Energy-Efficient Multiprocessor-Based Computation and Communication Resource Allocation in Two-Tier Federated Learning Networks

Rukhsana Ruby, *Member, IEEE*, Hailiang Yang, *Student Member, IEEE*, Felipe A. P. de Figueiredo, Thien Huynh-The, *Member, IEEE*, and Kaishun Wu, *Senior Member, IEEE*

***Abstract*—In conventional federated learning (FL), multiple edge devices holding local data jointly train a machine learning model by communicating learning updates with a centralized aggregator without exchanging their data samples. Owing to the communication and computation bottleneck at the centralized aggregator and inaccurate learning model caused by the non-independent and identically distributed (IID) data, we here consider a two-tier FL network, in which Internet of Things (IoT) nodes are the core clients that hold data, the model aggregators at the middle tier are the low altitude aerial platforms (UAVs), and the model aggregator at the top-most layer is the high-altitude aerial platform (UAV with relatively high altitude). Under the assumption that each IoT node has parallel computing ability, we study the energy-efficient computation and communication resource allocation in such a network within some time budget. Upon formulating the problem as an optimization problem, we solve the computation and communication resource allocation problems as the separate subproblems within a time frame, and then propose an iterative algorithm to solve the entire problem jointly. More specifically, we solve both the energy-efficient computation and communication resource allocation subproblems using the dual decomposition technique, and then apply a bisection search-based recursive technique to solve the entire energy efficiency problem jointly. Moreover, we propose offline and online client scheduling schemes that not only select the optimal edge nodes for association but also assign workload to each client based on the data quality and workload constraint. With real data, extensive simulations are conducted to verify the effectiveness of the proposed resource allocation scheme. The results further reveal that the learning performance not only is dependent on the computation and communication energy consumption of the FL process but also the model divergence weight owing to the non-IID data at client IoT nodes.**

***Index Terms*—Computation and communication resource allocation, dual decomposition technique, energy-efficient resource allocation, federated learning (FL).**

Manuscript received 4 October 2021; revised 29 December 2021; accepted 7 February 2022. Date of publication 4 March 2022; date of current version 24 March 2023. This work was supported in part by the China NSFC International Young Scientist under Grant 61850410538; in part by the China NSFC under Grant U2001207 and Grant 61872248; in part by the Guangdong NSF under Grant 2017A030312008; in part by the Shenzhen Science and Technology Foundation under Grant ZDSYS20190902092853047; in part by the Project of DEGP under Grant 2019KCXTD005 and Grant 2021ZDZX1068; and in part by the Guangdong "Pearl River Talent Recruitment Program" under Grant 2019ZT08X603. *(Rukhsana Ruby and Hailiang Yang contributed equally to this work.) (Corresponding author: Kaishun Wu.)*

Rukhsana Ruby, Hailiang Yang, and Kaishun Wu are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: ruby@szu.edu.cn; yanghailiang2019@email.szu.edu.cn; wu@szu.edu.cn).

Felipe A. P. de Figueiredo is with the Department of Communication, National Institute of Telecommunications (INATEL), Santa Rita do Sapucaí 37540-000, Brazil (e-mail: felipe.figueiredo@inatel.br).

Thien Huynh-The is with the ICT Convergence Research Center, Kumoh National Institute of Technology, Gumi 39177, Gyeongsangbuk-do, Republic of Korea (e-mail: thienht@kumoh.ac.kr).

Digital Object Identifier 10.1109/JIOT.2022.3153996

I. Introduction

In CONVENTIONAL ring-like federated learning (FL) networks [1]–[3], Internet of Things (IoT) clients train locally stored individual data based on the preliminary model obtained at the previous round, and then transmit the constructed models to the centralized aggregator. In such a type of FL architecture, the centralized aggregator could be the communication and computation bottleneck if the appropriate measure is not undertaken in order to achieve its reliable service. On the other hand, owing to the heterogeneity nature, a large number of clients in the training process may hold non-independent and identically distributed (IID) data, which may cause the construction of bad performing skewed model through the FL process. Therefore, among many solutions, training a machine learning (ML) model in a hierarchical architecture (e.g., two-tier architecture) is one of the promising solutions. Even with such a type of architecture, energy-constrained IoT clients still need to perform heavy-weight computation tasks in the FL process, which further necessitate to equip multiple processors with parallel computing ability to them [4]. However, energy-efficient completion of each FL round is entangled with several challenges compared to conventional single-tier networks. For example, sorting out the non-IID data may bring additional workload constraints in each client as well as there may arise total workload constraint over all the clients to avoid non-IID data in the training process as much as possible. On the other hand, bandwidth as well as the number of channels could be limited which, results in sharing a single channel by multiple nodes in the system. This brings interference effect in the communication phase and mitigating such phenomenon is another challenge in this problem.

To name, the communication resources are: bandwidth, time, and energy. Under the fixed quality of a communication link, the higher the bandwidth, time and energy, the higher the data rate. In the FL process, since the size of the model including its parameters is fixed, if we fix any two resources (e.g., bandwidth and time), the other one (e.g., energy) is required to be adjusted in order to achieve the model size-equivalent data rate. On the other hand, computation resources imply time, frequency scaling and workload partitioning [5]–[8]. The higher the computation resources, the higher the energy consumption. On the other hand, if the learning nodes have multiple processors with parallel computing ability, energy-efficient partitioning of workload among multiple processors, under a given timing budget and frequency scaling constraint [8], is not a straightforward problem. Unless the synchronization among client nodes is handled in a special manner, all client nodes in the FL process require to train a model locally and then upload to their aggregator node over a fixed time. Typically, client IoT nodes are energy-constrained unless they have any special quick batter-refilling facility. Therefore, they cannot use excessive energy for the computation and communication purposes to keep pace with the prespecified timing budget of one learning round. Moreover, if more energy is used at the beginning of training process, almost nothing will be available for the later periods [9]. Therefore, there should be a balanced usage of energy for the computation and communication tasks of the FL process. To the best of our knowledge, energy-efficient joint computation and communication resource allocation in two-tier hierarchical FL networks has not been studied yet.

There are already many evidence that the performance of FL is not only dependent on semantic of the model and convergence condition but also dependent on the computation and communication resource allocation. Owing to this fact, there are numerous works, on the resource allocation in conventional ring-like FL networks, appeared over the last couple of years. For example, Shi *et al.* [10] proposed a joint device scheduling and resource allocation policy to maximize the model accuracy within a given training time budget for latency constrained wireless FL networks. Chen *et al.* [11] also proposed communication resource allocation and client selection scheme based on the convergence condition of the learning model. A joint central processing unit (CPU) frequency and power control was studied in [12] with the objective of minimizing the energy consumption of all IoT devices. Wang *et al.* [13] proposed a control algorithm that determines the best tradeoff between local update and global parameter aggregation to minimize the loss function under a given computation and communication resource budget. He *et al.* [14] developed an importance-aware joint client selection and communication resource allocation algorithm to maximize the learning efficiency. Luo *et al.* [15] proposed a bandwidth allocation scheme based on the sliding differential scheduling concept while considering the energy minimization and model loss function. Upon characterizing the convergence upper bound in terms of the convergence rate and global rounds, Nguyen *et al.* [16] proposed an efficient path-following algorithm to minimize either total energy consumption or FL completion time. A similar type of another work is [17], in which the authors solve a resource allocation problem in three steps that captures the tradeoff between convergence time and energy consumption of the client nodes. Huang *et al.* [18] proposed a fairness-aware client scheduling algorithm using the Lyapunov optimization technique. Based on the 2-D contract theory, a greedy client selection is proposed in [19] knowing the data asymmetry among different clients. Zhang *et al.* [20] proposed a client selection scheme based on the degree of weight divergence among different non-IID data-equipped clients. Sun *et al.* [21] proposed a client selection scheme based on the gradient sparsifying concept as well as the optimization of communication resources. Under the assumption that channel state information (CSI) of the network is unknown, Xia *et al.* [22] proposed a client selection scheme based on the multiarmed bandit-based learning tool. Ren *et al.* [23] proposed a probabilistic scheduling framework that exploits multiuser diversity in wireless channels as well as the diversity in importance of FL clients. The FL process is also studied in cell-free multi-input-multioutput MIMO [24] and interference-prone [25] networks, and the optimization of computation and communication resources, such as transmit power, data processing, and client scheduling, of such networks are explored.

Numerous collaborative FL process using different network topologies, such as star-like, grid-like, and complete ones, have appeared in [26] and [27]. Among these different network architecture, two-tier hierarchical networks are reported to have higher learning performance, especially when the client IoT nodes hold non-IID data [28]. Over the last one year, there are some works on the optimization of control variables of such a network architecture. For example, Wu *et al.* [29] applied the concept of regional slack factors to select reliability-agnostic clients using a probabilistic approach without identifying the state of client nodes. For the same type of network, Mhaisen *et al.* [28] found an optimal client–edge assignment based on statistical properties of client data and network topology constraints such that edge-level data distributions turn to be similar (i.e., close to IID), which enhances the federated averaging performance. Luo *et al.* [30] also considered a cost-efficient resource allocation scheme in such type of networks, which has slight similarity to our work. However, the major difference is that the client nodes in this work do not have multiprocessor-based parallel computing capability. Moreover, the edge computing nodes conduct several learning rounds with client IoT nodes before sending the model to the centralized aggregator. In our case, each edge computing node transmits the gradient information to the centralized aggregator at tier-2 as soon as it receives this from the client nodes at tier-1, which insightfully should have better learning performance.

Despite some works on hierarchical two-tier FL networks, there is no any work on the energy-efficient computation and communication resource allocation, especially when client IoT nodes and edge servers are equipped with multiple processors with parallel computing capability. Inspired by the beneficial feature of a two-tier hierarchical network in the FL process, we first formulate the energy-efficient computation and communication resource allocation problem under the frequency scaling constraint of each processor, and the workload constraint at each client as well as the total workload constraint among all clients. The workload constraint at each client as well as the overall one are incorporated into the problem to avoid non-IID data in the model construction. Moreover, we provide some weight to each client based on the quality of the data it holds. As of the communication resource allocation, we consider that spectrum is divided into subchannels, and the number of subchannels is much lower compared to the number of client IoT nodes. Since each subchannel may be shared by multiple nodes at a time for the purpose of communication, interference is obvious and this is captured in the communication resource allocation of this problem. The entire model updating task happens over a fixed-time constraint, and all the nodes work in a time-synchronous manner. Under such a setup and with all constraints, it appears that the problem is a mixed integer programming one and, hence, intractable to solve. As a result, the problem is decoupled into two subproblems: 1) energy-efficient computation resource allocation and 2) energy-efficient communication resource allocation. Then, we provide an external bisection search-based iterative algorithm to solve the entire problem jointly. To summarize, the contributions of this work are listed as follows.

1) To solve the energy-efficient computation resource allocation problem, we first alter the objective function of the problem such that it becomes amenable to the dual decomposition technique [31], [32]. We then adopt this technique to find the optimal computation resource assignment (i.e., frequency scaling and workload partitioning) in an analytical manner.
2) To solve the energy-efficient communication resource allocation problem at each training round, we adopt the dual decomposition method [31], [32] as well. Since the complexity of this method to find the optimal solution is intractable, we propose a suboptimal algorithm with relatively lower complexity to solve this problem.
3) Under the constraint that a certain number of clients can be selected, we provide two scheduling schemes for the online and offline client–edge assignment, which not only take care of the energy consumption aspect of communication and computation tasks but also the importance of the clients based on their model distribution.
4) Extensive simulation is conducted to verify the effectiveness and efficiency of the proposed energy-efficient computation and communication resource allocation scheme. The results reveal that client–edge assignment in two-tier hierarchical networks should not only be determined based on the energy consumption at each round but also the model distribution at each client IoT node.

The remainder of this article is organized as follows. Upon providing necessary background information, in Section II, we describe the system model as well as formulate the entire energy-efficient computation and communication resource allocation problem. In Section III, we provide a solution strategy to solve the formulated problem. We evaluate the performance of the proposed energy-efficient resource allocation scheme in Section III. Finally, we draw the conclusion in Section V.

II. System Model and Problem Formulation

Consider an IoT network, as illustrated in Fig. 1, that consists of M IoT nodes (denoted by the nodes in set $\mathcal{M}$), K ($M \gg K$) UAV-based edge computing nodes at the middle tier (denoted by the nodes in set $\mathcal{K}$) to collect the models from individual IoT nodes and a cloud aggregator at the top-most tier to collect the constructed model from the edge computing nodes. The distribution of the model from the cloud aggregator to the edge computing nodes and that from the edge computing nodes to the IoT nodes occur in a synchronous manner within some time frame. In the similar manner, model collection from the IoT nodes to the edge computing nodes and that from the edge computing nodes to the cloud aggregator happen in the synchronous manner within some time interval. We assume that the IoT nodes in set $\mathcal{M}_k$ are connected to edge computing node k and all the edge computing nodes are connected to the cloud aggregator. As of the communication media, we consider that there are N orthogonal-frequency-division-multiple-access (OFDMA) channels (residing in set $\mathcal{N}$) in the system and $M >> N$ holds. As a result, for transmitting the model to the edge computing nodes, IoT nodes may require to share channels among each other which cause interference. It is assumed that the cloud aggregator is the centralized entity in the system who has perfect knowledge of the multiuser channel gains as well as the local computation characteristics at each client IoT node. These information are achievable by the well-known feedback mechanism over control channels. Upon obtaining all information at each learning round, the centralized entity determines the energy-efficient strategies for client node selection, and computation and communication resource allocation. Note that all the client IoT nodes and the UAVs could be mobile over the entire FL process. However, their locations remain constant or mobile within a very short range over the duration of a single learning round.

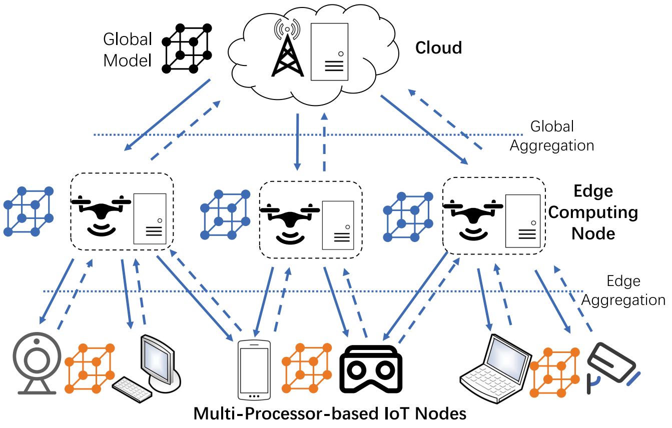


Fig. 1. Illustration of hierarchical two-tier FL networks.

A. FL Process in Two-Tier Networks

A hierarchical FL technique is considered here (shown in Fig. 2). Let consider an ML model, $\mathbf{w}$, is trained using the data sets at M client IoT nodes and K edge computing nodes in a hierarchical and collaborative manner. Let us denote the data set at IoT node m as $\mathcal{D}_m$ and define the local loss function as $F_m(\mathbf{w}) = (1/|\mathcal{D}_m|) \sum_{(x_j,y_j)\in\mathcal{D}_m} l(\mathbf{w} : x_j, y_j)$, where $l(\mathbf{w} : x_j, y_j)$ is the samplewise loss function quantifying the prediction error of the model $\boldsymbol{w}$ on the training sample x_j with respect to (w.r.t.)

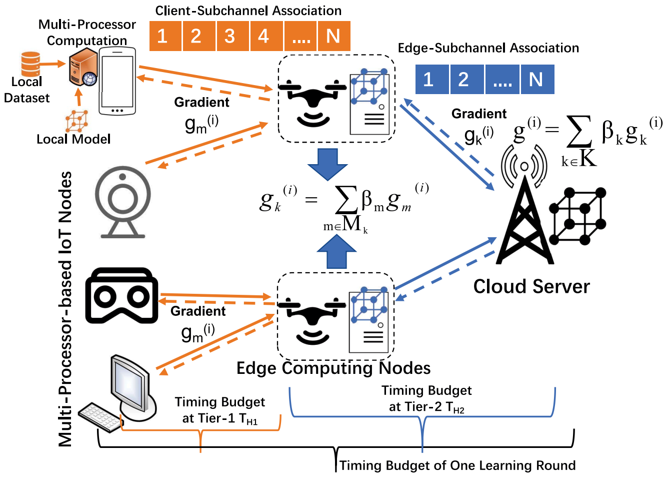

Client-Subchannel Association
1
2
3
4
....
N
Multi-Processor Computation
Local Dataset
Local Model
Gradient
$g_m^{(i)}$
Edge-Subchannel Association
1
2
....
N
Gradient
$g_k^{(i)}$
$g^{(i)} = \sum_{k\in K} \beta_k g_k^{(i)}$
$g_k^{(i)} = \sum_{m\in M_k} \beta_m g_m^{(i)}$
Cloud Server
Gradient
$g_m^{(i)}$
Edge Computing Nodes
Multi-Processor-based IoT Nodes
Timing Budget at Tier-1 T_{H1}
Timing Budget at Tier-2 T_{H2}
Timing Budget of One Learning Round

Fig. 2. Illustration of the hierarchical FL process.

its label y_j. Then, the global loss function based on all the distributed data sets can be written as

$$F(\boldsymbol{w}) = \frac{\sum_{(x_j,y_j)\in\bigcup_m\mathcal{D}_m} l\big(\boldsymbol{w} : x_j, y_j\big)}{|\bigcup_m\mathcal{D}_m|} = \sum_{m=1}^{M} \varsigma_m F_m(\boldsymbol{w}) \tag{1}$$

where ς_m is the weighted contribution of each client IoT node to the global model, owing to the heterogeneous size of their data sets. The learning process is to minimize $F(\boldsymbol{w})$, that is, $\boldsymbol{w}^* = \arg\min F(\boldsymbol{w})$. In this article, we consider the federated gradient averaging concept instead of the model averaging one as the core of the model building process. Let consider an arbitrary learning round i, at which M client IoT nodes (in set $\mathcal{M}^{(i)}$) as well as $\mathcal{K}^{(i)}$ edge computing nodes are selected to participate in the learning process. The cloud server broadcasts the global model $\mathbf{w}^{(i)}$ to all the selected edge computing nodes. Consequently, each client edge computing node k passes that model toward its associated client IoT nodes, denoted by set $\mathcal{M}_k$. Based on the received model $\boldsymbol{w}^i$, each scheduled IoT node m calculates the gradient $\nabla F_m(\boldsymbol{w}^i)$ using its local data set. Upon completion, the local gradients are transmitted to the corresponding edge computing nodes (to which they are associated) for aggregation. As a result, the aggregated gradient at the kth edge computing node results in

$$\boldsymbol{g}_k^{(i)} = \sum_{m\in\mathcal{M}_k^{(i)}} \nabla\varsigma_m F_m\Big(\boldsymbol{w}^{(i)}\Big). \tag{2}$$

Upon constructing the aggregated gradient, each edge computing node $k \in \mathcal{K}$ transmits this to the cloud server for the final aggregation as follows:

$$\boldsymbol{g}^{(i)} = \sum_{k\in\mathcal{K}^{(i)}} \nabla\varsigma_k \boldsymbol{g}_k^{(i)}. \tag{3}$$

The transmission of all IoT nodes as well as the edge computing nodes happen in a synchronized manner. The cloud server updates the global following the stochastic gradient descent (SGD) technique as $\mathbf{w}^{(i+1)} = \boldsymbol{w}^{(i)} - \eta \boldsymbol{w}^{(i)}$, where η is the learning rate. The process is continued until the model convergence is achieved.

B. Computation Resource Consumption

We adopt the multiprocessor-based parallel computing concept at both the IoT nodes and the edge computing nodes [8], [33]. Each client IoT node m (which is associated to edge computing node k) is equipped with Q_{mk} processors (residing in set $\mathcal{Q}_{mk}$). Since the IoT nodes are the data collectors and require to handle a huge data set, we assume that $Q_{mk} > 1$ holds for these nodes. On the other hand, the edge computing nodes collect the locally computed gradients from the IoT nodes and then aggregate these through some simple mathematical manipulation. Consequently, these nodes are relatively less computationally intensive and, hence, $Q_k = 1\ \forall k \in \mathcal{K}$ is a reasonable settings for these nodes. We define $f_{mk}^{q(c)}$ and $f_k^{q(c)}$ (in cycle/s) as the clock frequency of processor q at IoT node m (which is associated to edge computing node k) and UAV k, respectively. It follows that the computing speeds of processor q at those nodes can be defined as $f_{mk}^q = f_{mk}^{q(c)} \times \vartheta_{mk}$ and $f_k^q = f_k^{q(c)} \times \vartheta_k$, respectively, where ϑ_{mk} and ϑ_k denote the number of floating point operations (FLOPs) per cycle at those nodes, respectively. The workload at each processor of the IoT nodes is proportional to the data set assigned to itself. If the data set of the IoT node m is $\mathcal{D}_{mk}$, this becomes partitioned among Q_{mk} processors such that $\bigcup_{q=1}^{Q_{mk}} \mathcal{D}_{mk}^q = \mathcal{D}_{mk}$ holds, where $\mathcal{D}_{mk}^q$ is the assigned data set to processor q. If the workload of processor q at IoT node m is given by W_{mk}^q, the total workload of this node can be written as $W_{mk} = \sum_{q=1}^{Q_{mk}} W_{mk}^q$. According to [5], the workload at each processor q of IoT node m is defined as $W_{mk}^q = \mathcal{D}_{mk}^q \times N_{FLOP}$, where N_{FLOP} is the number of FLOPs at any of the processors. Once the gradient is computed at each processor, all the gradients are collected by any of the processors for taking average of all these. Since averaging of the gradient is the outcome of a simple mathematical manipulation, ignoring this, the computational time of IoT node m (associated to edge computing node k) can be given by

$$t_{mk}^{\text{cmp}} = \max\left\{\left\{\frac{W_{mk}^q}{f_{mk}^q}\right\}_{q=1}^{Q_{mk}}\right\} \quad \forall m \in \mathcal{M}_k.$$

According to [6] and [34], the energy consumption of each processor q at IoT node m can be given by $E_{mk}^q = C_{mk}^q (f_{mk}^q)^3$, where $C_{mk}^q = \Psi_{mk}^q/(\vartheta_{mk})^3$. Here, the unit of coefficient Ψ_{mk}^q is W/(cycle/s)^3 and dependent on the chip architecture. Overall, C_{mk}^q characterizes the computation efficiency of processor q, defined as the rate of power growth in response to the increase of the cubed computing speeds. Given the time duration W_{mk}^q/f_{mk}^q for processor $q\ \forall q \in \mathcal{Q}_{mk}$ to complete their tasks with the workloads W_{mk}^q, the resultant energy consumption at client m can be written as

$$E_{mk}^{\text{cmp}} = \sum_{q=1}^{Q_{mk}} C_{mk}^q W_{mk}^q \big(f_{mk}^q\big)^2 \quad \forall m \in \mathcal{M}_k.$$

C. Communication Energy Consumption

Consider that spectrum is divided into N OFDMA channels at each learning round. After the gradiaents are calculated and aggreagted at each IoT node m, these are transmitted to the edge servers by these channels. We denote the channel gain of IoT node m to the its associated edge server k by h_{mk} and that of edge server k toward the cloud server by h_k, respectively. The nodes (IoT nodes or UAVs) in the system are static or slightly floating without significant deviation, which can be captured by slow block fading model. Consequently, we can say that the channels in the system remain unchanged within one learning round but varies independently and identically over rounds. The channel between any two nodes in the system suffers path loss effect along with small-scale Rayleigh fading. For the sake of simplicity, it is assumed that each node obtains only one frequency channel for transmission. If a_{mk}^n is the binary indicator that channel n is assigned to node m or not, this constraint can be given represented by

$$\sum_{n=1}^{n} a_{mk}^n \leq 1 \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k.$$

In practical systems, it is possible that the condition $M > N$ holds. In such circumstances, a single subchannel is required to be shared by multiple nodes, in which case interference is an obvious phenomenon. Let denote the transmit power of each IoT node m and edge server k at each learning round are given by P_m^{TX} and P_k^{TX}, respectively. Consequently, achievable bitrate in each communication phase for IoT node m can be given by

$$r_{mk} = B \sum_{n=1}^{N} a_m^n \log_2\left(1 + \frac{h_{mk} P_{mk}^{TX}}{\sum_{k=1}^{K} \sum_{\substack{m' \in \mathcal{M}_k \\ m' \neq m}} a_{m'}^n P_{m'k}^{TX} h_{m'k}^n + \sigma^2}\right)$$

where B is the bandwidth of each subchannel. On $L = |\boldsymbol{g}|\alpha$, where α is the buffer bitrate to accurately extract the gradient information at the receiver side. If the duration of each communication round for IoT node m is t_{mk}^{cmm}, the condition $r_{mk}^{\text{cmm}} \geq L$ must hold to achieve a successful training round.

D. Problem Formulation

Let denote the timing budget of computation and communication task at tier-1 and tier-2 are $T_{H_1}^{\text{cmp}}$, $T_{H_1}^{\text{cmm}}$, $T_{H_2}^{\text{cmp}}$, and $T_{H_2}^{\text{cmm}}$, respectively. If the timing budget of one learning round is T, the energy-efficient computation and communication resource allocation problem of each learning round in two-tier hierarchical networks can be formulated as

$$\begin{aligned}
\text{Min} \ & \sum_{k=1}^{K} \sum_{m \in \mathcal{M}_k} \sum_{q=1}^{Q} \omega_{mk} C_{mk}^q W_{mk}^q \left(f_{mk}^q\right)^2 + \sum_{k=1}^{K} C_k M_k \Gamma(W) f_k^2 \\
& + \sum_{k=1}^{K} \sum_{m \in \mathcal{M}_k} t_{mk}^{\text{cmm}} P_{mk}^{TX} + \sum_{k=1}^{K} t_k^{\text{cmm}} P_k^{TX}
\end{aligned} \tag{4}$$

$$\begin{aligned}
& \sum_{q=1}^{Q} W_{mk}^q = W_{mk} \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \\
& f_{mk}^{q,\min} \leq f_{mk}^q \leq f_{mk}^{q,\max} \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \\
& f_k^{\min} \leq f_k \leq f_k^{\max} \quad \forall k \in \mathcal{K} \\
& \sum_{k=1}^{K} \sum_{m \in \mathcal{M}_k} W_{mk} = W
\end{aligned}$$

$$\begin{aligned}
& t_m^{\text{cmp}} = \max\left\{\left\{\frac{W_m^q}{f_m^q}\right\}_{q=1}^{Q}\right\} \quad \forall m \in \mathcal{M} \\
& 0 \leq t_m^{\text{cmp}} \leq T_{H_1}^{\text{cmp}} \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \\
& 0 \leq t_k^{\text{cmp}} \leq T_{H_1}^{\text{cmp}} \quad \forall k \in \mathcal{K} \\
& \sum_{n=1}^{N} a_{mk}^n t_{mk}^{\text{cmm}} B \\
& \quad \times \log_2\left(1 + \frac{P_{mk}^{TX} h_{mk}^n}{\sum_{k=1}^{K} \sum_{\substack{m' \in \mathcal{M}_k \\ m' \neq m}} a_{m'k}^n P_{m'k}^{TX} h_{m'k}^n + \sigma^2}\right) \\
& \qquad \geq L \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \\
& \sum_{n \in \mathcal{N}} a_{mk}^n \leq 1 \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \\
& 0 \leq t_{mk}^{\text{cmm}} \leq T_{H_1}^{\text{cmm}} \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \\
& \sum_{n=1}^{N} a_k^n t_k^{\text{cmm}} B \log_2\left(1 + \frac{P_k^{TX} h_k^n}{\sum_{\substack{k' \in \mathcal{K} \\ k' \neq k}} a_{k'}^n P_{k'}^{TX} h_{k'}^n + \sigma^2}\right) \geq L \quad \forall k \\
& \sum_{n \in \mathcal{N}} a_k^n \leq 1 \quad \forall k \in \mathcal{K} \\
& 0 \leq t_k^{\text{cmm}} \leq T_{H_2}^{\text{cmm}} \quad \forall k \in \mathcal{K} \\
& T_{H_1}^{\text{cmp}} + T_{H_1}^{\text{cmm}} + T_{H_2}^{\text{cmp}} + T_{H_2}^{\text{cmm}} \leq T \\
& a_{mk}^n \in \{0, 1\} \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \quad \forall n \in \mathcal{N} \\
& P_{mk}^{TX} \geq 0, \ P_k^{TX} \geq 0 \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k
\end{aligned} \tag{5}$$

where ω_{mk} is the model divergence weight for IoT node m, which is the indicator of the quality of its data set (i.e., weight divergence).[1] The lower the weight divergence, the closer the distribution of its local model with regard to the global model. $f_{mk}^{q,\min}$ and $f_{mk}^{q,\max}$ are the minimum and maximum frequency scaling of processor q at IoT node m. Moreover, $f_k^{\min}$ and $f_k^{\max}$ are also the minimum and maximum frequency scaling of edge server k. Because of the sorting process owing to the non-IID distribution of the data set, the workload is constrained at each IoT node. Moreover, knowing the fact that a large number of non-IID data may produce low-performing global model, the centralized controller restricts on the total volume of data across all IoT nodes, which further brings total workload constraint of all IoT clients. Note that $\Gamma(W)$ is a known function of the total workload W.

III. Proposed Solution Strategy

As we see in the problem in (4) and (5), the control variables in the problem are coupled with each other, especially computational time and communication time between computation and communication resource allocation. Consequently, we decouple the problem for the computation and communication tasks separately, and solve each problem while isolating the other problem. Then, we provide an iterative approach to solve the entire problem, in which the solution strategies of the

[1] Similar to the model divergence weight, we can incorporate another weight that can capture the Quality-of-Service (QoS) requirement of the client IoT nodes.

subproblems are applied in one alternate manner.[2] Because of the heterogeneous frequency scaling of multiple processors, the completion time of the computation task among different client IoT nodes could be heterogeneous, which results in heterogeneous communication among them. Based on this finding, we propose a greedy client selection scheme while considering both the energy consumption and the importance of client IoT nodes owing to the non-IID data. Finally, we provide an online client–edge node association problem with the objective of achieving better energy consumption as well as better learning efficiency.

A. Energy-Efficient Computation Resource Allocation at Tier-1

We first formulate the computation resource allocation problem at the client IoT nodes targeting on minimizing the computation energy consumption. In this formulation, the association between the client IoT nodes and the edge nodes are known. All the IoT nodes are required to complete the computation task within some time frame $T^{\text{cmp}}_{H_1}$. More specifically, we aim to find workload partitioning among the processors of each IoT node as well as their frequency scaling and workload partitioning among all the client IoT nodes under the total workload constraint at the system

$$\underset{\{W^q_{mk}\},\{W_{mk}\},\{f^q_{mk}\}}{\arg\min} \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k}\sum_{q=1}^{Q}\omega_{mk}C^q_{mk}W^q_{mk}\left(f^q_{mk}\right)^2 \tag{6}$$

$$\sum_{q=1}^{Q} W^q_{mk} = W_{mk} \quad \forall k\in\mathcal{K} \quad \forall m\in\mathcal{M}_k \tag{7}$$

$$f^{q,\min}_{mk} \le f^q_{mk} \le f^{q,\max}_{mk} \tag{8}$$

$$\sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k} W_{mk} = W \tag{9}$$

$$t^{\text{cmp}}_{mk} = \max\left\{\left\{\frac{W^q_m}{f^q_m}\right\}_{q=1}^{Q}\right\} \quad \forall m\in\mathcal{M} \tag{10}$$

$$0 \le t^{\text{cmp}}_m \le T^{\text{cmp}}_{H_1}. \tag{11}$$

From the structure of problems (6)–(11), it is clear that the larger the duration of one learning round (i.e., $T^{\text{cmp}}_{H_1}$), the lower the computation energy consumption. Therefore, assuming that all the clients can take the opportunity to complete the task within time $T^{\text{cmp}}_{H_1}$, the constraint in (10) can be relaxed. As a result, the entire problem can be rewritten as

$$\underset{\{W^q_{mk}\},\{W_{mk}\},\{f^q_{mk}\}}{\arg\min} \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k}\sum_{q=1}^{Q}\omega_{mk}\frac{C^q_{mk}\left(W^q_{mk}\right)^3}{\left(T^{\text{cmp}}_{H_1}\right)^2} \tag{12}$$

$$\sum_{q=1}^{Q} W^q_{mk} = W_{mk} \quad \forall k\in\mathcal{K} \quad \forall m\in\mathcal{M}_k \tag{13}$$

$$\sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k} W_{mk} = W. \tag{14}$$

The problem in (12)–(14) does not have duality gap. Therefore, utilizing the duality concept, by associating the dual variables $\boldsymbol{\lambda} = \{\lambda_{mk}\}_{k\in\mathcal{K},m\in\mathcal{M}_k}$ and β with the constraints in (13) and (14), respectively, the resultant Lagrangian becomes

$$\begin{aligned}\Omega\left(\left\{W^q_{mk}\right\},\{W_{mk}\},\boldsymbol{\lambda},\beta\right) &= \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k}\sum_{q=1}^{Q_{mk}}\frac{\omega_{mk}C^q_{mk}\left(W^q_{mk}\right)^3}{\left(T^{\text{cmp}}_{H_1}\right)^2}\\ &\quad+\sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k}\lambda_{mk}\left[\sum_{q=1}^{Q}W^q_{mk}-W_{mk}\right]\\ &\quad+\beta\left[\sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k}W_{mk}-W\right].\end{aligned}$$

Differentiating $\Omega(.)$ w.r.t. W^q_{mk} and W_{mk} and then equating to 0, we obtain the following two relations:

$$\frac{3\omega_{mk}C^q_{mk}\left(W^q_{mk}\right)^2}{\left(T^{\text{cmp}}_{H_1}\right)^2} + \lambda_{mk} = 0 \quad \forall k \quad \forall m\in\mathcal{M}_k \tag{15}$$

$$-\lambda_{mk} + \beta = 0 \quad \forall k \quad \forall m\in\mathcal{M}_k. \tag{16}$$

From the relation in (15), we can conclude the following relation:

$$C^1_{mk}\left(W^1_{mk}\right)^2 = C^2_{mk}\left(W^2_{mk}\right)^2 = \cdots = C^Q_{mk}\left(W^Q_{mk}\right)^2 \quad \forall m\in\mathcal{M}_k.$$

Then, from the relation in (7), we can conclude the following optimal outcome:

$$\begin{aligned} W^1_{mk} &= \frac{W_{mk}}{A_{mk}} \quad \forall k \quad \forall m\in\mathcal{M}_k\\ W^q_{mk} &= \frac{\sqrt{\frac{C^1_{mk}}{C^q_{mk}}}W_{mk}}{A_{mk}} \quad \forall k \quad \forall m\in\mathcal{M}_k \quad \forall q=2,\ldots,Q \end{aligned} \tag{17}$$

where

$$A_{mk} = 1 + \sum_{q=2}^{Q}\sqrt{\frac{C^1_{mk}}{C^q_{mk}}} \quad \forall k \quad \forall m\in\mathcal{M}_k.$$

With these results, from the relation in (15), we can obtain

$$\lambda_{mk} = -\frac{\omega_{mk}C^1_{mk}(W_{mk})^2}{(TA_{mk})^2} \quad \forall k \quad \forall m\in\mathcal{M}_k.$$

Thus, using the relation in (16) and the constraint in (9), we can derive the optimal workload partitioning among different clients as follows:

$$\begin{aligned} W_{11} &= \frac{W}{\Upsilon}\\ W_{mk} &= \frac{\frac{A_{mk}}{A_{11}}\sqrt{\frac{\omega_{11}C^1_{11}}{\omega_{mk}C^1_{mk}}}}{\Upsilon} \quad \forall k \quad \forall m\in\mathcal{M}_k, m\neq 1 \end{aligned} \tag{18}$$

where

$$\Upsilon = 1 + \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k,m\neq 1}\frac{A_{mk}}{A_{11}}\sqrt{\frac{\omega_{11}C^1_{11}}{\omega_{mk}C^1_{mk}}}.$$

[2] Since we decompose the main problem into subproblems and then solve the original problem by aggregating the solution outcome of all the subproblems, the proposed solution strategy is scalable to multitier networks.

Algorithm 1 Energy-Efficient Computation Resource Allocation Scheme At Tier-1

```
1:  repeat
2:    Compute the optimal $W_{mk}$ and $W_{mk}^{q} \forall k \in \mathcal{K} \forall m \in \mathcal{M}_k \forall q = 1 \cdots Q$ following the relations in (18) and (17).
3:    Computer frequency scaling client $m$ at processor $q$ as $f_{mk}^{q} = W_{mk}^{q}/T_{H_1}^{cmp}$.
4:    if $f_{mk} > f_{mk}^{max} \exists k \in \mathcal{K} \exists m \in \mathcal{M}_k$ then
5:      The problem is infeasible with this setup.
6:      $\Psi \leftarrow \{mk | f_{mk} > f_{mk}^{max}\}$.
7:      $\mathcal{F}_{mk} \leftarrow \{q | f_{mk} > f_{mk}^{max}\}, \ \forall mk \in \Psi$.
8:      $\mathcal{Q}_{mk} \leftarrow \mathcal{Q}_{mk}/\mathcal{F}_{mk}$.
9:      Under this setup, the entire process of this algorithm is called again from line 1.
10:   else
11:     The problem is feasible. However, for the condition $f_{mk}^{q} < f_{mk}^{q,min}$, it implies that process $q$ of node $m$ can accomplish the assigned workload before the timing budget, $T_{H_1}^{cmp}$, and such phenomenon is feasible although further optimization could be applied to such type of clients for the proper distribution of the workload.
12:   end if
13: until The problem becomes feasible.
```

In order to justify the optimality and feasibility of the analytically optimal solution W_{mk} and W_{mk}^{q} $\forall k \in \mathcal{K}$ v$\forall m \in \mathcal{M}_k$ $\forall q = 1 \cdots Q$ using the relations in (18) and (17), we develop a procedure in Algorithm 1.

Lemma 1: Algorithm 1 eventually converges.

Proof: This statement can be proved by the worst case analysis. In the worst case, through discarding the processors from each client one by one (lines 6–8), there will be only single client with one single processor. By assigning the entire workload of the system to this single client, if the frequency scaling constraint of this processor does not satisfy the workload, the original problem becomes infeasible. Therefore, the algorithm will eventually converge even though there is no any obvious break condition for the loop in between lines 1 and 13. ■

Lemma 2: Without any restriction on maximum frequency scaling, only with the minimum frequency scaling constraint, the analytical workload partitioning outcome in (18) and (17) are optimal.

Proof: The proof is simple if we look at the optimal expressions in (18) and (17). These expressions are independent of time duration, only a function of processor coefficient and total workload constraint in the system. After obtaining the workload, if the resultant frequency scaling is less than the minimum one, by reducing the computational time of that processor, frequency scaling can be adjusted as required. This adjustment is feasible, which implies the corresponding processor can complete its task ahead of the timing budget. ■

B. Energy-Efficient Communication Resource Allocation at Tier-1

The objective of this problem is to choose subchannels for the client IoT nodes at tier-1 and their power level for transmitting the gradient information to the associated edge computing nodes under some timing budget ($T_{H_1}^{\text{cmm}}$) such that the overall energy consumption is minimized. Mathematically, we can write the problem as follows:

$$\underset{\{P_{mk}^{TX}\},\{t_{mk}\}}{\arg\min} \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k}\sum_{n=1}^{N} t_{mk} a_{mk}^{n} P_{mk}^{TX} \tag{19}$$

$$\sum_{n=1}^{N} a_{mk}^{n} t_{mk} B\log_2 \times \left(1 + \frac{a_{mk}^{n} P_{mk}^{TX} g_{mk}^{n}}{\sum_{k=1}^{K}\sum_{\substack{m'\in\mathcal{M}_k\\ m'\neq m}} a_{m'k}^{n} P_{m'k}^{TX} g_{m'k}^{n} + \sigma^2}\right) = L \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \tag{20}$$

$$0 \leq t_{mk} \leq T_{H_1}^{\text{cmm}} \tag{21}$$

$$\sum_{n\in\mathcal{N}} a_{mk}^{n} \leq 1 \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \tag{22}$$

$$a_{mk}^{n} \in \{0, 1\} \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k \quad \forall n \in \mathcal{N} \tag{23}$$

$$P_{mk}^{TX} \geq 0 \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k. \tag{24}$$

It is clear from the constraints in (20) and (21) that communication energy consumption will be lower with the larger communication duration and, hence, $t_{mk} = T_{H_1}^{\text{cmm}}$ should be the optimal outcome for this problem. Consequently, the objective function and the constraint in (20) can be written as

$$\underset{\{P_{mk}^{TX}\},\{t_{mk}\}}{\arg\min} \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k}\sum_{n=1}^{N} T_{H_1}^{\text{cmm}} a_{mk}^{n} P_{mk}^{TX} \text{ and} \tag{25}$$

$$\sum_{n=1}^{N} a_{mk}^{n} T_{H_1}^{\text{cmm}} B\log_2 \times \left(1 + \frac{a_{mk}^{n} P_{mk}^{TX} g_{mk}^{n}}{\sum_{k=1}^{K}\sum_{\substack{m'\in\mathcal{M}_k\\ m'\neq m}} a_{m'k}^{n} P_{m'k}^{TX} g_{m'k}^{n} + \sigma^2}\right) = L \quad \forall k \in \mathcal{K} \quad \forall m \in \mathcal{M}_k, \text{ respectively.} \tag{26}$$

The problem in (19)–(24) does not have any duality gap, and thus it can be solved by the dual formulation. By associating the constraint in (26) and (21) by $\boldsymbol{\lambda} = \{\lambda_{mk}\}_{k\in\mathcal{K} m\in\mathcal{M}_k}$ and $\boldsymbol{\gamma} = \{\gamma_{mk}\}_{k\in\mathcal{K} m\in\mathcal{M}_k}$, respectively, the resultant Lagrangian becomes

$$\begin{aligned}
&\Omega\left(\{P_{mk}^{TX}\}, \{a_{mk}^{n}\}, \boldsymbol{\lambda}, \boldsymbol{\gamma}\right) \\
&= \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k}\sum_{n=1}^{N} T_{H_1}^{\text{cmm}} a_{mk}^{n} P_{mk}^{TX} + \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k} \lambda_m \\
&\times \left[- \sum_{n=1}^{N} T_{H_1}^{\text{cmm}} a_{mk}^{n} B\log_2 \right. \\
&\left. \times \left(1 + \frac{a_{mk}^{n} P_{mk}^{TX} g_{mk}^{n}}{\sum_{k=1}^{K}\sum_{\substack{m'\in\mathcal{M}_k\\ m'\neq m}} a_{m'k}^{n} P_{m'k}^{TX} g_{m'k}^{n} + \sigma^2}\right) + L \right] \\
&+ \sum_{k=1}^{K}\sum_{m\in\mathcal{M}_k} \gamma_{mk}\left(\sum_{n=1}^{N} a_{mk}^{n} - 1\right).
\end{aligned}$$

Therefore, the optimal solution to (19)–(24) becomes

$$\max_{\boldsymbol{\lambda},\boldsymbol{\gamma}} \min_{\{P_{mk}^{TX}\}\{a_{mk}^{n}\}} \Omega\left(\{P_{mk}^{TX}\}\{a_{mk}^{n}\}, \boldsymbol{\lambda}, \boldsymbol{\gamma}\right).$$

We solve this problem through a specific procedure. For the given values of dual variables, we first derive the closed form optimal solution of $\{a_{mk}^{n}\}$ and $\{P_{mk}^{TX}\}$. Then, we conduct a search procedure to find the optimal values of the dual variables. For the given values of $\boldsymbol{\lambda}$ and $\{a_{mk}^{n}\}$, we can obtain the optimal value of P_{mk}^{TX} as follows (by differentiating $\Omega(.)$ w.r.t. P_{mk}^{TX}):

$$a_{mk}^{n} - \lambda_m \frac{a_{mk}^{n} B}{1 + \frac{a_{mk}^{n} P_{mk}^{TX} g_{mk}^{n}}{\sum_{k=1}^{K} \sum_{\substack{m' \in \mathcal{M}_k \\ m' \neq m}} a_{m'k}^{n} P_{m'k}^{TX} g_{m'k}^{n} + \sigma^2}}$$

$$\times \frac{a_{mk}^{n} g_{mk}^{n}}{\sum_{k=1}^{K} \sum_{\substack{m' \in \mathcal{M}_k \\ m' \neq m}} a_{m'k}^{n} P_{m'k}^{TX} g_{m'k}^{n} + \sigma^2} = 0$$

$$a_{mk}^{n} - \lambda_m B a_{mk}^{n} \frac{a_{mk}^{n} g_{mk}^{n}}{\sum_{k=1}^{K} \sum_{m \in \mathcal{M}_k} a_{mk}^{n} P_{mk}^{TX} g_{mk}^{n} + \sigma^2} = 0.$$

Solving these $\sum_{k=1}^{K} M_k$ equations, we can obtain the optimal value of P_{mk}^{TX} as follows:

$$\hat{P}_{mk}^{TX} = f\left(B, T_{H_1}^{\text{cmm}}, \{g_{mk}^{n}\}, \boldsymbol{\lambda}\right).$$

To find the optimal values of a_{mk}^{n}, we denote the following expression:

$$\mu_{mk}^{n} = \log_2\left(1 + \frac{a_{mk}^{n} P_{mk}^{TX} g_{mk}^{n}}{\sum_{k=1}^{K} \sum_{\substack{m' \in \mathcal{M}_k \\ m' \neq m}} a_{m'k}^{n} P_{m'k}^{TX} g_{m'k}^{n} + \sigma^2}\right) - L.$$

From the structure of the Lagrangian $\Omega(.)$, we can deduce the optimal value of a_{mk}^{n} as follows:

$$\hat{a}_{mk}^{n} = \begin{cases} 1, & \arg\min_{mk,n} \left|\mu_{mk}^{n}\right| \\ 0, & \text{Otherwise.} \end{cases}$$

As a result, the resultant Lagrangian becomes

$$\Omega(\boldsymbol{\lambda}) = \sum_{k=1}^{K} \sum_{m \in \mathcal{M}_k} \sum_{n=1}^{N} T_{H_1}^{\text{cmm}} \hat{a}_{mk}^{n} \hat{P}_{mk}^{TX} + \sum_{k=1}^{K} \sum_{m \in \mathcal{M}_k} \lambda_m \mu_{mk}^{n}.$$

Now, the solution to (19)–(24) is given by numerically maximizing $\Omega(\boldsymbol{\lambda})$ over $\boldsymbol{\lambda}$. For this, we adopt a subgradient-based search technique and update λ_{mk} as follows:

$$\lambda_{mk}(t+1)$$
$$= \lambda_{mk}(t) - \kappa(t)\Bigg[-\sum_{n=1}^{N} T_{H_1}^{\text{cmm}} \hat{a}_{mk}^{n} B \log_2$$
$$\times \left(1 + \frac{\hat{a}_{mk}^{n} \hat{P}_{mk}^{TX} g_{mk}^{n}}{\sum_{k=1}^{K} \sum_{\substack{m' \in \mathcal{M}_k \\ m' \neq m}} \hat{a}_{m'k}^{n} \hat{P}_{m'k}^{TX} g_{m'k}^{n} + \sigma^2}\right) + L\Bigg].$$

The convergence of this procedure is achieved if $\kappa(t)$ is chosen in an appropriate manner [31]. Given an optimal $\boldsymbol{\lambda}^*$, from the definition of duality, the value of $\Omega(\boldsymbol{\lambda}^*)$ becomes the optimal objective value of (19)–(24). With this, we can easily obtain optimal $\{a_{mk}^{n}\}$ and $\{P_{mk}^{TX}\}$. However, the complexity of this process is very high due to the exhaustive numerical search and, hence, we provide a suboptimal solution of this energy-efficient communication resource allocation problem in Algorithm 2.

Algorithm 2 Energy-Efficient Communication Resource Allocation Scheme at Tier-1

```
1: $\mathcal{M}_n \leftarrow \emptyset,\ \forall n \in \mathcal{N}$.
2: $\mathcal{N}_m \leftarrow \emptyset,\ \forall k \in \mathcal{K} \forall m \in \mathcal{M}_k$.
3: $\mathcal{M}' \leftarrow \mathcal{M}$.
4: $\mathcal{N}' \leftarrow \mathcal{N}$.
5: for $i = 1 \cdots N$ do
6:     Find the best subchannel from set $\mathcal{N}'$ for client IoT node $m \in \mathcal{M}'$ based on $g_{mk}^{n}$ and store this in the variable $\zeta_m^n$. $\forall k \in \mathcal{K} \forall m \in \mathcal{M}_k$
7:     $\{m^*, n^*\} \leftarrow \arg\max_{m,n} \zeta_m^n$.
8:     Set power level $P_{m^*}^{TX}$ for client IoT node $m^*$ at subchannel $n^*$ such that bitrate equivalent to $L$ is achieved.
9:     $\mathcal{M}' \leftarrow \mathcal{M}'/m^*$.
10:    $\mathcal{N}' \leftarrow \mathcal{N}'/n^*$.
11:    $\mathcal{M}_{n^*} \leftarrow \mathcal{M}_{n^*} \bigcup m^*$.
12:    $\mathcal{N}_{m^*} \leftarrow \mathcal{N}_{m^*} \bigcup n^*$.
13: end for
14: for $m \in |\mathcal{M}/\mathcal{M}'|$ do
15:    Consider subchannel $n$ for transmission and compute sum-power required for the transmission of model parameters with size $L$ on this subchannel by all the assigned clients including the ones in set $\mathcal{M}_n$, and store this to $\sum_n,\ \forall n \in \mathcal{N}$.
16:    $n^* \leftarrow \arg\min_{n} \sum_n$.
17:    $\mathcal{N}_{m^*} \leftarrow \mathcal{N}_{m^*} \bigcup n^*$.
18: end for
```

Lemma 3: Algorithm 2 eventually converges.

Proof: From the structure, it is obvious that the loop in between lines 5 and 13 is finite, which runs at most N. In this loop, N clients obtain the subchannels for transmitting their model parameters. The remaining clients $(M - N)$ obtain their subchannels in the loop between lines 14 and 18. Therefore, we can conclude that this algorithm eventually converges. ■

Computational Complexity of Algorithm 2: The outer loop (the lines in between 5 and 13) of this algorithm runs at most N times. Inside this loop, the worst-case computational complexity of line 6 is $O((\sum_{k \in \mathcal{K}} M_k)N)$ (linear search technique is applied). Other steps in this loop have constant complexity. Hence, the overall complexity of this loop becomes $O((\sum_{k \in \mathcal{K}} M_k)N^2)$. Note that inside this loop N users are assigned to subchannels as the decision of one client–subchannel assignment is made in each iteration. Therefore, the loop in between lines 14 and 18 runs at most $O(M - N)$ times. The complexity of line 15 is at most $O(N)$ times. The complexity of the other steps in this loop is constant. Consequently, the complexity of this loop becomes $O(M - N)N$. Combining these two loops, we can deduce that the complexity of this algorithm as $O((\sum_{k \in \mathcal{K}} M_k)N^2 + (M - N)N)$.

C. Energy-Efficient Computation and Communication Resource Allocation at Tier-2

The energy-efficient computation resource allocation problem at tier-2 can be formulated as

$$\begin{aligned} \min \quad & \sum_{k=1}^{K} C_k M_k \Gamma(W) f_k^2 \\ & f_k^{\min} \le f_k \le f_k^{\max} \quad \forall k \in \mathcal{K} \\ & 0 \le t_k^{\mathrm{cmp}} \le T_{H_1}^{\mathrm{cmp}} \quad \forall k \in \mathcal{K}. \end{aligned}$$

From the formulation, it is clear that the computation energy consumption of each edge computing node k is proportional to the number of clients it supports as well as some constant value (which is very small fraction of total workload in the system). Since the edge computing nodes are equipped with a single processor, it is simple to find their workload ($W_k = M_k \Lambda(W)$) as well as the resultant frequency scaling (i.e., $f_k = W_k / T_{H_2}^{\mathrm{cmp}}$) given the timing budget (i.e., $T_{H_2}^{\mathrm{cmp}}$). If the required frequency scaling is not supported by the processor, the problem is either infeasible (i.e., $f_k > f_k^{\max}$) or the timing budget for the computation task should be increased (i.e., $f_k < f_k^{\min}$). For the energy-efficient communication resource allocation, we can adopt the same strategy that is adopted in tier-1.

D. Energy-Efficient Joint Computation and Communication Resource Allocation

We denote computation and communication power consumption at tier-1 and tier-2 by $E_{H_1}^{\mathrm{cmp}}(T_1)$, $E_{H_1}^{\mathrm{cmm}}(T_2)$, $E_{H_1}^{\mathrm{cmp}}(T_1)$, and $E_{H_1}^{\mathrm{cmm}}(T_2)$, respectively. Consequently, we can write the energy minimization problem of each FL round that should be completed by T time units as follows. The objective of this problem is to find the optimal values of $T_{H_1}^{\mathrm{cmp}}$, $T_{H_1}^{\mathrm{cmm}}$, $T_{H_2}^{\mathrm{cmp}}$, and $T_{H_2}^{\mathrm{cmm}}$, such that the overall energy consumption is minimized

$$\begin{aligned} (\chi 1)\text{:} \ \min \quad & E_{H_1}^{\mathrm{cmp}}\left(T_{H_1}^{\mathrm{cmp}}\right) + E_{H_1}^{\mathrm{cmm}}\left(T_{H_1}^{\mathrm{cmm}}\right) \\ & + E_{H_2}^{\mathrm{cmp}}\left(T_{H_2}^{\mathrm{cmp}}\right) + E_{H_2}^{\mathrm{cmm}}\left(T_{H_2}^{\mathrm{cmm}}\right) \\ & T_{H_1}^{\mathrm{cmp}} + T_{H_1}^{\mathrm{cmm}} + T_{H_2}^{\mathrm{cmp}} + T_{H_2}^{\mathrm{cmm}} \le T \\ & T_{H_1}^{\mathrm{cmp}} \ge 0, T_{H_1}^{\mathrm{cmm}} \ge 0, T_{H_2}^{\mathrm{cmp}} \ge 0, T_{H_2}^{\mathrm{cmm}} \ge 0. \end{aligned}$$

We further notice that the computation and communication energy consumption at tier-1 and tier-2 is positively proportional to $T_{H_1}^{\mathrm{cmp}}$, $T_{H_1}^{\mathrm{cmm}}$, $T_{H_2}^{\mathrm{cmp}}$, and $T_{H_2}^{\mathrm{cmm}}$, respectively. Because of this fact, we can shorten the problem as follows. Let denote $E_{H_1}(T_{H_1}^{\mathrm{cmp}}, T_{H_1}^{\mathrm{cmm}}, T_{H_1})$ and $E_{H_2}(T_{H_2}^{\mathrm{cmp}}, T_{H_2}^{\mathrm{cmm}}, T_{H_2})$ be the overall energy consumption (both the computation and communication) at tier-1 and tier-2, respectively, which results in

$$\begin{aligned} (\chi 1.2)\text{:} \ & E_{H_1}\left(T_{H_1}^{\mathrm{cmp}}, T_{H_1}^{\mathrm{cmm}}, T_{H_1}\right) + E_{H_2}\left(T_{H_2}^{\mathrm{cmp}}, T_{H_2}^{\mathrm{cmm}}, T_{H_2}\right) \\ & T_{H_1} + T_{H_2} \le T \\ & T_{H_1} \ge 0, T_{H_2} \ge 0. \end{aligned}$$

This problem has two parts, which are formulated as (χ1.2.1) and (χ1.2.2), which are for the overall energy consumption at tier-1 and tier-2, respectively

$$\begin{aligned} (\chi 1.2.1)\text{:} \ & E_{H_1}^{\mathrm{cmp}}\left(T_{H_1}^{\mathrm{cmp}}\right) + E_{H_1}^{\mathrm{cmm}}\left(T_{H_1}^{\mathrm{cmm}}\right) \\ & T_{H_1}^{\mathrm{cmp}} + T_{H_1}^{\mathrm{cmm}} \le T_{H_1} \\ & T_{H_1}^{\mathrm{cmp}} \ge 0, T_{H_1}^{\mathrm{cmm}} \ge 0. \\ (\chi 1.2.2)\text{:} \ & E_{H_2}^{\mathrm{cmp}}\left(T_{H_2}^{\mathrm{cmp}}\right) + E_{H_2}^{\mathrm{cmm}}\left(T_{H_2}^{\mathrm{cmm}}\right) \\ & T_{H_2}^{\mathrm{cmp}} + T_{H_2}^{\mathrm{cmm}} \le T_{H_2} \\ & T_{H_2}^{\mathrm{cmp}} \ge 0, T_{H_2}^{\mathrm{cmm}} \ge 0. \end{aligned}$$

From the structure of the problems in (χ1), (χ1.2), (χ1.2.1), and (χ1.2.2) w.r.t. the optimization variables $T_{H_1}^{\mathrm{cmp}}$, $T_{H_1}^{\mathrm{cmm}}$, $T_{H_2}^{\mathrm{cmp}}$ and $T_{H_2}^{\mathrm{cmm}}$, we can develop an algorithm (i.e., *Algorithm 3*) to find the optimal time division across different tasks (computation and communication) and different tiers, so that the overall energy consumption is minimized. The proposed algorithm is based on the bisection search technique. The output of this algorithm is the optimal T_{H_1} and T_{H_2}. Note that this algorithm is recursive as the same procedure is called again in lines 9 and 10, however, replacing the problem in line 1 by (χ1.2.1) and (χ1.2.1). While setting the problem in line 1, time duration in line 2 is changed as well as required. In order to find $E_{H_1}^{\mathrm{cmp}}(.)$, $E_{H_1}^{\mathrm{cmm}}(.)$, $E_{H_2}^{\mathrm{cmp}}(.)$, and $E_{H_2}^{\mathrm{cmm}}(.)$ for the given time duration, while solving the problems in (χ1.2.1) and (χ1.2.1), we resort to the contents in Sections III-A–III-C.

Lemma 4: Algorithm 3 eventually converges.

Proof: The algorithm is designed based on the bisection search technique. Therefore, no matter the time duration of a granular time slot is, the condition in line 24 is satisfied, which results in termination of the loop. On the other hand, although the algorithm is recursively called in lines 10 and 11, there are base conditions in lines 12 and 15, which results in the termination of this recursive algorithm. Basically, the algorithm is called recursively only four times due to lines 10 and 11. Thus, we can conclude that the algorithm will eventually converge. ■

Computation Complexity of Algorithm 3: If the time duration of the granular time unit is δ, the complexity of this algorithm is $O(\log_2(T/\delta))$.

Remark 1: In this article, we have optimized energy efficiency of a single FL round. If the energy consumption of one single FL round is $\sum_p$ and total learning rounds is Λ, the total energy consumption of the entire process can be written as $\Lambda \sum_p$. Another observation from our study here is, lower the duration of one learning round, the higher the energy consumption (although the energy consumption and the duration of one learning round is not linear or straightforward). If we reduce the duration of one learning round, the total learning time is reduced. From this aspect, the learning performance is increased if the duration of one learning round is decreased. However, with the reduced duration of one learning, more energy is consumed at each round, which results in possible scarcity of energy for the future rounds as both the IoT clients and UAV-based edge computing nodes are energy constrained. Consequently, further learning process may not be continued due to the scarcity of energy,

Algorithm 3 Energy-Efficient Joint Computation and Communication Resource Allocation Scheme

```
1: Problem: $(\chi 1.2)$.
2: Time Duration: T.
3: $I_{lo} \leftarrow 0$.
4: $I_{hi} \leftarrow T$.
5: repeat
6:     $I_{mid} \leftarrow \frac{I_{lo}+I_{hi}}{2}$.
7:     $I+ \leftarrow I_{mid} + \Delta$.
8:     $I- \leftarrow I_{mid} - \Delta$.
9:     if Problem in line 1 is $(\chi 1.2)$ then
10:        $E+ \leftarrow$ The sum of the objective values of $(\chi 1.2.1)$ and $(\chi 1.2.2)$, which can be obtained through calling the same procedure with the replacement of the problem in line 1 and Time Duration in line 2 by $(\chi 1.2.1)$ and $I+$, and $(\chi 1.2.2)$ and $T - I+$, respectively.
11:        $E- \leftarrow$ The sum of the objective values of $(\chi 1.2.1)$ and $(\chi 1.2.2)$, which can be obtained through calling the same procedure with the replacement of the problem in line 1 and Time Duration in line 2 by $(\chi 1.2.1)$ and $I-$, and $(\chi 1.2.2)$ and $T - I-$, respectively.
12:    else if Problem in line 1 is $(\chi 1.2.1)$ then
13:        $E+ \leftarrow E_{H_1}^{cmp}(I+) + E_{H_1}^{cmm}(T - I+)$.
14:        $E- \leftarrow E_{H_1}^{cmp}(I-) + E_{H_1}^{cmm}(T - I-)$.
15:    else if Problem in line 1 is $(\chi 1.2.1)$ then
16:        $E+ \leftarrow E_{H_2}^{cmp}(I+) + E_{H_2}^{cmm}(T - I+)$.
17:        $E- \leftarrow E_{H_2}^{cmp}(I-) + E_{H_2}^{cmm}(T - I-)$.
18:    end if
19:    if $E+ > E-$ then
20:        $I_{hi} \leftarrow I-$.
21:    else
22:        $I_{lo} \leftarrow I+$.
23:    end if
24: until $I_{lo} > I_{hi}$
25: $I_{H_1} \leftarrow \frac{I_{lo}+I_{hi}}{2}$.
26: $I_{H_2} \leftarrow T - I_{H_1}$.
27: The output of this algorithm is the computed outcome from the objective function in the problem stated at line 1 and split time duration $I_{H_1}$ and $I_{H_2}$ considering the timing budget stated at line 2.
```

which results in poor learning performance. Therefore, besides necessity of the energy-efficient resource allocation in each learning round within a time budget, there is an acute necessity for setting the duration of each learning round so that the tradeoff, between the reduction of total learning duration and availability of energy resource for the further rounds, is achieved. On the other hand, client IoT nodes may hold non-IID data because of their restriction-free data collection process. According to existing many studies, we see that under such condition, not necessarily more the client nodes, better the learning performance. Rather, the global model can be worse if the number of participating clients with non-IID data increases. On the other hand, the higher the number of clients, the more the energy consumption for the FL process. However, for the sake of green communications to save the world, the number of clients should be determined in an appropriate manner so that a robust FL model can be constructed.

E. Offline Energy-Efficient Client Scheduling Scheme

Let denote E_m^{cmp} and E_k^{cmp} be the computation energy consumption of client IoT node $m \in \mathcal{M}$ and edge computing node $k \in \mathcal{K}$, respectively. Moreover, we define two types of binary variables $x_m, m \in \mathcal{M}$ and y_{mk}, $m \in \mathcal{M}, k \in \mathcal{K}$. x_m indicates that client m is selected or not, y_{mk} indicates whether client m is associated to edge computing node k or not. Based on this, we define the communication energy consumption of client m under the association with edge computing node k by $E_{mk}^{\text{cmm}}(.)$. Moreover, we have divergence weight for each client m under the association with edge node k as ω_{mk}. Note that the divergence weight of each client m is not fixed. Rather, for different associated edge nodes, each client m holds different divergence weight according to [28]. Based on this, the client–edge assignment as well as the energy-efficient client selection problem can be formulated as

$$\begin{aligned}\min\ & \sum_{m\in\mathcal{M}}\sum_{k=1}^{K} x_m y_{mk}\omega_{mk}\left[E_m^{\text{cmp}}(.) + E_{mk}^{\text{cmm}}(.)\right]\\ & + \sum_{k=1}^{K}\left[E_k^{\text{cmp}}(.) + E_k^{\text{cmm}}(.)\right]\\ & \sum_{m\in\mathcal{M}} x_m \le M^{\max}\\ & \sum_{m\in\mathcal{M}} x_m y_{mk} \le M_k^{\max} \quad \forall k \in \mathcal{K},\end{aligned}$$

where $M^{\max}$ is the maximum number of scheduled clients demanded by the FL authority, and $M_k^{\max}$ is the maximum number of clients that edge computing node k can support owing to its constraints on frequency scaling or processing speed. The problem consists of other common constraints in the problem in (4) and (5). Clearly, due to the coupling of discrete and continuous variables, the formulation is a mixed integer programming problem and, hence, intractable. Using the sophisticated optimization tools, such as branch and bound technique, the solution of this problem may be achievable. However, the complexity of these techniques is very high and, hence, we propose a low-complexity suboptimal algorithm (shown in Algorithm 4) for client–edge assignment and client scheduling while considering the overall energy minimization and model divergence weight of each client. To facilitate this algorithm, we would like to note that the computation energy consumption of each client m is independent of the edge computing node to which it is associated and, hence, a constant workload $\tilde{W} = W/M$ (W is the total workload constrained by the FL authority) is considered for all the scheduled clients. Based on this, we solve the workload partitioning and frequency scaling problem for each client m following the strategy in Section III-A, which results in the computation energy consumption of each client m as follows:

$$\begin{aligned}W_m^1 &= \frac{\tilde{W}}{A_m}\\ W_m^q &= \frac{\sqrt{\frac{C_m^1}{C_m^q}}\tilde{W}}{A_m} \quad \forall q = 2, \ldots, Q\end{aligned} \tag{27}$$

where

$$A_m = 1 + \sum_{q=2}^{Q}\sqrt{\frac{C_m^1}{C_m^q}}.$$

Algorithm 4 Energy-Efficient Client–Edge Association and Communication Resource Allocation Scheme

```
1: $\mathcal{M}_n \leftarrow \emptyset,\ \forall n \in \mathcal{N}$.
2: $\mathcal{N}_m \leftarrow \emptyset,\ \forall k \in \mathcal{K} \forall m \in \mathcal{M}_k$.
3: $a_{mk}^{n} \leftarrow 0, \forall k \in \mathcal{K} \forall m \in \mathcal{M}_k \forall n \in \mathcal{N}$.
4: $\mathcal{M}' \leftarrow \mathcal{M}$.
5: $\mathcal{N}' \leftarrow \mathcal{N}$.
6: for $i = 1 \cdots N$ do
7:     Find the best subchannel from set $\mathcal{N}'$ and the best edge computing node from set $\mathcal{K}$ (while considering the constraint $M_k^{max}$) for client IoT node $m \in \mathcal{M}'$ based on $\omega_{mk}(E_m^{cmp} + E_{mk}^{cmm})$ and store this in the variable $\xi_{mk}^{n}$, $\forall k \in \mathcal{K} \forall n \in \mathcal{N}$
8:     $\{m^*, k^*, n^*\} \leftarrow \arg\max_{m,k,n} \xi_{mk}^{n}$.
9:     $a_{m^*k^*}^{n^*} \leftarrow 1$.
10:    $\mathcal{M}' \leftarrow \mathcal{M}'/m^*$.
11:    $\mathcal{N}' \leftarrow \mathcal{N}'/n^*$.
12:    $\mathcal{M}_{n^*} \leftarrow \mathcal{M}_{n^*} \bigcup m^*$.
13:    $\mathcal{N}_{m^*} \leftarrow \mathcal{N}_{m^*} \bigcup n^*$.
14: end for
15: for $m \in |\mathcal{M}/\mathcal{M}'|$ do
16:    Consider subchannel $n$ $\forall n \in \mathcal{N}$ and edge computing node $k$, $\forall k \in \mathcal{K}$ (while considering the constraint $M_k^{max}$) for transmission and compute sum-power required for the transmission on this subchannel by all the assigned clients including the ones in set $\mathcal{M}_n$, and store this to $\sum_{kn}$, $\forall k \in \mathcal{K} \forall n \in \mathcal{N}$. Let denote the previous communication energy consumption before this assignment on subchannel $n$ is given by $\sum_n^{prev}$.
17:    $(k^*, n^*) \leftarrow \arg\min_{k,n} \omega_{mk}(E_m^{cmp} + \sum_{kn} - \sum_{n})$.
18:    $a_{mk^*}^{n^*} \leftarrow 1$.
19:    $\mathcal{N}_{m^*} \leftarrow \mathcal{N}_{m^*} \bigcup n^*$.
20: end for
21: Check the feasibility of the computation energy consumption for edge computing node $k \in \mathcal{K}$ whether its assigned workload satisfies the frequency scaling constraint or not. If not, $M_k^{max}$ of the corresponding edge computing node is reduced by 1, and the client-edge association algorithm is called again from line 1 with the new setup. In the case of feasibility achievement, the algorithm is terminated here with the obtained resource assignment solution.
```

However, the communication energy consumption of each client node m is related to the edge computing node to which it is associated with. Consequently, inside the loop in between lines 7 and 8, each client m finds its best edge–subchannel pair based on its both computation and communication energy consumption and model divergence weight. Upon finding the best subchannel-edge pair for all the client IoT nodes, each iteration of the loop (lines $6-15$) finds the best client–edge–subchannel triplet with the lowest energy consumption. This loop is continued until all the N subchannels are assigned. Then, inside the loop (in between lines 15 and 20), the remaining clients try to find the best subchannel-edge pair based on the overall energy consumption and model divergence weight. Eventually, in each iteration of this loop, each client receives its best subchannel-edge pair. Note that the maximal number of serving clients $M_k^{\max}$ is considered while choosing the edge computing node for each client. Once all the $M^{\max}$ clients are scheduled, the feasibility of all the edge computing nodes is studied in terms of their constraints on frequency scaling and processing speed. If the feasibility is not achieved, $M_k^{\max}$ is reduced by one from the corresponding edge computing node, and the entire tasks in lines 1–20 are continued again. Finally, when the solution becomes feasible after all the client–edge–subchannel assignment, the energy-efficient computation resource allocation scheme in Section III-A is called again for the refined workload partitioning and frequency scaling.

Lemma 5: Algorithm 4 eventually converges.

Proof: In the loops (lines between 6 and 14 and 15 and 20), all the clients obtain their edge nodes as well as the subchannels for the transmission of model parameters, which are decided based on both the computation and communication energy consumption. Upon obtaining this resource assignment, each edge computing node checks its workload which is mainly based on the frequency scaling constraint. If any edge computing node finds the infeasible workload assignment to itself, it reduces maximum allowable clients (i.e., $M_k^{\max}$) by 1, and then the entire process is called again from line 1. If the system has feasible settings, all edge computing nodes eventually pass their workload constraints, and then the algorithm terminates. ■

Computational Complexity of Algorithm 4: The complexity of the loop in between lines 6 and 14 is $O(MN^2K)$ while considering all the detailed steps. In the similar manner, the complexity of the loop in between lines 15 and 20 is $O((M^{\max} - N)(M - N)K)$. If the edge computing nodes are not feasible (line 21), the process may be continued multiple times (say Ψ). In the case of feasibility, the complexity for the energy-efficient resource allocation tasks of the edge computing nodes becomes $O(KN^2)$ (following Algorithm 2). Thus, the overall complexity of this algorithm is $O(MN^2K + (M^{\max} - N)(M - N)K + KN^2)$.

F. Online Energy-Efficient Client Scheduling Scheme

The discussed client–edge assignment and client scheduling scheme in the previous section is offline. However, it is possible that the clients may enter into the system one by one, not at a time altogether. In such a type of systems, the client scheduling scheme will be different. For each incoming client, its feasibility to participate in the FL process is given as follows. Based on the size of the data set, each client is associated with some workload although all these may not be due to IID data set. Consequently, we calculate its computation energy consumption following the relation in (27) and the associated workload. Then, combining this energy consumption, it finds its best edge–subchannel pair based on the overall energy consumption and model divergence weight. Once the incoming client enters into the system, FL performance of the system is evaluated in terms of model quality. If the model quality is not improved, this implies that the entering client has a portion of non-IID data and, hence, removing those data in the FL training process results in less workload. Thus, with the reduced workload, the incoming client finds again its best edge–subchannel pair. This process is continued until its workload becomes 0 (i.e., the client's all data are non-IID) or the model quality is improved due to the portions of IID data the client has.

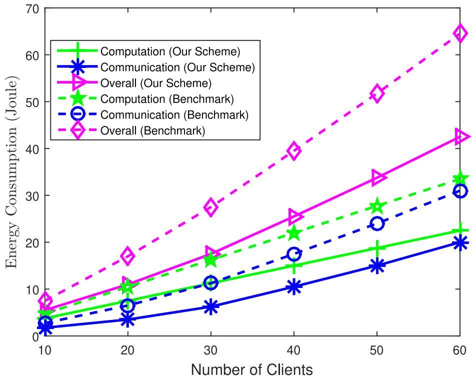


Fig. 3. Comparison of energy consumption with the increasing number of clients.

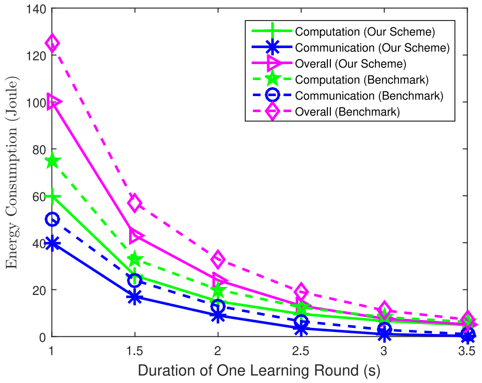


Fig. 5. Comparison of energy consumption with the increasing duration of one learning round when $M = 40$.

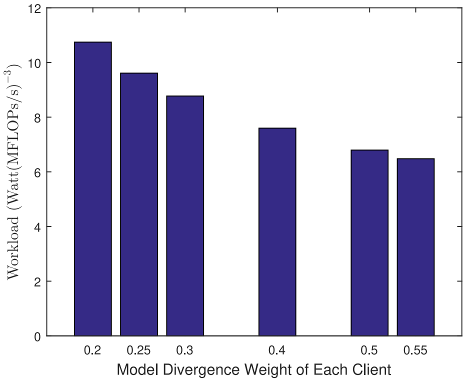
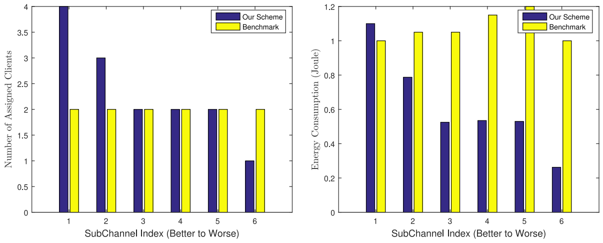


Fig. 4. Energy consumption across different clients for the purpose of computation when $M = 40$.



(a) (b)

Fig. 6. (a) Number of clients and (b) assigned energy level to each subchannel for the purpose of communication when $M = 40$.

IV. Performance Evaluation

Followed by the description of the methodology, we demonstrate the simulation results to verify the effectiveness of the proposed energy-efficient computation and communication resource allocation scheme for two-tier hierarchical FL networks.

A. Simulation Setup

We consider a two-tier hierarchical FL network as depicted in Fig. 1. Unless otherwise specified, there are 60 client IoT nodes and 12 edge computing nodes and one cloud server for the final aggregation of the model. Unless otherwise specified, all the client IoT and edge computing nodes are located randomly located over a region and the cloud resides at the corner of this region. There are four processors in each client IoT node and one processor in the UAV-based edge computing node. The efficiency of all the processors at each client IoT node are not same, rather the coefficient (C_{mk}^{q} $\forall k \in \mathcal{K}$ $\forall m \in \mathcal{M}_k$ $\forall q$) of these processors is taken randomly from the ranges of [0.01, 0.03] [0.03, 0.05], [0.05, 0.07] and [0.07, 0.09] W(MFLOPs/s)$^{-3}$, respectively. The minimum and maximum frequency scaling of these processors is taken from the ranges of [1, 10], respectively. All these parameters are consistent with that of Samsung Exynos 5422 System on Chip (SoC) [4]. As of the communication resource, we consider a system bandwidth of 5 MHz consisting of 512 OFDM tones, grouped into 32 subchannels (eight adjacent tones per subchannel [35], that corresponds to the Band AMC mode of the 802.16 d/e standard. Small-scale fading component of the channel gain between two nodes follow the Rayleigh distribution with mean 1, and to capture the large-scale fading component we consider the path loss exponent as 2. We further consider the noise over all channels are uniform and set to $N_0 = 10^{-9}$ W/Hz. The ML task that we consider here is to classify handwritten digits of the widely used MNIST data set. Each device is randomly assigned with 20 samples, which include both IID and non-IID data. The classification model is constructed based on a convolutional neural network (CNN), which consists of two 5×5 convolution layers equipped with ReLU activation function. Each of the convolution layers is followed by 2×2 pooling unit, a fully connected ReLU activation function-equipped layer with 50 units and a final softmax output layer. The total number of parameters in this model is 21 840. Considering the size of the samples that each client holds, the total computation workload is considered here as 500 MFLOPs. Furthermore, we suppose that each parameter of the training model gradient is quantized into 16 bits, which results in the transmission overhead L as 3.49×10^55 bits. In the following simulation results, we consider a benchmark scheme, the computation and communication resource allocation strategies of which are given as follows. For the computation task, the total workload of the system is equally subdivided among all the client IoT nodes and then the workload of each client is subdivided equally

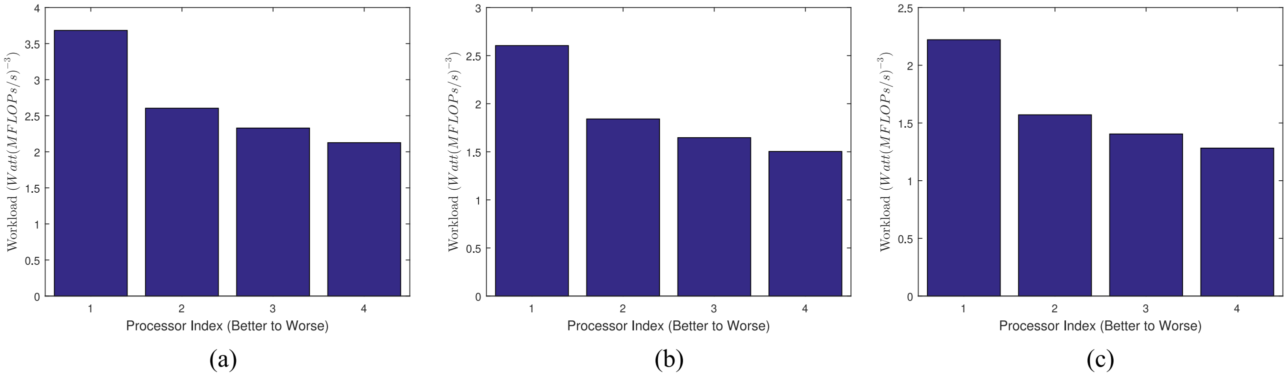


Fig. 7. Allocated workload across different processors. (a) Client with divergence weight 0.2. (b) Client with divergence weight 0.4. (c) Client with divergence weight 0.55.

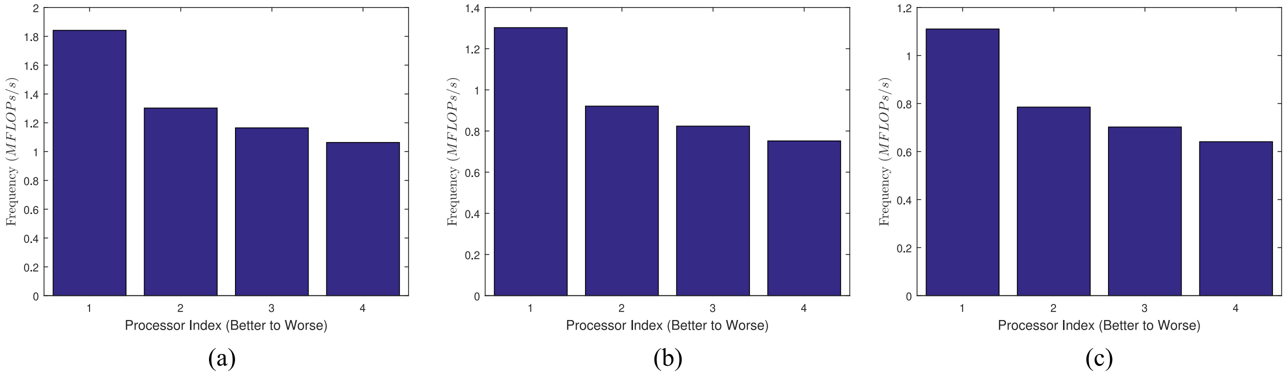


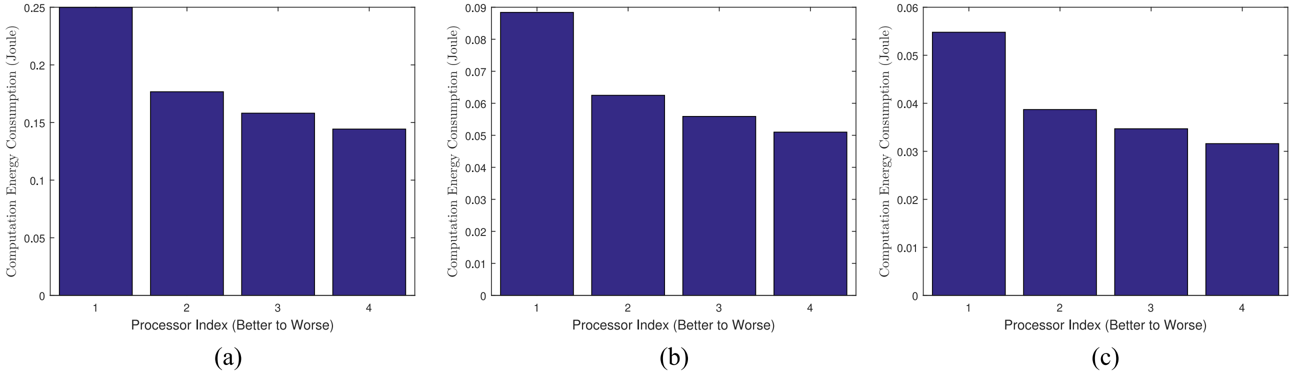
Fig. 8. Frequency scaling across different processors. (a) Client with divergence weight 0.2. (b) Client with divergence weight 0.4. (c) Client with divergence weight 0.55.

among its all processors. On the other hand, for the communication task, the subchannels are assigned to the clients in a round robin fashion.

B. Simulation Results

In Fig. 3, we plot the computation and communication energy consumption as well as the overall one with the increasing number of users. Note that the number of edge computing nodes is the (1/5)th of the number of client IoT nodes in this figure. Both the clients and edge computing nodes are distributed over a region in a uniform manner, and the cloud server is at the corner of the region for the global aggregation. The larger the number of clients, the larger the energy consumption no matter the task is for computation or communication. This trend is common and obvious for our resource allocation scheme and the benchmark one. In order to understand the results in a detailed manner, we plot Fig. 4, in which workloads are projected for different clients with difference model divergence weight. We see that the smaller the model divergence weight the higher the workload volume assigned to the clients. On the other hand, to see the detailed distribution of workload among different processors across clients with difference model divergence weight, we plot Fig. 7. Since all the clients in the system have similar type of processors with similar type of efficiency, the workload distribution among different processors across different clients are symmetric although different clients are assigned to different level of workload owing to their different model divergence weight. Similar phenomenon is observed in Figs. 8 and 9 for the aforementioned discussed reason. The higher the workload the higher the frequency scaling of the processors owing to their definition. The higher the workload and frequency scaling, the higher the energy consumption which is obvious in Fig. 9. In these detailed figures, we have not plot data from the benchmark scheme as this equally subdivides workload among different clients and processors, which does not carry much information. On the other hand, to understand the detailed mechanism of the energy-efficient communication resource allocation scheme, we plot Fig. 6(a) and (b). In these figures, we plot the number of clients assigned to each subchannel and the corresponding energy consumption. We see that the best quality subchannel can serve even four clients without consuming much energy for the communication purpose compared to the other subchannels. As the quality of the subchannel gets worse, the lower the clients they can serve in order to avoid much energy consumption. On the other hand, the benchmark scheme is oblivious about subchannel quality or energy consumption and, hence, the number of clients assigned to each subchannel is random which results in higher energy consumption.

In Fig. 5, we plot the computation and communication energy consumption, as well as the overall one, with the increasing duration of one learning round. The higher the



(a)

(b)

(c)

Fig. 9. Computation energy consumption across different processors. (a) Client with divergence weight 0.2. (b) Client with divergence weight 0.4. (c) Client with divergence weight 0.55.

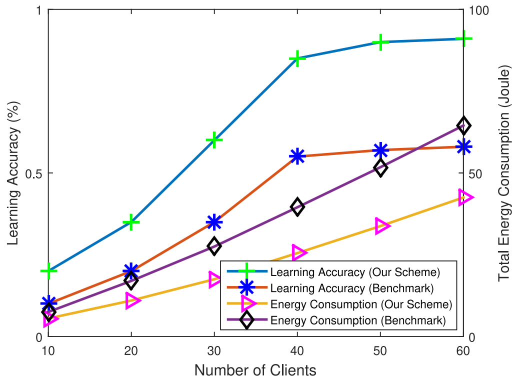


Fig. 10. Comparison of energy consumption and learning accuracy with the increasing number of scheduled users.

duration of one learning round, the lower the computation and communication energy consumption, which is true for the benchmark and our resource allocation schemes. This observation quite fits with our previous analysis. However, since our scheme assigns computation and communication resources among the entities and components of the network in an energy-efficient manner, the resultant energy consumption is much lower compared to the benchmark scheme, which is oblivious about energy efficiency in the case of resource allocation. We see that the energy consumption by our scheme can be reduced even by 25% compared with the benchmark scheme.

In Fig. 10, we plot the outcome of the energy-efficient client scheduling scheme (proposed in Section III-E) on the overall energy consumption of the system as well as the test accuracy. In this figure, the number of edge computing nodes is fixed no matter the number of client IoT nodes is. We assume that all the clients and edge computing nodes are randomly distributed over a region in an uniform manner. Moreover, the benchmark scheme considers that the clients are scheduled in a random fashion. The higher the number of scheduled clients, the lower the energy consumption and the higher the learning accuracy, which confers with our previous analysis. One of the reasons for the skewed performance of the benchmark scheme is high energy consumption, which results in no energy being available to continue the training process. Non-IID data are also another valid reason of the bad performing model by the benchmark scheme. On the other hand, our proposed client scheduling scheme chooses the client–edge assignment in an appropriate manner such that model distribution of different edge computing nodes align with each other. We see that our resource allocation scheme outperforms the benchmark scheme by 40% in terms of both the energy consumption and the learning accuracy.

V. Conclusion

In order to avoid the computation and communication bottleneck situation in the case of single centralized aggregator and skewed global model owing to the non-IID data, we considered a two-tier hierarchical FL network in this article. We proposed an energy-efficient joint computation and communication resource allocation scheme for such a type of networks, the objective of which is to allocate computation and communication resources in such a way that the overall energy consumption is minimized within one learning round. Upon formulating the problem, we noted that the problem is intractable due to the coupling of several continuous and discrete variables. As a result, we decoupled the computation and communication resource allocation problems at each tier and then solved the entire problem jointly through an iterative bisection search-based algorithm. Extensive simulation on real data set was conducted to verify the effectiveness and efficiency of the proposed energy-efficient resource allocation scheme. The results further revealed that the learning performance is not only dependent on the computation and communication energy consumption of the FL process but also the model divergence weight owing to the non-IID data at client IoT nodes. As of the future work, we can extend the proposed solution strategy for general multitier hierarchical or scattered FL networks.

References

[1] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6380–6391, Jul. 2020.

[2] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.

[3] S. R. Pokhrel and J. Choi, "Improving TCP performance over WiFi for Internet of Vehicles: A federated learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6798–6802, Jun. 2020.

[4] A. K. Singh, K. R. Basireddy, A. Prakash, G. V. Merrett, and B. M. Al-Hashimi, "Collaborative adaptation for energy-efficient heterogeneous mobile SoCs," *IEEE Trans. Comput.*, vol. 69, no. 2, pp. 185–197, Feb. 2020.

[5] J. Liu, P. H. Chou, and N. Bagherzadeh, "Communication speed selection and functional partitioning for low-energy on-chip networked multiprocessor," *IEEE Micro*, early access, Oct. 2002. [Online]. Available: https://dl.acm.org/doi/10.1145/581199.581205

[6] Y. Lee, K. G. Shin, and H. S. Chwa, "Thermal-aware scheduling for integrated CPUs–GPU platforms," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5, p. 1–25, 2019.

[7] S. Naffziger. "AMD's Commitment to Accelerating Energy Efficiency." [Online]. Available: https://www.amd.com/system/files/documents/energy-efficiency-whitepaper.pdf (Accessed: Jun. 2014).

[8] P. Devriend, "Multi-processor and frequency scaling," in *Proc. Linux Symp.*, 2004, pp. 168–180.

[9] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," 2019, *arXiv:1911.02417*.

[10] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.

[11] 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," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[12] J. Yao and N. Ansari, "Enhancing federated learning in fog-aided IoT by CPU frequency and wireless power control," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3438–3445, Mar. 2021.

[13] 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.

[14] Y. He, J. Ren, G. Yu, and J. Yuan, "Importance-aware data selection and resource allocation in federated edge learning system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13593–13605, Nov. 2020.

[15] Y. Luo, J. Xu, W. Xu, and K. Wang, "Sliding differential evolution scheduling for federated learning in bandwidth-limited networks," *IEEE Commun. Lett.*, vol. 25, no. 2, pp. 503–507, Feb. 2021.

[16] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless IoT networks," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3394–3409, Mar. 2021.

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

[18] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.

[19] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23920–23935, 2020.

[20] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang, "Client selection for federated learning with non-IID data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24462–24474, 2021.

[21] H. Sun, S. Li, F. R. Yu, Q. Qi, J. Wang, and J. Liao, "Toward communication-efficient federated learning in the Internet of Things with edge computing," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11053–11067, Nov. 2020.

[22] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, Nov. 2020.

[23] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.

[24] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive MIMO for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[25] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[26] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 48–54, Jan. 2021.

[27] Z. Yang, M. Chen, K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6G: Applications, challenges, and opportunities," 2021, *arXiv:2101.01338*.

[28] N. Mhaisen, A. A. Abdellatif, A. Mohamed, A. Erbad, and M. Guizani, "Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 55–66, Jan./Feb. 2022.

[29] W. Wu, L. He, W. Lin, and R. Mao, "Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1539–1551, Jul. 2021.

[30] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, "HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.

[31] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Sci., 1999.

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[33] J. Lee, M. Samadi, Y. Park, and S. Mahlke, "Transparent CPU-GPU collaboration for data-parallel kernels on heterogeneous systems," in *Proc. IEEE PACT*, 2013, pp. 245–256.

[34] C. Liu, J. Li, W. Huang, J. Rubio, E. Speight, and F. X. Lin, "Power-efficient time-sensitive mapping in heterogeneous systems," in *Proc. IEEE PACT*, Minneapolis, MN, USA, 2012, pp. 23–32.

[35] J. Huang, V. G. Subramanian, R. Agrawal, and R. Berry, "Joint scheduling and resource allocation in uplink OFDM systems for broadband wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 2, pp. 226–234, Feb. 2009.