Socially-Aware-Clustering-Enabled Federated Learning for Edge Networks

Latif U. Khan, Zhu Han, *Fellow, IEEE*, Dusit Niyato, *Fellow, IEEE*, and Choong Seon Hong, *Senior Member, IEEE*

***Abstract*—Edge Intelligence based on federated learning (FL) can be considered to be a promising paradigm for many emerging, strict latency Internet of Things (IoT) applications. Furthermore, a rapid upsurge in the number of IoT devices is expected in the foreseeable future. Although FL enables privacy-preserving, on-device machine learning, it still exhibits a privacy leakage issue. A malicious aggregation server can infer the sensitive information of other end-devices using their local learning model updates. Furthermore, centralized FL aggregation server might stop working due to security attack or a physical damage. To address the aforementioned issues, we propose a novel concept of socially-aware-clustering-enabled dispersed FL. First, we present a novel framework for socially-aware-clustering-enabled dispersed FL. Second, we formulate a problem for minimizing the loss function of the proposed FL scheme. Third, we decompose the formulated problem into three sub-problems, such as local devices relative accuracy minimization (i.e., end-devices local accuracy maximization) sub-problem, clustering sub-problem, and resource allocation sub-problem, due to the NP-hard nature of the formulated problem. The clustering and resource allocation sub-problems are solved using low complexity schemes based on a matching theory. The end devices' relative accuracy minimization problem is solved by using a convex optimizer. Finally, numerical results are provided for validation of the proposed FL scheme. Furthermore, we show the convergence of the proposed FL scheme for image classification tasks using the MNIST dataset.**

***Index Terms*—Edge computing, Internet of Things, federated learning.**

I. Introduction

THE EXPECTED upsurge of Internet of Everything (IoE) applications, such as flying vehicles, haptics, extended reality, and brain-computer interfaces, is spurring the development towards the new generation of wireless systems, namely, Sixth Generation (6G) wireless systems [1]. Although the Fifth Generation (5G) of wireless networks were aimed to support IoE applications, there are certain applications (e.g., brain-computer interfaces) that cannot be effectively enabled by 5G [2]. Therefore, 6G will be considered as a key enabler of IoE applications. To enable 6G-enabled smart IoE applications, machine learning (ML) will play the role of a key enabler [1], [3], [4]. However, ML based on centralized training suffers from the drawback of privacy leakage due to the migration of end-devices data to a centralized location. Furthermore, continuous interaction with the data generating devices is necessary for some applications. For instance, autonomous driving cars generate a 4,000 gigaoctet of data everyday [5]. Therefore, conventional ML based on one-time training does not consider the newly generated data, and thus might suffer from performance degradation. To address the aforementioned issues, one can use federated learning (FL). FL does not migrate end-devices data to the centralized server, and thus enables learning of a globally shared model using on-device ML in a distributed manner [6], [7].

FL can be divided into two types as per the location of global model aggregation: edge-based FL and cloud-based FL [6]. In edge-based FL, global model is computed at the network edge, whereas global model is computed at a remote cloud for the cloud-based FL. Specifically, edge-based FL deals with the local-context-aware applications that have a set of devices in a small geographical area. On the other hand, cloud-based FL deals with global-context-aware learning for end-devices located over a large geographically distributed area. Although FL was introduced to preserve the users' privacy through decentralized and on-device ML, it has few prominent challenges.

- In conventional FL, a malicious aggregation server can use devices local learning models for inferring their private information, thus causing privacy leakage [8]–[10]. Therefore, it is necessary to tackle this privacy leakage issue of FL. To address this issue, differential privacy was introduced [8]. However, this scheme achieve privacy preservation at the cost slowing down the convergence speed of FL.
- In 6G wireless systems, the device's connectivity density will be more than $10^6/km^2$ [4]. Therefore, enabling such a massive number of devices with FL will require a significant amount of communication resources. FL will

Manuscript received November 18, 2020; revised March 1, 2021 and May 13, 2021; accepted May 28, 2021. Date of publication June 18, 2021; date of current version September 9, 2021. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1A4A1018607). The associate editor coordinating the review of this article and approving it for publication was S. Ayoubi. *(Corresponding author: Choong Seon Hong.)*

Latif U. Khan and Choong Seon Hong are with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, South Korea (e-mail: cshong@khu.ac.kr).

Zhu Han is with the Electrical and Computer Engineering Department and the Computer Science Department, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, South Korea.

Dusit Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

Digital Object Identifier 10.1109/TNSM.2021.3090446

generally suffer from high convergence time if the number of end devices increases under wireless resources constraints. The reason for the increase in convergence time is due to the high delay in the sharing of the FL updates between aggregation server and the devices. On the other hand, generally, FL performance in terms of learning accuracy degrades if the number of participating devices decreases [11]. Therefore, an efficient resource allocation scheme must be used to enable the participation of more devices in FL.

- The centralized aggregation server in FL might stop working due to physical damage or security attack by a malicious user, thus interrupting the FL process (i.e., robustness issue).

To address the aforementioned challenges, we propose a novel socially-ware device-to-device (D2D) communication-enabled distributed aggregation-based dispersed FL (DDFL) framework. The DDFL framework is based on true decentralization to enable robust operation. The DDFL framework enables the efficient reuse of communication resources. Furthermore, the DDFL framework offers better privacy preservation than traditional FL. Our paper has the following contributions.

- We present a novel socially-aware-clustering-enabled DDFL framework for edge networks to offer robustness, privacy-awareness (better than traditional FL), and efficient communication resource usage.
- We propose a clustering for the DDFL framework based on the social relationship between devices. The cluster heads are selected mainly considering edge betweenness, social similarity, and physical distance between devices. The devices with the high social dissimilarity, high edge betweenness, and less physical distance are preferable candidates for the cluster heads (more details will be given in Section IV).
- We define a loss function that simultaneously takes into account the effect of packet error rate (PER) due to wireless channel uncertainties and local learning accuracy on a global FL model. The PER mainly depends on the signal-to-interference-plus-noise ratio (SINR). The SINR in our problem depends on clustering and resource allocation, whereas relative local accuracy depends on local device parameters, such as local dataset size and local operating frequency. Furthermore, we formulate an optimization problem for minimizing the global DDFL loss function by optimizing clustering, resource allocation, and relative local accuracy.
- Due to the NP-hard nature of the formulated problem, we decompose the main problem into three sub-problems: clustering sub-problem, resource allocation sub-problem, and the local accuracy sub-problem. Simultaneously solving these problems is very difficult. Therefore, to solve these sub-problems, an iterative scheme (an overview is given in Fig. 1) is proposed that is based on solving one sub-problem while fixing the other sub-problem variables. One-sided one-to-many matching game is used for clustering, whereas resource allocation is performed using a one-sided one-to-one matching game. The reason for using matching-theory-based solutions is due to their low complexity compared to traditional heuristic algorithms [12]. For minimizing the relative local accuracy, we use a convex optimization solver (i.e., CVX solver). Note that minimizing the relative local accuracy is actually maximizing the local accuracy.
- Finally, we provide numerical results to validate our proposed algorithm. Furthermore, we show the convergence of the proposed FL scheme for image classification tasks using the MNIST dataset. We study the impact of local learning model iterations, sub-global iterations, and global communication rounds on the performance of the global DDFL model. Furthermore, a comparison of DDFL with traditional FL shows its superior performance in terms of learning model accuracy for the MNIST dataset.

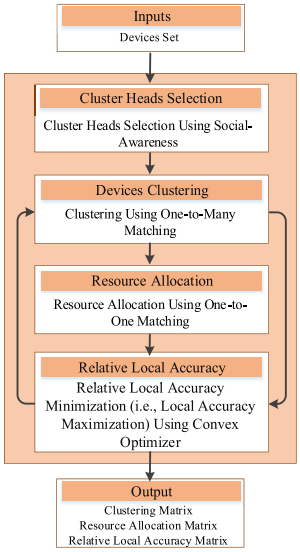

Fig. 1. Proposed solution approach.

The rest of the paper is organized as follows. Section II provides an overview of various works regarding resource optimization in FL, socially-aware D2D communication, matching theory for resource allocation, and hierarchical FL. Section III presents the proposed DDFL framework. System model and problem formulation are given in Section IV. Section V presents the proposed solution. Numerical results are given in Section VI. Section VII concludes this paper.

II. Related Works

A. Resource Optimization in Federated Learning

Several papers studied resource optimization in FL [6], [13], [14]. The authors in [6] surveyed resource optimization. Additionally, a Stackelberg game-enabled incentive mechanism for FL at the network edge is presented. Finally, few open challenges and future directions were presented. Another work jointly considered learning and communication for wireless FL [14]. The authors formulated a problem to jointly optimize power allocation, resource allocation, and user selection for wireless FL. The authors

TABLE I
COMPARISON OF RELATED WORKS WITH PROPOSED WORK

Reference	Resource optimization	Robustness	Privacy-awareness	Remark
Khan *et al.*, [6]	✓	✗	✗	Reviewed resource optimization and incentive mechanism for FL.
Dinh *et al.*, [13]	✓	✗	✗	Presented optimization model and resource allocation for wireless FL.
Chen *et al.*, [14]	✓	✗	✗	Presented a joint learning and communication framework for FL.
Wang *et al.*, [15]	✗	✗	✗	Optimized local learning model iterations for a global rounds to achieve a certain FL performance.
Li *et al.*, [16]	✗	✗	✗	Socially-aware device-to-device communication.
Zhao *et al.*, [17]	✗	✗	✗	Socially-aware resource allocation.
Ashraf *et al.*, [18]	✗	✗	✗	Proposed clustering and resource allocation for wireless networks.
Khan *et al.*, [19]	✓	✗	✗	Proposed a heuristic resource allocation scheme for wireless FL.
Kazmi *et al.*, [20]	✗	✗	✗	Proposed resource allocation for wireless networks.
Kazmi *et al.*, [12]	✗	✗	✗	Presented a learning and game theory-based on resource allocation for device-to-device networks.
Bairagi *et al.*, [21]	✗	✗	✗	Considered matching game for resource allocation in co-existence of eMBB and URLLC.
Abad *et al.*, [22]	✓	✗	✗	Proposed hierarchical FL scheme.
Liu *et al.*, [23]	✗	✗	✗	Edge-cloud hierarchical FL.
Wang *et al.*, [24]	✗	✗	✗	Convergence analysis for hierarchical FL.
Our paper	✓	✓	✓	N.A

modeled PER for wireless FL. A closed-form expression was derived for the expected FL convergence rate. Based on derived expression for the expected convergence rate, a mixed-integer non-linear programming problem (MINLP) was obtained. For resource allocation, a Hungarian algorithm was proposed. Although the work presented in [14] provided extensive analysis, it is preferable to consider multiple base station scenarios with security analysis. Wang *et al.* in [15] studied FL for edge networks. They proposed a control algorithm that offered a tradeoff between local model update and global model aggregation for minimizing the loss function under resource budget constraints. Finally, they validated their proposal using a real dataset. Tran *et al.* in [13] analyzed wireless FL. An optimization problem was derived that offered two kinds of tradeoffs. One tradeoff is computation and communication latencies as per the learning accuracy. Another tradeoff is the local energy consumption and FL model computing time. All the works presented in [6], [13]–[15] considered FL with a centralized aggregation which can be malfunctioned either due to physical damage or security attack.

B. Socially-Aware Device-to-Device Communication

Numerous works [16]–[19] considered socially-aware D2D networks. In [16], the authors proposed a socially-aware D2D communication network architecture. A real-time trace was studied for finding the social characteristics between different users in a social network. They identified weak and strong social ties depending on how frequently the nodes communicate with each other. Another work in [17] considered social relationships in resource allocation for devices in cellular networks. First, a utility based on social characteristics and the physical location of cellular users and D2D users was defined. Next, a social group utility maximization game (SGUM) was proposed to maximize the utilities of users. For SGUM, a Nash Equilibrium was investigated, and proposed a distributed algorithm to perform efficient socially-aware resource allocation. Finally, numerical results were provided to show the effectiveness of their distributed scheme. In [18], the authors proposed a socially-aware clustering for D2D communication. A utility function to maximize the achievable throughput of the social D2D network was proposed. To efficiently maximize the utility, a many-to-one matching game-based solution was proposed. Numerical results were provided to show the effectiveness of their matching-based scheme. The works presented in [16]–[18] considered only socially-aware D2D communication without considering FL. In contrast, FL over the D2D network was studied in [19]. In [19], one cluster with a cluster head acting as an aggregation server is formed by considering social relationship value (only social closeness parameter) among the devices. However, one cluster will suffer from a robustness issue. The device (i.e., cluster head) performing aggregation of local learning models might suffer from malfunctioning issues due to physical damage or security attack [25], [26]. To address this limitation, we will use multiple clusters. Every cluster uses its own cluster head as

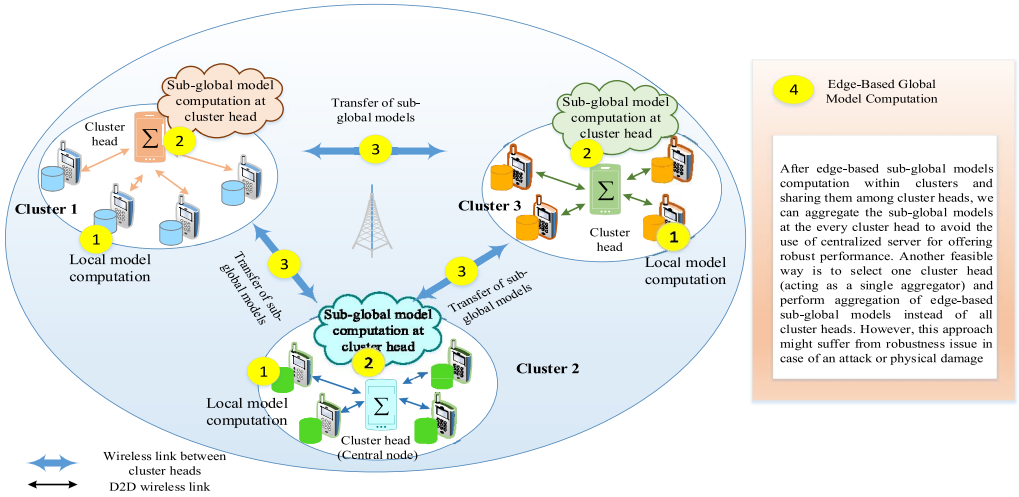

Fig. 2. Socially-aware-clustering-enabled DDFL framework.

an aggregation server in current work. The use of multiple cluster heads performing sub-global aggregations will offer robustness. Additionally, we can efficiently reuse resource blocks occupied by cellular users for various clusters while not exceeding the maximum allowed interference on cellular users [27]. However, selecting multiple cluster heads requires proper attention. Therefore, we consider various factors in determining the multiple cluster heads. These factors are social relationship value (i.e., social closeness parameter), social similarity, edge betweenness, and the Euclidean distance. The devices with less social similarity, high edge betweenness, and low Euclidean distance are preferred to be cluster heads (More details about these parameters are given in Section IV).

C. Matching Theory for Resource allocation

In [12], [20], and [21] matching theory was considered for resource allocation. In [12], the authors considered D2D communication and defined a utility to maximize the overall throughput. Two cases such as partial reuse and full reuse were considered. The partial reuse case is based on the usage of resource block by a single D2D pair, whereas the full reuse model is based on using a single resource block by multiple D2D pairs. For partial reuse mode, a one-to-one matching game was proposed, whereas a one-to-many matching game was proposed for full reuse mode. In [20], a hierarchical matching game-based approach was proposed for efficient resource allocation. A two-stage combinatorial problem for modeling service selection and resource purchasing was formulated. The formulated problem was solved using a matching game-based scheme. Another work [21] considered fair coexistence of WiFi and LTE-U. Specifically, LTE-U sum-rate maximization is considered under WAP-LTE-U coexistence and user quality of service constraints. To solve the formulated, a one-sided matching game and cooperative Nash bargaining game (NBG) were proposed. The NBG was considered for co-existence, whereas one-sided matching performed efficient resource allocation.

D. Hierarchical Federated Learning

Few works [22]–[24] considered hierarchical FL. In [24], the authors studied local averaging in FL before a central aggregation takes place. They studied the effect of various local aggregations on the performance of the global FL model. Furthermore, the authors provided convergence analysis for the considered schemes. In another work [22], the authors considered hierarchical FL for a heterogeneous cellular network consisting of a macro base station and few small cell base stations. In their approach, the first aggregation takes place at the small cell base stations which are followed by aggregation at the macro base station. Finally, the authors provided simulation results for the CIFAR-10 dataset. Liu *et al.* in [23] proposed client-edge-cloud FL. In their proposed approach, several local aggregations at the edge take place before a cloud aggregation. In all the works [22]–[24], centralized aggregation was considered which suffers from the robustness issue. A malicious user can attack the centralized aggregation server or it can stop working due to physical damage. Furthermore, the works in [22]–[24] did not consider the privacy concerns of FL. Furthermore, they did not effectively consider the effect of channel uncertainties on the FL model performance. Different from the works in [22]–[24], we consider a decentralized aggregation-enabled dispersed FL that enables robustness and minimizes the effect of channel uncertainties on the FL performance.

III. Distributed-Aggregation-Based Dispersed Federated Learning Framework

In this section, we present our DDFL framework as shown in Fig. 2. In our framework (the sequence diagram is shown in Fig. 3), first, a cluster is formed based on social relationships. These social relationships can be social closeness, social similarity, and edge betweenness centrality. Social closeness determines whether the devices can communicate or not because of social trust. The devices trust each other if their social closeness is high, and vice versa [16], [19]. Therefore,

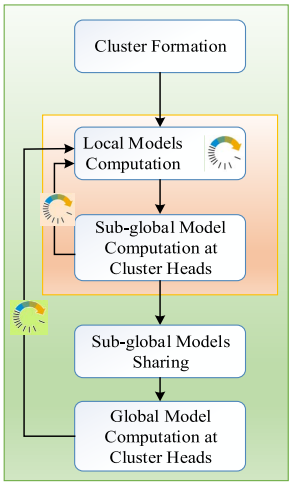

Fig. 3. DDFL framework sequence diagram.

the first condition for trustworthy communication in our framework is social closeness greater than the predefined threshold. Normally, we choose threshold value between 0 and 1 for the case when devices social closeness is taken between (0, 1) [28]. Greater the value of a threshold more trustworthy will be the communication, and vice versa. Next, social similarity and edge betweenness are used to compute the social distance between nodes (will be discussed in detail in Section IV). Socially dissimilar devices are good for data dissemination, and vice versa. Edge betweenness centrality indicates the central nature of the edge to the social graph. Using social distance (i.e., social similarity and edge betweenness centrality) and physical distance, we compute the cluster heads (i.e., central devices) that have a higher ability of data dissemination due to their dissimilar nature, centrality, and low physical distance than other devices.

Next to cluster head selection, clusters are formed to minimize the proposed DDFL loss. A sub-global model for every cluster is computed in a similar fashion to traditional FL. Local learning models are computed at devices and sent to the cluster heads of their clusters where sub-global aggregations take place. This iterative process of sub-global model computation continues for a predefined sub-global iterations. It should be noted here that the choice of sub-global iterations must be done intelligently. In DDFL, there are two kinds of model weight divergences, such as (a) among aggregation servers at sub-global level, and (b) among the global server and aggregation servers at sub-global level. The model weight divergences strictly depend on the fashion of data heterogeneity [23]. Another important aspect for DDFL is that we must choose an effective aggregation scheme that accounts for data heterogeneity within groups used for sub-global model computations. Therefore, FedAvg [7] might not perform significantly well when modified to work with the DDFL scheme for a higher number of sub-global iterations. In [23], FedAvg was modified to HierFedAvg and applied two different scenarios of non-IID and IID. Performance analysis of the HierFedAvg has shown improvement when the number of aggregations at the edge was increased while correspondingly reducing the number of local iterations for their considered cases of the MNIST data (i.e., the product of the edge aggregations and local iterations should remain the same). In another work, the authors analyzed hierarchical FL where multiple aggregations take place at the small cell base stations before a central aggregation takes place [22]. They have shown the general performance improvement with an increase in the number of sub-global aggregations per global aggregation (i.e., for 2, 4, and 6 sub-global iterations) at the small cell base stations for the CIFAR-10 dataset. From the above discussion, we conclude that there must be an effective aggregation scheme and intelligent selection of local iterations, sub-global iterations, and global iterations for DDFL. Furthermore, it might be possible to enable performance improvement of DDFL with an increase in the number of sub-global iterations after a careful design of the aggregation mechanism that truly accounts for data heterogeneity.

After sub-global model computation, global server aggregates sub-global models to compute a global FL model. There can be two possible ways to aggregate the sub-global models, such as centralized aggregation and distributed aggregation. In centralized aggregation, all the sub-global models will be aggregated at a centralized server located at the base station. However, this approach might suffer from a robustness issue due to failure in the working of the aggregation server (i.e., because of physical damage or security attacks) [10]. On the other hand, distributed aggregation of sub-global models involves the sharing of sub-global models between cluster heads where sub-global models aggregation takes place. This approach will offer more robustness compared to the centralized aggregation of sub-global models, but needs additional communication resources. On the other hand, one of the cluster-head based on specific criteria (i.e., high backup power and social connectivity) can perform aggregation of all sub-global models. However, this approach might suffer from the drawback of low robustness. A malicious user can attack one cluster head acting for aggregation of sub-global models from different clusters to produce a global model, and disturb the FL process. Therefore, our proposed DDFL uses a distributed fashion of sub-global model aggregation. Finally, after global model computation at the cluster heads, the cluster heads send back the global models to all the devices of their clusters. This process continues until the DDFL global model reaches a convergence.

The proposed DDFL framework offers better robustness compared to the conventional FL. Even if one or few but not significant number of the cluster heads stop working, the DDFL process will not be completely interrupted. In DDFL, there are many clusters for learning a sub-global model. The sub-global models of all the cluster heads other than the malfunctioned cluster head will be considered in the learning process. It must be noted here that there might be a slight loss in global DDFL model accuracy due to the non-participating cluster. However, the whole DDFL process will not be interrupted in contrast to conventional FL using a centralized aggregation server. To train a massive number of devices using the DDFL framework, there is a need to effectively use communication resources. In clusters (consists of nearby located

TABLE II
SUMMARY OF THE KEY NOTATIONS

Notation	Description	Notation	Description
$\overline{\mathcal{N}}$	Set of all devices.	$\overline{N}$	Total devices.
$\mathcal{M}$	Set of cluster heads (central devices).	M	Total clusters
$\mathcal{N}$	Set of devices involved in local learning.	$N = \overline{N} - M$	Number of devices involved in local learning.
$\mathcal{G}$	Social graph of devices.	$\mathcal{D}$	Vertices set of devices of a social graph.
$\mathcal{E}$	Set of edges of a social graph.	$\mathcal{S}$	Social graph weights set that denote social distance between vertices.
$\Gamma_{n,m}$	Social similarity between device n and cluster head m.	$\Upsilon_{n,m}$	Edge betweenness for device n and cluster head m.
$\gamma_{n,m}$	Social closeness indicator between device n and cluster head m.	$\Omega_{n,m}$	Matrix using social and physical distances for cluster heads computation.
x_n^m	Clustering binary variable.	θ_n	Relative local accuracy of device n.
$y_{n,m}^r$	Binary resource allocation variable.	f_{FL}	FL loss function.

devices), the DDFL framework can easily reuse resource blocks by protecting cellular users via low transmit power. Other than communication resources reuse, the DDFL framework offers better privacy preservation. Using local model updates, a malicious aggregation server has the ability to infer devices information [8], [29]. In DDFL clusters, cluster heads and devices communicate only when their social closeness is greater than a threshold (i.e., trustworthy devices), and thus offer privacy preservation [16], [19]. Furthermore, it is very difficult to infer the end-devices sensitive information from sub-global models. Therefore, we can say that the proposed DDFL can further enhance users' privacy.

Our proposed DDFL is based on true decentralization, therefore, it can be applied to various practical IoT applications, such as smart industries, augmented reality, and autonomous driving cars. For instance, consider an autonomous driving car scenario where a sub-global model is computed inside an autonomous driving car. Autonomous cars have higher mobility that makes it challenging to have a seamless connection with the roadside unit (RSU). Therefore, if we apply conventional FL, a device inside the car might not be able to connect to the RSU. In contrast, DDFL will allow the devices within the car to compute the sub-global model inside a car in an iterative manner. Next, the sub-global models of various cars will be transmitted to their nearby RSUs for sharing. The RSUs will then compute global model updates via aggregation of sub-global models received from other RSUs and finally, send them back to the cars. As in DDFL, we can tune between the sub-global iterations and global rounds, therefore, we can choose a higher number of sub-global rounds for autonomous driving cars and fewer global rounds. This will enable us to easily implement the DDFL for autonomous driving cars scenario than the conventional FL.

IV. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a system model comprising a set $\overline{\mathcal{N}}$ of $\overline{N}$ devices and a base station. Every device in the network has a social closeness with other devices in its social group. Such social closeness offers an opportunity to enable D2D communication for performance enhancement. The advantage of using D2D communication in clusters lies in the reuse of the already occupied spectrum by cellular users. To apply DDFL, we first choose a set $\mathcal{M}$ of M central devices (i.e., cluster heads) that will serve for sub-global model aggregation for other devices of their clusters. All the devices other than central devices are represented by a set $\mathcal{N}$ of $N = \overline{N} - M$ devices. A summary of key notations used in the paper is given in Table II.

A. Social Network Model

In this sub-section, we discuss the social distance between different devices to form clusters. The social distance depends on different parameters, such as social similarity and edge betweenness centrality. The social tie between end devices is formed only if their social relationship between devices n and m is greater than a certain threshold $\Psi_{n,m}$ [19]. The value of this social relationship varies between 0 (i.e., the least social relationship) and 1 (i.e., the highest social relationship). To explain the other factors of social distance, consider a social graph $\mathcal{G} = (\mathcal{D}, \mathcal{E}, \mathcal{S})$ to represents a socially-aware D2D network, where $\mathcal{D}$, $\mathcal{E}$, and $\mathcal{S}$ represent a set of vertices (i.e., devices), set of edges, and set of edges weights. The weight of the edges actually represents the social distance and euclidean between the two adjacent devices. The social distance $s_{n,m}$ between device n and central node m is defined as follows [30]:

$$s_{n,m} = \alpha \Gamma_{n,m} + \beta \Upsilon_{n,m}, \tag{1}$$

where $\Gamma_{n,m}$ and $\Upsilon_{n,m}$ denote the social similarity and edge betweenness between device n and central device m, respectively. The sum of the scaling constants $\alpha + \beta = 1$ must be equal to one. The choice of α and β tunes the impact of social similarity and edge betweenness during the selection of cluster heads. To equally consider the impact of both social similarity and edge betweenness, we consider $\alpha = 0.5$ and $\beta = 0.5$. The binary variable $\gamma_{n,m}$ restricts the communication between the

devices. The devices will be able to communicate in a trustworthy way if their social relationship is greater than a safety threshold, i.e.,

$$\gamma_{n,m} = \begin{cases} 1, & \text{If device } n \text{ and } m \text{ has social tie } \geq \Psi_{n,m}, \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

Next, we discuss the social similarity between devices and central devices. The social similarity between the two devices significantly affects the data dissemination among them. The lower the degree of social similarity, the better are the devices for data dissemination in a social network and vice versa [31]. The similarity measurement element between device *n* and central device *m* is given by:

$$\Gamma_{n,m} = \sum_{t \in \mathcal{N}_C^{n,m}} \frac{1}{d_t}, \tag{3}$$

where d_t represent the degree of device t from a common neighbors set $\mathcal{N}_C^{n,m}$ of devices *n* and *m*. On the other hand, edge betweenness represents the central nature of the edge. The greater the value of edge betweenness of an edge more will be its central nature (i.e., more devices will use this edge for connectivity among each other) and vice versa [32], [33]. The edge betweenness centrality is given by [18]:

$$\Upsilon_{n,m} = \frac{\sum_{n,m \in \mathcal{N}} \frac{\zeta_{n,m}|e}{\zeta_{n,m}}}{(N-1)(N-1)}, \tag{4}$$

where $\zeta_{n,m}$ and $\zeta_{n,m}|e$ represent the total number of the shortest paths between device *n* and central device *m*, and the number of the shortest paths between *n* and *m* passing through *e*, respectively. On the other hand, the Euclidean distance $\Lambda_{n,m}$ between devices *n* and *m* is computed as $\Lambda_{n,m} = \sqrt{(ax_n - ax_m)^2 + (ay_n - ay_m)^2}$. Then, the matrices Γ, Υ, and Λ are normalized using min-max normalization. Finally, the social matrix Ω is computed as follows:

$$\Omega_{n,m} = \frac{s_{n,m}}{\Lambda_{n,m}}. \tag{5}$$

Using matrix $\Omega_{n,m}$, we can compute the set of cluster heads (centralized devices). These cluster heads are the devices having higher values of social distance and lower physical distances. High social dissimilarity and edge betweenness centrality contribute to the high social distance, whereas lower physical distances reflect higher values of matrix $\Omega_{n,m}$. The devices with the highest values are preferred as centralized devices. Let the binary variable x_n^m represent the association of a device *n* with cluster *m*, i.e.,

$$x_n^m = \begin{cases} 1, & \text{If device } n \text{ is associated with cluster } m, \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

On the other hand, the number of devices associated with a cluster head must not exceed the maximum limit:

$$\sum_{n=1}^{N} x_n^m \leq \triangle_m, \ \forall m \in \mathcal{M}, \tag{7}$$

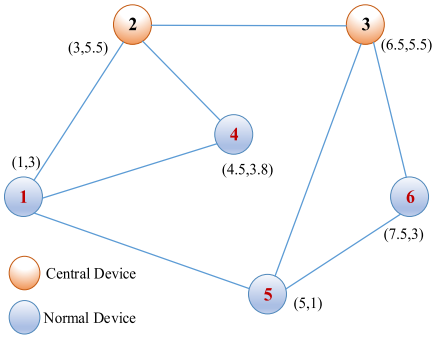

Fig. 4. Social graph scenario.

where the value of $\triangle_m$ depends on the capability of the cluster head regarding simultaneous communication with the other devices. In our paper, we use a fixed value for all cluster heads. Apart from the maximum number of devices assigned to a particular cluster, one device must not be assigned to more than one cluster:

$$\sum_{m=1}^{M} x_n^m \leq 1, \ \forall n \in \mathcal{N}. \tag{8}$$

It must be noted here that training a local learning model at the end-devices will consume significant energy compared to the aggregation (i.e., averaging for FedAvg) performed by the cluster head. However, the cluster head has to manage transmission with multiple end-devices that will cause higher transmission energy consumption than end-devices transmission energy. Overall, the energy consumption of cluster heads with a small number of end-devices in its cluster will not be higher than the end-devices energy consumption.

B. Example Scenario

In this sub-section, we will show how the cluster heads are selected based on social distance and physical distance. Consider Fig. 4 which shows 6 devices with their edges based on social closeness. The devices with social closeness greater than a threshold are allowed to communicate and are indicated by their connecting edge. The social closeness matrix can be given by.

$$\gamma = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{9}$$

Next, we compute the social similarity matrix which shows the dissemination capability of devices. It must be noted that we adopt and compute the min-max normalized metrics in this paper. The reason for normalization is to capture an equally proportional effect from different matrices on social matrix computation. The min-max normalization is given by.

$$c_{\text{norm}} = \frac{c - c_{\min}}{c_{\max} - c_{\min}}. \tag{10}$$

The normalized social similarity matrix for the graph of Fig. 4 is given by.

$$\Gamma = \begin{bmatrix} 0 & 0.7500 & 1.0000 & 0.5000 & 0 & 0.5000 \\ 0.7500 & 0 & 0 & 0.5000 & 1.0000 & 0.5000 \\ 1.0000 & 0 & 0 & 0.5000 & 0.7500 & 0.5000 \\ 0.5000 & 0.5000 & 0.5000 & 0 & 0.5000 & 0 \\ 0 & 1.0000 & 0.7500 & 0.5000 & 0 & 0.5000 \\ 0.5000 & 0.5000 & 0.5000 & 0 & 0.5000 & 0 \end{bmatrix} \tag{11}$$

The normalized edge betweenness matrix for graph of Fig. 4 is given by.

$$\Upsilon = \begin{bmatrix} 0 & 0.4444 & 0 & 0.5556 & 1.0000 & 0 \\ 0.4444 & 0 & 1.0000 & 0.5556 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0.4444 & 0.5556 \\ 0.5556 & 0.5556 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0.4444 & 0 & 0 & 0.5556 \\ 0 & 0 & 0.5556 & 0 & 0.5556 & 0 \end{bmatrix} \tag{12}$$

The social distance matrix obtained by adding Γ and Υ.

$$S = \begin{bmatrix} 0 & 0.5972 & 0.5000 & 0.5278 & 0.5000 & 0.2500 \\ 0.5972 & 0 & 0.5000 & 0.5278 & 0.5000 & 0.2500 \\ 0.5000 & 0.5000 & 0 & 0.2500 & 0.5972 & 0.5278 \\ 0.5278 & 0.5278 & 0.2500 & 0 & 0.2500 & 0 \\ 0.5000 & 0.5000 & 0.5972 & 0.2500 & 0 & 0.5278 \\ 0.2500 & 0.2500 & 0.5278 & 0 & 0.5278 & 0 \end{bmatrix} \tag{13}$$

The Euclidean distance matrix for graph of Fig. 4 is given by.

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & 0.4925 & 0.9295 & 0.5523 & 0.6880 & 1.0000 \\ 0.4925 & 0 & 0.5385 & 0.3488 & 0.7576 & 0.7920 \\ 0.9295 & 0.5385 & 0 & 0.4038 & 0.7298 & 0.4142 \\ 0.5523 & 0.3488 & 0.4038 & 0 & 0.4376 & 0.4777 \\ 0.6880 & 0.7576 & 0.7298 & 0.4376 & 0 & 0.4925 \\ 1.0000 & 0.7920 & 0.4142 & 0.4777 & 0.4925 & 0 \end{bmatrix} \tag{14}$$

The social matrix by element-wise dividing *S* by Euclidean distance matrix $\mathbf{\Lambda}$.

$$\Omega = \begin{bmatrix} 0 & 1.2125 & 0.5379 & 0.9555 & 0.7267 & 0.2500 \\ 1.2125 & 0 & 0.9286 & 1.5132 & 0.6600 & 0.3157 \\ 0.5379 & 0.9286 & 0 & 0.6191 & 0.8184 & 1.2741 \\ 0.9555 & 1.5132 & 0.6191 & 0 & 0.5713 & 0 \\ 0.7267 & 0.6600 & 0.8184 & 0.5713 & 0 & 1.0715 \\ 0.2500 & 0.3157 & 1.2741 & 0 & 1.0715 & 0 \end{bmatrix} \tag{15}$$

It is clear from the above matrix that the devices 2 and 3 has the highest values of the social matrix. Therefore, they can be considered as central devices (i.e., cluster heads).

C. Computation and Communication Model

1) Computation Model: Within every cluster, all the devices first compute their local learning model. The computation time of a device *n* of cluster *m* with local dataset size k_n^m, workload ψ (CPU-cycles required to process one data point), and operating frequency (CPU-cycles/sec) f_n is given by:

$$t_{\text{comp}}^{n,m}(f_n) = I_{\text{local}}\left(\frac{k_n^m \psi}{f_n}\right), \tag{16}$$

where I_{local} denotes the number of local training model iterations. The number of local iterations significantly determines the number of global FL model rounds for a fixed accuracy. Generally, the accuracy of the local learning model is high for high local iterations [6]. Therefore, we need a few global rounds of FL for a higher number of local iterations and vice versa. The number of local iterations determines the computation cost, whereas the global rounds determine the communication cost. Therefore, we must make a trade-off between the number of local iterations and global iterations for wireless FL. For a fixed global FL model accuracy ε, relative local accuracy θ, and a constant χ, the global iterations can be given as [34]:

$$I_g(\varepsilon, \theta) = \frac{\chi \log(1/\varepsilon)}{1-\theta}. \tag{17}$$

Every device transmits its local updates to the central device of its cluster for aggregation at sub-global level after computing local model. The operating frequency of a device (with CPU dependent constant, $\rho_{n,m}$) for a given number of local data set points and workload, determines the energy consumption, which is given by:

$$E_{\text{comp}}^{n,m}(\boldsymbol{f}) = I_{\text{local}}\Big(\rho_{n,m} k_u^m \psi (f_n)^2\Big). \tag{18}$$

From (18), it is clear that the local device energy consumption is proportional to the square of the device operating frequency. End-devices generally have limited available backup power. Therefore, we must choose the operating frequency for optimizing the overall performance of the FL system, so as to make a trade-off between the local learning model accuracy within allowable computation time and local device energy consumption. Alternatively, by minimizing θ, i.e., to improve local device local learning model performance within the allowable time, there is a need to increase the local operating frequency for a fixed local learning model and dataset size. However, increasing local operating frequency causes an increase in energy consumption. Therefore, to account for the constraint that all the end-devices energy consumption must be within the maximum allowable limit, we can use the notion as follows.

$$\sum_{n=1}^{N} \theta_n \geq O_{\min}, \tag{19}$$

where the value of $O_{\min}$ depends mainly on the energy consumption due to local training model computation. Lower the value of $O_{\min}$, more will be the energy consumption and vice versa. As θ reflects the local operating frequency for a fixed local learning model and local dataset, therefore, all devices must select a continuous value of θ within limits, i.e.,

$$\theta_{\min} \leq \theta_n \leq \theta_{\max}. \tag{20}$$

2) Communication Model: We use orthogonal frequency division multiple access (OFDMA) in this paper for communication. The set $\mathcal{R}$ of R resource blocks are used by both cellular users and D2D users. Therefore, there will be interference between cellular users and D2D users. However, distinct resource blocks are assigned for D2D pairs to avoid interference between the devices. We use binary variable $y^r_{n,m}$ to assign the resource block r to device n of cluster m:

$$y^r_{n,m} = \begin{cases} 1, & \text{If device } n \text{ is assigned resource block } r \\ 0, & \text{otherwise}. \end{cases} \tag{21}$$

In many works, only one resource block is considered for a device [12], [14]. Additionally, the size of local learning model updates in FL depends strictly on the local learning model used. Therefore, we assign only resource block for transmission of local learning model updates similar to [14]. To do so, We assume that devices cannot be assigned more than one resource block:

$$\sum_{r=1}^{R}\sum_{m=1}^{M} x^m_n y^r_{n,m} \leq 1, \ \forall n \in \mathcal{N}. \tag{22}$$

On the other hand, one resource block must not be assigned to more than one device:

$$\sum_{m=1}^{M}\sum_{n=1}^{N} x^m_n y^r_{n,m} \leq 1, \ \forall r \in \mathcal{R}. \tag{23}$$

The assignment of resource blocks to all devices must follow the maximum available limit:

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{r=1}^{R} x^m_n y^r_{n,m} \leq R. \tag{24}$$

D. Federated Learning Model

The set $\mathcal{N}$ of devices with local datasets $Q_n, \forall n \in \mathcal{N}$ wants to train a globally shared FL model. The objective of all the devices participating in FL is to minimize the following function f_{FL}:

$$\underset{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_N}{\text{minimize}} \ \frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{k_n} f_{\text{FL}}(\boldsymbol{w}_n, d_{nk}, \Theta_{nk}), \tag{25a}$$

$$s.t. \ \ \boldsymbol{w}_1 = \boldsymbol{w}_2 = \cdots = \boldsymbol{w}_N = \boldsymbol{z}, \forall n \in \mathcal{N}, \tag{25b}$$

where k_n local dataset points in device n. The collection of data points for FL strictly depends on the application. For instance, training of a smart keyword suggestion by FL use inputs of the mobile device users as training data. On the other hand, a weather monitoring system can use the output of the IoT sensors as training data. The local weights w_n of device n determines the output Θ_{nk} for input d_{nk}. The global model is given by:

$$\boldsymbol{z} = \frac{\sum_{n=1}^{N} k_n w_n}{K}, \tag{26}$$

where K represents the total data points. The performance of wireless FL is significantly affected by channel uncertainties. In our model, we use the PER to study the effect of degradation due to the wireless channel on the devices involved in FL. The PER of a device n of cluster m for a resource block r is given by:

$$e^{(r)}_{n,m}(\boldsymbol{X}, \boldsymbol{Y}) = x^m_n y^r_{n,m} C^n_p, \ \forall m \in \mathcal{M}, \forall r \in \mathcal{R}, \tag{27}$$

where $C^n_p = 1 - e^{\left(\frac{-\vartheta(\sum_{c\in\mathcal{C}} h^r_{c,m} P^r_{c,m} + \sigma^2)}{p^r_{n,m} h^r_{n,m}}\right.}$. ϑ and σ^2 denote the waterfall constant [35] and noise, respectively. The variables $p^r_{n,m}$ and $h^r_{n,m}$ denote the transmitted power of the device n of cluster m and channel gain between device n and cluster-head of cluster m for a resource block r, respectively. $\sum_{c\in\mathcal{C}} h^r_{c,m} P^r_{c,m}$ represents the interference due to cellular users on the resource block r. We consider a free space path loss model where the path loss is given by $P_{\text{loss}} = 20\log_{10}(d) + 20\log_{10}(f) + 33.4$ [12]. Where d and f denote the distance between antennas (i.e., transmitting and receiving antennas) and frequency, respectively. In our model, we consider a standard gradient descent algorithm for training of the local learning model at the devices and assume that $F(z) = \frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{k_n} f_{\text{FL}}(\boldsymbol{z}, d_{nk}, \Theta_{nk})$. Furthermore, we consider the following conditions [13], [14]:

- The gradient $\nabla(F(\boldsymbol{z}))$ with respect to $\boldsymbol{z}$ is uniform Lipschitz continuous [36].

$$\|\nabla(F(\boldsymbol{z}_{t+1})) - \nabla(F(\boldsymbol{z}_t))\| \leq L\|\boldsymbol{z}_{t+1} - \boldsymbol{z}_t\|, \tag{28}$$

  where L and $\|*\|$ denote the positive constant and norm of $*$, respectively.
- Similar to many works [13]–[15], for a positive constant μ, $F(\boldsymbol{z})$ is strongly convex (e.g., regression loss function):

$$F(\boldsymbol{z}_{t+1}) \geq F(\boldsymbol{z}_t) + (\boldsymbol{z}_{t+1} - \boldsymbol{z}_t)^T \nabla F(\boldsymbol{z}_t) + \frac{\mu}{2}\|\boldsymbol{z}_{t+1} - \boldsymbol{z}_t\|^2. \tag{29}$$

- F($\boldsymbol{z}$) is twice-continuously differentiable. Using (28) and (29) we can write:

$$\mu \boldsymbol{I} \preceq \nabla^2(F(\boldsymbol{z})) \preceq L\boldsymbol{I}. \tag{30}$$

- For positive constants $\zeta_1 \geq 0$ and $\zeta_2 \geq 0$, $\|f(\boldsymbol{z}_t, d^i_{nk}, \Theta^i_{nk})\|^2 \leq \zeta_1 + \zeta_2\|\nabla F(\boldsymbol{z}_t)\|^2$.

The effect of PER on the performance of FL function satisfying the above-mentioned assumptions and exactly one resource block per device can be given by the loss function [14]:

$$f_{\text{FL}}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{r=1}^{R} x^m_n y^r_{n,m} k^m_n \big(e_{n,m}\big). \tag{31}$$

In (31), k^m_n is a constant and it is taken equal to 1 for simplicity. From (31), we have the intuition of loss function for reflecting the effect of PER on FL model accuracy as follows:

$$f_{\text{FL}}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{r=1}^{R} x^m_n y^r_{n,m} \big(e_{n,m}\big). \tag{32}$$

Generally, more communication rounds are needed in FL for low local accuracy, and vice versa. More communication

rounds will highly expose the FL parameters to wireless channel uncertainties. Therefore, the effect of local accuracy on the loss function of (31) must be considered. To reflect this effect, relative local accuracy θ is used in our work. Relative local accuracy has low values for high local learning model accuracy and vice versa. To reflect the effect of local relative accuracy in (32) on the loss function, we can modify as follows.

$$f_{\mathrm{FL}}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{r=1}^{R} x_n^m y_{n,m}^r \frac{e_{n,m}}{(1-\theta_n)}. \quad (33)$$

The inverse relationship of relative local accuracy to loss function is adopted from (17) [37]. By using first-order Taylor's approximation, we can re-write (33) as follows.

$$f_{\mathrm{FL}}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{r=1}^{R} x_n^m y_{n,m}^r (1+\theta_n) e_{n,m} \quad (34)$$

It must be noted here that f_{FL} denotes the loss in FL performance for one global DDFL round. In FL, the aggregation server takes an average of the end-devices local learning model updates, and thus the performance of FL for one global round is significantly affected by local learning model accuracy. Therefore, the relative local accuracy term $1+\theta_n$ in (34) reflects the effect of local learning model accuracy on global FL performance. Next, we formulate a D2D communication-enabled DDFL loss minimization problem.

E. DDFL Loss Minimization Problem

In this sub-section, we formulate an optimization problem to minimize the DDFL loss due to local device heterogeneity and channel uncertainties. The problem can be given by.

$$\mathbf{P}: \underset{\mathbf{X},\mathbf{Y},\boldsymbol{\theta}}{\text{minimize}} \; f_{\mathrm{FL}}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) \quad (35)$$

$$\text{subject to:} \sum_{n=1}^{N} x_n^m \leq \triangle_m, \; \forall m \in \mathcal{M}, \quad (35a)$$

$$\sum_{m=1}^{M} x_n^m \leq 1, \; \forall n \in \mathcal{N}, \quad (35b)$$

$$\sum_{r=1}^{R} \sum_{m=1}^{M} x_n^m y_{n,m}^r \leq 1, \; \forall n \in \mathcal{N}, \quad (35c)$$

$$\sum_{m=1}^{M} \sum_{n=1}^{N} x_n^m y_{n,m}^r \leq 1, \; \forall r \in \mathcal{R}, \quad (35d)$$

$$\sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{r=1}^{R} x_n^m y_{n,m}^r \leq R,, \quad (35e)$$

$$\sum_{n=1}^{N} \theta_n \geq O_{\min},, \quad (35f)$$

$$\theta_{\min} \leq \theta_n \leq \theta_{\max}, \; \forall n \in \mathcal{N}, \quad (35g)$$

$$x_n^m \in \{0,1\}, \; \forall n \in \mathcal{N}, m \in \mathcal{M}, \quad (35h)$$

$$y_{n,n}^r \in \{0,1\}, \; \forall n \in \mathcal{N}, m \in \mathcal{M}, r \in \mathcal{R}. \quad (35i)$$

Algorithm 1 Dispersed Federated Learning Algorithm

```
Weights initialization ω_0
for i = 1, 2, ..., Global Rounds do
    Step 1: Parallel run for all clusters
    for k = 1, 2, ..., Sub-global iterations do
        for i = 1, 2, ..., Local iterations do
            ω_n(i) ⟵ ω_n(i−1) − η∇F_n(ω_n(i−1))
        end for
        ω_s(k) ⟵ Sub − global aggregation(ω_n)
        ω_n ⟵ ω_s
    end for
    Step 2: Global aggregation
    ω_g(k) ⟵ Global aggregation(ω_s)
    ω_n ⟵ ω_g
end for
```

In problem **P**, the constraints (35a) and (35b) restricts the maximum number of devices in a cluster and limit the association of a device to a maximum of a single cluster, respectively. Constraints (35c) and (35d) denote that a device cannot be assigned more than one resource block and one resource block must not be assigned to more than one device, respectively. The total number of resource blocks assigned to all the devices must be less than or equal to the total number of available blocks (constraint(35e)). Furthermore, all devices must be assigned local relative accuracy within the range specified by the constraint (35g). Constraint (35f) shows that the sum of relative local accuracy of all devices must be greater or equal than the minimum threshold $O_{\min}$. Finally, the constraints (35h) and (35i) represent that clustering variable x_n^m and resource allocation variable $y_{n,m}^r$ must take binary values. The formulated problem is a MINLP problem and is NP-hard for a higher number of devices and central devices.

V. Proposed DDFL Loss Minimization Algorithm

In this section, we present our proposed DDFL framework (the summary is given in *Algorithm 1*) loss minimization algorithm. First, we analyze the loss function f_{FL} (defined in (34)) for different values of relative local accuracy (i.e., [0 1]) and SINR (i.e., [0 1]) in Fig. 5. It is clear from Fig. 5 that f_{FL} has the lowest value for lower values of relative local accuracy and higher values of signal-to-interference-plus-noise ratio (SINR). However, simultaneously achieving local relative accuracy and higher SINR requires more end-devices resources and throughput, which is costly. Therefore, we must make a tradeoff between the performance and resources used in the training of a FL model. Next, we solve the problem **P**. As our formulated DDFL loss function depends on three parameters, such as clustering, resource allocation, and relative local accuracy. Therefore, we decompose problem **P** into three sub-problems: relative local accuracy minimization sub-problem, socially-aware clustering sub-problem, and the resource allocation sub-problem, due to the NP-hard nature of the formulated problem. To provide a simple solution, we perform decomposition by fixing two variables (e.g., resource allocation and clustering) and solving the remaining one (e.g., relative local

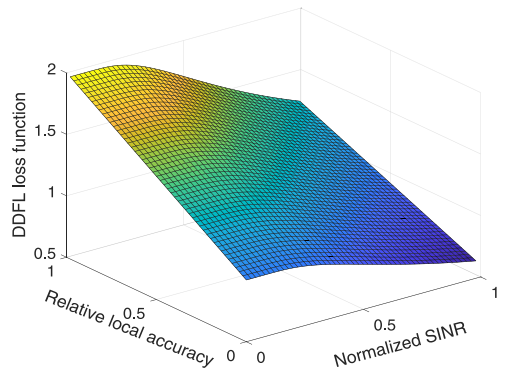

Fig. 5. DDFL Loss function.

accuracy minimization). Similarly, the other two sub-problems are solved iteratively to get the solution of the formulated NP-hard problem. For a given clustering and resource allocation variables, the relative local accuracy minimization problem can be written as:

$$\mathbf{P}-\mathbf{1}: \underset{\boldsymbol{\theta}}{\text{minimize}} \ f_{\text{FL}}(\boldsymbol{\theta}) \tag{36}$$

$$\text{subject to:} \ \sum_{n=1}^{N} \theta_n \geq O_{\min},, \tag{36a}$$

$$\theta_{\min} \leq \theta_n \leq \theta_{\max}, \ \forall n \in \mathcal{N}. \tag{36b}$$

Resource allocation problem **P-2** for a fixed clustering and relative local accuracy matrices can be written as:

$$\mathbf{P}-\mathbf{2}: \underset{\mathbf{Y}}{\text{minimize}} \ f_{\text{FL}}(\boldsymbol{Y}) \tag{37}$$

$$\text{subject to:} \ \sum_{r=1}^{R}\sum_{m=1}^{M} x_n^m y_{n,m}^r \leq 1, \ \forall n \in \mathcal{N}, \tag{37a}$$

$$\sum_{m=1}^{M}\sum_{n=1}^{N} x_n^m y_{n,m}^r \leq 1, \ \forall r \in \mathcal{R}, \tag{37b}$$

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{r=1}^{R} x_n^m y_{n,m}^r \leq R, \tag{37c}$$

$$y_{n,n}^r \in \{0,1\}, \ \forall n \in \mathcal{N}, m \in \mathcal{M}, r \in \mathcal{R}. \tag{37d}$$

Problem **P-2** has combinatorial nature and NP-hard for a large number of devices and resource blocks. To solve problem **P-2**, we use one-sided one-to-one matching. Our matching-based solution achieves local sub-optimal results with low complexity (Proof will be given in Theorem 1). On the other hand, for a fixed relative local accuracy and resource block matrices, the clustering sub-problem can be written as:

$$\mathbf{P}-\mathbf{3}: \underset{\mathbf{X}}{\text{minimize}} \ f_{\text{FL}}(\boldsymbol{X}) \tag{38}$$

$$\text{subject to:} \ \sum_{n=1}^{N} x_n^m \leq \triangle_m, \ \forall m \in \mathcal{M}, \tag{38a}$$

$$\sum_{m=1}^{M} x_n^m \leq 1, \ \forall n \in \mathcal{N}, \tag{38b}$$

$$x_n^m \in \{0,1\}, \ \forall n \in \mathcal{N}, m \in \mathcal{M}. \tag{38c}$$

Algorithm 2 DDFL Loss Minimization Algorithm

Inputs
Social tie matrix $\boldsymbol{\gamma}, \forall n \in \bar{\mathcal{N}}$
Euclidean distance matrix $\boldsymbol{\lambda}$.
Devices set $\bar{\mathcal{N}}$.
Social Matrix Computation
Compute the social similarity matrix $\boldsymbol{\Gamma}$.
Compute edge betweenness centrality matrix $\boldsymbol{\Upsilon}$.
Compute the Euclidean distance matrix $\boldsymbol{\Lambda}$.
Compute social matrix $\boldsymbol{\Omega}$ using (5).
Compute M cluster heads using social matrix.
repeat
  Relative Local Accuracy Minimization
  For fixed $\boldsymbol{X}$ and $\boldsymbol{Y}$, solve **P-1** via convex optimization.
  Resource Allocation
  Run *Alg. 3* to yield $\boldsymbol{Y}$ for fixed $\boldsymbol{X}$ and $\boldsymbol{\theta}$.
  Socially-Aware Clustering
  Run *Alg. 4* to yield $\boldsymbol{X}$ for fixed $\boldsymbol{Y}$ and $\boldsymbol{\theta}$.
  Compute f_{FL} for $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{\theta}$.
until f_{FL} converges.

Problem **P-3** has combinatorial nature and is difficult to solve using convex optimization schemes. We use a one-to-many matching game for computing the clustering matrix $\boldsymbol{X}$. The matching-based solution achieves a low complexity local sub-optimal result (Proof will be given in Theorem 2). Our proposed matching theory and optimization-based iterative algorithm is summarized in *Algorithm 2*. First, we compute the social matrix using various social parameters, such as the social similarity matrix, edge betweenness matrix, and physical distance matrix. Based on these matrices, we select M devices as cluster heads. These cluster heads will act for both sub-global and global aggregation. After the computation of cluster heads, we use an iterative approach to minimize the DDFL loss. First, relative local accuracy is minimized for a fixed resource allocation and clustering matrices. Then, the resource allocation sub-problem is solved using a one-to-one matching game for a fixed relative local accuracy and clustering matrices. Finally, the clustering sub-problem is solved for a fixed relative local accuracy and resource allocation matrices. This process continues in an iterative manner until convergence.

A. Relative Local Accuracy Minimization Algorithm

In this sub-section, we propose relative local minimization algorithm. For a given $\boldsymbol{X}$ and $\boldsymbol{Y}$ in problem $\mathbf{P}-\mathbf{1}$, $C_{FL}(\boldsymbol{\theta})$ is a convex optimization problem. We can use the standard convex optimization toolkits, e.g., CVXPY to solve **P - 1**.

Lemma 1: For a given $\boldsymbol{X}$ and $\boldsymbol{Y}$, $\mathbf{P}-\mathbf{1}$ is a convex optimization problem.

Proof: We first prove the convexity of the objective function $f_{\text{FL}}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Then, we prove the convexity of the feasible region. We can notice that $f_{\text{FL}}(\boldsymbol{\theta})$ is linear function in θ. Therefore, it is a convex function for $\theta_{\min} \leq \theta \leq \theta_{\max}$. Moreover, the constraints in (36a) and (36b) are

linear constraints. Therefore, (36) is a convex optimization problem. ■

B. Matching-Based Resource Allocation

In this sub-section, we propose a matching theory-based resource allocation algorithm for the allocation of resource blocks to end devices. A matching game can be either one-sided or two-sided depending on the preferences of players. In a two-sided matching game, both players rank each other in their preference lists according to some utility functions, whereas one-sided matching has only one preference list for one of the players [38]. In this paper, we use a one-sided one-to-one matching game for the allocation of resource blocks to the devices because of its near to optimal and low complexity solution [21]. The resource allocation problem shows similarity with the house allocation problem (HAP) denoted by a tuple (preferences of agents, houses set, and agent set) [21], [39], [40]. In problem **P-2**, the sets of devices and resource blocks are equivalent to houses and agents, respectively. A one-sided game for problem **P-2** assigns devices to resource blocks and is defined as follows.

Definition 1: A matching μ is a function from the set $\mathcal{N} \cup \mathcal{R}$ into the set of elements of $\mathcal{N} \cup \mathcal{R}$ such that

1) $|\mu(r)| \leq 1$ and $\mu(r) \in \mathcal{N}$,
2) $|\mu(n)| \leq 1$ and $\mu(n) \in \mathcal{R} \cup \phi$,
3) $\mu(r) = n$ if and only if r is in $\mu(n)$,

where $\mu(r) = n \Leftrightarrow \mu(n) = r$ for $\forall n \in \mathcal{N}, \forall r \in \mathcal{R}$ and $|\mu(.)|$ denotes the cardinality of matching outcome. The intuition of properties (1) and (2) is because of the constraints (37b) and (37a) that restricts the assignment of a resource block to a maximum of one device and assignment of a device to a maximum of one resource block, respectively. Next, we define the preference profile for our one-sided matching game. The preference profile $\mathcal{P}_r$ for resource blocks for a fixed clustering matrix $\boldsymbol{X}$ and relative local accuracy matrix $\boldsymbol{\theta}$ is based on the following function:

$$\mathcal{U}_r(n) = (1 + \theta_n) e_{n,m}. \tag{39}$$

The summary of the proposed resource allocation algorithm is given in *Algorithm 3*. The two main phases in *Algorithm 3* are the matching phase and initialization phase. In *initialization phase*, $\mathcal{P}_r$ is computed for a fixed clustering matrix $\boldsymbol{X}$ and relative local accuracy matrix $\boldsymbol{\theta}$ (line-7). In *matching phase*, every resource block proposes a device according to the preference profile (line 12). The proposed device is allocated the resource block if the given resource block is not assigned to another device or newly proposed device have a higher preference than the existing one and then, existing devices are rejected (lines 13-16). The newly proposed device is rejected if its preference is less than the existing one and the rejection list is updated (line 18). Finally, the preference list is updated by removing the rejected devices. The above process continues until stable matching is obtained.

Theorem 1: The one-sided matching μ produces local sub-optimal result for resource allocation problem **P-2**.

Proof: To prove this theorem, we use contradictions. The outcome of *Algorithm 3* is the matrix $\mu^{(t)} \mapsto \boldsymbol{Y}^{(t)}$ and want to minimize the DDFL cost f_{FL}. The $f_{\mathrm{FL}}^{(t)} \leq f_{\mathrm{FL}}^{(t-1)}$ is guaranteed by the matching *Algorithm 3*, because it is based on defer/acceptance. For a resource allocation matrix with binary values, the cost f_{FL} has non-increasing nature. Other than the non-increasing nature of the cost f_{FL} for the proposed algorithm, resource allocation matrix $\boldsymbol{Y}$ might not be optimal. The reason for this can be due to fact that any device can be at the same priority for more than one resource block. Therefore, it is not always possible that the resource block associated with the device will be the one with a lower cost f_{FL} compared to the other resource block. Based on the aforementioned observations, we can say that *Algorithm 3* produces sub-optimal results. ■

Algorithm 3 Matching-Based Resource Allocation Algorithm

1: ***Inputs***
2: Resource blocks preference profile $\mathcal{P}_r, \forall r \in \mathcal{R}$
3: Resource blocks set $\mathcal{R}$, devices set $\mathcal{N}$
4: ***Output***
5: Matching function $\mu^{(t)}$
6: ***Step 1: Initialization***
7: ${\mathcal{P}_r}^{(0)} = \mathcal{P}_r, \forall r \in \mathcal{R}$
8: $\mu^{(t)} \triangleq \{\mu(r)^{(t)}, \mu(n)^{(t)}\}_{r \in \mathcal{R}, n \in \mathcal{N}} = \emptyset$, ${\mathcal{L}_r}^{(t)} = \emptyset$, $t = 0$
9: ***Step 2: Matching phase***
10: **repeat**
11: $t \leftarrow t + 1$
12: **for** $r \in \mathcal{R}$, propose n according to $\mathcal{P}_r^{(t)}$ **do**
13: **if** $n \succ_r \mu(r)^{(t)}$ **then**
14: $\mu(r)^{(t)} \leftarrow \mu(r)^{(t)} \setminus n'$
15: $\mu(r)^{(t)} \leftarrow n$
16: $\mathcal{P}' r^{(t)} = \{n' \in \mu(r)^{(t)} | n \succ_r n'\}$
17: **else**
18: $\mathcal{P}'' r^{(t)} = \{n \in \mathcal{N} | \mu(r)^{(t)} \succ_r n\}$
19: **end if**
20: ${\mathcal{L}_r}^{(t)} = \{\mathcal{P}' r^{(t)}\} \cup \{\mathcal{P}'' r^{(t)}\}$
21: **for** $l \in {\mathcal{L}_r}^{(t)}$ **do**
22: ${\mathcal{P}_r}^{(t)} \leftarrow {\mathcal{P}_r}^{(t)} \setminus \{l\}$
23: **end for**
24: **end for**
25: **until** $\mu^{(t)} = \mu^{(t-1)}$

C. Matching-Based Clustering Algorithm

In this sub-section, we present a matching theory-based clustering algorithm for devices. First of all, cluster heads are selected based on the criteria of social awareness (already explained in Section IV). Then, the clusters are formed using a one-sided one-to-many matching game. In our model, all the clusters are made by the association of the central device with a fixed number of devices. Therefore, we use a one-to-many matching game to enable clustering. The problem **P-3** is similar to problem **P-2**, except for the number of devices that can be associated with a central device to enable clustering. Therefore, we can modify the house allocation problem to enable matching in problem **P-3**. The reason for using a matching game-based solution is due to its near to optimal

and low complexity solution [21]. In problem **P-3**, a cluster head can be associated with many devices restricted by limit Δ_m, in such a way to minimize the global FL cost.

Definition 2: A one-to-many matching μ is a function from the set $\mathcal{N} \cup \mathcal{M}$ into the set of elements of $\mathcal{N} \cup \mathcal{M}$ such that

1) $|\mu(n)| \leq 1$ and $\mu(n) \in \mathcal{M}$,
2) $|\mu(m)| \leq q_m$ and $\mu(m) \in 2^{\mathcal{N}} \cup \phi$,
3) if n $\in \mu(m)$ then $\mu(n) = m$,

where $\mu(m) = n \Leftrightarrow \mu(n) = m$ for $\forall n \in \mathcal{N}, \forall m \in \mathcal{M}$ and $|\mu(.)|$ denotes the cardinality of matching outcome. To use one-sided one-to-many matching game, we first define a preference profile for end-devices. For a fixed resource allocation matrix $\boldsymbol{Y}$ and relative local accuracy matrix $\boldsymbol{\theta}$, the preference profile for end-devices is based on the following function:

$$\mathcal{U}_n(m) = (1 + \theta_n) e_{n,m}. \tag{40}$$

The summary of our proposed clustering algorithm is given in *Algorithm 4*. Similar to the matching-based resource allocation algorithm, the clustering algorithm has two main phases, such as initialization and matching phase. In *initialization phase*, a preference profile for devices is created for a fixed resource allocation matrix $\boldsymbol{Y}$ and relative local accuracy $\boldsymbol{\theta}$ (line 7). In *matching phase*, each device proposes a cluster head (central device) according to its preference profile. If the device is not matched already, then the number of devices associated with the suggested cluster head is checked. The new device is associated if the cluster head has a fewer number of devices than its quota, otherwise, the preference of the newly suggested devices is checked with the existing matched devices. This iterative process continues until a stable matching is obtained for all devices.

Theorem 2: The one-sided one-to-many matching μ produces local sub-optimal result for clustering sub-problem **P-3**.

Proof: To prove this theorem, we use contradictions. The outcome of *Algorithm 4* is the matrix $\mu^{(t)} \mapsto \boldsymbol{X}^{(t)}$ and want to minimize the DDFL loss f_{FL}. The $f_{\mathrm{FL}}^{(t)} \leq f_{\mathrm{FL}}^{(t-1)}$ is guaranteed by the matching, because it is based on defer/acceptance. For a fixed resource allocation matrix and relative local accuracy, the loss f_{FL} has non-increasing nature for binary values of clustering matrix $\boldsymbol{X}$. Other than the non-increasing nature of the loss f_{FL} for the proposed algorithm, the clustering matrix $\boldsymbol{X}$ might not be optimal. The reason for this can be due to fact that any device can be at the same priority for more than one cluster. Subsequently, our matching-based clustering algorithm will select one cluster for the device for which the loss might be little more than the other cluster. Therefore, it is not always possible that the device is associated with the cluster will be the one with a lower loss f_{FL} compared to the other cluster. Based on the aforementioned observations, we can say that *Algorithm 4* produces sub-optimal results. ■

Theorem 3: Algorithm 3 and Algorithm 4 results in stable resource allocation and stable clustering, respectively.

Proof: To prove the stability of both *Algorithm 3* and *Algorithm 4*, we will use notion of blocking pairs in matching games. A matching is said to be stable if there is not blocking pair (*n*, *m*) such that $m \succ_n \mu(n)$ and $n \succ_m \mu(m)$, where $\mu(n)$ and $\mu(m)$ are current matched pairs of *n* and *m*, respectively. For *Algorithm 4*, let the new matching $\mu(n)^{t+1}$ is unstable then it will be replaced by a new stable matching (i.e., blocking pair) that will further minimize the DDFL loss function. There can be two cases.

- If the quota is not fulfilled, then unstable matching $\mu(n)^{t+1}$ will be replaced by at least one blocking pair without affecting the previously matched pairs.
- If the quota is fulfilled, then new stable matching will replace the existing matching $\mu(n)^t$ pair only if it minimizes the DDFL loss more than the matching $\mu(n)^t$.

Based on the aforementioned notion of the existence of blocking pairs, one can say that both *Algorithm 3* and *Algorithm 4*

Algorithm 4 Matching-Based Clustering Algorithm

1: ***Inputs***
2: Devices preference profile $\mathcal{P}_n, \forall n \in \mathcal{N}$
3: Central devices set $\mathcal{M}$, devices set $\mathcal{N}$
4: ***Output***
5: Matching function $\mu^{(t)}$
6: ***Step 1: Initialization***
7: ${\mathcal{P}_n}^{(0)} = \mathcal{P}_n, \forall n \in \mathcal{N}$
8: $\mu^{(t)} \triangleq \{\mu(n)^{(t)}, \mu(m)^{(t)}\}_{n \in \mathcal{N}, m \in \mathcal{M}} = \emptyset$, ${\mathcal{K}_m}^{(t)} = \emptyset$, $t = 0$
9: ***Step 2: Matching phase***
10: **repeat**
11: $t \leftarrow t + 1$
12: **for** $n \in \mathcal{N}$, propose *m* according to $\mathcal{P}_n^{(t)}$ **do**
13: **while** $m \notin \mu(n)^{(t)}$ and $\mathcal{P}_n^{(t)} \neq \emptyset$ **do**
14: **if** $q_m^{(t)} \geq l_n^m$ **then**
15: $\mu(m)^{(t)} \leftarrow \mu(m)^{(t)} \cup n$
16: $q_m^{(t)} \leftarrow q_m^{(t)} - l_n^m$
17: **else**
18: ${\mathcal{K}'}_m^{(t)} = \{n' \in \mu(m)^{(t)} | n \succ_m n'\}$
19: $n_{lp} \leftarrow$ the least preferred $n' \in {\mathcal{K}'}_m^{(t)}$
20: **while** ${\mathcal{K}'}_m^{(t)} \neq \emptyset \cup q_m^{(t)} \leq l_n^m$ **do**
21: $\mu(m) \leftarrow \mu(m) \setminus \{n'\}$
22: ${\mathcal{K}'}_m^{(t)} \leftarrow {\mathcal{K}'}_m^{(t)} \setminus \{n_{lp}\}$
23: $q_m^{(t)} = q_m^{(t)} + l_n^m$
24: $n_{lp} \leftarrow$ the least preferred $n' \in {\mathcal{K}'}_m^{(t)}$
25: **end while**
26: **if** $q_m^{(t)} \geq l_n^m$ **then**
27: $\mu(m) \leftarrow \mu(m) \cup \{n\}$
28: $q_m^{(t)} = q_m^{(t)} - l_n^m$
29: **else**
30: $n_{lp} \leftarrow n$
31: **end if**
32: $\mathcal{K}_m^{(t)} = \{n \in \mathcal{N} | n_{lp} \succ_m n\} \cup \{n_{lp}\}$
33: **for** $k \in \mathcal{K}_m^{(t)}$ **do**
34: $\mathcal{P}_n \leftarrow \mathcal{P}_n \setminus k$
35: **end for**
36: **end if**
37: **end while**
38: **end for**
39: **until** $\mu^{(t)} = \mu^{(t-1)}$

will result in stable resource allocation and stable clustering, respectively. ■

D. Complexity Analysis

We use Big $\mathcal{O}$ notation to analyze the complexity of the proposed scheme for the DDFL framework. As the cluster head selection is performed once, and therefore, we first analyze its computation complexity. The complexity of cluster heads mainly depends on edge betweenness and social similarity matrix complexity. Betweenness centrality can be computed using many techniques with a reasonable computational complexity [32], [41]. Additionally, the social similarity matrix can be easily computed by first finding the common nodes and then, using their degrees as given in (3). Therefore, finding the cluster for our proposal has a reasonable complexity and it is computed once. Next, for the iterative scheme, such as *Algorithm 2*, we individually compute the computational complexity of clustering, resource allocation, and relative local accuracy schemes for one global round. For both one-sided and two-sided matching-based schemes, there is a need to first compute the complexity of their preference profiles. The complexity of constructing preference profiles is that of any standard sorting algorithm because of their sorting operations. For one-sided one-to-one matching-based resource allocation, the complexity of constructing a preference profile for a resource block *r* using *N* devices is given by $\mathcal{O}(N\log(N))$ [12]. Similarly, for one-sided one-to-many matching-based clustering, the complexity of computing preference profile for a device *n* for a *M* cluster heads is given by $\mathcal{O}(M\log(M))$. *Algorithm 3* terminates after a finite number of iterations [42]. For the worst case, complexity is $\mathcal{O}(NR)$. Similar to *Algorithm 3*, *Algorithm 4* also terminates after a finite number of iterations and the worst-case complexity is $\mathcal{O}(NM)$. On the other hand, CVX optimizer for solving sub-problem **P-1** generally converges fast after a finite number of iterations for problems whose solution exists [43], [44]. Overall, the proposed iterative scheme *Algorithm 2* will converge after finite iterations with reasonable complexity. Although the proposed DDFL scheme can provide us with a robust operation because of its decentralization nature, this will be at the cost of complexity in finding the cluster heads. Additionally, sharing the sub-global models between the cluster heads will use extra communication resources compared to conventional FL. Therefore, one must appropriately choose the number of clusters for DDFL. For instance, consider DDFL with one sub-global iteration and conventional FL for *N* devices and *M* cluster heads (only for DDFL). For conventional FL, we have only one aggregation server in contrast to DDFL with *M* cluster heads acting for sub-global aggregations. The number of communication links required for transferring the local learning models to cluster heads are *N* links. Additionally, cluster heads in DDFL will require $M(M-1)$ communication links for transferring the sub-global models. The total communication links in DDFL are $N + M(M-1)$. On the other hand, conventional FL will use *N* communication links prior to global model aggregation.

TABLE III
SIMULATION PARAMETERS [45]

Simulation Parameter	Value
Total area	$1000 \times 1000m^2$
Cellular users	50
Devices	50
Frame Structure	Type 1 (FDD)
Carrier frequency (f)	2 GHz
Devices transmit power	23 dBm
Sub carriers per resource block	12
Resource block bandwidth (W)	180 kHz
Thermal noise for 1 Hz at 20. C	-174 dBm

VI. PERFORMANCE EVALUATION

In this section, we present numerical results to evaluate the performance of our proposed DDFL scheme for different scenarios. Simulation parameters used for numerical analysis are given in Table III. We consider an area of 1000×1000, where devices are randomly deployed. Furthermore, we consider two baseline schemes for performance comparison with the proposed DDFL scheme. Baseline-1 considers the proposed clustering algorithm and relative local accuracy with random resource allocation, whereas baseline-2 uses proposed resource allocation and relative local accuracy with random clustering.

A. Performance Analysis for DDFL Loss Minimization

In this sub-section, we analyze the performance of our proposed DDFL loss minimization algorithm for different scenarios. We use the keyword *performance gain* that is equal to $(1/f_{\mathrm{FL}})$ for performance evaluation. Fig. 6a shows the performance gain vs. iterations for proposed, baseline-1, and baseline-2 schemes. It is clear from Fig. 6a that the proposed algorithm outperforms both baseline-1 and baseline-2. Baseline-1 outperforms baseline-2 due to the fact that clustering has more impact on DDFL performance gain than resource allocation. In Fig. 6b, performance gain vs. clusters for 40 devices for various schemes is shown. Our proposed algorithm outperforms baseline-1 and baseline-2 for various clusters for 40 devices. The DDFL loss function depends on SINR and local devices' performance. SINR depends on clustering and resource allocation, whereas relative accuracy generally depends on end-devices computational resources resource with backup energy usage for a particular local dataset and local learning model. Therefore, the reason for performance improvement is due to fact that the proposed scheme considered relative local accuracy minimization (local accuracy maximization), efficient clustering, and resource allocation. Similar to Fig. 6a, Fig. 6b confirms the order of performance for different schemes. On the other hand, Fig. 6c shows performance gain vs. devices for fixed 4 clusters. Various numbers of devices for a fixed number of clusters of 4 confirms the performance trend of Fig. 6a and Fig. 6b.

B. Learning Performance Analysis of DDFL for Real-Time Dataset

In this sub-section, we consider three types of data distributions among the devices such as group-IID, group non-IID, and non-IID for the MNIST dataset.

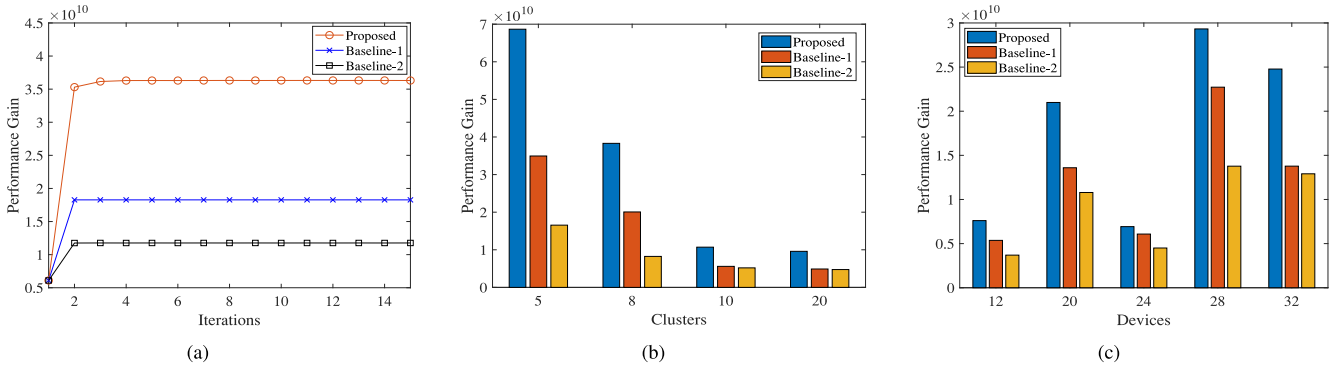

Fig. 6. (a) Performance gain vs. iterations for 50 devices and 5 clusters, (b) Performance gain vs. clusters for 40 devices, (c) Performance gain vs. devices for 4 clusters.

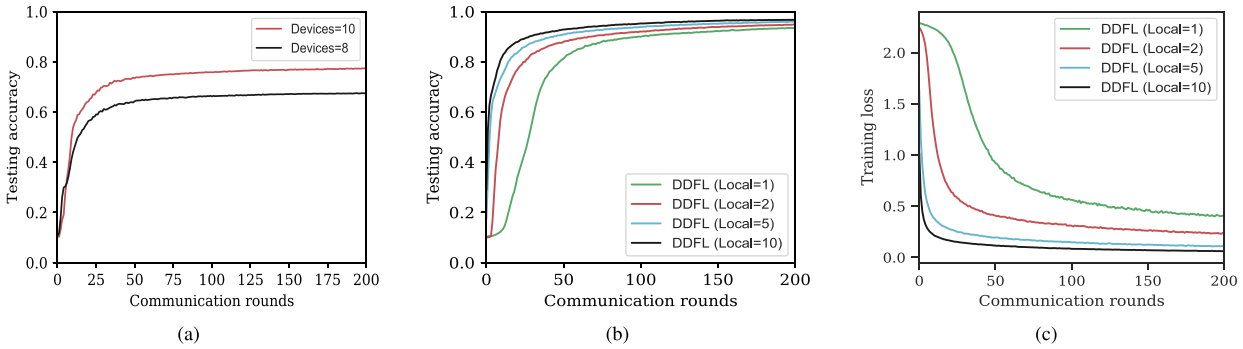

Fig. 7. (a) Accuracy vs. communication rounds for 10 and 8 devices for non-IID data, (b) DDFL accuracy v.s communication rounds for non-IID data and 5 clusters with 10 devices each, (c) DDFL training loss v.s communication rounds for non-IID data and 5 clusters with 10 devices each.

- *Group-IID:* For the group-IID case, we assign a single class of images to every device within groups of 10 devices each [23].
- *Group non-IID:* Group non-IID case deals with the two groups (each of 5 devices) having image classes of 0, 1, 2, 3, 4 and 5, 6, 7, 8, 9 [24].
- *Non-IID:* For the non-IID case, we divide the whole data into 200 shards of 300 images each and assign 2 shards to every device [7].

The data distribution and aggregation scheme adopted mainly determine the performance of the DDFL [23], [24]. Here, in this sub-section, we consider modification of FedAvg for DDFL by performing averaging at two levels, such as sub-global and global level. The DDFL uses computing of multiple sub-global models by reusing resource blocks. This will enable the more devices to participate in the FL process. Generally, increasing the number of participating devices in FL causes performance improvement. Therefore, first, we analyze the effect of increasing the number of participating devices in FL. Fig. 7a shows testing accuracy vs. communication rounds for non-IID settings for 10 and 8 devices participation. It is clear from Fig. 7a that FL with 10 devices outperforms FL with 8 devices. Therefore, we can conclude that generally, the participation of more devices in FL is preferable for better performance.

Next, we discuss the performance of the DDFL framework for various scenarios. First, we use the sub-global iterations equal to 1 and FedAvg within groups used for sub-global model computation [23]. As mentioned earlier in Section III that there are two kinds of weight divergences in DDFL, such as sub-global level and global level. Therefore, simply averaging the might not work well. We consider two cases, first has one sub-global iteration (one-time sub-global aggregation before global aggregation) and multiple sub-global aggregations. For a single sub-global aggregation case, the result will be similar to conventional FedAvg. However, the communication will be less expensive due to the fact that only sub-global models will be sent to the global aggregation server after one-time sub-global aggregations. Figs. 7b and 7c shows that increasing the number of local iterations results in performance improvement for one sub-global aggregation case. However, this will be at the cost of local computation resources. One thing must be noted that even, if we consider 1 sub-global iteration, still there is efficient communication compared to centralized aggregation-based FL. For instance, communication with the nearby device by reusing the already occupied channel resources is less expensive than direct communication with the centralized aggregation server located at the remote cloud. In dispersed FL, if the centralized cloud-based aggregation is considered, still only cluster heads send their sub-global models rather than all the devices. Therefore, there will be fewer communicating devices with a remote cloud.

Next, we analyze the performance of DDFL using different data distributions for multiple sub-global aggregation in Figs. 8, 9, and 10. Additionally, we compare the performance of DDFL with that of conventional FL [7]. The number of

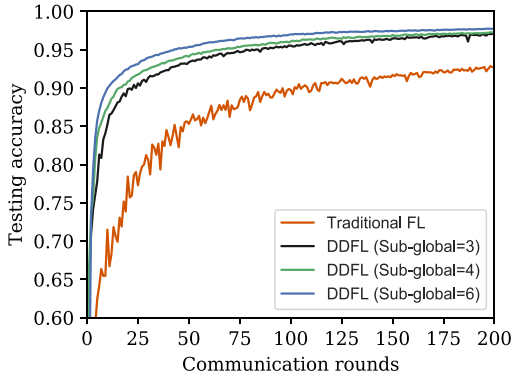

Fig. 8. Testing accuracy vs. communication rounds for group-IID data using $T = 12$ and 5 clusters with 10 devices each.

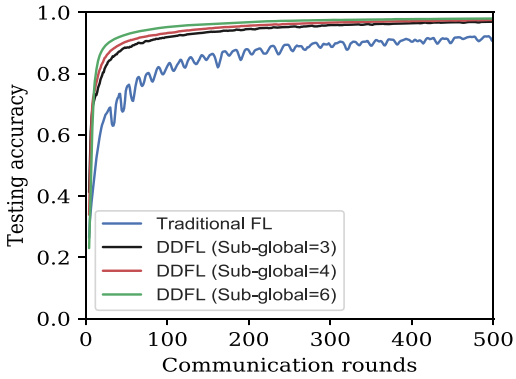

Fig. 9. Testing accuracy vs. communication rounds for group non-IID data using $T = 12$ and 2 clusters with 5 devices each.

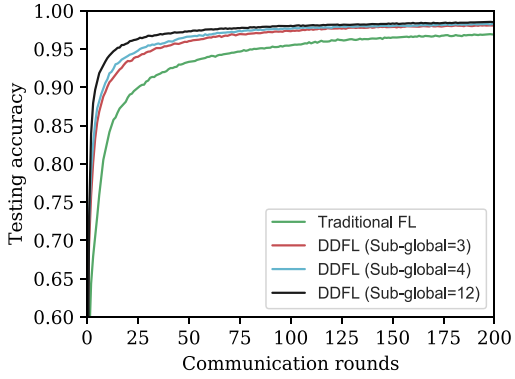

Fig. 10. Testing accuracy v.s communication rounds for non-IID data using $T = 12$ and 5 clusters with 10 devices each.

local iterations before a global aggregation takes place is the same for conventional FL and DDFL with various settings. For Fig. 8, we use T which is equal to the product of local iterations and sub-global aggregations. Fig. 8 shows the testing accuracy vs. communication rounds for group-IID case and $T = 12$. It is clear from Fig. 8 that the performance of the DDFL improves with an increase in the number of sub-global aggregations for a fixed value of $T = 12$. Furthermore, DDFL outperforms conventional FL for various configurations. On the other hand, Fig. 9 shows the testing accuracy of the group non-IID case for DDFL with sub-global iterations 3, 4, and 6 using a fixed value of $T = 12$. It is clear from Fig. 9 that 6 sub-global iterations significantly improves the performance of DDFL by allowing faster convergence than the DDFL with 3 sub-global iterations. Finally, we show the performance of DDFL for non-IID case in Fig. 10. The trend of results for various sub-global aggregations is same as the Figs. 8 and 9.

VII. Conclusion

In this paper, we have presented a socially-aware-clustering-enabled DDFL framework. The proposed DDFL framework can offer robustness, efficient communication resource usage, and privacy preservation. Therefore, we can say that DDFL can serve for the practical implementation of FL for various applications. Additionally, we formulated a joint relative local accuracy minimization, clustering, and resource allocation problem to minimize the DDFL loss function. Then, we propose an iterative solution to minimize the cost of DDFL. For relative local accuracy minimization, we have used a convex optimization solver, whereas for resource allocation we have used a one-to-one matching game. However, for clustering, we have used a one-to-many matching game. Finally, we have validated our proposal using numerical results.

References

[1] L. U. Khan, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, "6G wireless systems: A vision, architectural elements, and future directions," *IEEE Access*, vol. 8, pp. 147029–147044, 2020.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[3] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 96–103, Jun. 2020.

[4] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6G: Vision, architectural trends, and future directions," 2021. [Online]. Available: arXiv:2102.12169.

[5] *Federated Learning, a Step Closer Towards Confidential AI*. Accessed: Jan. 24, 2020. [Online]. Available: https://medium.com/frstvc/tagged/thoughts

[6] L. U. Khan *et al.*, "Federated learning for edge networks: Resource optimization and incentive mechanism," 2019. [Online]. Available: arXiv:1911.05642.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016. [Online]. Available: arXiv:1602.05629.

[8] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[9] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Security Privacy*, San Francisco, CA, USA, May 2019, pp. 739–753.

[10] L. U. Khan, W. Saad, Z. Han, and C. S. Hong, "Dispersed federated learning: Vision, taxonomy, and future directions," 2020. [Online]. Available: arXiv:2008.05189.

[11] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning in the Internet of Things," 2020. [Online]. Available: arXiv:2006.02499.

[12] S. M. A. Kazmi *et al.*, "Mode selection and resource allocation in device-to-device communications: A matching game approach," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3126–3141, Nov. 2017.

[13] 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 INFOCOM Conf. Comput. Commun.*, May 2019, pp. 1387–1395.

[14] 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. [Online]. Available: arXiv:1909.07972.

[15] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
[16] Y. Li, T. Wu, P. Hui, D. Jin, and S. Chen, "Social-aware D2D communications: Qualitative insights and quantitative analysis," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 150–158, Jun. 2014.
[17] Y. Zhao, Y. Li, Y. Cao, T. Jiang, and N. Ge, "Social-aware resource allocation for device-to-device communications underlaying cellular networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6621–6634, Dec. 2015.
[18] M. I. Ashraf, M. Bennis, W. Saad, M. Katz, and C.-S. Hong, "Dynamic clustering and user association in wireless small-cell networks with social considerations," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6553–6568, Jul. 2017.
[19] L. U. Khan, M. Alsenwi, Z. Han, and C. S. Hong, "Self organizing federated learning over wireless networks: A socially aware clustering approach," in *Proc. Int. Conf. Inf. Netw.*, Barcelona, Spain, Jan. 2020, pp. 453–458.
[20] S. M. A. Kazmi, N. H. Tran, T. M. Ho, and C. S. Hong, "Hierarchical matching game for service selection and resource purchasing in wireless network virtualization," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 121–124, Jan. 2018.
[21] A. K. Bairagi, N. H. Tran, W. Saad, Z. Han, and C. S. Hong, "A game-theoretic approach for fair coexistence between LTE-U and Wi-Fi systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 442–455, Jan. 2019.
[22] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning ACROSS heterogeneous cellular networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 8866–8870.
[23] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun.*, Dublin, Ireland, Jul. 2020, pp. 1–6.
[24] J. Wang, S. Wang, R.-R. Chen, and M. Ji, "Local averaging helps: Hierarchical federated learning and convergence analysis," 2020. [Online]. Available: arXiv:2010.12998.
[25] D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman, "A survey on secure data analytics in edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4946–4967, Jun. 2019.
[26] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.
[27] A. Moubayed, A. Shami, and H. Lutfiyya, "Power-aware wireless virtualized resource allocation with D2D communication underlaying LTE network," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.
[28] M. Ahmed, Y. Li, M. Waqas, M. Sheraz, D. Jin, and Z. Han, "A survey on socially aware device-to-device communications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2169–2197, 3rd Quart., 2018.
[29] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," 2020. [Online]. Available: arXiv:2009.13012.
[30] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proc. 8th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Sep. 2007, pp. 32–40.
[31] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, Oct. 2009.
[32] P. De Meo, E. Ferrara, G. Fiumara, and A. Ricciardello, "A novel measure of edge centrality in social networks," *Knowl. Based Syst.*, vol. 30, pp. 136–150, Jun. 2012.
[33] S. Narayanan, "The betweenness centrality of biological networks," Ph.D. dissertation, Dept. Comput. Sci., Virginia Polytechn. Inst. State Univ., Blacksburg, VA, USA, 2005.
[34] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016. [Online]. Available: arXiv:1610.02527.
[35] Y. Xi, A. Burr, J. Wei, and D. Grace, "A general upper bound to evaluate packet error rate over quasi-static fading channels," *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1373–1377, May 2011.
[36] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. A1380–A1405, May 2012.
[37] 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, May 2020.
[38] S. Bayat, Y. Li, L. Song, and Z. Han, "Matching theory: Applications in wireless communications," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 103–122, Nov. 2016.
[39] L. Zhou, "On a conjecture by gale about one-sided matching problems," *J. Econ. Theory*, vol. 52, no. 1, pp. 123–135, Oct. 1990.
[40] A. Abdulkadiroğlu and T. Sönmez, "Random serial dictatorship and the core from random endowments in house allocation problems," *Econometrica*, vol. 66, no. 3, pp. 689–701, May 1998.
[41] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.
[42] M. Hasan and E. Hossain, "Distributed resource allocation in 5G cellular networks," in *Towards G*, vol. 5. Hoboken, NJ, USA: Wiley, 2014, pp. 129–161.
[43] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[44] M. Grant, S. Boyd, and Y. Ye, "*CVX: MATLAB Software for Disciplined Convex Programming*, CVX, Austin, TX, USA, 2008.
[45] *Evolved Universal Terrestrial Radio Access (E-UTRA): Physical Layer Procedures, Release 11*, 3GPP Standard TS 36.213, Dec. 2012.

Latif U. Khan received the M.S. degree (with Distinction) in electrical engineering from the University of Engineering and Technology (UET), Peshawar, Pakistan in 2017. He is currently pursuing the Ph.D. degree in computer science and engineering with Kyung Hee University (KHU), South Korea. He is working as a leading Researcher with the intelligent Networking Laboratory under a project jointly funded by the prestigious Brain Korea 21st Century Plus and Ministry of Science and ICT, South Korea. Prior to joining the KHU, he has served as a Faculty Member and as a Research Associate with UET, Peshawar, Pakistan. He has authored/coauthored two conference best paper awards. He has also authored two books, such as *Network Slicing for 5G and Beyond Networks* and *Federated Learning for Wireless Networks*. His research interests include analytical techniques of optimization and game theory to edge computing, end-to-end network slicing, and federated learning for wireless networks.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively. From 2000 to 2002, he was an Research and Development Engineer with JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department as well as with the Computer Science Department, University of Houston, Houston, TX, USA. He is also a Chair Professor with National Chiao Tung University, Taiwan. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received the NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *EURASIP Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS) in 2016, and several best paper awards in IEEE conferences. He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018, and has been an AAAS Fellow since 2019 and an ACM Distinguished Member since 2019. He is 1% highly cited researcher since 2017 according to Web of Science.

Dusit Niyato (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has published more than 400 technical articles in the area of wireless and mobile computing. He received the Best Young Researcher Award of the IEEE Communications Society Asia Pacifica and the 2011 IEEE Communications Society Fred W. Ellersick Prize Paper Award. He is also serving as a Senior Editor of the IEEE WIRELESS COMMUNICATION LETTERS, an Area Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS, and an Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He was a Distinguished Lecturer of the IEEE Communications Society from 2016 to 2017. He was named as a highly cited researcher in computer science.

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, Gyeonggi, South Korea, where he was involved in broadband networks as a Member of the Technical Staff. Since 1993, he has been with Keio University. He was with the Telecommunications Network Laboratory, KT, as a Senior Member of Technical Staff and as 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. His research interests include future Internet, intelligent edge computing, network management, and network security. He has served as the General Chair, the TPC Chair/Member, or an Organizing Committee Member of international conferences, such as the Network Operations and Management Symposium, International Symposium on Integrated Network Management, Asia–Pacific Network Operations and Management Symposium, End-to-End Monitoring Techniques and Services, IEEE Consumer Communications and Networking Conference, Assurance in Distributed Systems and Networks, International Conference on Parallel Processing, Data Integration and Mining, World Conference on Information Security Applications, Broadband Convergence Network, Telecommunication Information Networking Architecture, International Symposium on Applications and the Internet, and International Conference on Information Networking. He was an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He currently serves as an Associate Editor for the *International Journal of Network Management*. He is a member of the Association for Computing Machinery, the Institute of Electronics, Information and Communication Engineers, the Information Processing Society of Japan, the Korean Institute of Information Scientists and Engineers, the Korean Institute of Communications and Information Sciences, the Korean Information Processing Society, and the Open Standards and ICT Association.