Self-Organizing Democratized Learning

Toward Large-Scale Distributed Learning Systems

Nguyen, Minh N.H.; Pandey, Shashi Raj; Dang, Tri Nguyen; Huh, Eui-Nam; Tran, Nguyen H.; Saad, Walid; Hong, Choong Seon

Published in:
IEEE Transactions on Neural Networks and Learning Systems

DOI (link to publication from Publisher):
10.1109/TNNLS.2022.3170872

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Self-Organizing Democratized Learning: Toward Large-Scale Distributed Learning Systems

Minh N. H. Nguyen, *Member, IEEE*, Shashi Raj Pandey, *Member, IEEE*, Tri Nguyen Dang, Eui-Nam Huh, *Member, IEEE*, Nguyen H. Tran, *Senior Member, IEEE*, Walid Saad, *Fellow, IEEE*, and Choong Seon Hong, *Senior Member, IEEE*

***Abstract*— Emerging cross-device artificial intelligence (AI) applications require a transition from conventional centralized learning systems toward large-scale distributed AI systems that can collaboratively perform complex learning tasks. In this regard, democratized learning (Dem-AI) lays out a holistic philosophy with underlying principles for building large-scale distributed and democratized machine learning systems. The outlined principles are meant to study a generalization in distributed learning systems that go beyond existing mechanisms such as federated learning (FL). Moreover, such learning systems rely on hierarchical self-organization of well-connected distributed learning agents who have limited and highly personalized data and can evolve and regulate themselves based on the underlying duality of specialized and generalized processes. Inspired by Dem-AI philosophy, a novel distributed learning approach is proposed in this article. The approach consists of a self-organizing hierarchical structuring mechanism based on agglomerative clustering, hierarchical generalization, and corresponding learning mechanism. Subsequently, hierarchical generalized learning problems in recursive forms are formulated and shown to be approximately solved using the solutions of distributed personalized learning problems and hierarchical update mechanisms. To that end, a distributed learning algorithm, namely DemLearn, is proposed. Extensive experiments on benchmark MNIST, Fashion-MNIST, FE-MNIST, and CIFAR-10 datasets show that the proposed algorithm demonstrates better results in the generalization performance of learning models in agents compared to the conventional FL algorithms. The detailed analysis provides useful observations to further handle both the generalization and specialization performance of the learning models in Dem-AI systems.**

***Index Terms*— Democratized learning, distributed artificial intelligences (AIs), hierarchical learning, self-organization.**

Manuscript received 21 February 2021; revised 22 December 2021; accepted 21 April 2022. Date of publication 10 May 2022; date of current version 1 December 2023. This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2020R1A4A1018607, in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) Grant funded by the Korean Government [Ministry of Science and ICT (MSIT)] (Evolvable Deep Learning Model Generation Platform for Edge Computing) under Grant 2019-0-01287, and in part by the IITP Grant funded by the Korean Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068. *(Corresponding authors: Choong Seon Hong; Minh N. H. Nguyen; Eui-Nam Huh.)*

Minh N. H. Nguyen is with the Faculty of Computer Engineering and Electronics, The University of Danang–Vietnam-Korea University of Information and Communication Technology, Da Nang 550000, Vietnam, and also with the Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea (e-mail: nhnminh@vku.udn.vn).

Shashi Raj Pandey is with the Connectivity Section, Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark, and also with the Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea (e-mail: srp@es.aau.dk).

Tri Nguyen Dang, Eui-Nam Huh, and Choong Seon Hong are with the Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea (e-mail: trind@khu.ac.kr; johnhuh@khu.ac.kr; cshong@khu.ac.kr).

Nguyen H. Tran is with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: nguyen.tran@sydney.edu.au).

Walid Saad is with the Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA 22203 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea (e-mail: walids@vt.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3170872.

Digital Object Identifier 10.1109/TNNLS.2022.3170872

I. Introduction

NOWADAYS, artificial intelligence (AI) has grown to be successful in solving complex real-life problems such as decision support in healthcare systems, advanced control in automation systems, robotics, and telecommunications. Numerous existing mobile applications incorporate AI modules that leverage users' data for personalized services such as Gboard mobile keyboard on Android, the QuickType keyboard, and the vocal classifier for Siri on iOS [1]. By exploiting the unique features and personalized characteristics of users, these applications not only improve the personal experience of the users, but also helps to better control their devices. Moreover, the rising concern of data privacy in existing machine learning frameworks fueled a growing interest in developing distributed machine-learning paradigms such as federated learning (FL) frameworks [1]–[11]. FL was first introduced in [2], where the learning agents coordinate via a central server to train a global learning model in a distributed manner. These agents receive the global learning model from the central server and perform local learning based on their available datasets. Then, they send back the updated learning models to the server for updating the global model via an aggregation operation without revealing the private training data to the others.

In practice, the private dataset collected at each agent is unbalanced, highly personalized for some applications such as handwriting and voice recognition, and exhibits non-independent and non-identically distributed (i.i.d.) characteristics. Therefore, the iterative process of updating the global model improves the generalization of the model, but also hurts the personalized performance of the agents [1]. Hence, existing FL algorithms cannot efficiently handle the underlying

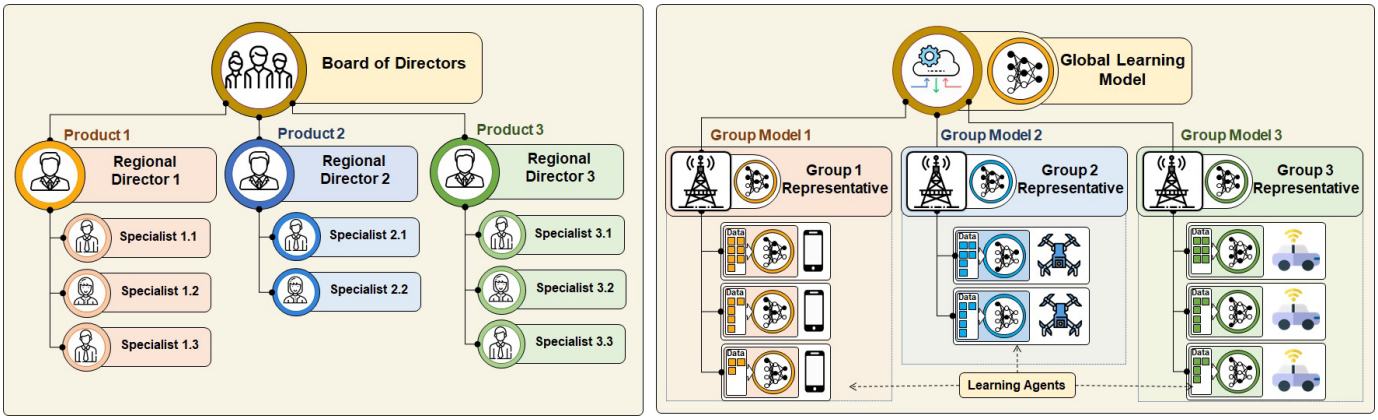


Fig. 1. Analogy of a hierarchical distributed learning system.

cohesive relation between generalization and personalization (or specialization) abilities of the trained learning model [1]. To the best of our knowledge, the work in [9] was the first attempt to study and improve the personalized performance of FL using a personalized federated averaging (Per-FedAvg) algorithm based on a meta-learning framework (MLF). Furthermore, in a recent work [10], the authors propose an adaptive personalized FL framework where a mixture of the local and global models was adopted to reduce the generalization error. However, similar to [9], the cohesive relation between generalization and personalization was not adequately analyzed. Recently, [12] proposed a pFedMe algorithm by studying a bi-level learning optimization problem such as global problem and personalized problems.

To better analyze the personalized and generalized learning performance for learning models in the FL framework, the Dem-AI philosophy, discussed in [13], introduces a holistic approach and general guidelines to develop distributed and democratized learning systems. The approach refers to observations about the generalization and specialization capabilities of biological intelligence, the hierarchical structure of society, and swarm intelligence in large-scale distributed learning systems. Fig. 1 illustrates the analogy of the Dem-AI system and the hierarchical structure in an organization. The specialists from different domain knowledge are grouped into teams to perform common products or targets. These groups in an organization need to collaborate toward the common goals under the supervision of a board of directors. Similarly, learning agents in different groups perform collaborative learning for group models. The outputs of these groups in a Dem-AI system are the specialized learning models that are created by group members. In this article, inspired by Dem-AI guidelines, we develop a novel distributed learning framework that can directly extend the conventional FL scheme for collectively solving a common learning task at learning agents. Different from existing FL algorithms for building a single generalized model (a.k.a global model), we maintain self-organizing hierarchical group models. Accordingly, we adopt the agglomerative hierarchical clustering [14] and periodically update the hierarchical structure based on the similarity in the learning characteristic of users. In particular, we propose the hierarchical generalization and learning problems for each generalized level in a recursive form. To solve the complex formulated problem due to its recursive structure, we develop a distributed learning algorithm, DemLearn. The proposed algorithm uses the bottom-up scheme to iteratively performs the local learning by solving personalized learning problems and hierarchical updating the generalized models for groups at higher levels. With extensive experiments, we validate both specialization and generalization performance of all learning models on benchmark MNIST, Fashion-MNIST, federated extended MNIST (FE-MNIST), and CIFAR-10 datasets.

To that end, we discuss the preliminaries of democratized learning in Section II. Based on the Dem-AI guidelines, we formulate hierarchical generalized, personalized learning problems, and propose a novel distributed learning algorithm in Section III. We validate the efficacy of our proposed algorithm for both specialization and generalization performance of clients, groups, and global models compared to the conventional FL algorithms in Section IV. Finally, Section V concludes the article.

II. Democratized Learning: Preliminaries

Different from FL, the Dem-AI framework [13] introduces a self-organizing hierarchical structure for solving common single/multiple complex learning tasks by mediating contributions from a large number of learning agents in collaborative learning. Moreover, it unlocks the following features of democracy in the future distributed learning systems. According to the differences in their characteristics, learning agents form appropriate groups that can be specialized for similar agents to deal with the learning tasks. These specialized groups are self-organized in a hierarchical structure and collectively construct the shared generalized learning knowledge to improve their learning performance by reducing individual biases due to the unbalanced, highly personalized local data. In particular, the learning system allows new group members to 1) speed up their learning process with the existing group knowledge and 2) incorporate their new learning knowledge in expanding the generalization capability of the whole group. In Dem-AI systems, learning agents are free to join any of the appropriate groups and exhibit equal power in the construction of their groups' generalized learning model. Here, the power of each group can be

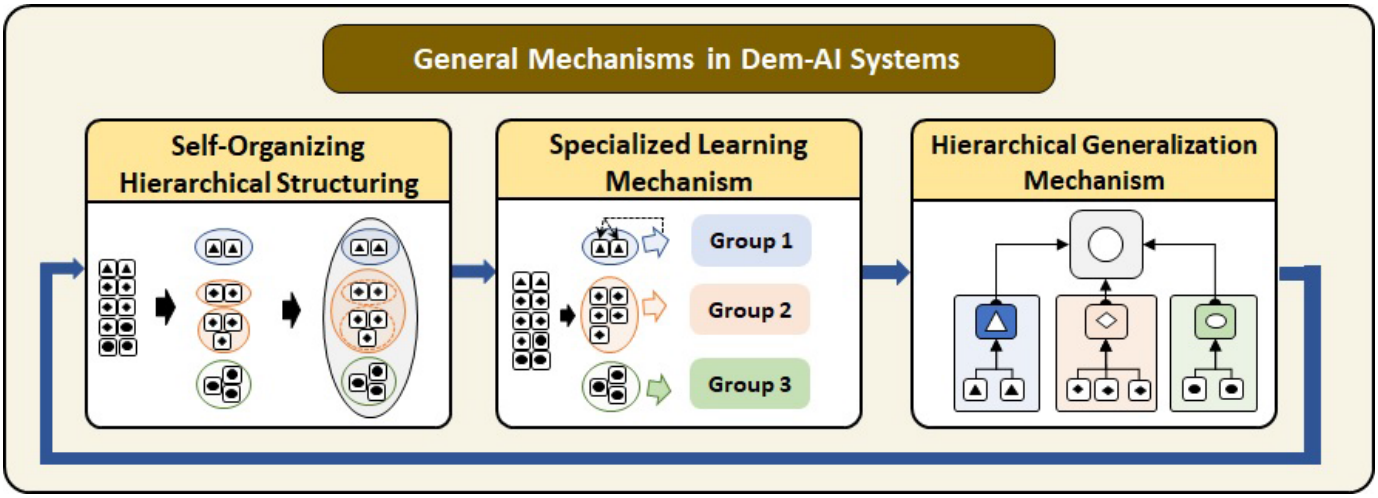


Fig. 2. Three general mechanisms for designing Dem-AI systems.

represented by the number of its members which varies over the training time. We introduce a brief summary of Dem-AI concepts and principles [13] in the following discussion.

A. Definition and Goal

Democratized learning (Dem-AI in short) studies dual (coupled and working together) specialized–generalized processes in a self-organizing hierarchical structure of large-scale distributed learning systems. The specialized and generalized processes must operate jointly toward an ultimate learning goal identified as performing collective learning from biased learning agents, who are committed to learning from their own data using their limited learning capabilities. As such, the ultimate learning goal of the Dem-AI system is to establish a mechanism for collectively solving common (single or multiple) complex learning tasks from a large number of learning agents.

B. Specialized Process

This process is used to leverage the specialized learning capabilities of the learning agents and specialized groups by exploiting their collected data. By incorporating the generalized knowledge of higher-level groups created by the generalization mechanism, the learning agents can update their model parameters to reduce biases in their personalized learning. Thus, the personalized learning objective has two goals: 1) to perform specialized learning and 2) to reuse the available hierarchical generalized knowledge.

C. Generalized Process

The generalization mechanism encourages group members to share knowledge when performing learning tasks with similar characteristics and construct hierarchical levels of generalized knowledge. The hierarchical generalized knowledge helps the Dem-AI system maintain the generalization ability for reducing biases of learning agents and efficiently dealing with environmental changes or performing new learning tasks.

D. Self-Organizing Hierarchical Structure

The hierarchical structure of specialized groups and the relevant generalized knowledge are constructed and regulated following a self-organization principle based on the similarity of learning agents. In particular, this principle governs the union of small groups to form a bigger group that eventually enhances the generalization capabilities of all members. Thus, specialized groups at higher levels in the hierarchical structure have more members and can construct more generalized (less biased) knowledge faster adaptation to new environments in [15].

E. Transition in the Dual Specialized–Generalized Process

The specialized process becomes increasingly important compared to the generalized process during the training time. As a result, the learning system not only evolves to gain specialization capabilities from the learned tasks, but also loses the capabilities to deal with environmental changes such as new learning agents and new learning tasks. Meanwhile, the hierarchical structure of the Dem-AI system is self-organized and evolved from a high level of plasticity to a high level of stability, that is, from unstable specialized groups to well-organized specialized groups. The transition of the Dem-AI learning system is illustrated in Fig. 2 with three iterative sub-mechanisms such as generalization, specialized learning, and hierarchical structuring mechanism. Accordingly, the transition of the dual specialized–generalized process represents the steps in a typical democratized learning framework [13]. In that transition, the learning agents are grouped according to the similarities of their learning tasks at the early stage. Then, the generalized process helps in the construction of a hierarchical generalized knowledge for the specialized groups from the bottom-up and encourages the group members to be close together. In the meantime, the specialized learning processes leverage personalized learning to exploit their biased datasets by incorporating higher-level generalized group knowledge from top-level to lower-level groups. In doing so, the group members deviate from the common

generalized knowledge. After that, the hierarchical structure will be updated according to the new learning models.

In the next section, we develop a democratized learning design that results in a hierarchical generalized learning problem. To that end, we propose a novel democratized learning algorithm, DemLearn, to realize as an initial implementation of Dem-AI philosophy.

III. Democratized Learning Design

Dem-AI philosophy and guidelines in [13] envision different designs for a variety of applications and learning tasks. In this work, we focus on developing a novel distributed learning algorithm that consists of the following hierarchical clustering, hierarchical generalization, and learning mechanisms with a common learning task for all learning agents.

A. Hierarchical Clustering Mechanism

To construct the hierarchical structure of the Dem-AI system with relevant specialized learning groups, we adopt the commonly used agglomerative hierarchical clustering algorithm (i.e., dendrogram implementation from scikit-learn [14], [16]), based on the similarity or dissimilarity of all learning agents. The dendrogram method is used to examine the similarity relationships among individuals and is often used for cluster analysis in many fields of research. During implementation, the dendrogram tree topology is built-up by merging the pairs of agents or clusters having the smallest distance between them, following the bottom-up scheme. Accordingly, the measured distance is considered as the differences in the characteristics of learning agents (e.g., local model parameters or gradients of the learning objective function). Since we obtain a similar performance implementing clustering based on model parameters or gradients, in what follows, we only present a clustering mechanism using the local model parameters. Additional discussion for gradient-based clustering is provided in the supplementary material.

Given the local model parameters $\boldsymbol{w_n} = (\boldsymbol{w}_{n,1}, \ldots, \boldsymbol{w}_{n,M})$ of learning agent n, where M is the number of learning parameters, the measure distance between two agents $\boldsymbol{\phi}_{n,l}$ is derived based on the Euclidean distance such as $\boldsymbol{\phi}_{n,l} = \|\boldsymbol{w_n} - \boldsymbol{w_l}\|$. In addition, we consider the average-linkage method [17] for distance calculation between an agent and a cluster using the Euclidean distance between the model parameters of the agent and the average model parameters of the cluster members. Accordingly, the hierarchical tree structure is in the form of a binary tree with many levels. In consequence, it will require unnecessarily high storage and computational cost to maintain and be also an inefficient way to maintain a large number of low-level generalized models for small groups. As a result, we keep only the top K levels in the tree structure and discard the lower-level structure. Therefore, at the top-level K, the system could have two big groups that have a large number of learning agents.

B. Hierarchical Generalization and Learning Mechanism

The K level hierarchical structure emerges via agglomerative clustering. Accordingly, the system constructs K levels of the generalization, as in Fig. 2. As such, we propose hierarchical generalized learning problems (HGLPs) to build these generalized models for specialized groups in a recursive form, starting from the global model $\boldsymbol{w}^{(K)}$ construction at the top-level K as follows.

HGLP problem at level K

$$\min_{\boldsymbol{W}^{(K)}} L^{(K)} = \sum_{i \in \mathcal{S}_K} \frac{N_{g,i}^{(K-1)}}{N_g^{(K)}} L_i^{(K-1)}\left(\boldsymbol{w}_i^{(K-1)} \middle| \mathcal{D}_i^{(K-1)}\right) \quad (1)$$

$$s.t. \quad \boldsymbol{w}^{(K)} = \boldsymbol{w}_i^{(K-1)}, \ \forall i \in \mathcal{S}_K \quad (2)$$

where $\boldsymbol{W}^{(K)} = (\boldsymbol{w}^{(K)}, \boldsymbol{w}_1^{(K-1)}, \ldots, \boldsymbol{w}_{|\mathcal{S}_K|}^{(K-1)})$, $\mathcal{S}_K$ is the set of subgroups of the top-level group, and $L_i^{(K-1)}$ is the loss function of subgroup i given its collective dataset D_i. The objective function is weighted by a fraction of the number of learning agents $N_g^{(K-1)}$ of the subgroup i, and the total number of learning agents $N_g^{(K)}$ in the system. Hence, the subgroups which have more learning agents have higher impact to the generalized model at level K. The hard constraints in (2) enforce these subgroups to share a common learning model (i.e., a global variable $\boldsymbol{w}^{(K)}$). To preserve the specialization capabilities of each subgroup, these constraints (2) could be relaxed by using additional proximal terms in the objective. In this way, the problem encourages the subgroup learning models to become close to the global model but not necessarily equal. Thus, the relaxed problem HGLP' is defined as follows.

HGLP' problem at level K

$$\min_{\boldsymbol{W}^{(K)}} \sum_{i \in \mathcal{S}_K} \frac{N_{g,i}^{(K-1)}}{N_g^{(K)}} \left(L_i^{(K-1)}\left(\boldsymbol{w}_i^{(K-1)} \middle| \mathcal{D}_i^{(K-1)}\right) + \frac{\mu_K}{2}\left\|\boldsymbol{w}^{(K)} - \boldsymbol{w}_i^{(K-1)}\right\|^2\right) \quad (3)$$

where μ_K denotes the tradeoff between the learning loss and the generalization constraint enforcing the group learning models to be close to the global model $w^{(K)}$. Since the dataset is distributed and only available at the learning agents, the problem (3) at the top-level K can be solved starting from its members' problem first. Accordingly, the hierarchical generalized structure emerged naturally following the bottom-up scheme where the learning models at lower levels are updated before solving the higher-level generalized problems of its upper group. Specifically, the problem (3) can be decentralized and solved by the following problem of each subgroup i at the level $K-1$.

HGLP problem for each group i at level $K-1$

$$\min_{\boldsymbol{W}^{(K-1)}} \sum_{j \in \mathcal{S}_{i,K-1}} \frac{N_{g,j}^{(K-2)}}{N_g^{(K)}} \left(L_j^{(K-2)}\left(\boldsymbol{w}_j^{(K-2)} \middle| \mathcal{D}_j^{(K-2)}\right) + \frac{\mu_{K-1}}{2}\left\|\boldsymbol{w}_j^{(K-2)} - \boldsymbol{w}_i^{(K-1)}\right\|^2\right) + \frac{\mu_K N_{g,i}^{(K-1)}}{2N_g^{(K)}}\left\|\boldsymbol{w}^{(K)} - \boldsymbol{w}_i^{(K-1)}\right\|^2$$

where $\boldsymbol{W}^{(K-1)} = (\boldsymbol{w}_i^{(K-1)}, \boldsymbol{w}_1^{(K-2)}, \ldots, \boldsymbol{w}_{|\mathcal{S}_{i,K-1}|}^{(K-2)})$. Therefore, we make a general approximation form of the generalized

learning problem for the group i at the level k given the prior higher generalized model $\boldsymbol{w}^{(k+1)}$ as follows.

HGLP problem for each group i at level k

$$\min_{\boldsymbol{W}^{(k)}} \sum_{j\in\mathcal{S}_{i,k}} \frac{N_{g,j}^{(k-1)}}{N_g^{(K)}} \left(L_j^{(k-1)}\left(\boldsymbol{w}_j^{(k-1)} \middle| \mathcal{D}_j^{(k-1)}\right) + \frac{\mu_k}{2}\left\|\boldsymbol{w}_j^{(k-1)} - \boldsymbol{w}_i^{(k)}\right\|^2 \right) + \frac{\mu_{k+1} N_{g,i}^{(k)}}{2N_g^{(K)}} \left\|\boldsymbol{w}^{(k+1)} - \boldsymbol{w}_i^{(k)}\right\|^2 \quad (4)$$

where $\boldsymbol{W}^{(k)} = (\boldsymbol{w}_i^{(k)}, \boldsymbol{w}_1^{(k-1)}, \ldots, \boldsymbol{w}_{|\mathcal{S}_{i,k}|}^{(k-1)})$, $N_{g,i}^{(k)}$ is the number of learning agents of group i, and $\boldsymbol{w}^{(k+1)}$ is the learning model of the upper group at level $k+1$ in which group i belongs. Since there exists coupling between the upper and lower levels, and the training dataset is decentralized, the learning problem (4) of the group i at level k cannot be solved directly. Therein, similar to FL, the learning loss of the group can be distributed amongst the group members [2]. As a result, the objective of the group problem has the remaining proximal terms forcing the learning models in different levels to be close to each other. Therefore, the learning model is constructed with the model of upper group at level $k+1$ and group members at level $k-1$ models by solving the following problem:

$$\min_{\boldsymbol{w}} \sum_{j\in\mathcal{S}_{i,k}} \frac{\mu_k N_{g,j}^{(k-1)}}{N_g^{(K)}} \left\|\boldsymbol{w}_j^{(k-1)} - \boldsymbol{w}\right\|^2 + \frac{\mu_{k+1} N_{g,i}^{(k)}}{N_g^{(K)}} \left\|\boldsymbol{w}^{(k+1)} - \boldsymbol{w}\right\|^2. \quad (5)$$

The closed-form of the optimal solution of the problem (5) can be handily derived by setting the gradient to zero as follows:

$$\sum_{j\in\mathcal{S}_{i,k}} \mu_k N_{g,j}^{(k-1)} \left(\boldsymbol{w}_j^{(k-1)} - \boldsymbol{w}^*\right) = \mu_{k+1} N_{g,i}^{(k)} \left(\boldsymbol{w}^* - \boldsymbol{w}^{(k+1)}\right).$$

Thus, the learning model of group i can be updated as

$$\boldsymbol{w}_i^{(k)} = \alpha \boldsymbol{w}^{(k+1)} + (1-\alpha) \sum_{j\in\mathcal{S}_{i,k}} \frac{N_{g,j}^{(k-1)}}{N_{g,i}^{(k)}} \boldsymbol{w}_j^{(k-1)} \quad (6)$$

where $\alpha = \mu_{k+1}/(\mu_k + \mu_{k+1})$. The tradeoff parameter α can be tuned later in the experiments to control the contribution from the learning models of upper-group and group members.

Given the closed-form solution (6), the coupling between different levels can be approximated by splitting the model updates at each level via the bottom-top, and then, top-bottom scheme. In Fig. 3, we show the illustration of the proposed hierarchical updates. In particular, the lower groups and members are updated first before the upper groups. And then, the updated upper groups broadcast the parameters to their group members to finish one update cycle.

At the lowest level, each learning agent n can actually perform the local training process to fit its private data with the personalized learning problem using the latest hierarchical generalized models as follows.

PLP at level 0

$$\boldsymbol{w}_n^{(0)} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} L_n^{(0)}\left(\boldsymbol{w}|\mathcal{D}_n^{(0)}\right) + \frac{\mu}{2}\left\|\boldsymbol{w} - \boldsymbol{w}_n^{(1)}\right\|^2 \quad (7)$$

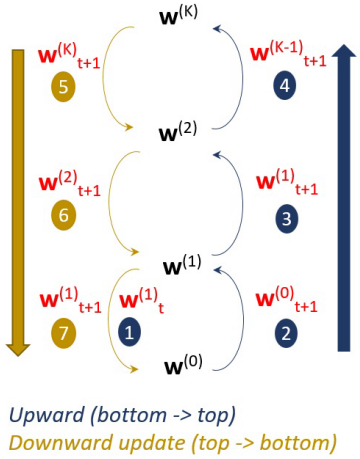


Fig. 3. Hierarchical update of the generalized models.

where $L_n^{(0)}$ is the personalized learning loss function for the learning task (e.g., cross entropy loss for classification [18]) given its personalized dataset $\mathcal{D}_n^{(0)}$, $N_{n,g}^{(k)}$ is the number of learning agents of the level-k group in which the agent n belongs. Solving the PLP problem, the learning agent can update their personalized model $w_n^{(0)}$ belonging to the parameterized deep learning model set $\mathcal{W}$. In this personalized level, the number of group member is 1.

C. Democratized Learning Algorithm

Inspired by the FedAvg [2] and FedProx [19] algorithms, we adopt the aforementioned recursive analysis and hierarchical clustering mechanism to develop a novel democratized learning algorithm, namely DemLearn. The details of our proposed algorithm are presented in Algorithm 1. Each agent n uses the upper-group model at level 1 (i.e., $\boldsymbol{w}_{n,t}^{(1)}$) as the initial learning model. Thereafer, the agent iteratively solves the PLP problem in equation (8) based on the gradient method. The updated client model will be sent to the central server to perform hierarchical clustering and update from the generalized level 1 to the level K. After every τ global round, the hierarchical structure is reconstructed according to the changes in the personalized learning model of agents. The generalized learning models of groups are updated, respectively, in bottom-top, and then top-bottom fashion, following (9) and (10) as the approximation of the closed-form solution (6). This allows the lower-level subgroups to contribute their knowledge for updating the group model. In return, they receive (and incorporate) the better generalized knowledge from the upper groups that enhances the generalization capacity of their local learning models. Additionally, we introduce an amplification trick in the bottom-top update for the first five rounds to speed up the initial stage of the learning process. Accordingly, the update from group members [i.e., $\boldsymbol{w}_{t+1}^{(k)}$ in (9)] is multiplied with a scaling constant 1.15 in the first five rounds.

IV. Experiments

A. Setting

In this section, we validate the efficacy of the DemLearn algorithm with the MNIST [20], Fashion-MNIST [21],

Algorithm 1 Democratized Learning (DemLearn)

Input: K, T, τ.
for $t = 0, \ldots, T-1$ **do**
  for learning agent $n = 1, \ldots, N$ **do**
    Agent n uses the upper-group model $\boldsymbol{w}_{n,t}^{(1)}$ as the initial model;
    Agent n iteratively updates the personalized learning model $\boldsymbol{w}_{n,t+1}^{(0)}$ as an in-exact minimizer (i.e., *gradient based*) of the following problem:

$$\min_{\boldsymbol{w}\in\mathcal{W}} L_n^{(0)}(\boldsymbol{w}|\mathcal{D}_n) + \tfrac{\mu}{2}\|\boldsymbol{w} - \boldsymbol{w}_{n,t}^{(1)}\|^2; \tag{8}$$

    Agent n sends updated learning model to the server;
  end for
  if ($t \mod \tau = 0$) **then**
    Server reconstructs the hierarchical structure by the clustering algorithm;
  end if
  Hierarchical Update: Each group i at each level k performs an update for its learning model from bottom to top for updating the contribution of group members

$$\boldsymbol{w}_{i,t+1}^{(k)} = \sum_{j\in\mathcal{S}_{i,k}} \frac{N_{g,j}^{(k-1)}}{N_{g,i}^{(k)}} \boldsymbol{w}_{j,t+1}^{(k-1)}. \tag{9}$$

  After the top-level model is updated, the lower-levels starts updating top to bottom for the contribution of the upper group as follows:

$$\boldsymbol{w}_{i,t+1}^{(k)} = \alpha \boldsymbol{w}_{t+1}^{(k+1)} + (1-\alpha)\boldsymbol{w}_{i,t+1}^{(k)}. \tag{10}$$

  The updated learning models at level 1 (i.e., $\boldsymbol{w}_{t+1}^{(1)}$) are then broadcast to all agents to update their local models following equation (10).
end for

Federated Extended MNIST [22], and CIFAR-10 [23] datasets for handwritten digits and fashion images recognition, and objects recognition, respectively. We conduct the experiments with 50 clients, where each client has median numbers of data samples at 64, 70, and 785 with MNIST, Fashion-MNIST dataset, and CIFAR-10, respectively. Different from these three datasets, we also experiment with the FE-MNIST dataset which has more number of classes such as 10 digits and 25 lowercase and 25 uppercase. Accordingly, we select 50 clients from 3559 users who have at least 50 data samples in the FE-MNIST dataset. Using these datasets, 20% of data samples on each client are used for evaluating the model testing performance. We divide the total dataset such that each client has a small amount of data from two specific labels among the overall ten in both datasets. In doing so, we replicate a scenario of biased personal datasets, that is, highly unbalanced data and a small number of training samples can be collected at agents. The learning models consist of two convolution layers followed by two pooling and two fully connected layers, whereas three convolution layers are used in the CIFAR-10 dataset. We set the update period $\tau = 1$ and validate the performance of the proposed algorithm with $K = 4$ generalized levels. Our implementation is developed based on the available code of FedProx in [19]. DemLearn, FedAvg, and FedProx use the common learning rate $\eta = 0.05$, local epoch $E = 2$, batch size $B = 10$. For FedProx, we set the parameter $\mu = 0.5$. Meanwhile, pFedMe needs detailed tuning to obtain a competitive accuracy for different datasets. The Python implementation of our proposed algorithm using Pytorch and datasets are available at https://github.com/nhatminh/Dem-AI.

B. Results

Existing FL approaches such as FedProx and FedAvg focus more on the learning performance of the global model rather than the learning performance of clients. Therefore, for forthcoming personalized applications, we implement DemLearn and measure the learning performance of all clients and the group models. In particular, we conduct evaluations for specialization (C-SPE) and generalization (C-GEN) of learning models at agents on average that are defined as the performance in their local test data only, and the collective test data from all agents in the region, respectively. Accordingly, we denote Global as global model performance; and G-GEN and G-SPE are the average generalization and specialization performance of group models, respectively. In addition to the standard C-SPE performance for local models, the introduced C-GEN performance is an important metric that shows the generalized capabilities of local models. Even though the biased local models can achieve high C-SPE values from very early, particularly due to their small local datasets, they still have very low generalized capabilities which can help to produce good predictions according to the frequent changes of users. Meanwhile, the global and group models have the highest generalized capabilities, but lower specialized capabilities during deployment to the clients.

In Figs. 4 and 5, we conducted performance comparisons of our proposed methods, DemLearn with the three FL methods, FedAvg [2], FedProx [19], and pFedMe [12] as baselines on four benchmark datasets, MNIST, Fashion-MNIST, FE-MNIST and CIFAR-10. Fig. 4(a) depicts the performance comparison of DemLearn with FedProx and FedAvg with the MNIST dataset. Experimental evaluations show that the proposed approach outperforms the baselines in terms of the convergence speed, especially to obtain better client generalization performance. We observe that the local model requires only 40 rounds to reach the C-GEN performance level of 80% using the proposed algorithm, whereas existing FL algorithms such as FedProx, FedAvg, and pFedMe take more than 80 global rounds to achieve a near-competitive level of performance as ours. Furthermore, after 100 rounds, DemLearn obtains better average client generalization performance (i.e., 88.77%) across client models and comparable C-SPE and Global performance as of FedAvg and pFedMe.

Following similar trends, Figs. 4(b) and (c), and 5 depict that the DemLearn algorithm performs better than FedProx and FedAvg with the Fashion-MNIST, FE-MNIST, and CIFAR-10 datasets in terms of the client generalization performance. In Fig. 4(c), the pFedMe algorithm found difficulties in

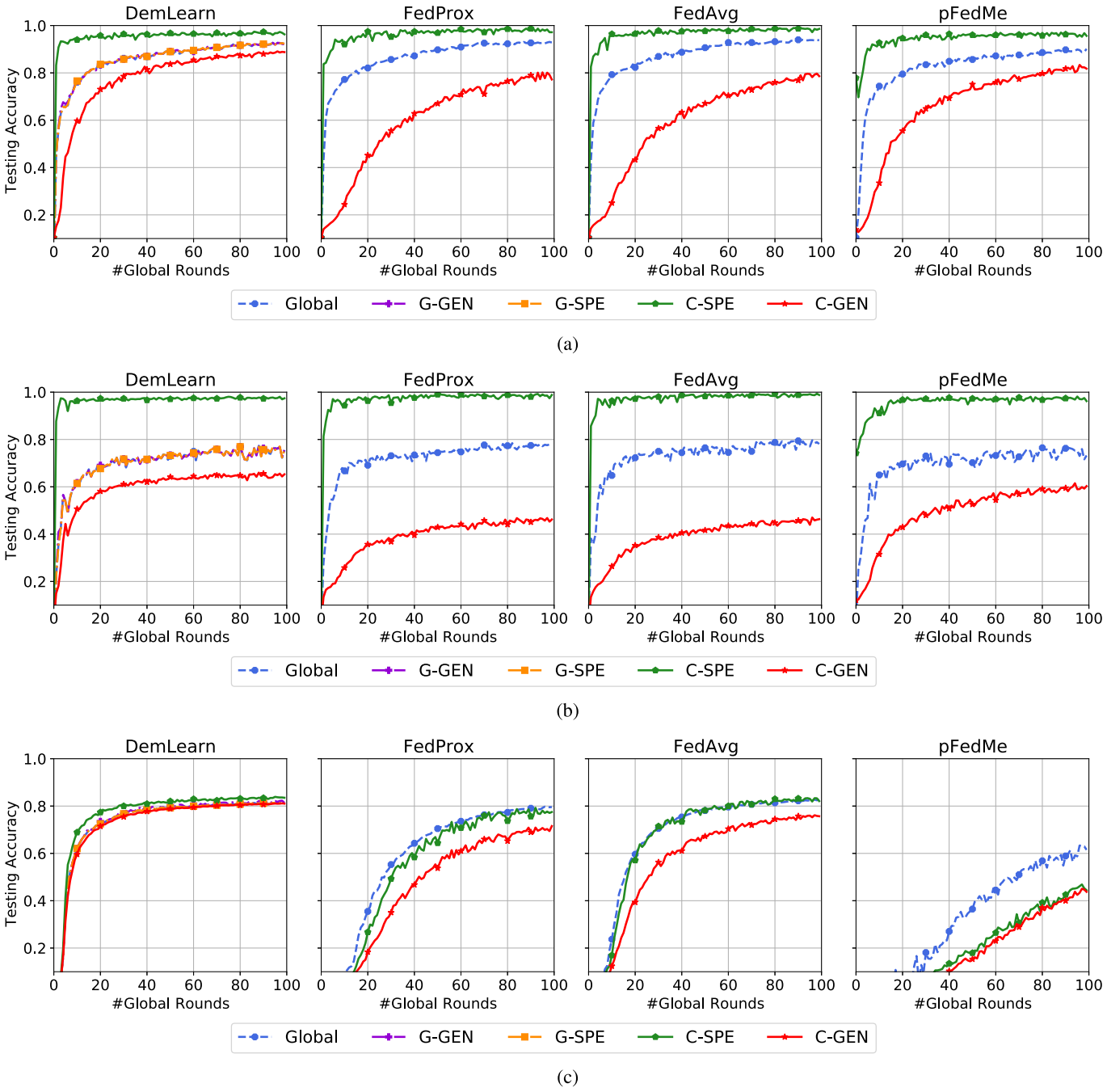


Fig. 4. Performance comparison of DemLearn versus FedAvg, FedProx, and pFedMe. (a) Experiment with MNIST dataset. (b) Experiment with Fashion-MNIST dataset. (c) Experiment with Federated Extended MNIST dataset.

tuning parameters to obtain a comparable performance and showed lower performance and more fluctuated rather than that of the other algorithms for the FE-MNIST dataset. Thus, pFedMe obtains a slow improvement of the client models in both specialization and generalization. Meanwhile, our proposed algorithm exhibits stable convergence speed and efficiency to achieve consistently high performance of learning models at all levels. In Fig. 5, we observe DemLearn suffers a slight degradation in C-SPE and Global performance to gain high C-GEN performance. After 100 rounds, DemLearn demonstrates good tradeoff learning capabilities of client models with high C-SPE (79.09%) and C-GEN (57%) performance, while other baseline algorithms only produce biased local models with low generalized capabilities.

In Fig. 6, we evaluate and compare the performance of the proposed algorithm for a fixed and self-organizing hierarchical structure via periodic reconstruction (i.e., $\tau = 1$). We observe that DemLearn benefits from the self-organizing mechanism and can provide slightly better generalization capability of client models. In addition, the amplification in the first five rounds helps the proposed algorithm can speed up the initial performance of the DemLearn algorithm.

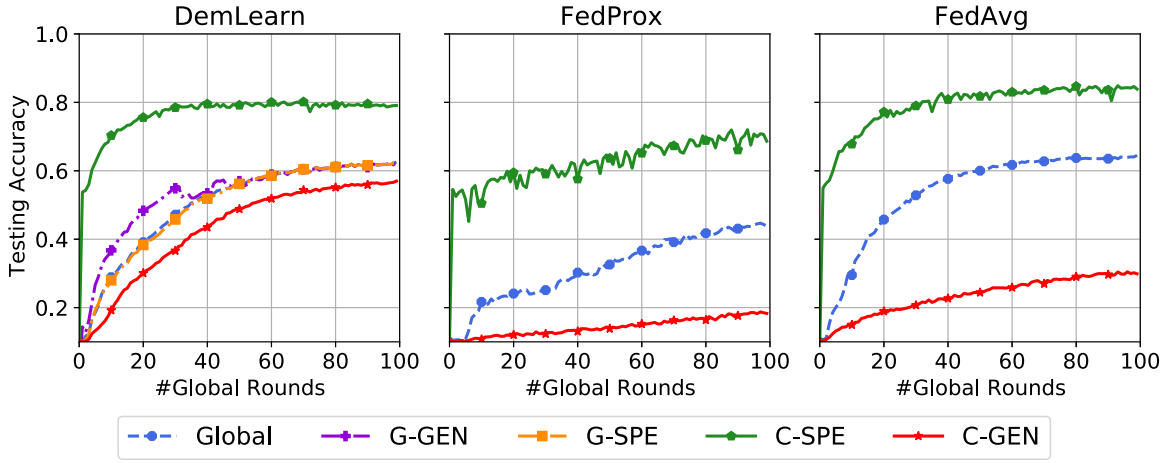
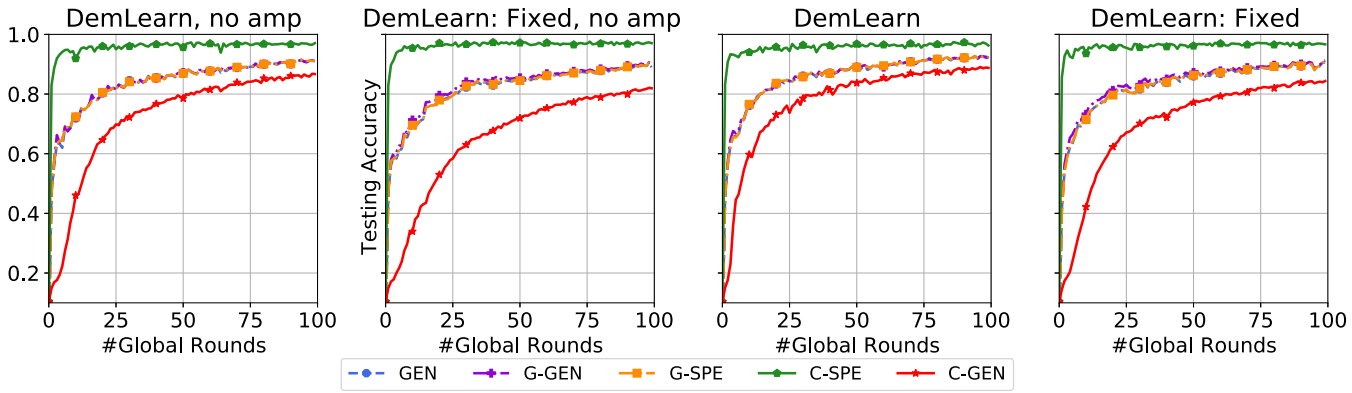
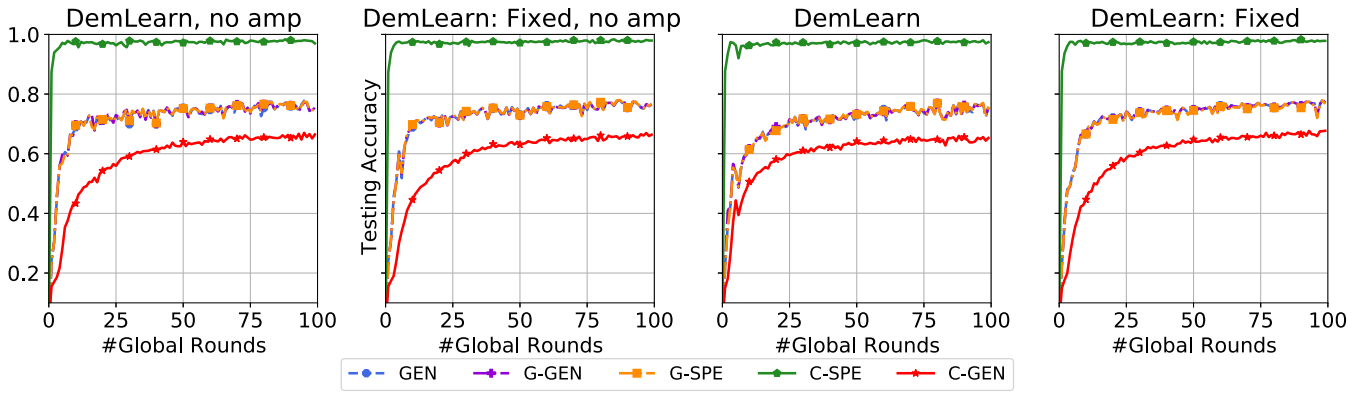


Fig. 5. Performance comparison of DemLearn versus FedAvg and FedProx with the CIFAR-10 dataset.



(a)



(b)

Fig. 6. Comparison of algorithms for a fixed and a self-organizing hierarchical structure. (a) Experiment with the MNIST dataset. (b) Experiment with the Fashion-MNIST dataset.

C. Tuning Parameters of the Proposed Algorithm

In this section, we show the impact of parameter α in the testing accuracy of MNIST and Fashion-MNIST datasets, as shown in Fig. 7. As can be seen in the results, when α is large, we obtain high performance of the client generalization while slight degradation in their specialized capabilities. As such, increasing the value of α enhances generalization, but also reduces the specialization performance of client models to a marginal extent. Since $\alpha = \mu_{k+1}/(\mu_k + \mu_{k+1})$ controls the contribution from the learning models of the upper group and group members, tuning α produces the impact on disseminating the generalized knowledge from upper groups to the lower-level groups and clients. At each level k, the higher value of α signifies the objective is more focused on minimizing the gap with the upper-group model at $k+1$ than the lower-group models at $k-1$. Furthermore, we evaluate the impact of other parameters, such as μ, K, τ, but they do not show any clear effects on the performance of DemLearn,

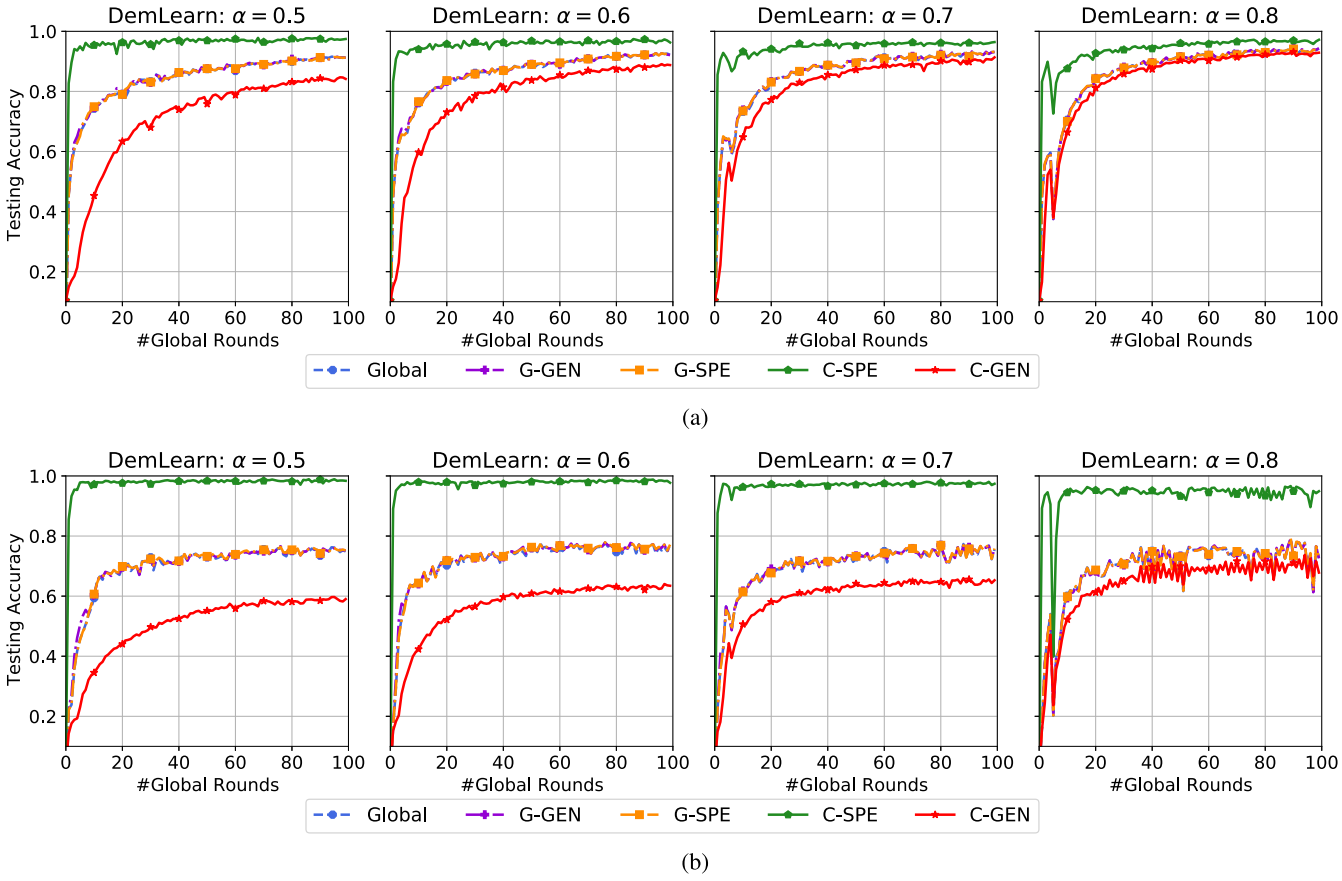


Fig. 7. Performance of DemLearn by varying α with MNIST and Fashion-MNIST datasets. (a) Experiment with the MNIST dataset. (b) Experiment with the Fashion-MNIST dataset.

likely due to the small-scale simulation settings. In addition to this, reducing the frequency of cluster updates by increasing the parameter τ can help the algorithm obtain comparable performance with a lower reorganizing cost of the hierarchical structure. However, due to the limited scope of the simulated datasets and settings, we advocate the impact of defined parameters on the performance of the proposed algorithm is better realized when evaluating the algorithm with more practical/experimental data. In particular, we keep it as our future work to evaluate our algorithm on the dataset that has a hierarchical structure based on groups of similar users, or exceedingly large and different users.

D. Clustering Approach

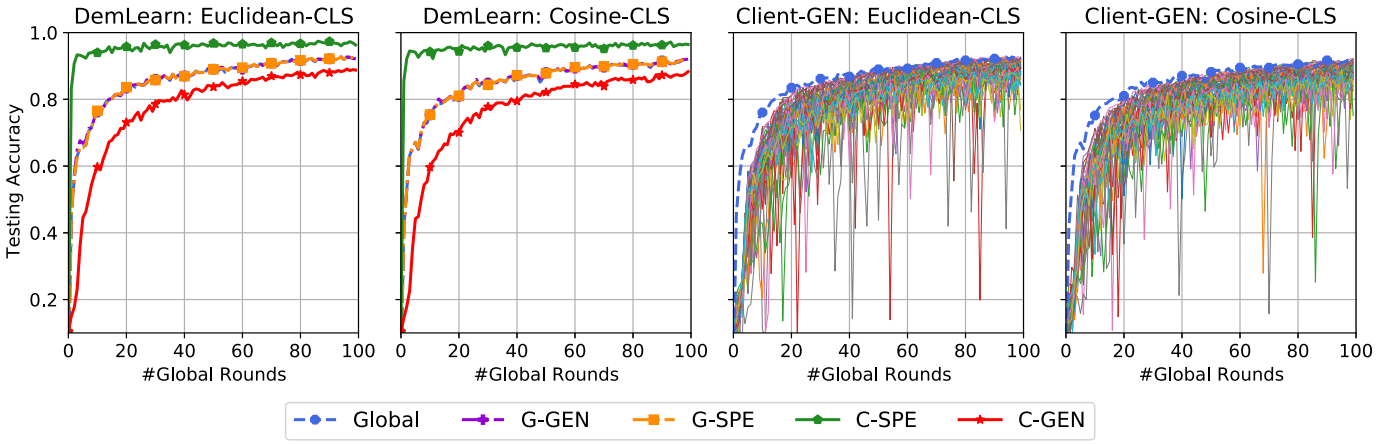
In addition to the Euclidean distance between learning parameters in the hierarchical clustering algorithm, we can also evaluate the measured distance $\phi_{n,l}^{(\cos)}$ between two learning models based on the cosine similarity as follows:

$$\phi_{n,l}^{(\cos)} = \cos(\boldsymbol{w}_n, \boldsymbol{w}_l) = \frac{\sum_{m=1}^{M} \boldsymbol{w}_{n,m} \boldsymbol{w}_{l,m}}{\sqrt{\sum_{m=1}^{M} \boldsymbol{w}_{n,m}^2} \sqrt{\sum_{m=1}^{M} \boldsymbol{w}_{l,m}^2}}. \quad (11)$$

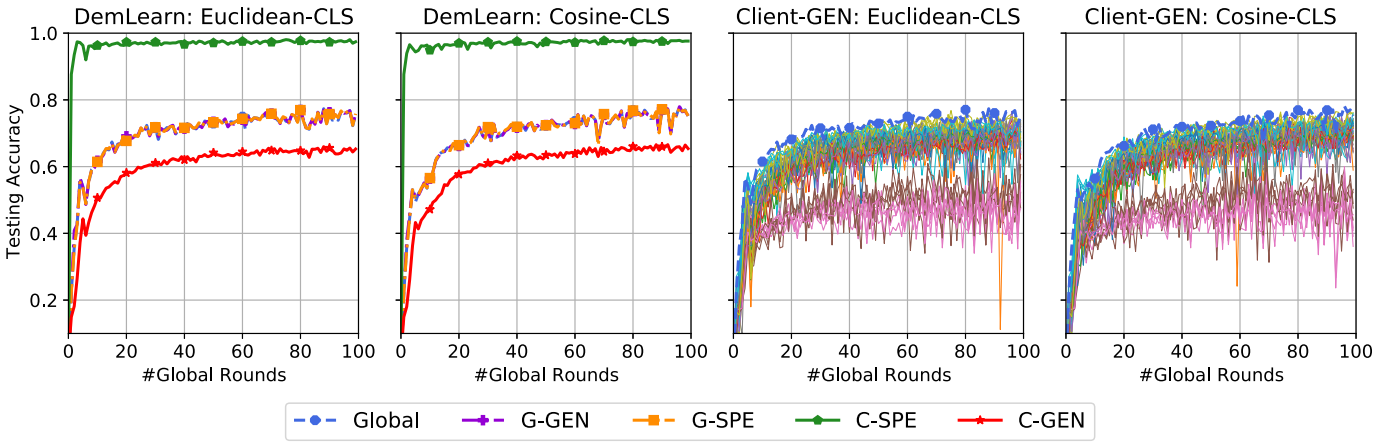
Our experimental results demonstrate that the clients and group models show almost similar learning performance (see Fig. 8) and different trends of cluster topology (see Fig. 9) using the DemLearn algorithm. As shown in the Client-GEN subfigure, some outliers contribute to the drop in client-GEN performance throughout the training. At the same time, most clients have comparable generalized capacities with the global model. Accordingly, the illustration further helps us figure out the exceedingly different clients with low C-GEN performance. That means, in each round, the outliers are detected (and classified) by a clustering mechanism that is exceedingly different from other client model parameters. However, we note that the outliers are not always the same and change throughout their local model updates. But some appear several times that provide meaningful information for the learning system. Furthermore, as shown in Fig. 9, the difference in cluster topology does not affect much on the overall learning performance.

E. Discussion

Compared to the FedAvg and FedProx, we have a similar on-device computational cost for solving PLP problem using the gradient descent method. Different from the others, pFedMe requires extra steps for the θ approximation and hence requires more computation time on the client device. On the other hand, for the computation time requirement at the server, the DemLearn algorithm requires extra computation cost for hierarchical clustering and a negligible cost for hierarchical updates. We note that the cost of the hierarchical clustering depends on the model size and the number of learning



(a)



(b)

Fig. 8. Comparison of different metrics using hierarchical clustering. (a) Experiment with the MNIST dataset. (b) Experiment with the Fashion-MNIST dataset.

agents. In principle, the standard algorithm for hierarchical agglomerative clustering has a time complexity of $O(n^3)$ and requires $O(n^2)$ memory [24]. In this work, we evaluate the algorithm with $n = 50$ users and obtain a running time of 0.0015 second per step on average when using a computer with CPU i7-7700K and a memory of 32 GB. For the communication cost, all algorithms require similar costs to send the model parameters.

In practice, to deploy a typical hierarchical structure, the Dem-AI systems can include three entities.

1) A cloud server that handles the global model (root node) and its sublevel generalized groups.
2) Distributed regional servers, which are edge servers deployed within each region and whose role is to manage the subgroups and learning agents.
3) Learning agents.

Our implementation for hierarchical clustering is in a centralized manner in this article. However, the agglomerative hierarchical clustering mechanism merges the learning agents and groups in a bottom-up manner, and then it is possible to implement it in a decentralized manner. Therefore, we can operate the hierarchical clustering in the cloud server for top levels and decentralized implementation for regions at edge servers for lower-level groups. Also, this grouping mechanism can mitigate the negative effects of the aggregation of exceedingly different learning agents to construct hierarchical generalized models. In that way, we maintain three or more levels of generalized models instead of a single global model. To the best of our knowledge, this design has not been considered yet in recent studies of personalized FL.

V. Conclusion and Future Works

The novel Dem-AI philosophy has provided general guidelines for specialized, generalized, and self-organizing hierarchical structuring mechanisms in large-scale distributed machine learning systems. Inspired by these guidelines, we have formulated the hierarchical generalized learning problems and developed a novel distributed learning algorithm, DemLearn. In this work, based on the similarity in the learning characteristics, the agglomerative clustering enables the self-organization of learning agents in a hierarchical structure, which gets updated periodically. Detailed analysis of experimental evaluations has shown the advantages and disadvantages of the proposed algorithm. Compared to conventional FL, we show that DemLearn significantly

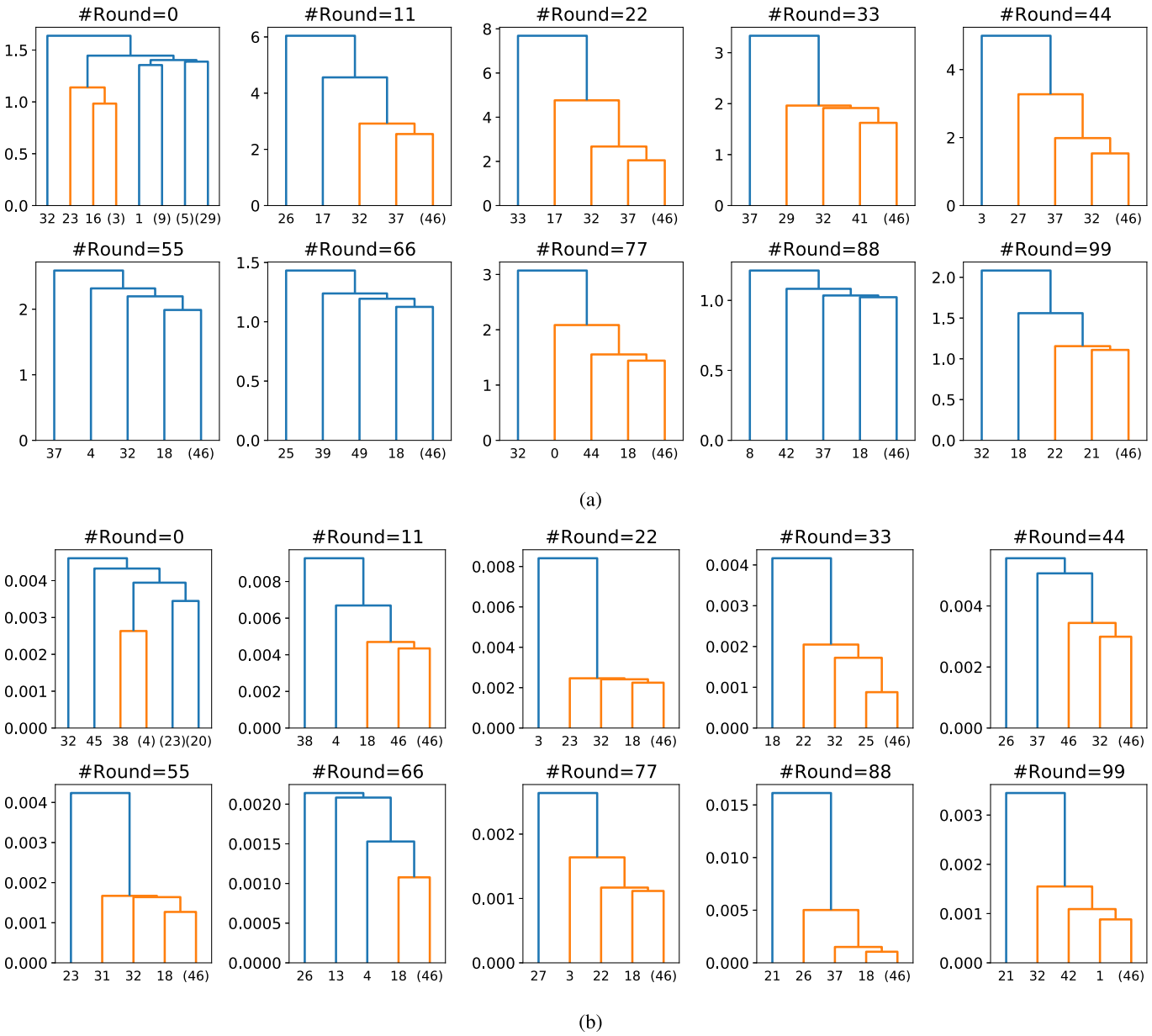


Fig. 9. Topology changes via hierarchical clustering in the DemLearn algorithm with the MNIST dataset. (a) Hierarchical clustering based on Euclidean distance. (b) Hierarchical clustering based on Cosine similarity distance.

improves the generalization performance of client models without largely compromising the specialization performance of clients' models. As a result, DemLearn enables good tradeoff learning capabilities of client models with high C-SPE and C-GEN performance, while other algorithms can only produce biased local models with low generalized capabilities. These observations benefit a better understanding and improvement in specialization and generalization performance of the learning models in future Dem-AI systems.

Democratized learning provides unique ingredients to develop future distributed personalized intelligent systems. To that end, the learning design could be further studied with personalized datasets, extended for multi-task learning capabilities, and validated with actual generalization capabilities in practice for new users and environmental changes. We advocate that our current design has not coped with multiple learning tasks [3] and the adaptability of the system due to environmental changes as general intelligent systems. Turning the general distributed learning systems into reality, we need to profoundly analyze the Dem-AI from a variety of perspectives such as the robustness and diversity of the learning models and novel knowledge transfer and distillation mechanisms [25], [26]. Also, it is possible to incorporate our flexible design with current approaches such as meta-learning and optimization-based methods, to further improve the personalization in FL. We consider our work as an orthogonal contribution to the design architecture of distributed AIs with recent approaches. Besides, we believe it is necessary to experiment with these design approaches

in common personalized settings using realistic datasets for personalized learning tasks.

REFERENCES

[1] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artif. Intell. Statist.*, vol. 54, Apr. 2017, pp. 1273–1282.

[3] V. Smith, C. K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 4424–4434.

[4] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 1387–1395.

[5] M. N. H. Nguyen, N. H. Tran, Y. K. Tun, Z. Han, and C. S. Hong, "Toward multiple federated learning services resource sharing in mobile edge networks," 2020, *arXiv:2011.12469*.

[6] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3241–3256, Feb. 2020.

[7] C. T. Dinh *et al.*, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.

[8] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," 2019, *arXiv:1909.07972*.

[9] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," 2020, *arXiv:2002.07948*.

[10] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.

[11] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," 2019, *arXiv:1909.11875*.

[12] C. Dinh, N. Tran, and T. D. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21394–21405.

[13] M. N. H. Nguyen *et al.*, "Distributed and democratized learning: Philosophy and research challenges," *IEEE Comput. Intell. Mag.*, vol. 16, no. 1, pp. 49–62, Feb. 2021.

[14] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[15] H. Mengistu, J. Huizinga, J.-B. Mouret, and J. Clune, "The evolutionary origins of hierarchy," *PLOS Comput. Biol.*, vol. 12, no. 6, Jun. 2016, Art. no. e1004829.

[16] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, 2011.

[17] S. Dasgupta and P. M. Long, "Performance guarantees for hierarchical clustering," *J. Comput. Syst. Sci.*, vol. 70, no. 4, pp. 555–569, 2005.

[18] J. Shore and R. Johnson, "Properties of cross-entropy minimization," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 4, pp. 472–482, Jul. 1981.

[19] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[21] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.

[22] S. Caldas *et al.*, "LEAF: A benchmark for federated settings," 2018, *arXiv:1812.01097*.

[23] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, CA, USA, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf, https://www.cs.toronto.edu/ kriz/cifar.html

[24] F. Nielsen, *Introduction to HPC With MPI for Data Science*. Cham, Switzerland: Springer, 2016.

[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. 27th Int. Conf. Artif. Neural Netw. (ICANN)*, in Lecture Notes in Computer Science, vol. 11141. Greece. Rhodes: Springer-Verlag, Oct. 2018, pp. 270–279.

[26] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data," 2018, *arXiv:1811.11479*.

Minh N. H. Nguyen (Member, IEEE) received the B.E. degree in computer science and engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2013, and the Ph.D. degree in computer science and engineering from Kyung Hee University, Seoul, South Korea, in 2020.

He is currently working as a Lecturer with The University of Danang–Vietnam–Korea University of Information and Communication Technology, Da Nang, Vietnam, and a collaborator with the Networking Intelligence Laboratory, Kyung Hee University. His research interests include wireless communications, mobile edge computing, federated learning, and distributed machine learning.

Dr. Nguyen received the Best KHU Ph.D. Thesis Award in engineering in 2020.

Shashi Raj Pandey (Member, IEEE) received the B.E. degree in electrical and electronics with a specialization in communication from Kathmandu University, Dhulikhel, Nepal, in 2013, and the Ph.D. degree in computer science and engineering from Kyung Hee University, Seoul, South Korea, in August 2021.

He has served as a Network Engineer with Huawei Technologies Nepal Company Pvt. Ltd., Lalitpur, Nepal, from 2013 to 2016. He is currently working as a Post-Doctoral Researcher with the Connectivity Section, Aalborg University, Aalborg, Denmark. His research interests include network economics, game theory, wireless communications, data markets, and distributed machine learning.

Tri Nguyen Dang received the B.S. degree in information teaching from the Hue University's College of Education, Hue, Vietnam, in 2014. He is currently pursuing the Ph.D. degree in computer science and engineering with Kyung Hee University, Yongin, South Korea, under a fully-funded scholarship.

His professional experiences include mobile application developer and middleware programming. His research interests include network optimization, mobile cloud computing, mobile-edge computing, and the Internet of Things.

Dr. Dang won a consolation prize of Olympiad in Informatics ACM ICPC Vietnam in 2013.

Eui-Nam Huh (Member, IEEE) received the B.S. degree from Pusan National University, Busan, South Korea, in 1990, the master's degree in computer science from The University of Texas at Austin, Austin, TX, USA, in 1995, and the Ph.D. degree from Ohio University, Athens, OH, USA, in 2002.

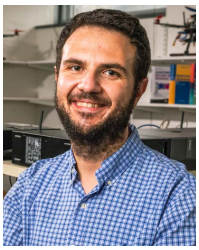
He is currently a Professor with the Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea. His research interests include cloud computing, the Internet of Things, future Internet, distributed real-time systems, mobile computing, big data, and security.

Dr. Huh is also a member of the Review Board of the National Research Foundation of Korea. He has also served many community services for ICCSA, WPDRTS/IPDPS, APAN Sensor Network Group, ICUIMC, ICONI, APIC-IST, ICUFN, and SoICT as various types of chairs. He is also the Vice-Chairperson of the Cloud/Bigdata Special Technical Group of TTA and an Editor of ITU-T SG13 Q17.

Nguyen H. Tran (Senior Member, IEEE) received the B.S. degree in electrical and computer engineering from the HCMC University of Technology, Ho Chi Minh City, Vietnam, in 2005, and the Ph.D. degree in electrical and computer engineering from Kyung Hee University, Seoul, South Korea, in 2011.

He was an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University, from 2012 to 2017. Since 2018, he has been with the School of Computer Science, The University of Sydney, Sydney, NSW, Australia, where he is currently a Senior Lecturer. His research interests include distributed computing, machine learning, and networking.

Dr. Tran received the Best KHU Thesis Award in engineering in 2011 and several best paper awards, including IEEE ICC 2016 and ACM MSWiM 2019. He receives the Korea NRF Funding for Basic Science and Research 2016–2023 and ARC Discovery Project 2020–2023. He was the Editor of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING from 2016 to 2020, and the Associate Editor of IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS 2020 in the area of distributed machine learning/federated learning.



Walid Saad (Fellow, IEEE) received the Ph.D. degree from the University of Oslo, Oslo, Norway, in 2010.

He is currently a Professor with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, where he leads the Network sciEnce, Wireless, and Security (NEWS) Laboratory. His research interests include wireless networks (5G/6G/beyond), machine learning, game theory, cybersecurity, unmanned aerial vehicles, semantic communications, and cyber-physical systems.

Dr. Saad was the author/coauthor of ten conference best paper awards and the 2015 and 2022 IEEE ComSoc Fred W. Ellersick Prize. He was the coauthor of the 2019 IEEE Communications Society Young Author Best Paper and the 2021 IEEE Communications Society Young Author Best Paper. He also serves as an Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING and the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also an Area Editor of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, an Associate Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC)–Special Issue on Machine Learning for Communication Networks, and an Editor-at-Large of the IEEE TRANSACTIONS ON COMMUNICATIONS.

Choong Seon Hong (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University, Tokyo, Japan, in 1997.

In 1988, he joined KT, where he was involved in broadband networks as a Member of Technical Staff. He was with the Telecommunications Network Laboratory, KT, as a Senior Member of Technical Staff and the Director of the Networking Research Team until 1999. Since 1999, he has been a Professor with the Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea. His research interests include machine learning and mobility management.

Dr. Hong is currently a member of ACM, IEICE, IPSJ, KIISE, KICS, KIPS, and OSIA. He has served as the General Chair, the TPC Chair/Member, or an Organizing Committee Member for international conferences, such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, TINA, SAINT, and ICOIN. He was an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and *Journal Communications and Networks*, and an Associate Technical Editor of *IEEE Communications Magazine*. He also serves as an Associate Editor for the *International Journal of Network Management* and *Future Internet*.