Future Generation Computer Systems 128 (2022) 406–419

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data

Alaa Awad Abdellatif [a], Naram Mhaisen [a], Amr Mohamed [a], Aiman Erbad [b,*], Mohsen Guizani [a], Zaher Dawy [c], Wassim Nasreddine [d]

[a] College of Engineering, Qatar University, Qatar
[b] College of Science and Engineering, Hamad Bin Khalifa University, Qatar
[c] Electrical and Computer Engineering Department, American University of Beirut, Lebanon
[d] Department of Neurology, American University of Beirut, Lebanon

ARTICLE INFO

Article history:
Received 16 August 2021
Received in revised form 8 October 2021
Accepted 18 October 2021
Available online 28 October 2021

Keywords:
Distributed deep learning
Edge computing
Non-IID data
Internet of Things (IoT)
Intelligent health systems

ABSTRACT

Federated Learning (FL) is a distributed learning methodology that allows multiple nodes to cooperatively train a deep learning model, without the need to share their local data. It is a promising solution for telemonitoring systems that demand intensive data collection, for detection, classification, and prediction of future events, from different locations while maintaining a strict privacy constraint. Due to privacy concerns and critical communication bottlenecks, it can become impractical to send the FL updated models to a centralized server. Thus, this paper studies the potential of hierarchical FL in Internet of Things (IoT) heterogeneous systems. In particular, we propose an optimized solution for user assignment and resource allocation over hierarchical FL architecture for IoT heterogeneous systems. This work focuses on a generic class of machine learning models that are trained using gradient-descent-based schemes while considering the practical constraints of non-uniformly distributed data across different users. We evaluate the proposed system using two real-world datasets, and we show that it outperforms state-of-the-art FL solutions. Specifically, our numerical results highlight the effectiveness of our approach and its ability to provide 4–6% increase in the classification accuracy, with respect to hierarchical FL schemes that consider distance-based user assignment. Furthermore, the proposed approach could significantly accelerate FL training and reduce communication overhead by providing 75–85% reduction in the communication rounds between edge nodes and the centralized server, for the same model accuracy.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rapid evolution of Artificial Intelligence (AI), Internet of Things (IoT) [1], and big data is paving the path towards a plethora of interactive applications that can inspire substantial transformations in the industrial services. However, these new trends come with several challenges due to the latency and reliability constraints, and the need to process and transmit enormous amount of data, generated by IoT devices, with guaranteed quality of service. Emerging AI technologies, such as deep learning and active learning, can turn the vision of building real-time interactive systems into reality [2,3]. For instance, in light of the recent pandemic, deep learning has proven its crucial role in analyzing and understanding the outbreaks evolution. Indeed, efficient deep learning techniques can provide intelligent healthcare services, such as event detection and characterization, real-time remote monitoring, and identification of patients with high mortality risks. However, deep learning relies on the availability of large datasets, provided by medical and IoT devices, for training and improving deep learning models [4]. This puts a major challenge on maintaining data privacy, since medical devices are typically collecting private data. Hence, it is commonly not a practical solution to forward this data to a centralized entity for conducting the training [5]. Centralized processing of massive amount of data puts users' privacy at high risk, while imposing enormous load on wireless networks. Thus, we envision that bringing the intelligence close to the IoT devices, using Multi-access Edge Computing (MEC) [6,7] along with Federated Learning (FL) [8], is a key for supporting data privacy, while jointly training large deep learning models.

FL is a promising solution for training deep learning models without storing the data at a centralized location. It allows

* Corresponding author.
E-mail addresses: alaa.abdellatif@ieee.org (A.A. Abdellatif), naram@qu.edu.qa (N. Mhaisen), amrm@qu.edu.qa (A. Mohamed), aerbad@ieee.org (A. Erbad), mguizani@qu.edu.qa (M. Guizani), zd03@aub.edu.lb (Z. Dawy), wn13@aub.edu.lb (W. Nasreddine).

https://doi.org/10.1016/j.future.2021.10.016
0167-739X/© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

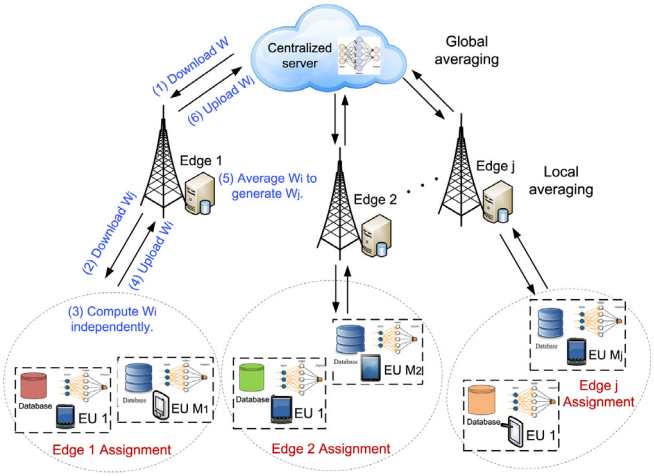


Fig. 1. Hierarchical federated learning: architecture and data flow.

multiple entities/users to jointly train a deep learning model on their combined data, without revealing their data to a centralized entity. This approach of privacy-preserving cooperative learning is carried out by three main steps: (i) all participating users receive the latest global model $\boldsymbol{W}$ from the centralized server (also called a broker); (ii) they train the received model using their local data; and (iii) they upload their locally trained models $\boldsymbol{W}_i$ back to the centralized server to be aggregated and form an updated global model. These steps are repeated until a certain convergence criterion is obtained.

Following FL protocol, the local devices (or users) never transfer their local data, since only the updated models are communicated. However, a major challenge in centralized FL is the massive communication overhead that arises from sending the End Users (EUs) their full models' update to the centralized server at each iteration. Such updates are of the same size as the trained model, which can be in the range of gigabytes for deep learning models with thousands of parameters [9]. Moreover, frequent exchange of the updated models with the centralized server leads to increasing the FL convergence time, especially with non-Independent and Identically Distributed (non-IID) data [10]. Accordingly, in large scale networks, centralized FL can generate high communication overheads (in terms of latency, network bandwidth, and energy consumption), which turns the FL to be unproductive in resource-constrained environments.

To address the above challenges, we propose an optimized EUs assignment and resource allocation scheme over hierarchical federated learning architecture (see Fig. 1), in which multiple edge servers are employed to reduce the overhead resulting from information exchange between EUs and the centralized server. In the proposed hierarchical architecture, EUs are assigned to different edge nodes not only based on their locations (like the state-of-the-art solutions), but also based on their data distributions and communication constraints. Based on this, the EUs send their local updates to the selected edge node (e.g., small cell base station) for aggregation. Then, the edge node computes the aggregate model and transfers it back to their assigned EUs to update their models accordingly. After a specific number of iterations, the edge nodes forward their models' updates to the centralized server to maintain a global model. In this context, we propose a novel EUs assignment and resource allocation scheme that aims to minimize the FL convergence time, hence the consumed resources, by optimally assigning the EUs with different data distributions to the available edge nodes.

Without loss of generality, this paper considers, as a case study, applying the proposed hierarchical FL in large-scale intelligent healthcare systems. These systems include thousands of participants who need continuous monitoring at the same time, especially in case of pandemics. This entails major challenges and puts significant load on different healthcare entities. Thus, the strategic solution is to move large number of patients with mild symptoms into remote monitoring or home care. Deep learning models can enable home care services by acquiring and processing the patients' data to identify the patients' states. However, collecting and forwarding such private data from the patients to a central entity come at a risk of violating the patients' privacy. Thus, we envision that leveraging the proposed hierarchical FL within large-scale intelligent healthcare systems is a key for enabling remote monitoring. In this context, the main contributions of our work can be summarized as:

1. Design a distributed learning system with hierarchical FL for supporting intelligent healthcare services. In the hierarchical FL setting, the EUs are collaborating to train a deep learning model, without exchanging raw data, by connecting to the local edge nodes in their communication range, which are then connected to a centralized server.
2. Study the global imbalanced data effect on the obtained accuracy and convergence time of the deep learning models trained by hierarchical FL. This includes formulating an optimization problem for EUs assignment and resource allocation based on EUs' data distributions, while considering latency, bandwidth, and energy consumption constraints.
3. Solving the formulated problem using an efficient approach that considers both data distributions of different EUs and wireless environment conditions.

The developed approach has been tested extensively via comprehensive experiments using real-world datasets. Our results confirm that the proposed approach provides near-optimal performance for different data distributions and system configurations.

In the rest of the paper, Section 2 presents the related work on single-layer FL and hierarchical FL. Section 3 introduces the main challenges that will be tackled in this paper, and presents the considered intelligent healthcare system architecture with hierarchical FL. Section 4 introduces first the mathematical modeling of hierarchical FL. Then, it presents the communication and computation latency as well as energy consumption analysis of the proposed hierarchical FL framework. Section 5 presents the formulated optimization framework along with the solution for EUs assignment and resource allocation over hierarchical FL. Performance evaluation of our system is then discussed in Section 6. Finally, the paper is concluded in Section 7.

2. Related work

The proposed FL schemes in the literature can be broadly classified, based on the system architecture, into two categories: Single-layer FL and hierarchical FL.

2.1. Single-layer Federated Learning

The concept of FL was first proposed in [11], with its efficiency demonstrated through experiments on different datasets. The presented model in [11] considered a single-layer FL, where the users exchanged their updated models with a centralized server that aggregated and formed an updated global model with a fixed frequency. This was followed by several extensions, which can be categorized into:

- analyzing the convergence of distributed gradient descent and federated learning algorithms from a theoretical perspective, and optimizing the learning process given computation and communication resource budgets [8,10,12,13];
- selecting the users participating in the synchronization process of FL approaches [14–17];
- developing communication-efficient techniques to reduce the amount of exchanged data in FL systems by adopting various sparsification and compression techniques [5,18,19].

The effect of non-IID on the performance of FL has been studied in [12]. It has been shown, theoretically and empirically, that highly skewed non-IID data can significantly reduce the accuracy of the global learning model by up to 55%. As a solution to enhance the training on non-IID data, the authors proposed to share globally a small subset of data between all users. Combining this data with the local data of each user turns it to be less biased or skewed. However, exchanging data between different users is not always feasible due to the privacy constraint and communication overhead. In [10], the authors analyzed the convergence rate of Federated Averaging (FedAvg) on non-IID data for strongly convex and smooth problems. In [8], the authors studied the adaptation of global aggregation frequency for FL, while considering a fixed resource budget. They analyzed the convergence bound of gradient-descent based FL on non-IID data from a theoretical perspective. Then, they used this convergence bound to build a control algorithm that adapts the frequency of global aggregation in real time to minimize the learning loss under a fixed resource budget. In [13], the convergence analysis of FL within an Unmanned Aerial Vehicle (UAV) swarm was studied. Then, a joint power allocation and scheduling problem was formulated to optimize the convergence rate of FL while considering the energy consumption and delay requirements imposed by the swarm's control system.

It is shown in [10] that the participation of all users in the FL process forces the central server to wait for *stragglers*, i.e., users who have low-quality wireless links that can significantly slow down the FL process, which turns the FL to be unrealistic. Thus, to mitigate the impact of *stragglers*, the authors in [14] proposed a method to select a subset of users for the FL synchronization (or aggregation) process in a resource-constrained environment. They demonstrated the advantages of such technique on improving the FL learning speed. This work has been extended in [15], where a control scheme is proposed, based on reinforcement learning, to accelerate the FL process by actively selecting the best subset of users in each communication round that can counterbalance the bias introduced by non-IID data. In [16], a realistic wireless network model was considered to study the convergence time of FL. Then, given the limited wireless resources, a joint learning, resource allocation, and user selection optimization problem was formulated to minimize the FL convergence time and training loss. In [17], a joint optimization framework for sample selection and user selection was studied to keep a balance between the model accuracy and cost. However, the distribution distance between different users was optimized in this framework through adjusting the local batch size, which might lead to the under-utilization of data in strongly skewed users.

Alternatively, sparsification schemes have been studied to reduce the entropy of the exchanged data (i.e., models' updates) in FL communications. The authors in [18] presented an approach that accelerates the distributed stochastic gradient descent by exchanging sparse updates instead of dense updates. Indeed, they fixed the sparsity rate by only communicating the fraction of entries with the biggest magnitude for each gradient. In [5], the authors proposed a sparse ternary compression scheme that was designed to maintain the requirements of the FL environment. The proposed scheme compressed both the upstream and downstream communications of FL leveraging sparsification, ternarization, error accumulation, and optimal Golomb encoding. This study demonstrated the effect of communications compression and data distributions on the obtained performance. However, it neither considered the wireless resources allocation nor the hierarchical FL architecture. In [19], the FedAvg scheme was adjusted to use a distributed form of Adam optimization, along with the sparsification and quantization, in order to propose a communication-efficient FedAvg.

2.2. Hierarchical Federated Learning

In general, few studies have considered hierarchical FL [20]; specifically, few of them have been proposed so far to address the problem of non-IID data on hierarchical FL architecture. For instance, the authors in [21] extended the work in [8] to analytically prove the convergence of the hierarchical federated averaging algorithm. This work was further extended in [22] considering probabilistic user selection to avoid the impact of *stragglers*. In [23], a self-balancing FL framework, along with two strategies to prevent the bias of training caused by imbalanced data distribution, was proposed. The first strategy aimed to perform data augmentation, before training the model, in order to alleviate global imbalance. The second strategy exploited some mediators (which can be considered as edge nodes) to reschedule the training of the users based on the distribution distances between the mediators. In [24], the effect of the skewed data in hierarchical FL was studied and compared to the centralized FL. Indeed, this work identified the major parameters that affect the learning performance of hierarchical FL. However, this work ignored the resource allocation and wireless communications constraints, such as bandwidth, energy consumption, and latency. Although the aforementioned studies have advanced the area of hierarchical FL and user selection from the theoretical perspective, there are still interesting open research directions related to optimized hierarchical FL with wireless resource allocation under practical conditions.

Novelty. Our work is the first to address the problem of EUs' assignment to different edge nodes in hierarchical FL architectures, while considering the users' data distributions, wireless communication characteristics, and resource allocation. Specifically, unlike other studies, we consider the EUs dataset characteristics and computational capabilities, in addition to practical communication constraints including wireless channel quality, transmission energy consumption, and communication latency. Furthermore, the proposed framework for hierarchical FL is incorporated in an intelligent healthcare system and evaluated using two real-world health datasets to provide efficient remote monitoring services.

3. Hierarchical Federated Learning

In this section, we first highlight the key challenges of using FL in large-scale healthcare systems, then we present our Intelligent-Care (I-Care) system architecture that will be leveraged to address these challenges.

3.1. Challenges of FL in healthcare systems

Before proceeding with the discussion of the proposed framework, we highlight the uniqueness of FL compared to distributed training schemes. FL enables a large amount of EUs to jointly learn a model without sharing their local data. Hence, the distribution of both training data and computational resources is fixed and

linked to the learning environment. However, for efficient leveraging of FL within large scale healthcare systems, the following challenges have to be adequately addressed. We remark that these challenges are present in most of the FL scenarios, however, they are particularly amplified in health-related use-cases.

Imbalanced and non-IID data: Given the heterogeneity of healthcare systems, the distribution and size of the acquired datasets at each EU/patient significantly vary. Typically, the training data in FL is collected by diverse IoT devices near/attached to the EUs, which results in non-IID data distribution over different EUs. FedAvg, the leading FL algorithms for non-IID data distributions, suffers from the large number of communications rounds needed between EUs and the centralized server, especially in the case of imbalanced data [10].

Large number of participants: Given the evolution of healthcare systems with the increasing number of the chronically ill and elderly people, most of the hospitals are required to serve hundreds of patients daily (especially, in case of pandemic such as the recent COVID-19 outbreak). This puts a significant load on different health entities. A promising solution to face such a demand is to move a large number of patients with mild conditions to home care, while still being monitored remotely. This leads to a wide extension in the FL environment for large-scale healthcare systems.

Limited resources: Given the large number of EUs participating in FL, the availability of the needed computational and communication resources at the EUs and network nodes can be challenging. The generated traffic for the models' update grows linearly with the number of participating EUs. Hence, distributing the available bandwidth on different EUs in a communication-efficient way is not an easy task. Moreover, the heterogeneity of the EUs, in terms of computational capabilities and energy availability (i.e., battery level) introduces an extra constraint to optimize the latency and energy consumption.

3.2. I-Care system architecture

For supporting intelligent healthcare services while addressing the above challenges, we consider the I-Care system architecture presented in Fig. 2. The proposed architecture stretches from the EU layer, where data is generated/collected, to the centralized server (or cloud) layer. It includes the following major components:

EU layer: It is considered that each patient has a combination of IoT devices, which enable real-time monitoring and ubiquitous sensing for his/her health conditions and activities within the smart assisted environment. Examples include: body area sensor networks (i.e., implantable or wearable sensors that measure different bio-signals and vital signs), IP cameras, and external medical and non-medical devices. These IoT devices are connected with a local hub, which is responsible for: (1) gathering the health-related information, (2) processing these data, and (3) training the local learning model at the EU level. In our architecture, the EU can be a patient, in-home healthcare environment, or a hospital with several patients. In both cases, FL allows for cooperative training (at the patients or hospital level) without sharing any privacy-sensitive data.

Edge node layer: This layer is an intermediate layer between EUs and the centralized server that aims to reduce the massive communication overhead that arises from sending full EU model updates to the centralized server. In the proposed architecture, the EUs are assigned to different edge nodes, such that each group of EUs runs, with its serving edge node, Synchronous distributed Gradient Descent learning algorithm, also referred to as FedSGD[1] [25]. Under this learning model, an EU i updates its local model $\boldsymbol{W}_i$ based on its local dataset D_i. Then, the EUs' models are aggregated at the assigned edge node j to form the edge model $\boldsymbol{W}_j$. Then, all edge nodes synchronize their models at the centralized server through the FedAvg learning scheme; and the process is repeated until the convergence is attained.

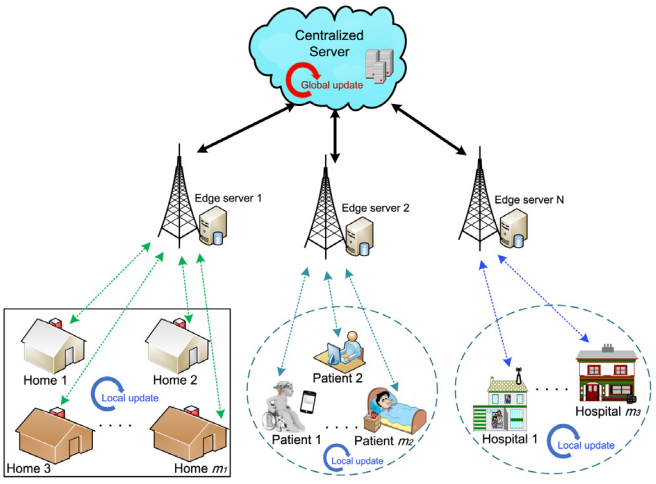


Fig. 2. The proposed I-Care system architecture.

Centralized server layer: This layer takes the comparative advantages of powerful computing resources to aggregate, average, and update the global learning model. Then, it sends back the global model to all participants to complete the FL process.

4. Distributed resources optimization

In this section, we first introduce the mathematical modeling that defines the problem of hierarchical FL with imbalanced and non-IID datasets. Then, we present the communication and computation latency as well as energy consumption analysis of the proposed hierarchical FL framework. Table 1 presents the main mathematical symbols that are used in the following sections.

4.1. Hierarchical FL problem formulation

We consider the hierarchical FL learning system presented in Fig. 2, which consists of one centralized server, a set $\mathcal{N}$ of N edge nodes, and a set $\mathcal{M}$ of M EUs. Each participating EU i stores a local dataset $\mathcal{D}_i$, with a size D_i. Hence, the *virtual* dataset at an edge node j is $\mathcal{D}^{(j)} = \cup_i \mathcal{D}^{(i,j)}, \forall i \in \mathcal{M}_j$, and the *virtual* global dataset at the centralized server is $\mathcal{D} = \cup_j \mathcal{D}^{(j)}, \forall j \in \mathcal{N}$. Note that we use the term *virtual* dataset to clarify the analysis, although these datasets are not stored physically on the edge or the centralized server.

The local dataset $\mathcal{D}_i$, at EU i, consists of the collection of data samples $\{x_i, y_i\}_{i=1}^{D_i}$, where x_i denotes the data acquired by EU i (i.e., features), and $y_i \in \mathcal{C}$ is the associated label, such that $\mathcal{C}$ is the set of all possible labels or data classes. Typically, the main task in any learning model is to find the model parameters $\boldsymbol{w}$ (i.e., weights) that minimize the loss function $\mathcal{L}(\boldsymbol{w})$ using the dataset D_i. We use the widely adopted cross-entropy loss function [26]:

$$\mathcal{L}(\boldsymbol{w}) = \mathbb{E}_{y\sim p(y),\boldsymbol{x}\sim q(\boldsymbol{x}|y)}\left[-\mathbb{1}_{y=k}\log d_k(\boldsymbol{x};\boldsymbol{w})\right]$$

[1] FedSGD is a gradient-based algorithm that is equivalent to federated learning when the global aggregation occurs at every step, where every local step is a full deterministic gradient.

Table 1
List of symbols used throughout the paper.

Symbol	Meaning
$\mathcal{N}$	A set of N edge nodes
$\mathcal{M}$	A set of M EUs
$\mathcal{D}_i$	The local dataset stored at EU i
$\mathcal{C}$	The set of all possible data classes
M_j	Number of EUs assigned to edge j
σ_j	The proportion of edge j dataset to the global dataset
$\|W_i\|$	Updated model length in bits
$r^u_{i,j}$	Upload data rate of EU i to edge j
$r^d_{j,i}$	Download data rate of EU i from edge j
T^c_i	Computation time for updating local model at EU i
L_{ij}	Transmission delay from EU i to edge j
B_{ij}	Allocated bandwidth from edge j to EU i
E_{ij}	Energy consumed by EU i for uploading to edge j
$\boldsymbol{w}_f$	Federated weights
$\boldsymbol{w}_c$	Virtual central weights
D_{KL}	Kullback–Leibler divergence (KLD)
λ_{ij}	Assignment indicator of EU i to edge j

$$= \sum_{k=1}^{C} -p(y=k)\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|y=k)}\left[\log d_k(\boldsymbol{x};\boldsymbol{w})\right] \tag{1}$$

Where $\boldsymbol{x}$ is a feature vector of a data point, C is the number of classes in the learning problem, $p(\cdot)$ is the global classes distribution (global distribution), $q(\cdot|\cdot)$ is the likelihood function, and $d_k(\boldsymbol{x}, \boldsymbol{w})$ is the probability of the kth class for input $\boldsymbol{x}$ under parameters $\boldsymbol{w}$. Then, the objective of the overall learning process is to reach a set of parameters $\boldsymbol{w}$ that minimize the loss function across the *virtual* global dataset:

$$\underset{\boldsymbol{w}}{\text{Minimize}} \sum_{k=1}^{C} p(y=k)\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|y=k)}\left[\log d_k(\boldsymbol{x};\boldsymbol{w})\right]. \tag{2}$$

The loss function in (1) is minimized using the regular gradient descent update:

$$\boldsymbol{w}_c^t = \boldsymbol{w}_c^{t-1} - \delta\nabla_{\boldsymbol{w}_c}\mathcal{L}(\boldsymbol{w}_c). \tag{3}$$

We refer to the parameters in (3) as the centralized model.

Following the proposed hierarchical FL, each group of EUs is associated with a specific edge. Then, in the first step of the hierarchical FL procedure, each EU calculates its learning parameters $\boldsymbol{w}_i$ that are optimized based on its dataset $\mathcal{D}_i$ using FedSGD. Since the *virtual* global data is distributed across different EUs, each EU i locally calculates the loss function over its own dataset to obtain its learning parameters $\boldsymbol{w}_i$.

$$\boldsymbol{w}_i^t = \boldsymbol{w}_i^{t-1} - \delta\nabla_{\boldsymbol{w}_i}\mathcal{L}(\boldsymbol{w}_i). \tag{4}$$

where

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}_i) &= \mathbb{E}_{y\sim p^i(y),\boldsymbol{x}\sim q(\boldsymbol{x}|y)}\left[\mathbb{1}_{y=k}\log d_k(\boldsymbol{x};\boldsymbol{w})\right] \\ &= \sum_{k=1}^{C} p^i(k)\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|y=k)}\left[\log d_k(\boldsymbol{x};\boldsymbol{w})\right].\end{aligned} \tag{5}$$

In (5), $p^i(\cdot)$ is the classes distribution of EU i.

In the hierarchical FL settings, the synchronization (i.e., aggregation) of EUs' parameters (weights) is periodically done by taking the weighted average of EUs' parameters with respect to their local datasets' size. The EUs then exploit the averaged parameters until the next aggregation round. Hence, the second step in the hierarchical FL procedure is to synchronize the local weights across all EUs belonging to a specific edge node every T' local gradients steps. Hence, the parameters of an edge node j at ath *edge aggregation* iteration are:

$$\boldsymbol{w}_j^a = \sum_{i=1}^{M_j} \sigma_{i,j}\boldsymbol{w}_i^{a\times T'}, \tag{6}$$

where

$$\sigma_{i,j} = \frac{|\mathcal{D}^{(i,j)}|}{|\cup_i \mathcal{D}^{(i,j)}|}. \tag{7}$$

In (6), M_j refers to the number of EUs associated with edge node j. Similarly, the learning models are synchronized across all edge nodes every $(T' \times T)$ steps, where T is the centralized aggregation frequency. Hence, at bth *centralized aggregation*, the learning parameters are averaged across all edge nodes to obtain the federated weights as [24]:

$$\boldsymbol{w}_f^b = \sum_{j=1}^{N} \sigma_j\boldsymbol{w}_j^{b\times T} \tag{8}$$

where

$$\sigma_j = \frac{|\mathcal{D}^{(j)}|}{|\cup_j \mathcal{D}^{(j)}|}. \tag{9}$$

Finally, the updated model at the centralized server (i.e., federated weights) is sent back to the edge nodes as well as EUs, and the process is repeated until maintaining the convergence. We remark here that the proposed framework considers the synchronous FL approach, which has shown its efficiency in [8,11] compared to asynchronous approaches. In synchronous FL, the computation steps are synchronized between all EUs, meaning that all EUs have to finish updating their local models before proceeding with the communication step (to forward their models to the edge nodes). Instead, asynchronous FL [27] is an alternative to the typically used synchronous FL, in which the edge nodes operate in an asynchronous manner. The main advantage of asynchronous FL is the ability to fully exploit the available computational resources, at each EU, by performing more gradient descent iterations at powerful (or faster) EUs [8]. However, in case of imbalanced and/or non-IID datasets, the aggregated models will be biased to the faster EUs, which may lead to considerable performance degradation. On the contrary, the performance of synchronous FL may be limited to the performance of the worst EU, i.e., a user with the largest computation and communication delay. However, this limitation can be addressed using the proposed solution, which aims to optimize the allocated resources for different EUs, such that it guarantees the best possible performance for all EUs (as will be shown in the next section).

4.2. Computation and communication latency analysis

In the proposed hierarchical FL framework, we also consider both the communication and computation time for optimizing the performance. In the local update step of our hierarchical FL (i.e., from the EUs to the edge nodes), we denote the minimum upload rate of EU i to edge j by $r^u_{i,j}$. Hence, the maximum uplink communication latency of models aggregation at different edge nodes is defined as

$$L^u = \max_i \left(\frac{|W_i|}{r^u_{i,j}} + \xi^u_i\right), \forall i \in \mathcal{M}, \tag{10}$$

where $|W_i|$ is the total amount of bits that have to be uploaded and downloaded by every EU during the training. In (10), $\frac{|W_i|}{r^u_{i,j}}$ and

ξ_i are, respectively, the transmission time and the access channel delay that EU i expects to experience when transmitting $|W_i|$ bits to edge node j. In other words, it refers to the estimated end-to-end delay when using a given technology [28]. Similarly, if $r^d_{j,i}$ is the minimum download rate from edge node j to EU i, the maximum downlink latency of receiving the updates at the EUs will be

$$L^d = \max_j \left(\frac{|W_i|}{r^d_{j,i}} + \xi^d_j \right), \forall j \in \mathcal{N}, \tag{11}$$

For the computation time per local iteration at EU i, it depends on the size of the FL model, number of CPU cycles that are needed to execute one sample of data, which is denoted by ψ_i, and size of the local dataset D_i at each EU [29]. Given that all samples $\{x_i, y_i\}_{i=1}^{D_i}$ have the same size (i.e., number of bits), the number of CPU cycles needed for EU i to run one local iteration is $(\psi_i \cdot D_i)$. The total number of iterations needed to update the local model, at EU i, is upper bounded by $\mathcal{O}(\log(1/\epsilon))$, where ϵ is the local accuracy. This upper bound is applicable for a wide range of iterative algorithms, such as gradient descent, coordinate descent, or stochastic dual coordinate descent [30]. Moreover, it is proven in [31] that whatever algorithm is used in the computation phase, FL performance will not be affected as long as the convergence time of these algorithms is upper-bounded by $\mathcal{O}(\log(1/\epsilon))$. Hence, the computation time for updating the local model at EU i can be written as a function of $T^c_i = f(|W_i|, \upsilon, \epsilon, \psi_i, D_i, f_i)$, where f_i is the CPU-cycle frequency, and υ is a constant that depends on the dataset size and conditions of the local problem [30]. We remark here that our scheme focuses on the delay over the links between EUs and edge nodes, since the delay over the links between edge nodes and the centralized server can be considered constant for different edge node-EU associations.

4.3. Transmission energy consumption analysis

In the considered hierarchical architecture, it is assumed that the EU can be an IoT device (e.g., a smartphone) with a battery-limited capacity. Hence, it is important to consider the energy consumption that the EU incurs for transferring its local updates to the edge. The transmitted energy consumption at EU i is mainly a function of its wireless channel state, transmission rate, and allocated bandwidth. The wireless channel between an EU i and an edge node j is characterized by the received signal to noise ratio (SNR), denoted by γ_{ij}, which is defined as

$$\gamma_{ij} = \frac{P^r_{ij} \cdot |h_{ij}|^2}{N_0 \cdot B_{ij}}, \tag{12}$$

where P^r_{ij} is the received power at the edge node, $|h_{ij}|$ is the fading channel magnitude, N_0 is the noise spectral density, and B_{ij} is the allocated bandwidth. Also, we consider a deterministic path loss model where the EU and the edge node are separated by a distance d_{ij}, hence the received power P^r_{ij} is attenuated with respect to the transmission power P^t_{ij} following $P^r_{ij} = P^t_{ij} \cdot \omega \cdot d^{-\alpha}_{ij}$, where α is the path loss exponent (such that $2 \leq \alpha \leq 6$), and ω is a constant that depends on the wavelength and transmitter/receiver antennas gain [32]. Given the SNR γ_{ij}, the transmission rate of EU i to edge node j is defined as:

$$r^u_{i,j} = B_{ij} \log_2(1 + \vartheta \gamma_{ij}) \tag{13}$$

where $\vartheta = -1.5/(\log(5 \cdot \text{BER}))$, and BER is the bit error rate target [28]. By Substituting from (12) in (13), the transmitted power can be written as

$$P^t_{ij} = \frac{N_0 \cdot B_{ij}}{g_{ij}} (2^{r_{ij}/B_{ij}} - 1), \tag{14}$$

where g_{ij} is the channel gain that is defined as

$$g_{ij} = \vartheta \cdot \omega \cdot d^{-\alpha}_{ij} \cdot |h_{ij}|^2. \tag{15}$$

Accordingly, the total energy consumed by EU i to send a data of length $|W_i|$ to edge node j is

$$E_{ij} = \frac{P^t_{ij} \cdot |W_i|}{r_{ij}} = \frac{|W_i| \cdot N_0 \cdot B_{ij}}{r_{ij} \cdot g_{ij}} (2^{r_{ij}/B_{ij}} - 1). \tag{16}$$

We remark that the equation in (16) considers the energy consumption due to the transmission, while ignoring the energy consumption due to processing since it is normally notably smaller. Moreover, the processing energy consumption, at each EU, is neither affected by the proposed EUs assignment nor the wireless resource allocation, hence it will not have an impact on the proposed solution.

5. Joint EU assignment and resource allocation

Although leveraging hierarchical FL allows for avoiding data exchange between different EUs, the FL procedure can also drain all resources in the system. Thus, it is crucial to optimize the amount of resources used during the learning process, to avoid overloading the system, while maintaining the accuracy and convergence time on an acceptable level. This paper paves the way for optimizing the hierarchical FL performance in resource-constrained heterogeneous systems through answering the following questions:

- To which edge node should the participating EUs be associated in order to speed up the convergence and maximize the accuracy?
- How can we optimally leverage the available amount of resources while adequately training the global model?

These questions are particularly important in our system, since we consider a real scenario of distributed nodes and IoT devices [33], where: (i) the global data distribution is imbalanced, and (ii) EUs' resources as well as edge computing resources are not as abundant as in the centralized server. Herein, we refer to "resources" as the computation and communication resources, including time, energy, and wireless network bandwidth.

5.1. Problem formulation

Typically, FL performance is measured by its convergence behavior, which depends on the divergence between the federated weights $\boldsymbol{w}^b_f$ and the central weights $\boldsymbol{w}^t_c$. Herein, the central weights represent the learning parameters that are calculated by the centralized gradient descent, when it considers the aggregated dataset of all EUs. This divergence represents the deviation caused due to the distribution of the datasets at different EUs while performing distributed learning. Hence, it is always mandatory to obtain the minimal divergence in order to generate a closer model to the conventional centralized solution.

Convergence bound of FedAvg or gradient-descent based FL on non-IID data has been studied in [10,12]. Such convergence analysis depicts two important facts:

- there is always a trade-off between the communication-efficiency and convergence rate, and
- heterogeneity of the data slows down the convergence.

By extending the analysis in [26] for hierarchical FL, the upper bound of the divergence between federated weights and *virtual* central weights can be approximated to

$$\|\boldsymbol{w}_f - \boldsymbol{w}_c\| \propto \sum_{j=1}^{N} \sigma_j \times \|D^{(j)}\|_1, \tag{17}$$

where σ_j is the proportion of edge j dataset to the global dataset, and $\|D^{(j)}\|_1$ is the classes distribution distance between the edge j dataset and the global aggregated *virtual* dataset. This upper bound can be obtained by adjusting the analysis in [26] to follow the hierarchical FL settings while writing the upper bound as a function of $\|D^{(j)}\|$. Hence, in order to minimize the divergence between federated weights and *virtual* central weights while accelerating FL convergence, it is crucial to maintain a balanced data distribution (similar to the global *virtual* dataset) at the edge nodes by uniformly distributing the acquired data across all edge nodes. This fact has been also proved mathematically in [23].

Based on the previous theoretical analysis, we formulate an optimization problem that considers the imbalanced data distribution and wireless communication resources to dynamically adapt the EUs assignment to the edge nodes in real time, while minimizing the learning loss. Our problem aims at optimally allocating EUs to different edge nodes, such that it obtains a balanced data distribution at different edge nodes, hence accelerating the FL convergence. To achieve this, we opt to minimize the Kullback–Leibler divergence (KLD) of the data distributions [34] in order to tackle the global imbalanced data problem. KLD, also called relative entropy, measures how a probability distribution $H(c_k)$ is different from a second, reference probability distribution $Q(c_k)$ over the same random variable c_k. Let $H_j(c_k)$ and $Q(c_k)$ be the data classes distribution at edge node j and uniform distribution of a discrete random variable $\mathcal{C}$ with possible values $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, respectively. The KLD is defined as

$$D_{KL}(H_j \,||\, Q) = \sum_{k=1}^{K} H_j(c_k) \log \frac{H_j(c_k)}{Q(c_k)}, \tag{18}$$

where $\sum_{k=1}^{K} H_j(c_k) = 1$ and $\sum_{k=1}^{K} Q(c_k) = 1$, as well as $H_j(c_k) > 0$ and $Q(c_k) > 0$, for any $c_k \in \mathcal{C}$, such that $\mathcal{C}$ is the set of all possible classes available at all EUs. Herein, $H_j(c_k)$ represents the actual data distribution at the edge, while $Q(c_k)$ representing the ideal data distribution at different edge nodes (i.e., uniform distribution or the global *virtual* dataset distribution). A KLD of zero indicates that the data classes are uniformly distributed on all edge nodes, i.e., the acquired data at different edge nodes is IID.

Thus, the main objective of our optimization problem is to optimally allocate the M EUs to the available N edge nodes, such that the KLD is minimized. However, given the wireless communication constraints, distributing the EUs to different edge nodes will not be an easy task, especially when considering the non-IID data distribution between different EUs. Thus, our optimization problem is formulated as follows:

$$\mathbf{P1}: \min_{\lambda_{ij}, B_{ij}} \sum_{j=1}^{N} D_{KL}(H_j \,||\, Q) \tag{19}$$

s.t.

$$T_i^c + \sum_{j=1}^{N} \lambda_{ij} \cdot L_{ij} \le T^m, \quad \forall i \in \mathcal{M} \tag{20}$$

$$\sum_{j=1}^{N} \lambda_{ij} \cdot E_{ij} \le E_i^m, \quad \forall i \in \mathcal{M} \tag{21}$$

$$\sum_{i=1}^{M} \lambda_{ij} \cdot B_{ij} \le B_j^m, \quad \forall j \in \mathcal{N} \tag{22}$$

$$\sum_{j=1}^{N} \lambda_{ij} = 1, \quad \forall i \in \mathcal{M} \tag{23}$$

$$\lambda_{ij} = \{0, 1\}, \quad \forall i \in \mathcal{M} \;\&\; \forall j \in \mathcal{N} \tag{24}$$

where λ_{ij} is the edge indicator, such that $\lambda_{ij} = 1$ when EU i is allocated to edge node j. The constraint in (20) ensures that the transmitted updates from all EUs will be received at the edge nodes with a maximum delay T^m. This constraint considers the computational delay, at each EU T_i^c, in addition to the transmission delay L_{ij} from an EU i to an edge j, where $L_{ij} = \frac{|W_i|}{r_{i,j}^u} + \xi_i$. The constraint in (21) considers the energy budget/limit at different battery-operated EUs. Hence, it ensures that the energy consumed by EU i to send its local updates to edge j cannot exceed the maximum transmission energy E_i^m. The constraint in (22) refers to the network capacity constraint, where B_{ij} is the maximum fraction of bandwidth B_j^m that can be used by EU i to communicate with edge j. We remark that B_{ij} depends on the number of EUs communicating with the edge, and it is notified by the edge nodes to the EUs. The constraints in (23) and (24) ensure that each EU will be connected with only one edge.

The optimization variables in (19) are the λ_{ij}'s and B_{ij}'s, i.e., each EU needs to be assigned to an edge node, and each edge node needs to allocate the bandwidth for the assigned EUs. Meanwhile, the proposed optimization problem assigns the EUs to the available edge nodes such that the edge nodes' data distributions turn to be the closest to the uniform distribution, while considering the network dynamics. In other words, our optimization problem aims to adjust the EU allocation at different edge nodes in order to obtain balanced data distribution over all edge nodes (by optimizing the KLD). This balanced data (or class) distribution, from a practical perspective, will lead to accelerating the learning process and promoting the swift convergence (as will be shown in our results).

Looking at problem formulation in (19), one can see that it is an integer programming problem, which is NP-complete problem [35]. Also, the well-known approaches of converting the problem into a convex problem or a Geometric Program (GP), would not work in this case due to the existence of the constraints in (23) and (24) with the non-linearity of the objective function (according to the definition of KLD in (18)). Thus, below we envision a methodology to solve this problem.

5.2. Proposed solution

To solve the formulated problem in (19), we first reformulate the NP-complete optimization problem to have a linear objective function; then we *relax* it to a Linear Program (LP) by removing the bandwidth constraint, and solve the linear program. After that, we *round* the LP solution to a solution that satisfies the original integer constraint. Finally, given the LP solution, we optimize the bandwidth allocation to the available EUs, such that we maintain the KLD minimization obtained by the LP solution. In what follows, we illustrate the proposed solution in details.

First, we rewrite the objective function as follows:

$$\begin{aligned} Z &= \min_{\lambda_{ij}, B_{ij}} \sum_{j=1}^{N} D_{KL}(H_j \,||\, Q) \\ &= \min_{\lambda_{ij}, B_{ij}} \sum_{j=1}^{N} \sum_{k=1}^{K} H_j \log \frac{H_j}{Q}, \\ &= \min_{\lambda_{ij}, B_{ij}} \left(\sum_{j=1}^{N} \sum_{k=1}^{K} H_j \log\left(H_j\right) - \sum_{j=1}^{N} \sum_{k=1}^{K} H_j \log\left(Q\right) \right). \end{aligned} \tag{25}$$

For the sake of brevity, we refer to $H_j(c_k)$ by H_j, and $Q(c_k)$ by Q. Given that Q is representing the uniform distribution, i.e., Q is fixed, Hence

$$Z = \min_{\lambda_{ij}, B_{ij}} \left(\sum_{j=1}^{N} \sum_{k=1}^{K} H_j \log\left(H_j\right) - \log\left(Q\right) \sum_{j=1}^{N} \sum_{k=1}^{K} H_j \right),$$

$$= \min_{\lambda_{ij}, B_{ij}} \left(\sum_{j=1}^{N} \sum_{k=1}^{K} H_j \log\left(H_j\right) - N \cdot \log\left(Q\right) \right). \tag{26}$$

Given that the term $N \cdot \log(Q)$ is fixed with respect to λ_{ij} and B_{ij}, thus the objective function can be written as

$$Z = \min_{\lambda_{ij}, B_{ij}} \left(\sum_{j=1}^{N} \sum_{k=1}^{K} H_j \log\left(H_j\right) \right) = \max_{\lambda_{ij}, B_{ij}} \left(\sum_{j=1}^{N} \chi_j(\mathcal{C}) \right). \tag{27}$$

Now, the objective function in (27) is turned to be equivalent to maximizing the information entropy (or Shannon entropy) χ_j at the edge nodes [36], where $\chi_j(\mathcal{C}) = -\sum_{k=1}^{K} H_j(c_k) \log H_j(c_k)$. It can be easily proved that the maximum entropy can be only achieved at the value associated with uniform distribution, which can be maintained by distributing all data classes equally on all edge nodes. This implies that H_j's should have the same value for all $j \in \mathcal{N}$. However, H_j depends on: (i) λ_{ij}'s, i.e., the allocated EUs at edge node j, and (ii) c_k^i's, i.e., the associated data classes for the EUs. Hence, $H_j(c_k)$ can be defined as

$$H_j(c_k) = \frac{\sum_{i=1}^{M} \lambda_{ij} \cdot c_k^i}{\sum_{k=1}^{K} \sum_{i=1}^{M} \lambda_{ij} \cdot c_k^i}. \tag{28}$$

Given that the maximum entropy is achieved with uniform distribution, our objective function can be written as:

$$Z = \min_{\lambda_{ij}, B_{ij}} \left(\sum_{k=1}^{K} \sum_{\{j,\tilde{j}\} \in \mathcal{S}} \left| \sum_{i=1}^{M} \lambda_{ij} \cdot c_k^i - \sum_{i=1}^{M} \lambda_{i\tilde{j}} \cdot c_k^i \right| \right), \tag{29}$$

where $\mathcal{S}$ is a subset of all possible subsets that can be selected from $\mathcal{M}$ without replacement. For example, if we have 3 edge nodes, the set $\mathcal{S}$ can be defined as $\mathcal{S} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$.

After simplifying our objective function, we still have the non-linear constraint in (20), due to the dependence of the transmission rate on the bandwidth, as well as the integer constraint in (24). Thus, to convert the formulated optimization problem into a LP problem, with a polynomial time complexity, that can be easily solved using many well-known techniques [37], we reformulate the problem as follows:

$$\mathbf{P2}: \min_{\lambda_{ij}} \left(\sum_{k=1}^{K} \sum_{\{j,\tilde{j}\} \in \mathcal{S}} \left| \sum_{i=1}^{M} \lambda_{ij} \cdot c_k^i - \sum_{i=1}^{M} \lambda_{i\tilde{j}} \cdot c_k^i \right| \right) \tag{30}$$

s.t.

$$T_i^c + \sum_{j=1}^{N} \lambda_{ij} \cdot L_{ij} \leq T^m, \quad \forall i \in \mathcal{M} \tag{31}$$

$$\sum_{j=1}^{N} \lambda_{ij} \cdot E_{ij} \leq E_i^m, \quad \forall i \in \mathcal{M} \tag{32}$$

$$\sum_{j=1}^{N} \lambda_{ij} = 1, \quad \forall i \in \mathcal{M} \tag{33}$$

$$0 \leq \lambda_{ij} \leq 1, \ \forall i \in \mathcal{M} \ \& \ \forall j \in \mathcal{N} \tag{34}$$

Now, to solve the original formulated problem in (19) and obtain the best assignment to the edge nodes with bandwidth allocation for all EUs, we propose an efficient EUs Assignment and Resource Allocation (EARA) algorithm (see Algorithm 1). The proposed EARA algorithm aims to solve our problem in two steps. Initially, it is assumed that each edge will have enough bandwidth to be allocated equally to all associated EUs. Then, in the first step, the problem in (30) is solved to obtain λ_{ij}. However, in (30), the integer constraint of λ_{ij} is relaxed, while assuming equal bandwidth allocation for all EUs. Thus, the obtained λ_{ij} is rounded to "0" or "1", such that the original constraints in (23) and (24) are maintained. We remark here that the proposed EARA algorithm allows for enhancing the hierarchical FL performance by enabling the EUs to have two configuration options: Single Connectivity Allocation (SCA), and Dual Connectivity Allocation (DCA). The former refers to assigning each EU to only one edge, while the later refers to the possibility of an EU to be connected to one or two edge nodes simultaneously, leveraging Dual Connectivity (DC)[2] and multicast transmission [38,39]. We remark that DC is one of the substantial technologies adopted by 5G to increase the network throughput and mobility robustness by enabling the users to simultaneously connect with macro and small-cell base stations [40]. In addition, multicast transmission provides significant bandwidth savings for sending duplicate model update from one EU to two edge nodes on a same resource share. It is important also to note that DCA configuration represents an extension to our problem formulation, since it relax the constraint in (33) to allow the EU to connect to one or two edge nodes simultaneously.

Using SCA configuration, we set $\lambda_{ij}^* = 1$, where

$$\lambda_{ij}^* = \arg\max_{j \in \mathcal{N}} \left(\lambda_{ij}\right), \tag{35}$$

while setting the others $\lambda_{ij}, \forall j \in \mathcal{N} \setminus \left\{\lambda_{ij}^*\right\}$ to 0. In DCA configuration, we consider the maximum two values of λ_{ij}'s, i.e., $\lambda_{ij}^1 = \arg\max_{j \in \mathcal{N}} \left(\lambda_{ij}\right)$, and $\lambda_{ij}^2 = \arg\max_{j \in \mathcal{N} \setminus \left\{\lambda_{ij}^1\right\}} \left(\lambda_{ij}\right)$, such that: (i) λ_{ij}^1 is set to 1, and (ii) λ_{ij}^2 is set to 1, only if $\lambda_{ij}^2 > \nu$, while setting the others $\lambda_{ij}, \forall j \in \mathcal{N} \setminus \left\{\lambda_{ij}^1, \lambda_{ij}^2\right\}$ to 0. Herein, ν is a predefined threshold that represents the EU ability to connect to two edge nodes.

In the second step, the available bandwidth B_j^m of an edge node j is allocated to the associated EUs based on their importance. Herein, *importance* of an EU refers to its weight (or role) in minimizing the KLD. For instance, EUs with data classes that are different from the available ones at edge node j will be weighted more than others, in order to maintain the KLD minimization. In this context, an edge j ranks all EUs M_j that are assigned to it, after solving (30). Then, it allocates the minimum bandwidth required for the most important EU to satisfy its latency constraint in (20). This bandwidth allocation process is repeated until all EUs receive their bandwidth allocation or the edge node consumes the available bandwidth budget, i.e., the original constraint in (22) is maintained. The algorithm complexity is closely related to the number of associated EUs to the edges; the worst-case complexity is $\mathcal{O}(M_j \log(M_j))$.

6. Performance evaluation

In this section, we first present the simulation environment that is used to derive our results. Then, we assess the performance of the proposed hierarchical FL framework compared to state-of-the-art techniques. In particular, our results assess the performance of the proposed EARA algorithm using two real-world healthcare datasets to measure the enhancement in the learning speed and accuracy.

[2] In Release 12 of the 3GPP LTE specifications, DC is initially designed to enable the users to connect with two cells at the same time, preferably in heterogeneous networks [38]. Then, the required non-standalone deployment of 5G technology has extended the concept of DC to be used with multiple radio technologies, and in particular, for connecting 5G cells to a 4G core network.

Algorithm 1 EUs Assignment and Resource Allocation (EARA) Algorithm.

Input: $B_{ij} = B_f$, where B_f is the fixed bandwidth assigned to EU i at edge j.
At the centralized server: (*EUs assignment*)
Solve the problem in (30) to obtain λ_{ij}.
if SCA is ON **then**
  Set $\lambda^*_{ij} = 1$, while setting $\lambda_{ij} = 0, \ \forall j \in \mathcal{N} \setminus \{\lambda^*_{ij}\}$.
end if
if DCA is ON **then**
  Set $\lambda^1_{ij} = 1$.
  if $\lambda^2_{ij} > \nu$ **then**
    $\lambda^2_{ij} = 1$.
  else
    $\lambda^2_{ij} = 0$.
  end if
  Set $\lambda_{ij} = 0, \quad \forall j \in \mathcal{N} \setminus \{\lambda^1_{ij}, \lambda^2_{ij}\}$.
end if
Send the values of λ_{ij}'s to all edge nodes.

At the Edge: (*bandwidth allocation*)
Receive λ_{ij}'s from the centralized server .
Sort the assigned EUs based on their weights in an descending order.
for $i = 1 \rightarrow M_j$ **do**
  Calculate the minimum B_{ij} that satisfies the constraint in (20).
  if $\sum_{i=1}^{M_j} \lambda_{ij} \cdot B_{ij} \geq B_j^m$ **then**
    Break ▷ the available bandwidth is consumed.
  end if
end for
return λ_{ij}, B_{ij}.

6.1. Environment setup

For our performance evaluation, we use two datasets, namely, *Heartbeat* dataset and *Seizure* dataset. *Heartbeat* dataset in [41] includes groups of heartbeat signals, i.e., electrocardiogram (ECG) signals, derived from MIT-BIH Arrhythmia dataset [42] for heartbeat classification. These signals represent the normal case and the cases affected by different arrhythmias and myocardial infarction, hence they are used in exploring heartbeat classification using deep neural network architecture. *Seizure* dataset represents electroencephalogram (EEG) measurements comprised of a mixture of waveforms recorded over time that reflect the electrical activity of the brain and obtained from electrodes placed on the scalp, typically in patients with epilepsy. The seizure dataset consists of EEG recordings in such patients that was acquired on a Nicolet machine at the American University of Beirut Medical Center (AUBMC) using the international 10–20 system for electrode placement. The occurrence, number and timing of seizure(s) in each EEG recording was annotated by an experienced electroencephalographer using established criteria that can be summarized as an abrupt change in frequency or amplitude of the waveforms that exhibit evolution in time or space and lasting for 10 s or more [43].

For the distributed settings, we consider different number of EUs while the data is distributed randomly into the EUs, such that we maintain non-IID data distribution between different EUs. For the *Seizure* dataset, the number of possible classes is 3. Hence, we consider 3 edge nodes with 13 users. These users are initially assigned to the edge nodes such that the edge nodes level distribution are obtained as in Table 2. As for the *Heartbeat* dataset, the number of classes is 5. Hence, we consider 5 edge nodes with 18 users. These users are initially assigned to the edge nodes such that each edge contains the distribution illustrated in Table 3.

Table 2
Initial edge-level distribution for the *Seizure* dataset.

Edge	Instances per Class		
	class 0	class 1	class 2
0	1459	25	25
1	25	1160	25
2	25	25	1238

Table 3
Initial edge-level distribution for the *Heartbeat* dataset.

Edge	Instances per Class$\times 10^3$				
	class 0	class 1	class 2	class 3	class 4
0	10	10	0	0	0
1	0	0	10	10	0
2	10	0	0	0	10
3	0	10	10	0	0
4	0	0	0	10	10

Models architecture: For the *Heartbeat* dataset, we use the model presented in [44], which expects 1 input channel and outputs probabilities for 5 classes. For the *Seizure* dataset, we also use a similar model, but we adapted it to accommodate the 19 input channels and the 3 output classes. Our models' details can be found in [45].

The training-related parameters are the same for both datasets, and they are: local epochs equal 1, local batch size equals 10, and learning rate equals 0.001, while selecting ADAM optimizer. For the centralized baseline, it is assumed that the batch size is 50 for the *Heartbeat* dataset and 30 for the *Seizure* dataset. This is because the local batch size should be multiplied by the number of edge nodes, so that at each communication round in the federated settings, the same number of data points are used for both the central and federated models [12].

6.2. Simulation results

In what follows, we compare the proposed EARA algorithm with its two configuration options, i.e., SCA and DCA, against the state-of-the-art solution, such as [21,46], hereinafter we referred to it as Distance-based Allocation (DBA). The DBA scheme considers the same hierarchical FL architecture in Fig. 2, while assigning different EUs to the nearest edge node, based on the distance between the EUs and the edge nodes. Furthermore, we compare our EARA algorithm against a baseline approach, namely, the centralized learning, which represents the benchmark of the FL performance since it assumes that all EUs' data are collected at a centralized server for the training.

The first aspect we investigate is the influence of the EUs dropping on the obtained classification accuracy. To this end, Fig. 3 presents the classification accuracy as a function of the communication rounds between the edge nodes and the centralized server, while considering the DBA scheme. In this figure, we consider different values for Users Participating Percentage (UPP), which represents the percentage of the EUs that the edge nodes receive their updates during the FL training process. Also, we consider two extreme cases, i.e., Single Class Dropping (SCD) and Dual Classes Dropping (DCD). The former refers to missing the whole samples of one data class by dropping their EUs, while the latter refers to missing the samples of two data classes. It is possible to observe that DBA scheme has significantly less performance than the centralized learning, even when considering all EUs' updates,

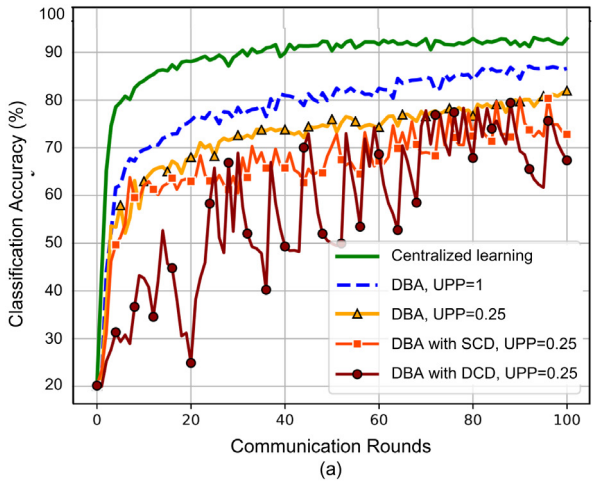


Fig. 3. Effect of varying UPP, i.e., (a) UPP=0.25, (b) UPP=0.5, on the obtained classification accuracy for DBA scheme, using *Heartbeat* dataset.

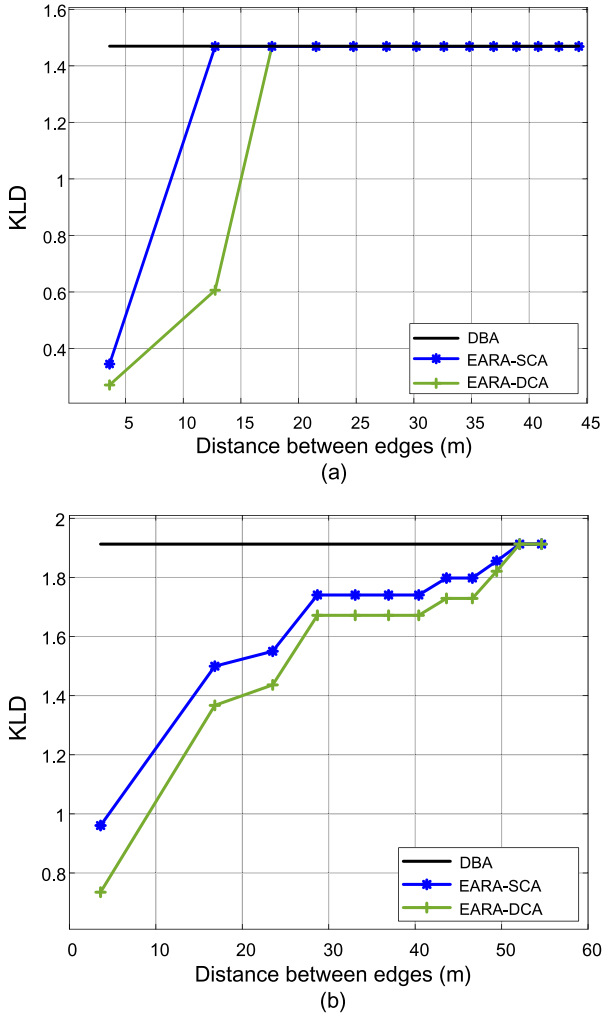


Fig. 4. KLD variations with increasing the distance between EUs and edge nodes, for different EUs assignment strategies using *Heartbeat* dataset, while considering: (a) 3 edge nodes and 13 EUs, (b) 5 edge nodes and 18 EUs.

i.e., UPP=1. An intuitive explanation is that the non-IID data on different edge nodes increases the bias in the learned pattern by the centralized FL server, which results in low prediction accuracy. This confirms the empirical observations in [10] that heterogeneity of the data harms the accuracy and slows down the convergence. More interestingly, decreasing the number of participating EUs in the training yields substantial reduction in the obtained accuracy, especially when the dropping of the EUs leads to missing the samples of one or two data classes. Thus, it is crucial to address the heterogeneity of the data at different edge nodes, to reduce the effect of non-IID data, while considering all EUs with important/unique data. These results are the main motivation behind the proposed EARA scheme.

The second aspect we are interested in is how the proposed EARA algorithm influences the performance of the hierarchical FL. Fig. 4 compares the proposed EARA algorithm with its two configuration options against the DBA approach. This figure depicts the KLD variations of all edge nodes, while varying the distance between EUs and edge nodes. It is clear that our EARA algorithm consistently outperforms the DBA approach. This figure indicates that the proposed EARA algorithm decreases the local data imbalance unlike DBA that obtains the highest KLD, while the dual connectivity option in EARA algorithm (EARA-DCA) obtaining the lowest KLD. However, with increasing the distance between EUs and edge nodes, the energy consumption increases significantly for the EUs that are assigned to an edge node other than the nearest one. This violates the energy constraint in (21), at large distances, hence the EUs are assigned to the nearest edge nodes, which results in converging the EARA performance to the performance of the DBA. In short, Fig. 4 suggests that the edge nodes can achieve better balanced data when more EUs participate in training or when the EUs are assigned uniformly to the edge nodes.

The better performance of the proposed EARA algorithm in decreasing the local data imbalance leads to a significant enhancement in the obtained classification accuracy, as shown in Fig. 5. Each marker therein corresponds to an EUs assignment strategy, and its x- and y- coordinates, respectively, correspond to the communication rounds and the achieved classification accuracy. This figure highlights how the performance of our algorithm approaches the centralized benchmark solution, while reducing the communication rounds between the edge nodes and the centralized server by 75–85% compared to DBA approach, for the same model accuracy. This is because of the adopted EUs assignment strategy that allows for maintaining balanced data distributions at different edge nodes. We highlight here that obtaining balanced data distribution or less-skewed data is an

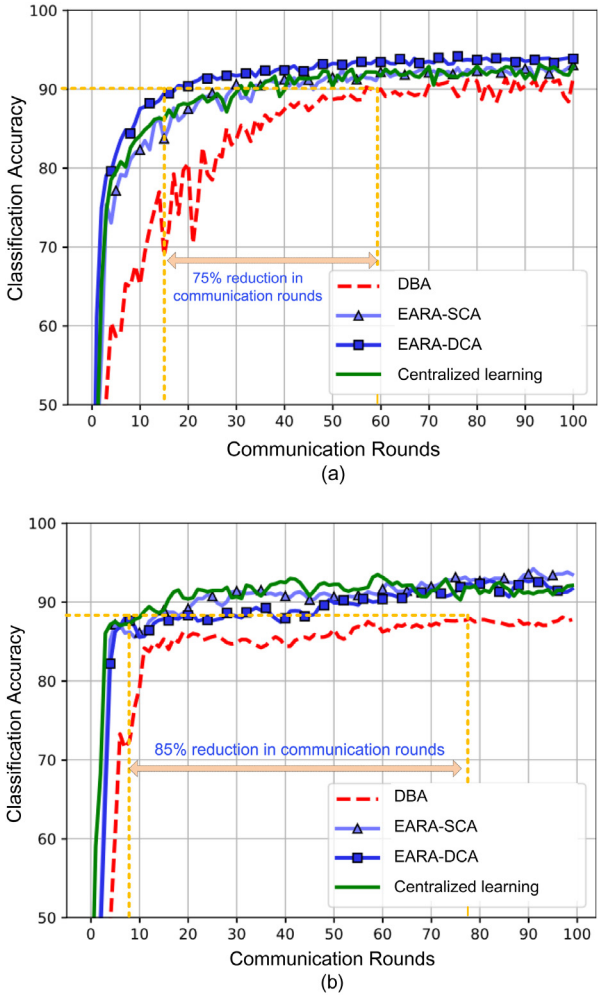


Fig. 5. Classification accuracy as a function of the communication rounds, for different EUs assignment strategies: (a) using *Heartbeat* dataset, (b) using *Seizure* dataset.

important issue to avoid the weights divergence of the learning model. Instead, the DBA approach obtains a slower convergence, while saturating at a lower accuracy value compared to our EARA algorithm due to the weights divergence at the edge nodes that are affected by their local skewed data distributions. More interestingly, leveraging EARA algorithm with dual connectivity option (EARA-DCA) allows for sending duplicate updates to two edge nodes concurrently, hence reducing more the effect of imbalanced data, which leads to an out-performance even compared to the centralized learning benchmark. This resembles the well-known machine learning fact that the accuracy improves as the scale of the training expands.

Fig. 5 highlights also that the differences in the obtained accuracy, leveraging different EUs assignment schemes, are obvious when considering *Heartbeat* dataset. This is because *Heartbeat* dataset has a more challenging classification problem compared to *Seizure* dataset; hence, the imbalanced data has a major effect on the obtained performance. Moreover, we remark that our focus in this paper is to minimize the gap between the hierarchical FL performance and the centralized learning performance, instead of obtaining the most accurate learning model. Hence, the adopted deep learning models, described earlier, are sufficient for this purpose.

Finally, in Fig. 6, we assess the communication traffic load per EU for our EARA approach compared to the DBA approach,

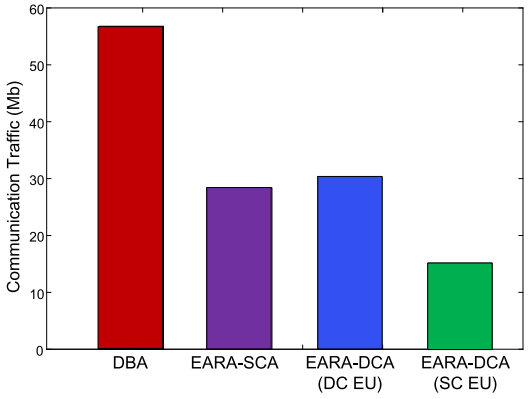


Fig. 6. Communication traffic per EU for different EUs assignment strategies, using *Heartbeat* dataset.

while achieving 90% classification accuracy on *Heartbeat* dataset. This figure depicts that the proposed approach is communication-efficient since our EARA-SCA reduces the communication traffic by 50% compared to DBA. For EARA-DCA, it significantly reduces the communication rounds, hence it reduces the communication traffic by 73% for EUs that have been assigned to one edge, i.e., single-connectivity EUs (SC EU), compared to DBA. For EUs that have been assigned to two edge nodes, i.e., dual-connectivity EUs (DC EU), the communication traffic is increased by 3% compared to EARA-SCA, while still obtaining less communication traffic by 47% compared to DBA. Herein, the communication traffic is calculated considering the deep learning models, described earlier, with 14,789 parameters, while assuming that each parameter is represented by 4 bytes, as in the PyTorch tutorial [47]. Moreover, we remark that by leveraging the recent technologies adopted by 5G, two edge nodes (such as macro cell base station and small cell base station) can simultaneously transmit the data to one EU with the help of millimeter-wave (mmWave) massive MIMO technology [48]. This can significantly decrease the overhead resulting from the dual connectivity.

7. Conclusion and future work

In this paper, we propose an efficient user assignment and resource allocation scheme for hierarchical FL. The proposed scheme allows for leveraging the massive data generated from IoT devices for training deep learning models, while effectively addressing the challenges and requirements posed by the data privacy and resource-constrained environment. In particular, the proposed scheme allows for rendering the distribution of EUs' data, at the edge, to be close to the uniform distribution, which significantly reduces the communication rounds between the edge nodes and the centralized server. The proposed scheme could significantly alleviate the impact of local imbalanced and non-IID data, and reduce the KLD compared to the state-of-the-art solutions that rely on distance-based user assignment. Hence, it accelerates the learning process and reduces the communication traffic by decreasing the global aggregation rounds on the centralized server. Our results show that the proposed approach can reduce 75–85% of the communication rounds between the edge nodes and the centralized server, compared to the state-of-the-art solutions, while preserving the same model accuracy.

The future research directions can include developing online algorithms to select a subset of outstanding users that accelerate the hierarchical FL process, while minimizing the communication overheads. Moreover, the tight synchronization needed among

nodes in FL and the presence of a centralized server are still challenging problems that require further research to make FL techniques suitable for highly dynamic scenarios like vehicular applications.

CRediT authorship contribution statement

Alaa Awad Abdellatif: Conceptualization, Methodology, Investigation, Software, Validation, Resources, Writing – original draft, Funding acquisition. **Naram Mhaisen:** Conceptualization, Methodology, Investigation, Software, Validation, Resources, Writing – original draft, Funding acquisition. **Amr Mohamed:** Conceptualization, Methodology, Investigation, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Aiman Erbad:** Conceptualization, Methodology, Investigation, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Mohsen Guizani:** Conceptualization, Methodology, Writing – review & editing. **Zaher Dawy:** Methodology, Writing – review & editing. **Wassim Nasreddine:** Data curation, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was made possible by NPRP grant # NPRP12S-0305-190231 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors. Open Access funding provided by the Qatar National Library.

References

[1] A. Ghasempour, Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges, Inventions 4 (1) (2019) 22.
[2] W. Samek, G. Montavon, A. Vedaldi, L.K. Hansen, K.-R. Müller, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Springer International Publishing, 2019.
[3] A.A. Abdellatif, C.F. Chiasserini, F. Malandrino, Active learning-based classification in automated connected vehicles, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops, 2020, pp. 598–603, http://dx.doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162964.
[4] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.
[5] F. Sattler, S. Wiedemann, K.R. Müller, W. Samek, Robust and communication-efficient federated learning from non-i.i.d. data, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2020) 3400–3413, http://dx.doi.org/10.1109/TNNLS.2019.2944481.
[6] M.G.R. Alam, M.M. Hassan, M.Z. Uddin, A. Almogren, G. Fortino, Autonomic computation offloading in mobile edge for IoT applications, Future Gener. Comput. Syst. 90 (2019) 149–157.
[7] A.A. Abdellatif, A. Mohamed, C.F. Chiasserini, A. Erbad, M. Guizani, Edge computing for energy-efficient smart health systems: Data and application-specific approaches, in: Energy Efficiency of Medical Devices and Healthcare Applications, Elsevier, 2020, pp. 53–67.
[8] S. Wang, T. Tuor, T. Salonidis, K. Leung, C. Makaya, T. He, K. Chan, Adaptive federated learning in resource constrained edge computing systems, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1205–1221.
[9] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 2261–2269, http://dx.doi.org/10.1109/CVPR.2017.243.
[10] X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, On the convergence of fedavg on non-IID data, 2019, arXiv:1907.02189.
[11] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A.Y. Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS, 2017.
[12] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-IID data, 2018, arXiv:1806.00582 [cs, stat], URL http://arxiv.org/abs/1806.00582.
[13] T. Zeng, O. Semiari, M. Mozaffari, M. Chen, W. Saad, M. Bennis, Federated learning in the sky: Joint power allocation and scheduling with UAV swarms, in: ICC 2020 - 2020 IEEE International Conference on Communications, ICC, 2020, pp. 1–6, http://dx.doi.org/10.1109/ICC40277.2020.9148776.
[14] T. Nishio, R. Yonetani, Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge, in: ICC 2019 - 2019 IEEE International Conference on Communications, ICC, 2019, pp. 1–7.
[15] H. Wang, Z. Kaplan, D. Niu, B. Li, Optimizing Federated Learning on Non-IID Data with Reinforcement Learning, in: INFOCOM, 2020, p. 10.
[16] M. Chen, H. Vincent Poor, W. Saad, S. Cui, Convergence time optimization for federated learning over wireless networks, IEEE Trans. Wireless Commun. (2020) 1, http://dx.doi.org/10.1109/TWC.2020.3042530.
[17] C. Feng, Y. Wang, Z. Zhao, T.Q.S. Quek, M. Peng, Joint optimization of data sampling and user selection for federated learning in the mobile edge computing systems, in: IEEE International Conference on Communications Workshops, ICC Workshops, 2020, pp. 1–6.
[18] A.F. Aji, K. Heafield, Sparse communication for distributed gradient descent, 2017, ArXiv:1704.05021, URL https://arxiv.org/abs/1704.05021.
[19] J. Mills, J. Hu, G. Min, Communication-efficient federated learning for wireless edge intelligence in IoT, IEEE Internet Things J. 7 (7) (2020) 5986–5994, http://dx.doi.org/10.1109/JIOT.2019.2956615.
[20] L.U. Khan, Z. Han, D. Niyato, C.S. Hong, Socially-aware-clustering-enabled federated learning for edge networks, IEEE Trans. Netw. Serv. Manag. 18 (3) (2021) 2641–2658, http://dx.doi.org/10.1109/TNSM.2021.3090446.
[21] L. Liu, J. Zhang, S.H. Song, K.B. Letaief, Client-Edge-Cloud Hierarchical Federated Learning, in: IEEE International Conference on Communications, ICC, 2020, pp. 1–6.
[22] W. Wu, L. He, W. Lin, R. Mao, Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems, 2020, arXiv:2007.14374 [Cs, Eess], URL http://arxiv.org/abs/2007.14374.
[23] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, L. Liang, Self-balancing federated learning with global imbalanced data in mobile systems, IEEE Trans. Parallel Distrib. Syst. 32 (1) (2021) 59–71, http://dx.doi.org/10.1109/TPDS.2020.3009406.
[24] N. Mhaisen, A. Awad, A. Mohamed, A. Erbad, M. Guizani, Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints, IEEE Trans. Netw. Sci. Eng. (2021) 1, http://dx.doi.org/10.1109/TNSE.2021.3053588.
[25] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, in: Proceedings of the 20 Th International Conference on Artificial Intelligence and Statistics, 2017, pp. 1273–1282.
[26] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-iid data, 2018, arXiv preprint arXiv:1806.00582.
[27] J. Chen, R. Monga, S. Bengio, R. Jozefowicz, Revisiting distributed synchronous SGD, in: International Conference on Learning Representations Workshop Track, 2016, URL https://arxiv.org/abs/1604.00981.
[28] A.A. Abdellatif, A. Mohamed, C.-F. Chiasserini, User-centric networks selection with adaptive data compression for smart health, IEEE Syst. J. 12 (4) (2018) 3618–3628, http://dx.doi.org/10.1109/JSYST.2017.2785302.
[29] N.H. Tran, W. Bao, A. Zomaya, M.N.H. Nguyen, C.S. Hong, Federated learning over wireless networks: Optimization model design and analysis, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019, pp. 1387–1395, http://dx.doi.org/10.1109/INFOCOM.2019.8737464.
[30] J. Konečný, Z. Qu, P. Richtárik, Semi-stochastic coordinate descent, in: Optimization Methods and Software, 32, (5) 2017, pp. 993–1005.
[31] C. Ma, J. Konečný, M. Jaggi, V. Smith, M. Jordan, P. Richtárik, M. Takáč, Distributed optimization with arbitrary local solvers, in: Optimization Methods and Software, 32, (4) 2017, pp. 813–848.
[32] T.S. Rappaport, Wireless Communications: Principles and Practice, second ed., Prentice Hall, 2001.
[33] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, G. Fortino, Agent-based internet of things: State-of-the-art and research challenges, Future Gener. Comput. Syst. 102 (2020) 1038–1053.
[34] S. Kullback, Information Theory and Statistics, in: A Wiley publication in mathematical statistics, Dover Publications, 1997, 97014382, URL https://books.google.com.qa/books?id=05LwShwkhFYC.
[35] S. Boyd, L. Vandenberghe, Convex Optimization, first ed., Cambridge University Press, 2003.
[36] T. M.Cover, J. A.Thomas, Elements of Information Theory, second ed., John Wiley & Sons, 2006.
[37] Jiří Matoušek, Bernd Gärtner, Understanding and using Linear Programming, first ed., Springer, Berlin, Heidelberg, 2007.
[38] J.F. Monserrat, F. Bouchmal, D. Martin-Sacristan, O. Carrasco, Multi-radio dual connectivity for 5g small cells interworking, IEEE Commun. Stand. Mag. 4 (3) (2020) 30–36, http://dx.doi.org/10.1109/MCOMSTD.001.1800030.

[39] Y. Sun, Z. Chen, M. Tao, H. Liu, Bandwidth gain from mobile edge computing and caching in wireless multicast systems, IEEE Trans. Wireless Commun. 19 (6) (2020) 3992–4007, http://dx.doi.org/10.1109/TWC.2020.2979147.
[40] L. Sharma, B.B. Kumar, S. Wu, Performance analysis and adaptive DRX scheme for dual connectivity, IEEE Internet Things J. 6 (6) (2019) 10289–10304, http://dx.doi.org/10.1109/JIOT.2019.2937345.
[41] ECG heartbeat categorization dataset: Segmented and preprocessed ECG signals for heartbeat classification, 2018, https://www.kaggle.com/shayanfazeli/heartbeat, Accessed 13 January 2021.
[42] G.B. Moody, R.G. Mark, The impact of the MIT-bih arrhythmia database, IEEE Eng. Med. Biol. Mag. 20 (3) (2001) 45–50, http://dx.doi.org/10.1109/51.932724.
[43] D.L. Schomer, F.H.L. da Silva, Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields, sixth ed., Lippincott Williams & Wilkins, 2011.
[44] An application of deep convolutional networks for the classification of ECG signal, 2019, https://github.com/eddymina/, Accessed 13 January 2021.
[45] Datasets models, 2021, https://github.com/Naram-m/Datasets-Models, Accessed 13 January 2021.
[46] M.S.H. Abad, E. Ozfatura, D. GUndUz, O. Ercetin, Hierarchical federated learning across heterogeneous cellular networks, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2020, pp. 8866–8870, http://dx.doi.org/10.1109/ICASSP40776.2020.9054634.
[47] Training a classifier — PyTorch tutorials 1.6.0 documentation, https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html.
[48] Y. Yang, X. Deng, D. He, Y. You, R. Song, Machine learning inspired codeword selection for dual connectivity in 5g user-centric ultra-dense networks, IEEE Trans. Veh. Technol. 68 (8) (2019) 8284–8288, http://dx.doi.org/10.1109/TVT.2019.2923314.

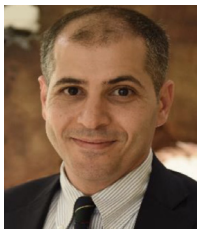
Alaa Awad Abdellatif received his Ph.D. in 2018 from Politecnico di Torino. He is currently a Postdoctoral Research Fellow in Qatar University. He received his B.Sc. and M.Sc. degrees with honors in Electronics and Electrical Communications Engineering from Cairo University, in 2009 and 2012, respectively. He also worked as a Postdoctoral Researcher, Senior Research Assistant, and Research Assistant at Politecnico di Torino, from 2019 till 2020, at Qatar University, from 2013 till 2014, and at Cairo University, from 2009 till 2012, respectively. Dr. Alaa Abdellatif has authored or co-authored over 40 refereed journal, magazine, and conference papers in reputable international journals and conferences. He has served as a technical reviewer for many international journals and magazines. His research interests include Edge computing, Blockchain, machine learning and resources optimization for wireless heterogeneous networks, smart-health, IoT applications, and vehicular networks. He was the recipient of the Postdoctoral Research Award (PDRA) and Graduate Student Research Award (GSRA) from Qatar National Research Fund in 2014 and 2021, respectively, in addition to the Best Paper Award from Wireless Telecommunications Symposium 2018, in USA, and the Quality Award from Politecnico di Torino in 2018.

Naram Mhaisen received a B.Sc. degree in Computer Engineering, with excellence, from Qatar University (QU) in 2017, and an M.Sc. in Computing from QU in 2020. He is currently a Research Assistant at the department of computer science and engineering, college of engineering, QU. His research interests include the design and optimization of distributed and multi-agent (learning) systems with applications to IoT.

Amr Mohamed (S' 00, M' 06, SM' 14) received his M.S. and Ph.D. in electrical and computer engineering from the University of British Columbia, Vancouver, Canada, in 2001, and 2006 respectively. He has worked as an advisory IT specialist in IBM Innovation Centre in Vancouver from 1998 to 2007, taking a leadership role in systems development for vertical industries. He is currently a professor in the college of engineering at Qatar University. He has over 25 years of experience in wireless networking research and industrial systems development. He holds 3 awards from IBM Canada for his achievements and leadership, and 4 best paper awards from IEEE conferences. His research interests include wireless networking, and edge computing for IoT applications. Dr. Amr Mohamed has authored or co-authored over 200 refereed journal and conference papers, textbooks, and book chapters in reputable international journals, and conferences. He is serving as a technical editor for two international journals and has served as a technical program committee (TPC) co-chair for many IEEE conferences and workshops.

Aiman Erbad (aerbad@ieee.org) is an Associate Professor and ICT Division Head in the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). Prior to this, he was an Associate Professor at the Computer Science and Engineering (CSE) Department and the Director of Research Planning and Development at Qatar University until May 2020. He also served as the Director of Research Support responsible for all grants and contracts (2016–2018) and as the Computer Engineering Program Coordinator (2014–2016). Dr. Erbad obtained a Ph.D. in Computer Science from the University of British Columbia (Canada) in 2012, a Master of Computer Science in embedded systems and robotics from the University of Essex (UK) in 2005, and a B.Sc. in Computer Engineering from the University of Washington, Seattle in 2004. He received the Platinum award from H.H. The Emir Sheikh Tamim bin Hamad Al Thani at the Education Excellence Day 2013 (Ph.D. category). He also received the 2020 Best Research Paper Award from Computer Communications, the IWCMC 2019 Best Paper Award, and the IEEE CCWC 2017 Best Paper Award. His research received funding from the Qatar National Research Fund, and his research outcomes were published in respected international conferences and journals. He is an editor for KSII Transactions on Internet and Information Systems, an editor for the International Journal of Sensor Networks (IJSNet), and a guest editor for IEEE Network. He also served as a Program Chair of the International Wireless Communications Mobile Computing Conference (IWCMC 2019), as a Publicity chair of the ACM MoVid Workshop 2015, as a Local Arrangement Chair of NOSSDAV 2011, and as a Technical Program Committee (TPC) member in various IEEE and ACM international conferences (GlobeCom, NOSSDAV, MMSys, ACMMM, IC2E, and ICNC). His research interests span cloud computing, edge intelligence, Internet of Things (IoT), private and secure networks, and multimedia systems. He is a senior member of IEEE and ACM.

Mohsen Guizani (S'85–M'89–SM'99–F'09) received the B.S. (with distinction) and M.S. degrees in electrical engineering, the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor at the Computer Science and Engineering Department in Qatar University, Qatar. Previously, he served in different academic and administrative positions at the University of Idaho, Western Michigan University, University of West Florida, University of Missouri-Kansas City, University of Colorado-Boulder, and Syracuse University. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is currently the Editor in-Chief of the IEEE Network Magazine, serves on the editorial boards of several international technical journals and the Founder and Editor-in-Chief of Wireless Communications and Mobile Computing journal (Wiley). He is the author of nine books and more than 600 publications in refereed journals and conferences. He guest edited a number of special issues in IEEE journals and magazines. He also served as a member, Chair, and General Chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He is the recipient of the 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award for outstanding contributions to the technological advancement of security. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is a Fellow of IEEE and a Senior Member of ACM.



Zaher Dawy received the BE Degree in Computer and Communications Engineering from American University of Beirut (AUB) in 1998 and the Doctoral Degree in Communications Engineering from Munich University of Technology (TUM) in 2004. He joined AUB in 2004 and is currently a Professor in the Electrical and Computer Engineering Department. His research interests are in the general areas of wireless communications and biomedical engineering. He has served on the Editorial Board of IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Communications Surveys and Tutorials, Computer Networks (Elsevier), and Physical Communications (Elsevier).

Wassim Nasreddine is appointed as Assistant Professor of Clinical Neurology effective January 1, 2013. He graduated with an M.D. degree from the American University of Beirut in 1990. He is a well-trained in the field of Epilepsy and Sleep Medicine, and has a very busy practice. He has 18 peer-reviewed publications, of which he is first/second author on 8. He involved in teaching medical students, residents and fellow in his areas of specialty.