Zhang, Yuanchun Li, Leye Wang, Ding Li, Yao Guo, and Xiangqun Chen. 2022. DistFL: Distribution-aware federated learning for mobile scenarios. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2022), 1–26.
[29] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. 2022. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis* 59 (2022), 85–116.
[30] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. 2018. Hierarchical representations for efficient architecture search. In *Proceedings of the ICLR*. 1–13.
[31] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the AISTATS*. 1273–1282.
[32] Rahul Mishra, Ashish Gupta, Hari Prabhat Gupta, and Tanima Dutta. 2022. A sensors based deep learning model for unseen locomotion mode identification using multiple semantic matrices. *IEEE Transactions on Mobile Computing* 21, 3 (2022), 799–810.
[33] Rahul Mishra, Hari Prabhat Gupta, and Tanima Dutta. 2020. Teacher, trainee, and student based knowledge distillation technique for monitoring indoor activities: Poster abstract. In *Proceedings of the SenSys*. 729–730.
[34] Rahul Mishra, Hari Prabhat Gupta, and Tanima Dutta. 2022. Noise-resilient federated learning: Suppressing noisy labels in the local datasets of participants. In *Proceedings of the IEEE INFOCOM WKSHPS*. 1–2.
[35] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Guoliang Xing, and Jianwei Huang. 2022. ClusterFL: A clustering-based federated learning system for human activity recognition. *ACM Transactions on Sensor Networks* 19, 1 (2022), 1–32.
[36] Jihong Park, Sumudu Samarakoon, Mehdi Bennis, and Mérouane Debbah. 2019. Wireless network intelligence at the edge. *Proceedings of the IEEE* 107, 11 (2019), 2204–2239.
[37] Mary Phuong and Christoph Lampert. 2019. Towards understanding knowledge distillation. In *Proc. ICML*. 5142–5151.
[38] Hanchi Ren, Jingjing Deng, Xianghua Xie, Xiaoke Ma, and Yichuan Wang. 2023. *FedBoosting: Federated Learning with Gradient Protected Boosting for Text Recognition.* arXiv:2007.07296 [cs.CV].
[39] Shihao Shen, Yiwen Han, Xiaofei Wang, and Yan Wang. 2019. Computation offloading with multiple agents in edge-computing-supported IoT. *ACM Transactions on Sensor Networks* 16, 1 (2019), 1–27.
[40] Yimin Shi, Haihan Duan, Lei Yang, and Wei Cai. 2022. An energy-efficient and privacy-aware decomposition framework for edge-assisted federated learning. *ACM Transactions on Sensor Networks* 18, 4 (2022), 1–24.
[41] SHL Challenge. 2022. Retrieved from http://www.shl-dataset.org/activity-recognition-challenge/. Accessed 10 September 2022.
[42] Chitranjan Singh, Rahul Mishra, Hari Prabhat Gupta, and Garvit Banga. 2022. A federated learning-based patient monitoring system in internet of medical things. *IEEE Transactions on Computational Social Systems* 10, 4 (2022), 1–7.
[43] Guangcong Wang, Xiaohua Xie, Jianhuang Lai, and Jiaxuan Zhuo. 2017. Deep growing learning. In *Proceedings of the ICCV*. 2812–2820.
[44] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221.
[45] Chuhan Wu, Fangzhao Wu, Ruixuan Liu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2021. Communication-efficient federated learning via knowledge distillation. *Nature Communication* 13, 2032 (2022).
[46] Qiong Wu, Xu Chen, Tao Ouyang, Zhi Zhou, Xiaoxi Zhang, Shusen Yang, and Junshan Zhang. 2023. HiFlash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association. *IEEE Transactions on Parallel and Distributed Systems* 34, 5 (2023), 1560–1579. DOI: https://doi.org/10.1109/TPDS.2023.3238049

[47] Wenyan Wu and Shuo Yang. 2017. Leveraging intra and inter-dataset variations for robust face alignment. In *Proceedings of the CVPR*. 150–159.
[48] Hao Yu and Rong Jin. 2019. On the computation and communication complexity of parallel SGD with dynamic batch sizes for stochastic non-convex optimization. In *Proceedings of the ICML*. 7174–7183.
[49] Rong Yu and Peichun Li. 2021. Toward resource-efficient federated learning in mobile edge computing. *IEEE Network* 35, 1 (2021), 148–155.
[50] Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the CVPR*. 3903–3911.
[51] Yufeng Zhan, Peng Li, and Song Guo. 2020. Experience-driven computational resource allocation of federated learning by deep reinforcement learning. In *Proceedings of the IPDPS*. 234–243.
[52] H. Zhao, X. Sun, J. Dong, C. Chen, and Z. Dong. 2020. Highlight every step: Knowledge distillation via collaborative teaching. *IEEE Transactions on Cybernetics* 52, 4 (2020), 1–12. DOI: 10.1109/TCYB.2020.3007506
[53] Shuxin Zheng, Qi Meng, Taifeng Wang, Wei Chen, Nenghai Yu, Zhi-Ming Ma, and Tie-Yan Liu. 2017. Asynchronous stochastic gradient descent with delay compensation. In *Proceedings of the ICML*. 4120–4129.
[54] G. Zhou, Y. Fan, R. Cui, W. Bian, X. Zhu, and K. Gai. 2018. Rocket launching: A universal and efficient framework for training well-performing light net. In *Proceedings of the AAAI*. 1–8.
[55] Yuhao Zhou, Qing Ye, and Jiancheng Lv. 2022. Communication-efficient federated learning with compensated overlap-FedAvg. *IEEE Transactions on Parallel and Distributed Systems* 33, 1 (2022), 192–205.
[56] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the ICML*. 1–12.
[57] Hayreddin Çeker and Shambhu Upadhyaya. 2016. Adaptive techniques for intra-user variability in keystroke dynamics. In *Proceedings of the BTAS*. 1–6.

Received 22 October 2022; revised 29 July 2023; accepted 7 October 2023