IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 21, 2021, accepted May 25, 2021, date of publication May 28, 2021, date of current version June 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3084689

An Energy-Efficient Fine-Grained Deep Neural Network Partitioning Scheme for Wireless Collaborative Fog Computing

EMRE KILCIOGLU, HAMED MIRGHASEMI, IVAN STUPIA, AND LUC VANDENDORPE, (Fellow, IEEE)
ICTEAM/ELEN, Université Catholique de Louvain, 1348 Ottignies-Louvain-la-Neuve, Belgium
Corresponding author: Emre Kilcioglu (emre.kilcioglu@uclouvain.be)

This work was supported by F.R.S.-FNRS under the EOS Research Project MUSEWINET under EOS Project 30452698.

ABSTRACT Fog computing is a potential solution for heterogeneous resource-constrained mobile devices to collaboratively operate deep learning-driven applications at the edge of the networks, instead of offloading the computations of these applications to the powerful cloud servers thanks to the latency reduction, decentralized structure, and privacy concerns. Compared to the mobile cloud computing concept where computation-intensive deep learning operations are offloaded to the powerful cloud servers, making use of the computing capabilities of resource-constrained devices can improve the delay performance and lessen the need for powerful servers to execute such applications by considering a collaborative fog computing scenario with deep neural network (DNN) partitioning. In this paper, we propose an energy-efficient fine-grained DNN partitioning scheme for wireless collaborative fog computing systems. The proposed scheme includes both layer-based partitioning where the DNN model is divided into layer by layer and horizontal partitioning where the input data of each layer operation is partitioned among multiple devices to encourage parallel computing. A convex optimization problem is formulated to minimize the energy consumption of the collaborative part of the system by optimizing the communication and computation parameters as well as the workload of each participating device and solved by using the primal-dual decomposition and Lagrange duality theory. As can be observed in the simulation results, the proposed optimized scheme makes a notable difference in the energy consumption compared to the non-optimized scenario where the workload distribution is equal for all participating devices but the communication and computation parameters are still optimized, so it is a quite challenging bound to be compared.

INDEX TERMS Convex optimization, deep convolutional neural network, energy efficiency, fog computing, DNN partitioning, wireless collaborative computing.

I. INTRODUCTION

Deep learning (DL) is pervasively utilized in a variety of latency-constrained applications including virtual/augmented reality, high-quality video processing, voice/face recognition and autonomous driving due to providing inferences with high accuracy [1]–[3]. However, its high accuracy comes together with the high computational demands [4]. Most DL-driven applications are computation-intensive because of the following two reasons:

- The depth of the DL model (i.e. massive number of layers),
- Big input data dimension.

Therefore, deploying these applications into a single resource-constrained mobile node (MN) is infeasible since it is not able to perform the whole DL operation by itself due to its limited computing capability, battery level and memory [5]–[7].

To fulfill the computational requirements of DL-driven applications, the initial idea is to use the concept of mobile cloud computing (MCC) [8]–[10], where the computation-intensive DL operations are implemented in the centralized powerful cloud servers [11]. In this approach,

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco Rafael Marques Lima.

MNs are only utilized as data collectors and they do not have any DL model implemented on them. However, MCC has some disadvantages to be used for DL-driven applications such as intolerable latency [11]–[14], backhaul network congestion [13]–[15], too centralized structure and some privacy concerns [15], [16]. An alternative solution to mitigate the effect of the aforementioned problems is the mobile edge computing (MEC) paradigm [17]–[24] in which the powerful servers are placed at the edge of the networks in close proximity to MNs [12], [13]. With MEC, much lower latency, reduced network congestion, less centralized structure, and more privacy can all be achieved [3], [4]. MNs can partially or fully offload DL operations to a MEC server close to them so that they make use of the power of MEC servers while saving their limited energy at the same time.

In the absence of powerful MEC servers nearby, a similar concept can be considered, called fog computing [25], that is a generalized form of MEC [26], in which lightweight DL computations can be operated in not only powerful MEC servers, but also the resource-constrained MNs, collaboratively. The formal definition of fog computing is introduced by Vaquero and Rodero-Merino [27] as *'a huge number of heterogeneous ubiquitous and decentralized devices [that] communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties'*. As there are lots of MNs connected to the networks like smartphones, wearable devices, and smart vehicles, some of them surely stay idle for a while [28] and their computing capabilities can be used by partitioning DL operations among them and executing them in parallel.

Some of the articles [4], [24], [29], [30] survey about deploying deep learning architectures in wireless communication and mobile edge computing perspectives. In order to distribute DL operations among multiple MNs, an idea of deep neural network (DNN) partitioning can be considered [31]–[37]. Some of the studies [3], [35]–[38] suggest layer-based DNN partitioning, where the DNN model is partitioned layer by layer and some of the layer computations are operated in MNs and some of them in the cloud. The layer-based DNN partitioning can solve the depth problem of the DL-driven applications since the massive amount of layers in the DNN model are divided and some of the layer operations are offloaded to the cloud, which provides the mobile devices to save a considerable amount of energy. However, the system becomes quite cloud-dependent in this case and there may be no powerful servers nearby to offload the operations given a limited latency constraint. Also, the layer-based DNN partitioning still lacks a solution for the big input data problem because it is not enough to partition the DNN model only layer by layer due to the fact that a single MN is not able to perform even a single layer operation. A solution for this problem is horizontal DNN partitioning in addition to layer-based DNN partitioning, meaning that the input data of each layer can be also partitioned among multiple MNs [39]–[42]. With this approach, all DL operations are collaboratively executed in MNs in parallel without the need for any external powerful server to offload the workload. In this scenario, due to the heterogeneous computing capabilities and different network conditions of MNs, distributing the workload equally makes the system inefficient. Therefore, the communication and computation parameters and the workload of each MN need to be jointly optimized to minimize the energy consumption of the collaborative part of the system.

The article [38] proposes NeuroSurgeon that includes layer-based DNN partitioning with a single cutting point, meaning that DNN operations are executed in a single MN up to a specific layer, the output of that specific layer is offloaded to the cloud and the remaining layer operations are executed in the cloud server. However, it is claimed in [35] that cutting the model from a single cutting point is not the most efficient way and it proposes JointDNN, which investigates the model layer by layer, operating some of the layers in a single MN and some of them in the cloud by having multiple cutting points. The authors of [37] extend the scheme by adding an edge server to the process and increasing the number of participating devices and they propose DDNN, a distributed DNN architecture that partitions the model among multiple MNs, edge servers, and the cloud server. However, they all make use of the cloud server power to offload part of DNN operations, which makes the system too centralized. Also, because a single MN may not be able to operate a single layer operation by itself, they still need a solution for the big input data problem.

As the first collaborative computing scenario, Mao *et al.* propose MoDNN [41] and MeDNN [42], which have a local distributed mobile device computing system to minimize the latency of the operations by cutting the 2D input data into slices and sharing them among multiple mobile devices. In MoDNN and MeDNN, each participating node sends the output of each layer operation (i.e. intermediate data) to the master node that hosts the DL-driven application and the master node sends the input data of the next layer operation to each node again, which introduces some communication burden on the system. However, instead of that, after each layer operation, each node can keep most of the output data itself and send only the boundary row of its output data of each layer operation to its neighboring nodes for next layer operations so that the communication between the nodes is reduced and the dependency of the overall system on the master node is mitigated, meaning that the system becomes more decentralized. Also, MoDNN and MeDNN only optimize the workload distribution among MNs by fixing all the other communication and computation parameters.

In order to obtain the most energy-efficient system, the workload distribution, communication, and computation parameters need to be jointly optimized. In [6], Zhao *et al.* introduce DeepThings, in which a Fused Tile Partitioning (FTP) method is proposed. Unlike partitioning DNN model among multiple MNs only layer by layer, FTP partitions it vertically in a grid fashion to reduce the memory footprint of MNs and makes the model slices independently distributable computation tasks [24], [30]. However, DeepThings is still dependent on a centralized gateway device to facilitate the operations between edge nodes.

Motivated by these ideas, we propose an energy-efficient fine-grained DNN partitioning scheme for wireless collaborative fog computing systems, considering both layer-based and horizontal DNN partitioning. We use the convolutional neural network (CNN) as the DNN model since it is the most popular one in many deep learning-driven applications. However, the system can be adapted to any DNN model with minor modifications. In the proposed CNN model, each layer operation is locally performed in a collaborative manner by partitioning the workload of each single layer operation among multiple MNs. Before the collaborative local computation, the MN that receives a 2D input data sample acts as a master node, and the other participating MNs become slaves. The master node horizontally cuts the 2D input data sample into slices and it optimally distributes these slices among all MNs including itself. This procedure is called the first round distribution (FRD). After that, MNs perform their local computations for each layer by using the assigned slices as inputs. After each layer operation, MNs exchange the boundary rows of the 2D output data with their neighboring MNs that are responsible for the adjacent slices above and below for next layer operations. The process continues until all collaborative layer operations are finished. Our contributions in this paper can be listed as follows:

- We propose a novel wireless collaborative distributed fog computing scheme considering a fine-grained partitioning that includes both layer-based and horizontal DNN partitioning for deep learning-driven applications.
- Because most of the output data of each layer operation stay in the same node and is not sent to another node throughout the whole DNN operation, the communication overhead between MNs after each layer operation is significantly mitigated. Only a single row (boundary row) of the output data after each layer operation needs to be sent to the neighboring nodes, which provides communication efficiency to the system. Also, not all nodes communicate with each other during the operation, only neighboring nodes do it, which provides even more communication efficiency.
- Instead of the pure workload distribution optimization by fixing the other dynamic parameters in the system, we jointly optimize the communication and computation parameters and the workload distribution.
- The structure makes use of all the computing resources of participating MNs without needing any powerful server, which makes the system decentralized.
- The final problem is formulated as a convex optimization problem where an analytical solution is obtained by using primal-dual decomposition and Lagrange duality theory, and the analytical results are shown to converge to the ones obtained in CVX software of MATLAB [43]. The resulting problem is proposed to be solved at two iterative stages by using primal-dual decomposition. In the external stage, the workload assigned to each MN, the time needed to operate for each layer operation, and the communication parameters of the first round distribution such as transmitted power of the master node and the time needed to transmit each slice are optimized whereas in the internal stage, CPU frequencies of MNs in the local computation for each layer, transmitted power and the times needed to communicate between the neighboring MNs to exchange boundary rows are the optimizing parameters.

The comparison of similar studies with our scheme is provided in Table 1 while the list of abbreviations used throughout the paper can be found in Table 2.

TABLE 1. Comparison of the similar studies with our scheme.

	DNN Partitioning	Collaboration between Nodes	Performance Parameter
Neurosurgeon [38]	Layer-based with single cutting point	No	Latency & energy consumption
JointDNN [35]	Layer-based with multiple cutting points	No	Latency & energy consumption
DDNN [37]	Layer-based with exit points	Depending on the input, yes or no	Accuracy
MoDNN [41], MeDNN [42]	Layer-based & horizontal	Yes	Latency
DeepThings [6]	Layer-based & horizontal	Yes	Latency & memory usage
Our Scheme	Layer-based & horizontal	Yes	Latency-constrained energy consumption

TABLE 2. List of abbreviations.

CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
FRD	First Round Distribution
FTP	Fused Tile Partitioning
MAC	Multiply-Accumulate
MCC	Mobile Cloud Computing
MEC	Mobile Edge Computing
MN	Mobile Node

Section II describes the CNN model considered in the proposed scheme whereas section III introduces the system model and the scenario. Section IV, V, and VI present the energy consumption model formulations for first round distribution, collaborative local computation, and exchange communication, respectively. Section VII shows the final problem formulation of the overall scheme and section VIII includes the solution of the final problem. Section IX demonstrates the simulation results and interpretation of them. Lastly, Section X discusses the results, the improvable points on the proposed scheme, and possible future research ideas.

II. CNN MODEL

A typical 2D CNN model can be seen in Fig. 1 where N^{LAY} is the number of layers that are locally operated among MNs collaboratively and N^{TOT} is the total number of layers in the entire CNN model. By using CNN properties, we can

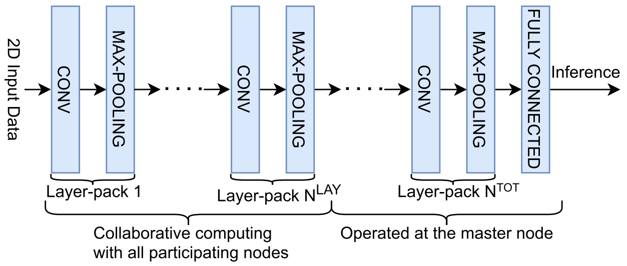

FIGURE 1. **CNN model.**

define the relationship between the input and output sizes of a single layer as

$$o^{\text{conv}}_{k,n} = \left\lfloor \frac{i^{\text{conv}}_{k,n} - f^{\text{conv}}_n + 2\,ZP_n}{s^{\text{conv}}_n} \right\rfloor + 1 \tag{1}$$

and

$$o^{\text{pool}}_{k,n} = \left\lfloor \frac{o^{\text{conv}}_{k,n} - f^{\text{pool}}_n}{s^{\text{pool}}_n} \right\rfloor + 1 \tag{2}$$

where $\{k, n\}$ corresponds to the layer-pack n in MN k, $i^{\text{conv}}_{k,n}$ and $o^{\text{conv}}_{k,n}$ are the input and output sizes of the convolutional layer n, respectively, $o^{\text{pool}}_{k,n}$ is the output size of the pooling layer n, f^{conv}_n and f^{pool}_n are the filter (kernel) sizes of the convolutional layer n and the pooling layer n, respectively, ZP_n is the zero-padding setting parameter of the convolutional layer n, s^{conv}_n and s^{pool}_n are the stride setting parameters of the convolutional layer n and the pooling layer n, respectively. In this paper, for convolutional layers, we have 3 × 3 filters ($f^{\text{conv}}_n = 3$), unity stride ($s^{\text{conv}}_n = 1$) and zero-padding is applied ($ZP_n = (f^{\text{conv}}_n - 1)/2$) so that the input and output sizes of a convolutional layer become equal. For pooling layers, we have 2 × 2 filter size ($f^{\text{pool}}_n = 2$) and two strides ($s^{\text{pool}}_n = 2$) so that both the number of rows and number of columns become half of their initial values while the total dimension of the output data becomes a quarter of its initial value. With these hyper-parameter assignments, (1) and (2) can be modified as

$$o^{\text{conv}}_{k,n} = i^{\text{conv}}_{k,n}, \quad o^{\text{pool}}_{k,n} = \left\lfloor \frac{o^{\text{conv}}_{k,n}}{2} \right\rfloor = \left\lfloor \frac{i^{\text{conv}}_{k,n}}{2} \right\rfloor \tag{3}$$

III. SYSTEM MODEL

We consider that there is a set of K active single antenna MNs, indexed by $k \in [K]$ and an access point/a base station (AP/BS) in the system, together with the cloud nearby. The cloud is only needed for training of the CNN model. After training, the system does not need any cloud server availability to facilitate the scenario. Each node is a wirelessly connected device and all nodes have an identical CNN model.

The proposed scenario which can be seen in Fig. 2 works as follows: (i) The CNN model is trained completely in the cloud by offline data sets before the operation starts. We work on already trained CNN models to only focus on the energy consumption of the inference phase of the CNN model since the training phase is operated only once at the beginning while the inference phase is a continuous process and the training is generally operated offline on powerful cloud servers [3]. Also, since we have a latency-constrained scenario, performing the training in a given limited time is not optimal since it needs to be operated with full performance without any time limitation to obtain maximum accuracy of the model. Thus, we consider the inference phase rather than the training phase in this paper. (ii) The trained model weights w are offloaded to the nodes. (iii) When node k receives a 2D input data sample d, having d^y number of rows and d^x number of columns, it starts to act as a master node whereas the other active nodes become slaves. Each node has a specific time frame allocated to it to be the master node of the system. Thus, no collision exists in the system when there is more than one node having a 2D input data sample to perform collaborative DNN execution. (iv) The master node horizontally cuts the 2D input data sample into slices and optimally shares out them among all participating nodes, including itself, by taking the computing capabilities and network conditions of all nodes into account. This operation is called the first round distribution (FRD). During FRD, the only parameters that the master node needs to have are the computing capability of each node and the channel gain between the master node and the other nodes. These parameters do not change and they are fixed during the deep learning operations, so this information is given to the nodes once before the deep learning operations. It is assumed that the placement of each node on the input data is assigned by the master node before the operation starts. Consequently, every node is aware of its adjacent nodes throughout the operation. After FRD, the master node sends the partial input data to each node with the additional boundary row of its neighboring node as shown in Fig. 3. Because the nodes have a single antenna, all of the data cannot be sent simultaneously by the master node, so all input data should be sent in a pre-determined time frame structure. For the sake of tractability, we assume that the sending order is pre-determined and the local computations start after all nodes receive their partial input data for the synchronization of the nodes for each layer operation. After a single convolutional layer operation in each node, if all bits are

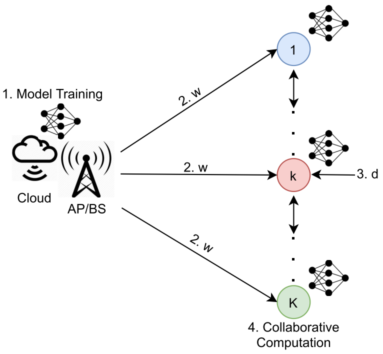

FIGURE 2. **Illustration of the system model.**

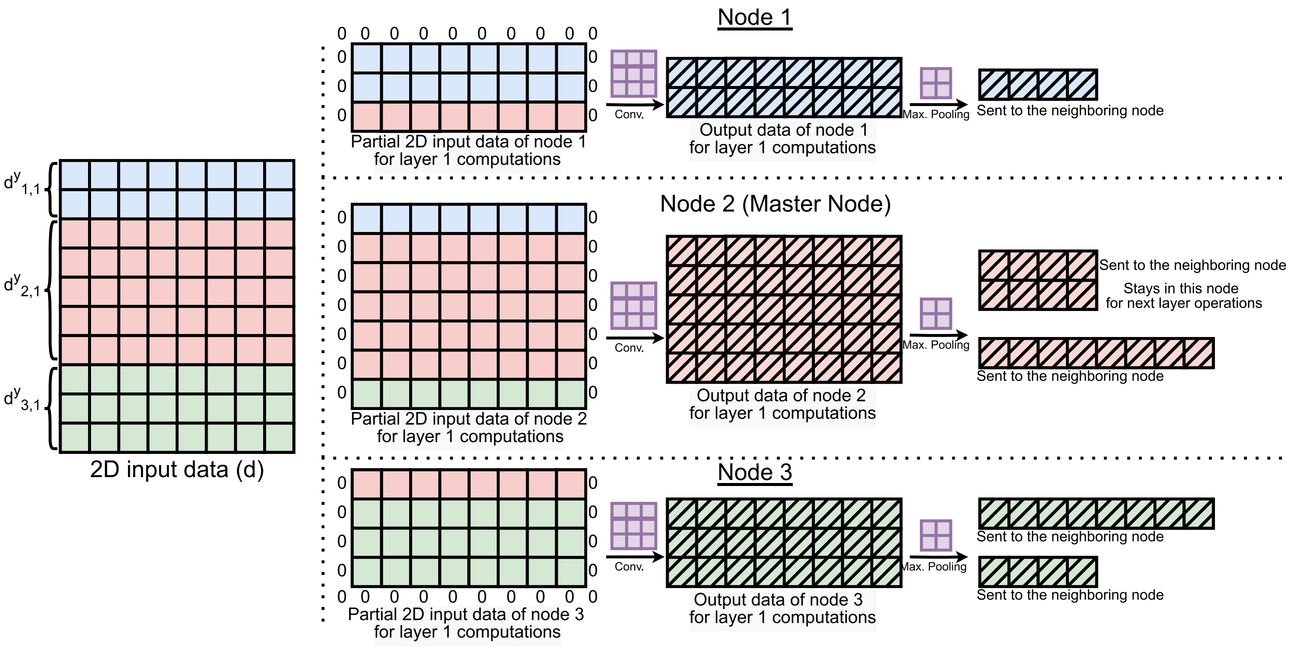
Node 1
Partial 2D input data of node 1 for layer 1 computations
Conv.
Output data of node 1 for layer 1 computations
Max. Pooling
Sent to the neighboring node
Node 2 (Master Node)
Partial 2D input data of node 2 for layer 1 computations
Output data of node 2 for layer 1 computations
Sent to the neighboring node
Stays in this node for next layer operations
Sent to the neighboring node
Node 3
Partial 2D input data of node 3 for layer 1 computations
Output data of node 3 for layer 1 computations
Sent to the neighboring node
Sent to the neighboring node
$d^y_{1,1}$
$d^y_{2,1}$
$d^y_{3,1}$
2D input data (d)
0

FIGURE 3. An example scenario for the first layer computation with 3 nodes, showing the first round distribution, collaborative local computation and the resulting intermediate output data (Zero padding activated).

max-pooled, the boundary rows of the output data are sent to the neighboring nodes for next layer operations since each node needs the boundary rows to operate next layer convolution operations. If there remains a boundary row that maximum pooling operation cannot be done because of lack of the other part of the output data, this row remains unchanged without operating any maximum pooling operation. This is the case between node 2 and node 3 in Fig. 3 where the boundary rows of the output data in node 2 and node 3 should be max-pooled but they both lack the partial bits to complete the max-pooling operation. Except for the boundary rows, there is no need to send the remaining non-boundary rows since the next layer operations of the data are performed in the same node. Sending only boundary row from the output data significantly reduces the communication overhead between the neighboring nodes.

During FRD, since the overall energy consumption of the collaborative part is a function of the communication and computation parameters of each participating node, the master node considers them to minimize the overall energy consumption of the collaborative part and distributes the rows of the input data accordingly. Also, the intermediate boundary rows without max-pooling need to be assigned one of the neighboring nodes to be operated for the next layer operations. This is the case between node 2 and node 3 in Fig. 3 in which the boundary rows without max-pooling in node 2 and node 3 together constitute the input data of the next layer, so the resulting row should be operated in either node 2 or node 3. There is a communication-computation trade-off in this point. If the undetermined row is operated in node 2 meaning that node 3 needs to send its boundary row without max-pooling together with the next additional max-pooled data to node 2, so it sacrifices from the communication while gaining from the computation. In this paper, for the sake of simplicity of the resulting optimization problem, it is assumed that always upper neighboring node operates the undetermined row for next layer operations. This assumption does not substantially change the optimization problem since the considered trade-off point consists of just one-row operation. As can be understood from Fig. 3, this undetermined row is assigned to node 2 which is the upper neighbor of node 3.

IV. ENERGY CONSUMPTION OF THE FIRST ROUND DISTRIBUTION

Since all rows in the input data should be assigned to the active nodes, the following equation is held

$$\sum_{k=1}^{K} d^y_{k,1} = d^y \tag{4}$$

where $d^y_{k,n}$ is the number of rows from which node k is responsible in layer n computations. Except itself, the master node (shown as $\widetilde{k}$) sends the input data slices to $K-1$ number of slave nodes at the achievable rate

$$r^{\mathrm{FRD}}_k = B\ln(1+\frac{p^{\mathrm{FRD}}_k h^{\mathrm{FRD}}_k}{N_0 B}) \tag{5}$$

where p^{FRD}_k is the RF transmit power of the master node $\widetilde{k}$ while sending the input data slice to node k, h^{FRD}_k is the channel gain between the master node $\widetilde{k}$ and node k, B is the communication bandwidth between the nodes and N_0 is the noise power. The energy consumption to transmit the input data slice to node k is

$$E_k^{\mathrm{FRD}} = t_k^{\mathrm{FRD}}(p_k^{\mathrm{FRD}} + P_{\tilde{k}}^c) \tag{6}$$

where P_k^c is the constant energy consumption of the communication circuitry in node k and t_k^{FRD} is the time needed to transmit the input data slice of node k. Because the transmission rate cannot be greater than the achievable rate and the RF transmit power cannot exceed the maximum RF power allowed for a node, we have

$$d_k^{\mathrm{FRD}} \leq t_k^{\mathrm{FRD}} r_k^{\mathrm{FRD}} \tag{7}$$

and

$$p_k^{\mathrm{FRD}} \leq p_{\tilde{k}}^{\max} \tag{8}$$

where $p_k^{\max}$ is the maximum RF transmit power of node k and d_k^{FRD} is the number of bits sent to node k by the master node $\widetilde{k}$ during FRD and it is defined as

$$\begin{aligned} d_k^{\mathrm{FRD}} &= \begin{cases} d^x(d_{k,1}^y + 1) & \text{for top \& bottom nodes} \\ d^x(d_{k,1}^y + 2) & \text{for other nodes} \end{cases} \\ &= d^x(d_{k,1}^y + 1 + I_{k,1}^{\mathrm{a}} I_{k,1}^{\mathrm{b}}) \end{aligned} \tag{9}$$

where $I_{k,n}^{\{\mathrm{a,b}\}}$ are the binary indicators such that they are 1 if node k has a neighbor above or below, respectively after layer n operation and they are 0 otherwise. Thus, the constraint in (7) becomes

$$d^x(d_{k,1}^y + 1 + I_{k,1}^{\mathrm{a}} I_{k,1}^{\mathrm{b}}) \leq t_k^{\mathrm{FRD}} r_k^{\mathrm{FRD}} \tag{10}$$

V. PER LAYER ENERGY CONSUMPTION OF LOCAL COMPUTATION

After FRD, all participating nodes execute their layer operations using their partial input data. Among each layer-pack, convolutional layers dominate the overall energy consumption for CNNs [41], [42], [44]. Also, although the fully connected layers are the most memory-consuming layers, convolutional layers consume more energy than fully connected layers [44]. Besides, since our CNN model has a single fully connected layer and it is placed at the end of the model, its input data dimension is so small. Therefore, the proposed scheme focuses on the energy consumption of convolutional layers.

The convolution operation in convolutional layers can be decomposed into lots of multiply-accumulate (MAC) operations [44], [45]. MAC is defined as taking one element from the filter and one from the shaded input map, i.e. the portion of the 2D input data that the filter is hovering, multiplying them, and accumulating to the previous sum. For example, for each element-wise dot product of two 3 × 3 matrices, there are $3^2 = 9$ MAC operations.

In order to find the per layer energy consumption of the local computation for a single node, we need to know how many CPU cycles are needed to accomplish it. Thus, we calculate the number of MACs needed for this computation and multiply it with c_k which is the number of CPU cycles to perform a single MAC operation for node k. From the properties of convolution operation, the number of MACs $N_{k,n}^{\mathrm{MAC}}$ to be performed by node k in layer n computations can be found as

$$N_{k,n}^{\mathrm{MAC}} = (f_n^{\mathrm{conv}})^2 d_n^x d_{k,n}^y = 9 d_n^x d_{k,n}^y \tag{11}$$

where d_n^x is the number of columns of the 2D input data at the beginning of layer n computations and we have $d_n^x = \lfloor d_1^x / 2^{n-1} \rfloor$. The energy consumption of layer n computations performed in node k under the assumption of low CPU voltage [46]–[48] is

$$E_{k,n}^{\mathrm{LOC}} = \frac{\kappa_k c_k^3 (N_{k,n}^{\mathrm{MAC}})^3}{(t_{k,n}^{\mathrm{LOC}})^2} = \frac{3^6 \kappa_k c_k^3 (d_n^x)^3 (d_{k,n}^y)^3}{(t_{k,n}^{\mathrm{LOC}})^2} \tag{12}$$

where κ_k is the effective capacitance coefficient depending on the hardware architecture and $t_{k,n}^{\mathrm{LOC}}$ is the local computation time for the layer-pack n in MN k. The number of CPU cycles used in the local computation cannot exceed the CPU cycles obtained by working at its maximum CPU frequency $\nu_k^{\max}$ such that

$$c_k N_{k,n}^{\mathrm{MAC}} = 9 c_k d_n^x d_{k,n}^y \leq t_{k,n}^{\mathrm{LOC}} \nu_k^{\max} \tag{13}$$

for $\forall k \in [K]$ and $\forall n \in [N^{\mathrm{LAY}}]$.

VI. PER LAYER ENERGY CONSUMPTION OF EXCHANGE COMMUNICATION

After convolutional layer operations, we have several scenarios depending on the distribution of the input data ($d_{k,n}^y$'s) for the example scenario in Fig. 3:

- If a node has a neighbor below:
  - If the row number index of the boundary row ends with an even number, max. pooled boundary row is sent to the neighbor below ($d_n^x/2$ bits). This is the case for node 1 in Fig. 3.
  - If the row number index of the boundary row ends with an odd number, the row that cannot be max-pooled because of lack of the adjacent row is sent without max-pooling to the neighbor below (d_n^x bits). This is the case for node 2 in Fig. 3.
- If a node has a neighbor above:
  - If the row number index of the boundary row starts with an odd number, max. pooled boundary row is sent to the neighbor above ($d_n^x/2$ bits). This is the case for node 2 in Fig. 3.
  - If the row number index of the boundary row starts with an even number, the row that cannot be max-pooled because of lack of the adjacent row and the previous max-pooled row are sent together to the neighbor above for next layer operations ($3\, d_n^x/2$ bits). This is the case for node 3 in Fig. 3.

Thus, there is a relationship between $d_{k,n}^y$ and $d_{k,1}^y$, depending on the previous starting row number index in the overall input data of a layer and the previous distributed number of rows $d_{k,n-1}^y$ assigned to node k. However, because this relationship cannot be analytically shown as a general expression, in average, it is assumed that $d_{k,n}^y = d_{k,1}^y / 2^{n-1}$. Therefore, in the previous section, we rewrite (11), (12) and (13) as

$$N_{k,n}^{\mathrm{MAC}} = \frac{9d_n^x}{2^{n-1}}\, d_{k,1}^y \tag{14}$$

$$E_{k,n}^{\mathrm{LOC}} = \frac{3^6 \kappa_k c_k^3 (d_n^x)^3 (d_{k,1}^y)^3}{(2^{n-1})^3 (t_{k,n}^{\mathrm{LOC}})^2} \tag{15}$$

and

$$\frac{9c_k d_n^x}{2^{n-1}}\, d_{k,1}^y \leq t_{k,n}^{\mathrm{LOC}} \nu_k^{\max} \tag{16}$$

In addition, it can be seen from the scenario itemized previously that each node sends either $d_n^x/2$, d_n^x or 3 $d_n^x/2$ number of bits to its neighbors after a single layer computation. Therefore, in average, the number of bits that should be sent to each neighbor after a single layer computation is assumed to be as d_n^x bits.

After the local computation, each node exchanges the boundary rows with its neighbors at the achievable rate

$$r_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} = B \ln\Big(1 + \frac{p_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} h_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}}{N_0 B}\Big) \tag{17}$$

where $\{\mathrm{a,b}\}$ index can become either a for the exchange communication expressions of neighbor above or b for the exchange communication expressions of neighbor below. Also, $p_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}$ and $h_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}$ are the RF transmit power of node k and the channel gain while transmitting the boundary row of its layer n output data to its neighbor above and below, respectively. Per layer energy consumption of the exchange communication is

$$E_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} = t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} (p_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} + P_k^c) \tag{18}$$

where $t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}$ are the exchange time for node k to transmit the boundary row to its neighbor above and below, respectively. Also, since the number of bits sent during exchange communication cannot exceed the number of bits that can be sent with the achievable rate, we have

$$I_{k,n}^{\{\mathrm{a,b}\}} d_n^x \leq t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} r_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} \tag{19}$$

and each node has a maximum transmitted power constraint such that

$$p_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} \leq p_k^{\max} \tag{20}$$

Finally, we can write the energy consumption of the exchange communication after layer n operations for node k as

$$E_{k,n}^{\mathrm{EXC}} = I_{k,n}^{\mathrm{a}} E_{k,n}^{\mathrm{EXC_a}} + I_{k,n}^{\mathrm{b}} E_{k,n}^{\mathrm{EXC_b}} \tag{21}$$

VII. PROBLEM FORMULATION

Because the participating nodes cannot start the next layer operation without receiving the previous layer boundary rows from their neighbors, for each layer, the total time t_n spent during local computation and exchange communication is assumed to be the same for all nodes, optimized by the master node and it is the upper bound of the following inequality:

$$t_{k,n}^{\mathrm{LOC}} + I_{k,n}^{\mathrm{a}} t_{k,n}^{\mathrm{EXC_a}} + I_{k,n}^{\mathrm{b}} t_{k,n}^{\mathrm{EXC_b}} \leq t_n \tag{22}$$

Also, we have a latency requirement such that the whole operation should be finished in less than τ seconds and we can express it mathematically as

$$\sum_{\substack{k=1 \\ k\neq\tilde{k}}}^{K} t_k^{\mathrm{FRD}} + \sum_{n=1}^{N^{\mathrm{LAY}}} t_n \leq \tau \tag{23}$$

The final optimization problem is as follows:

$$\begin{aligned} \min_{x_1}.\ & \sum_{\substack{k=1 \\ k\neq\tilde{k}}}^{K} E_k^{\mathrm{FRD}} + \sum_{k=1}^{K} \sum_{n=1}^{N^{\mathrm{LAY}}} (E_{k,n}^{\mathrm{LOC}} + E_{k,n}^{\mathrm{EXC}}) \\ \text{s.t. } & (4), (8), (10), (16), (19), (20), (22), (23) \\ & 0 \leq p_k^{\mathrm{FRD}}, p_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}, \quad 0 \leq d_{k,1}^y \leq d^y \\ & 0 \leq t_{k,n}^{\mathrm{LOC}}, t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} \leq t_n, \quad 0 \leq t_n, t_k^{\mathrm{FRD}} \leq \tau \end{aligned} \tag{24}$$

where $x_1 = \{\{t_n\}_{n=1}^{N^{\mathrm{LAY}}}, \{d_{k,1}^y\}_{k=1}^K, \{t_k^{\mathrm{FRD}}, p_k^{\mathrm{FRD}}\}_{\substack{k=1 \\ k\neq\tilde{k}}}^K,$ $\{\{p_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}, t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}, t_{k,n}^{\mathrm{LOC}}\}_{k=1}^K\}_{n=1}^{N^{\mathrm{LAY}}}\}$.

VIII. PROBLEM SOLUTION

The final optimization problem in (24) is not convex because of the communication energy expressions. Therefore, we can convexify the optimization problem by introducing two new parameters as $E_k^{\mathrm{FRD_{RF}}} = p_k^{\mathrm{FRD}} t_k^{\mathrm{FRD}}$ and $E_{k,n}^{\mathrm{EXC_{RF}}_{\{\mathrm{a,b}\}}} = t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} p_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}$ where $E_k^{\mathrm{FRD_{RF}}}$ and $E_{k,n}^{\mathrm{EXC_{RF}}_{\{\mathrm{a,b}\}}}$ are defined as the RF energy consumed during the first round distribution and exchange communication, respectively. After these assignments, we modify (5), (6), (8), (17), (18) and (20) as

$$r_k^{\mathrm{FRD}} = B \ln\Big(1 + \frac{(E_k^{\mathrm{FRD_{RF}}}/t_k^{\mathrm{FRD}}) h_k^{\mathrm{FRD}}}{N_0 B}\Big) \tag{25}$$

$$E_k^{\mathrm{FRD}} = E_k^{\mathrm{FRD_{RF}}} + t_k^{\mathrm{FRD}} P_{\tilde{k}}^c \tag{26}$$

$$E_k^{\mathrm{FRD_{RF}}} \leq t_k^{\mathrm{FRD}} p_{\tilde{k}}^{\max} \tag{27}$$

$$r_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} = B \ln\Big(1 + \frac{(E_{k,n}^{\mathrm{EXC_{RF}}_{\{\mathrm{a,b}\}}}/t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}) h_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}}}{N_0 B}\Big) \tag{28}$$

$$E_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} = E_{k,n}^{\mathrm{EXC_{RF}}_{\{\mathrm{a,b}\}}} + t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} P_k^c \tag{29}$$

and

$$E_{k,n}^{\mathrm{EXC_{RF}}_{\{\mathrm{a,b}\}}} \leq t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} p_k^{\max} \tag{30}$$

After these modifications, the optimization problem becomes convex and is expressed as

$$\begin{aligned} \min_{x_2}.\ & \sum_{\substack{k=1 \\ k\neq\tilde{k}}}^{K} E_k^{\mathrm{FRD}} + \sum_{k=1}^{K} \sum_{n=1}^{N^{\mathrm{LAY}}} (E_{k,n}^{\mathrm{LOC}} + E_{k,n}^{\mathrm{EXC}}) \\ \text{s.t. } & (4), (10), (16), (19), (22), (23), (27), (30) \\ & 0 \leq E_k^{\mathrm{FRD_{RF}}}, E_{k,n}^{\mathrm{EXC_{RF}}_{\{\mathrm{a,b}\}}}, \quad 0 \leq d_{k,1}^y \leq d^y \\ & 0 \leq t_{k,n}^{\mathrm{LOC}}, t_{k,n}^{\mathrm{EXC}_{\{\mathrm{a,b}\}}} \leq t_n, \quad 0 \leq t_n, t_k^{\mathrm{FRD}} \leq \tau \end{aligned} \tag{31}$$

where $x_2 = \{\{t_n\}_{n=1}^{N^{\mathrm{LAY}}}, \{d_{k,1}^{y}\}_{k=1}^{K}, \{t_k^{\mathrm{FRD}}, E_k^{\mathrm{FRD_{RF}}}\}_{\substack{k=1\\ k\neq\tilde{k}}}^{K}, \{\{E_{k,n}^{\mathrm{EXC_{RF_{\{a,b\}}}}}, t_{k,n}^{\mathrm{EXC_{\{a,b\}}}}, t_{k,n}^{\mathrm{LOC}}\}_{k=1}^{K}\}_{n=1}^{N^{\mathrm{LAY}}}\}$.

The problem can be easily solved via convex optimization techniques, but in order to make it easier to solve the problem and obtain some analytical insight for each optimizing parameter, we decompose the problem into two sub-problems with hierarchical order in an iterative manner. The decomposition type is the primal-dual decomposition consisting of an external master primal problem operated in the master node and an internal sub-problem operated in each slave node for each layer operation separately. Solving part of the optimization problem locally in the slave nodes makes the optimization distributed and this is also one of the reasons that the decomposition is preferred. The external problem optimizes the time duration t_n for each layer operation, the number of rows $d_{k,1}^{y}$ assigned to node k and the first round distribution communication parameters t_k^{FRD} and $E_k^{\mathrm{FRD_{RF}}}$ whereas the internal one optimizes the parameters of the local computation and exchange communication, i.e. $t_{k,n}^{\mathrm{LOC}}$, $t_{k,n}^{\mathrm{EXC_{\{a,b\}}}}$ and $E_{k,n}^{\mathrm{EXC_{RF_{\{a,b\}}}}}$ given the parameter values t_n and $d_{k,1}^{y}$. The internal optimization problem is shown as

$$\begin{aligned}
(E_{k,n}^{\mathrm{PL}}(t_n, d_{k,1}^{y}))^* = \min_{x_3}.\ & E_{k,n}^{\mathrm{LOC}} + E_{k,n}^{\mathrm{EXC}}\\
\text{s.t. } & (16), (19), (22), (30)\\
& 0 \leq E_{k,n}^{\mathrm{EXC_{RF_{\{a,b\}}}}}\\
& 0 \leq t_{k,n}^{\mathrm{LOC}}, t_{k,n}^{\mathrm{EXC_{\{a,b\}}}} \leq t_n
\end{aligned} \tag{32}$$

where $x_3 = \{t_{k,n}^{\mathrm{LOC}}, t_{k,n}^{\mathrm{EXC_{\{a,b\}}}}, p_{k,n}^{\mathrm{EXC_{\{a,b\}}}}\}$ and $(E_{k,n}^{\mathrm{PL}}(t_n, d_{k,1}^{y}))^*$ is the optimal per layer energy consumption of the sum of local computation and exchange communication in layer n operations for node k given the time duration t_n and the distribution $d_{k,1}^{y}$. The external optimization problem can be expressed as

$$\begin{aligned}
\min_{x_4}.\ & \sum_{\substack{k=1\\ k\neq\tilde{k}}}^{K} E_k^{\mathrm{FRD}} + \sum_{k=1}^{K}\sum_{n=1}^{N^{\mathrm{LAY}}} (E_{k,n}^{\mathrm{PL}}(t_n, d_{k,1}^{y}))^*\\
\text{s.t. } & (4), (10), (23), (27)\\
& 0 \leq E_k^{\mathrm{FRD_{RF}}}\\
& 0 \leq t_n, t_k^{\mathrm{FRD}} \leq \tau\\
& 0 \leq d_{k,1}^{y} \leq d^{y}
\end{aligned} \tag{33}$$

where $x_4 = \{\{t_n\}_{n=1}^{N^{\mathrm{LAY}}}, \{d_{k,1}^{y}\}_{k=1}^{K}, \{t_k^{\mathrm{FRD}}, E_k^{\mathrm{FRD_{RF}}}\}_{\substack{k=1\\ k\neq\tilde{k}}}^{K}\}$. Inspired by [49], the solution of the primal-dual decomposition of the convex optimization problem converges to the solution of the overall problem with the iterative convergence algorithm approach mentioned in Algorithm 1.

A. SOLUTION OF INTERNAL OPTIMIZATION PROBLEM

The detailed solution steps of the internal optimization problem is given in Appendix A. The optimal parameters of the internal optimization problem are found as

$$(t_{k,n}^{\mathrm{LOC}})^* = \begin{cases} \dfrac{m_{k,n}}{\nu_k^{\max}} & \beta_{k,n} \geq 2\kappa_k(\nu_k^{\max})^3\\[2ex] m_{k,n}\left(\dfrac{2\kappa_k}{\beta_{k,n}}\right)^{1/3} & \dfrac{2\kappa_k(m_{k,n})^3}{(t_n)^3} < \beta_{k,n} < 2\kappa_k(\nu_k^{\max})^3\\[2ex] t_n & \beta_{k,n} \leq \dfrac{2\kappa_k(m_{k,n})^3}{(t_n)^3} \end{cases} \tag{34}$$

$$(p_{k,n}^{\mathrm{EXC_b}})^* = \begin{cases} 0 & \mu_{k,n}^{\textcircled{1}} \leq \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}}\\[2ex] B\left(\mu_{k,n}^{\textcircled{1}} - \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}}\right) & \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} < \mu_{k,n}^{\textcircled{1}} < \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} + \dfrac{p_k^{\max}}{B}\\[2ex] p_k^{\max} & \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} + \dfrac{p_k^{\max}}{B} \leq \mu_{k,n}^{\textcircled{1}} \end{cases} \tag{35}$$

and

$$(t_{k,n}^{\mathrm{EXC_b}})^* = \begin{cases} t_n & \mu_{k,n}^{\textcircled{1}} \leq \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} e^{\frac{d_n^x}{Bt_n}}\\[2ex] \dfrac{d_n^x}{B\ln\left(\frac{h_{k,n}^{\mathrm{EXC_b}}}{N_0}\mu_{k,n}^{\textcircled{1}}\right)} & \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} e^{\frac{d_n^x}{Bt_n}} < \mu_{k,n}^{\textcircled{1}} < \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} + \dfrac{p_k^{\max}}{B}\\[2ex] \dfrac{d_n^x}{B\ln\left(1+\frac{h_{k,n}^{\mathrm{EXC_b}}}{BN_0}p_k^{\max}\right)} & \dfrac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} + \dfrac{p_k^{\max}}{B} \leq \mu_{k,n}^{\textcircled{1}} \end{cases} \tag{36}$$

where $\mu_{k,n}^{\textcircled{1}}$ and $\beta_{k,n}$ are the Lagrange multipliers associated with (19) and (22), respectively and $m_{k,n} = 9c_k d_n^x d_{k,1}^{y}/2^{n-1}$ is assigned to collect the parameters in a single term.

Algorithm 1 Iterative Convergence Algorithm to Solve the Internal and External Optimization Problems

```
1: Initialize the parameters $d_{k,1}^{y}$ and $t_n$.
2: Given the parameters $d_{k,1}^{y}$ and $t_n$, solve the convex internal optimization problem (32) for the parameters $t_{k,n}^{\mathrm{LOC}}, t_{k,n}^{\mathrm{EXC_{\{a,b\}}}}, E_{k,n}^{\mathrm{EXC_{RF_{\{a,b\}}}}}$.
3: By using $(E_{k,n}^{\mathrm{PL}}(t_n, d_{k,1}^{y}))^*$, solve the convex external optimization problem (33) for the parameters $t_n, d_{k,1}^{y}, t_k^{\mathrm{FRD}}, E_k^{\mathrm{FRD_{RF}}}$.
4: By using the parameters $d_{k,1}^{y}$ and $t_n$ found in item 3, repeat the process from item 2 until convergence is reached and the results do not change anymore.
```

B. SOLUTION OF EXTERNAL OPTIMIZATION PROBLEM

The optimal parameters of the external optimization problem can be obtained as

$$(t_n)^* = \begin{cases} 0 & \phi > \sum_{k=1}^{K} \beta_{k,n} \\ (0,\tau) & \phi = \sum_{k=1}^{K} \beta_{k,n} \\ \tau & \phi < \sum_{k=1}^{K} \beta_{k,n} \end{cases} \tag{37}$$

$$(d_{k,1}^{y})^* = \begin{cases} 0 & \psi_k \leq 0 \\ \sqrt{\dfrac{\psi_k}{3\left(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}}\right)}} & 0 < \psi_k < 3\left(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}}\right)(d^y)^2 \\ d^y & \psi_k \geq 3\left(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}}\right)(d^y)^2 \end{cases} \tag{38}$$

$$(p_k^{\mathrm{FRD}})^* = \begin{cases} 0 & \theta_k \leq \dfrac{N_0}{h_k^{\mathrm{FRD}}} \\ B\left(\theta_k - \dfrac{N_0}{h_k^{\mathrm{FRD}}}\right) & \dfrac{N_0}{h_k^{\mathrm{FRD}}} < \theta_k < \dfrac{N_0}{h_k^{\mathrm{FRD}}} + \dfrac{p_{\widetilde{k}}^{\max}}{B} \\ p_{\widetilde{k}}^{\max} & \dfrac{N_0}{h_k^{\mathrm{FRD}}} + \dfrac{p_{\widetilde{k}}^{\max}}{B} \leq \theta_k \end{cases} \tag{39}$$

and

$$(t_k^{\mathrm{FRD}})^* = \begin{cases} 0 & \theta_k < cons_k^{\textcircled{3}} \\ (0,\tau) & \theta_k = cons_k^{\textcircled{3}} \\ \tau & \theta_k > cons_k^{\textcircled{3}} \end{cases} \tag{40}$$

where θ_k and ϕ are the Lagrange multipliers associated with (10) and (23),

$$\psi_k = \begin{cases} \lambda - \sum_{n=1}^{N^{\mathrm{LAY}}} \gamma_{k,n}^{\textcircled{1}} cons_{k,n}^{\textcircled{2}} & k = \widetilde{k} \\ \lambda - \sum_{n=1}^{N^{\mathrm{LAY}}} \gamma_{k,n}^{\textcircled{1}} cons_{k,n}^{\textcircled{2}} - \theta_k d^x & k \neq \widetilde{k} \end{cases} \tag{41}$$

$$cons_{k,n}^{\textcircled{1}} = \frac{3^6 \kappa_k c_k^3 (d_n^x)^3}{(2^{n-1})^3((t_{k,n}^{\mathrm{LOC}})^*)^2} \tag{42}$$

$$cons_{k,n}^{\textcircled{2}} = \frac{9\, c_k d_n^x}{2^{n-1} \nu_k^{\max}} \tag{43}$$

$$cons_k^{\textcircled{3}} = \frac{P_{\widetilde{k}}^{c} + \phi - \xi_k^{\textcircled{2}} p_{\widetilde{k}}^{\max}}{\left(r_k^{\mathrm{FRD}}\left((p_k^{\mathrm{FRD}})^*\right) - \dfrac{\frac{h_k^{\mathrm{FRD}}}{N_0}(p_k^{\mathrm{FRD}})^*}{1+\frac{h_k^{\mathrm{FRD}}}{BN_0}(p_k^{\mathrm{FRD}})^*} \right)} \tag{44}$$

$$\xi_k^{\textcircled{2}} = \begin{cases} 0 & (p_k^{\mathrm{FRD}})^* < p_{\widetilde{k}}^{\max} \\ \theta_k \dfrac{h_k^{\mathrm{FRD}}}{N_0 + \frac{h_k^{\mathrm{FRD}}}{B} p_{\widetilde{k}}^{\max}} - 1 & (p_k^{\mathrm{FRD}})^* = p_{\widetilde{k}}^{\max} \end{cases} \tag{45}$$

and $\gamma_{k,n}^{\textcircled{1}}$ is the Lagrange multiplier associated with (16). The proof of the solutions can be found in Appendix B.

IX. SIMULATION RESULTS

In this section, the performance of the optimal analytical findings for the iterative two-stage primal-dual decomposed problem are observed and it is proved that the performance results of the decomposed approach converges to the results that MATLAB CVX obtains by using the overall optimization problem in (31). Also, the proposed algorithm is compared with a scenario in which the distribution $d_{k,1}^{y}$ is not optimized and shared out equally among the participating nodes, i.e. $d_{k,1}^{y} = d^y/K$. By this way, this scenario does not optimize the distribution $d_{k,1}^{y}$ but it can still optimize the communication and computation parameters as well as the time t_n allocated for each layer operation. We show this scenario as *NoOptDist* in the plots. Although this scenario does not optimize the distribution, it still optimizes the communication and computation parameters, which makes this bound still quite ideal to be compared.

Inspired mainly by the articles [11], [18], [22], [47], the parameters are selected as in Table 3. The minimized collaborative computing energy consumption E^{MAC} per MAC operation, including first round distribution, local computation, and exchange communication of all collaborative layer operations (i.e. the solution of the external optimization problem (33) / $\sum_{k=1}^{K}\sum_{n=1}^{N^{\mathrm{LAY}}} N_{k,n}^{\mathrm{MAC}}$) is observed for three different scenarios shown as *Opt − Decomp* (found by using analytical results obtained in Appendices), *Opt − Overall* (found by solving the overall optimization problem without decomposing it) and *NoOptDist* by varying different parameters in the system. The common result for all figures below is that the optimal distribution of the input data among the heterogeneous nodes makes a considerable improvement on the energy consumption per MAC operation compared to the scenario that the distribution is performed equally among the nodes. Since there could be millions of MAC operations to be performed in a typical CNN model, multiplying E^{MAC} values with the number of MAC operations in the entire model makes even the tiny differences between the curves in the figures quite important. Another finding is that the results of the decomposed approach converge to the results found by using the overall optimization problem without decomposing it, which means we can analytically find the minimized total energy consumption without using any toolbox and modify it

TABLE 3. Selected parameters for the simulation results.

Parameter	Value	Units
κ_k	Unif($[10^{-28}; 10^{-27}]$)	/
c_k	Unif($[250; 750]$)	[CPU cycles/MAC]
$\nu_k^{\max}$	Unif($[2; 4]$)	[GHz]
h_k	$\mathcal{CN}(0, 10^{-3})$(Rayleigh)	/
$p_k^{\max}$	Unif($[10; 25]$)	[mW]
P_k^c	Unif($[10; 25]$)	[mW]
B	1	[MHz]
N_0	10^{-16}	[W/Hz]

mathematically, depending on the energy and latency requirements of the application to be operated.

In Fig. 4, the optimal t_n values allocated to each layer operation are compared as a bar graph by changing the total number of rows d^y of a 2D input data sample. As the layer-pack number increases, the optimization allocates less time to complete the given layer-pack operations since the data dimension at the back-end layers is so small compared to the first layers of the model. Therefore, the time t_n allocated to each layer-pack operation exponentially decreases while the data passes through the layers.

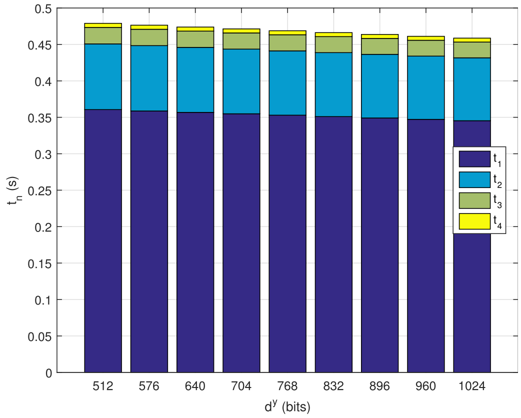

FIGURE 4. t_n **vs.** d^y **for** $N^{\text{LAY}} = 4$, $d^x = 512$ **bits,** $K = 10$ **and** $\tau = 0.5$ **s.**

In Fig. 5, unlike the optimal t_n values, the optimal distribution $d_{k,1}^y$ for each node is investigated for some d^y values. Each color represents the assignment of a single node and as can be seen, the optimal distribution does not converge to the equal sharing scenario due to having heterogeneous computing capabilities. The partial input data with a higher number of rows is assigned to the node having higher computing capability. The optimal distribution gives much more efficiency to the system compared to the equal distribution (*NoOptDist* scenario) as we observe in the next figures.

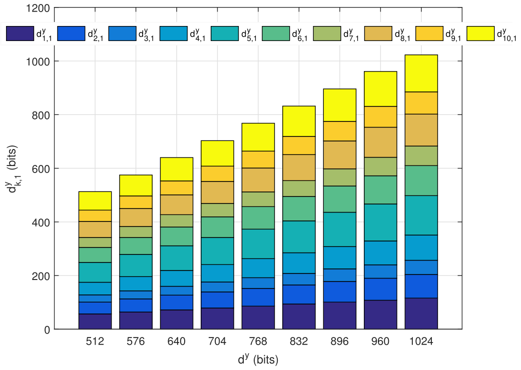

FIGURE 5. $d_{k,1}^y$ **vs.** d^y **for** $N^{\text{LAY}} = 4$, $d^x = 512$ **bits,** $K = 10$ **and** $\tau = 0.5$ **s.**

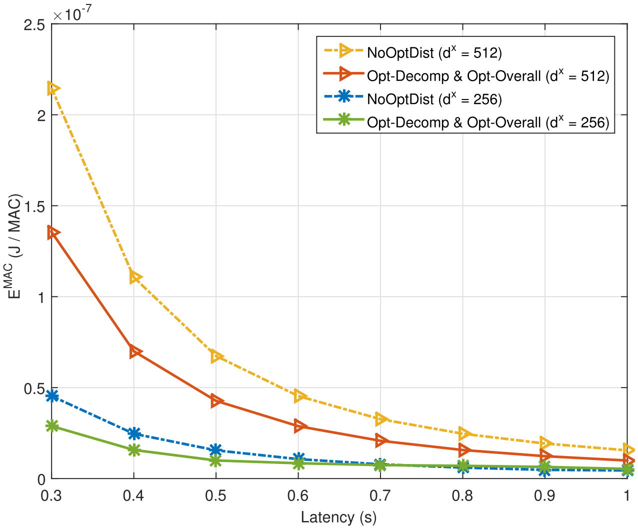

FIGURE 6. E^{MAC} **vs. latency** (τ) **for** $d^y = 1024$ **bits,** $K = 20$ **and** $N^{\text{LAY}} = 3$.

In Fig. 6, the maximum latency allowed for the whole operation is varied and the scenarios where $d^x = 256, 512$ bits are investigated. For $d^x = 512$ bits, it can be seen that the energy consumption E^{MAC} of *NoOptDist* scenario is approximately 50% larger than that of the proposed optimal distribution for small values of τ whereas these curves approach each other towards the end of the plot as they both have sufficient latency to achieve a very small energy consumption for large values of τ. For $d^x = 256$ bits, the optimal distribution still makes a difference on the energy consumption but not that much compared to the case where $d^x = 512$ bits.

In Fig. 7, we test the scheme with the different number of columns of the input data and place two different scenarios where $K = 10$ and $K = 20$. As the number of columns in the input data increases, the difference between optimized and non-optimized curves increases since the number of operations to be performed is larger and the optimal distribution becomes crucial for large values of d^x. Also, less energy consumption can be obtained with more participating MNs as compared between $K = 10$ and $K = 20$ curves in the figure where the curve for $K = 20$ has almost three times less energy consumption per MAC operation than the curve for $K = 10$. As the number of participating MNs is increased, the energy consumption difference between optimized and non-optimized cases decreases since the optimal distribution approaches to the equal distribution for a massive number of participating MNs.

In Fig. 8, the number of participating MNs K is changed to analyze its effect on the energy consumption. As K increases,

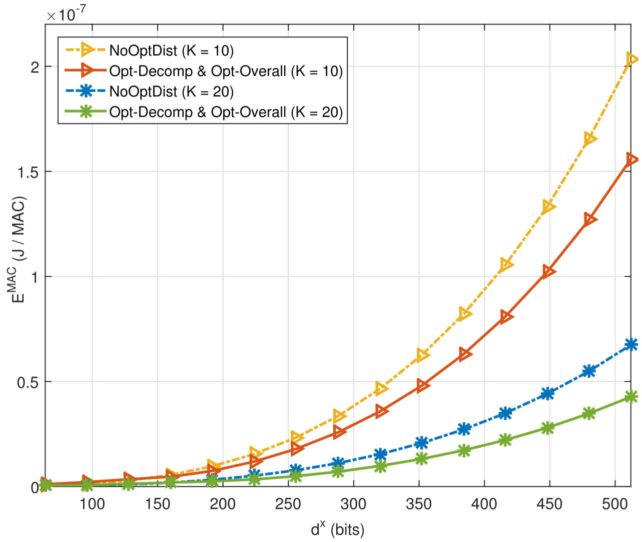

FIGURE 7. E^{MAC} **vs.** d^x **for** $d^y = 1024$ **bits,** $N^{\text{LAY}} = 3$ **and** $\tau = 0.5$ **s.**

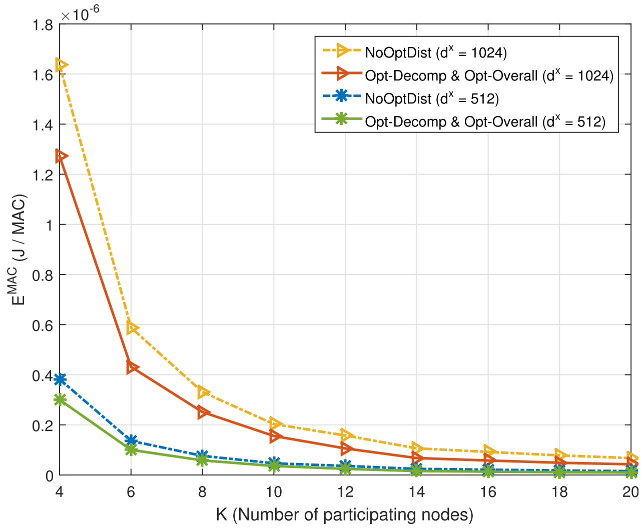

FIGURE 8. E^{MAC} **vs.** K **for** $d^y = 1024$ **bits,** $N^{\text{LAY}} = 3$ **and** $\tau = 1$ **s.**

the energy consumption E^{MAC} massively decreases and approaches very close to zero. The reason for it comes from the issue that when there are more nodes in the system, the effect of the less energy efficient nodes starts to disappear in the overall system performance because these less energy efficient nodes have less number of bits to perform with the increasing number of nodes. Also, the effectiveness of adding more nodes to the system decreases after some point in the plots since the number of nodes is increased by 50% from $K = 4$ to $K = 6$ while it is increased by 11% from $K = 18$ to $K = 20$. This is also one of the reasons that a massive decrease occurs at the beginning of the curves.

In Fig. 9, we observe the effect of the number of rows d^y of the input data on the energy consumption for two different scenarios where $N^{\text{LAY}} = 2$ and $N^{\text{LAY}} = 4$. The energy consumption difference between optimized and non-optimized scenario is evident, having almost 33% more efficient than the non-optimized ones for $d^y = 1024$ bits and $N^{\text{LAY}} = 4$. Also, as expected, increasing the number of layers N^{LAY} of the deep learning model makes the collaborative part of the system consume more energy but the overall energy consumption decreases when the number of layers N^{LAY} operated collaboratively increases as can be seen in the next figure (Fig. 10).

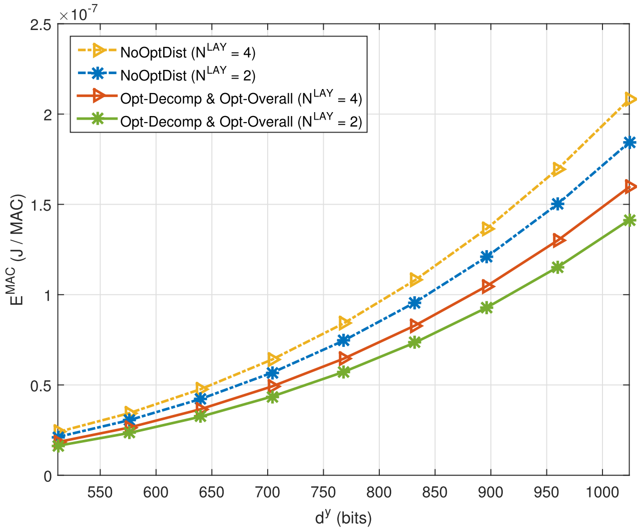

FIGURE 9. E^{MAC} **vs.** d^y **for** $d^x = 512$ **bits,** $K = 10$ **and** $\tau = 0.5$ **s.**

In Fig. 10, different than E^{MAC}, a new performance parameter E_{overall} is investigated. It is defined as the energy consumption of the overall system covering up the collaborative operations of the front-end layers (N^{LAY}) as well as the back-end layers which are completely operated in the master node (N^{TOT}-N^{LAY} number of layers). For this figure, the CNN model has $N_{\text{TOT}} = 5$ total number of layers and three scenarios are compared where $N^{\text{LAY}} = 3$, $N^{\text{LAY}} = 4$ and $N^{\text{LAY}} = 5$. As can be seen from Fig. 10, if the collaborative computation part is increased even one more layer for a 5-layer CNN model, a considerable amount of energy is saved in MNs (approximately 23% reduction in energy consumption between $N^{\text{LAY}} = 3$ and $N^{\text{LAY}} = 4$ for $d^y = 896$ bits), which shows the importance of the collaborative computation for this type of DNN applications. Increasing the number of layers that are collaboratively performed is more energy efficient than performing them in a single master node itself. If more layers are performed collaboratively, the master node becomes responsible for less number of local computation operations at the back-end layers so that it can easily deal with the reduced number of bits assigned to them. There is a trade-off at this point between the communication and computation. It is better for the front-end layer computations to be collaboratively performed in multiple nodes due to the high input data dimensions, while it is more advantageous to perform back-end layer computations at the master node alone since the input data dimensions of the back-end layers are so small and the cost of sending this data for collaborative computation becomes more expensive than performing these layer operations at the master node itself. Thus, the master

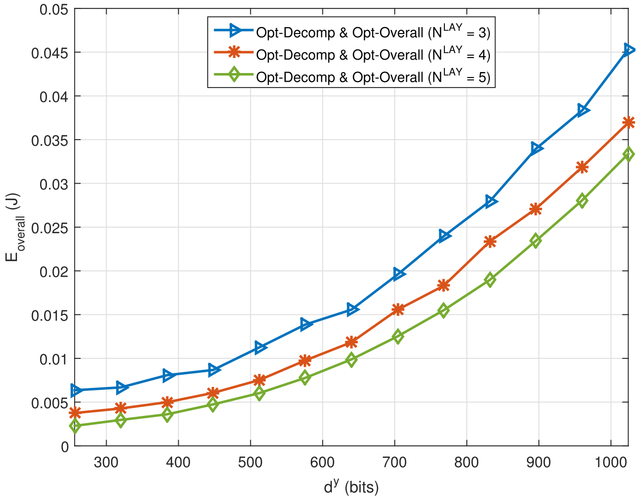

FIGURE 10. E_{overall} **vs.** d^y **for** $d^x = 256$ **bits,** $K = 10$**,** $\tau = 1$ **and** $N^{\text{TOT}} = 5$**.**

node sacrifices from its communication energy and saves its computational energy at the front-end layer operations, while it sacrifices from its computational energy and saves its communication energy at the back-end layer operations.

X. DISCUSSION, CONCLUSION AND FUTURE WORK

In this paper, we propose a novel scheme proposing a cloud-independent edge-based collaborative computing structure that can work efficiently in the absence of powerful MEC servers for deep learning-driven applications by considering both layer-based and horizontal DNN partitioning. The optimization over the distribution of the input data, communication, and computation parameters makes a considerable performance difference on the energy consumption of the system. Although the results are satisfying, there are some improvable points in this paper that we will work on as our future work. First of all, the interference between the mobile nodes in the communication model proposed in this paper is neglected. As the active number of mobile nodes increases considerably, the interference should be taken into account at some point. We also assume that the channel gains are known perfectly and the optimization is operated offline to purely observe the effect of the algorithm without the channel gain uncertainty, which is unrealistic for practical scenarios. Also, optimizing the placement of the mobile nodes on the input data, the time frame ordering during the first round distribution, the undetermined row assignment mentioned in Section III for next layer operations are some points that we plan to work on as our future research.

APPENDIX A
SOLUTION OF INTERNAL OPTIMIZATION PROBLEM

According to Lagrange duality theory, we have the following partial Lagrangian

$$\mathcal{L}(x_{k,n}, d^y_{k,1}, t_n, \beta_{k,n}) = \frac{3^6\kappa_k c_k^3 (d_n^x)^3 (d^y_{k,1})^3}{(2^{n-1})^3 (t^{\text{LOC}}_{k,n})^2}$$

$$+ I^{\text{a}}_{k,n}(E^{\text{EXC}_{\text{RF}_\text{a}}}_{k,n} + t^{\text{EXC}_\text{a}}_{k,n} P^c_k)$$
$$+ I^{\text{b}}_{k,n}(E^{\text{EXC}_{\text{RF}_\text{b}}}_{k,n} + t^{\text{EXC}_\text{b}}_{k,n} P^c_k)$$
$$+ \beta_{k,n}(t^{\text{LOC}}_{k,n} + I^{\text{a}}_{k,n} t^{\text{EXC}_\text{a}}_{k,n} + I^{\text{b}}_{k,n} t^{\text{EXC}_\text{b}}_{k,n} - t_n) \quad (46)$$

where $x_{k,n} = \{t^{\text{LOC}}_{k,n}, t^{\text{EXC}_{\{\text{a,b}\}}}_{k,n}, E^{\text{EXC}_{\text{RF}_{\{\text{a,b}\}}}}_{k,n}\}$ and $\beta_{k,n} \geq 0$ is the Lagrange multiplier associated with (22). The dual function is defined as

$$g(d^y_{k,1}, t_n, \beta_{k,n}) = \min_{x_{k,n}}. \mathcal{L}(x_{k,n}, d^y_{k,1}, t_n, \beta_{k,n})$$
$$\text{s.t. } (16), (19), (30)$$
$$0 \leq E^{\text{EXC}_{\text{RF}_{\{\text{a,b}\}}}}_{k,n}$$
$$0 \leq t^{\text{LOC}}_{k,n}, t^{\text{EXC}_{\{\text{a,b}\}}}_{k,n} \leq t_n \quad (47)$$

where the problem can be decomposed into three sub-problems. The first part is the local computation optimization problem,

$$\min_{t^{\text{LOC}}_{k,n}}. \frac{3^6\kappa_k c_k^3 (d_n^x)^3 (d^y_{k,1})^3}{(2^{n-1})^3 (t^{\text{LOC}}_{k,n})^2} + \beta_{k,n} t^{\text{LOC}}_{k,n}$$
$$\text{s.t. } \frac{9 c_k d_n^x d^y_{k,1}}{2^{n-1}\nu_k^{\max}} \leq t^{\text{LOC}}_{k,n} \leq t_n \quad (48)$$

The second one is the exchange communication with the neighbor below (assuming that $I^{\text{b}}_{k,n} = 1$ if there is any neighbor below),

$$\min_{t^{\text{EXC}_\text{b}}_{k,n}, E^{\text{EXC}_{\text{RF}_\text{b}}}_{k,n}}. E^{\text{EXC}_{\text{RF}_\text{b}}}_{k,n} + t^{\text{EXC}_\text{b}}_{k,n}(P^c_k + \beta_{k,n})$$
$$\text{s.t. } (19), (30),\ 0 \leq E^{\text{EXC}_{\text{RF}_\text{b}}}_{k,n},\ 0 \leq t^{\text{EXC}_\text{b}}_{k,n} \leq t_n \quad (49)$$

Finally, the exchange communication problem with the neighbor above can be solved just like (49).

A. LOCAL COMPUTATION SOLUTION OF PROBLEM (48)

The Lagrangian is

$$\mathcal{L}^{\textcircled{1}}_{k,n} = \frac{3^6\kappa_k c_k^3 (d_n^x)^3 (d^y_{k,1})^3}{(2^{n-1})^3 (t^{\text{LOC}}_{k,n})^2} + \beta_{k,n} t^{\text{LOC}}_{k,n}$$
$$+ \gamma^{\textcircled{1}}_{k,n}\left(\frac{9 c_k d_n^x d^y_{k,1}}{2^{n-1}\nu_k^{\max}} - t^{\text{LOC}}_{k,n}\right) + \gamma^{\textcircled{2}}_{k,n}\left(t^{\text{LOC}}_{k,n} - t_n\right) \quad (50)$$

where $\gamma^{\textcircled{1}}_{k,n}, \gamma^{\textcircled{2}}_{k,n} \geq 0$ are the Lagrange multipliers and with respect to the variable $t^{\text{LOC}}_{k,n}$, we have the KKT condition

$$\frac{\partial \mathcal{L}^{\textcircled{1}}_{k,n}}{\partial t^{\text{LOC}}_{k,n}} = -\frac{2\ 3^6\kappa_k c_k^3 (d_n^x)^3 (d^y_{k,1})^3}{(2^{n-1})^3 (t^{\text{LOC}}_{k,n})^3} + \beta_{k,n} - \gamma^{\textcircled{1}}_{k,n} + \gamma^{\textcircled{2}}_{k,n} = 0 \quad (51)$$

where the complementary slackness conditions are

$$\gamma^{\textcircled{1}}_{k,n}\left(\frac{9 c_k d_n^x d^y_{k,1}}{2^{n-1}\nu_k^{\max}} - t^{\text{LOC}}_{k,n}\right) = 0 \quad (52)$$
$$\gamma^{\textcircled{2}}_{k,n}(t^{\text{LOC}}_{k,n} - t_n) = 0 \quad (53)$$

We can find the optimal value of $t_{k,n}^{\mathrm{LOC}}$ as

- $t_{k,n}^{\mathrm{LOC}} = \frac{9\, c_k d_n^x d_{k,1}^y}{2^{n-1} \nu_k^{\max}} < t_n \Rightarrow \gamma_{k,n}^{\textcircled{1}} > 0, \gamma_{k,n}^{\textcircled{2}} = 0$: node k works at its highest CPU frequency and we substitute $t_{k,n}^{\mathrm{LOC}}$ into (51) $\Rightarrow \gamma_{k,n}^{\textcircled{1}} = -2\kappa_k (\nu_k^{\max})^3 + \beta_{k,n} > 0 \Rightarrow \beta_{k,n} > 2\kappa_k (\nu_k^{\max})^3$.
- $\frac{9\, c_k d_n^x d_{k,1}^y}{2^{n-1} \nu_k^{\max}} < t_{k,n}^{\mathrm{LOC}} < t_n \Rightarrow \gamma_{k,n}^{\textcircled{1}} = \gamma_{k,n}^{\textcircled{2}} = 0$: node k works not at its full CPU frequency. From (51), $t_{k,n}^{\mathrm{LOC}} = \frac{9\, c_k d_n^x d_{k,1}^y}{2^{n-1}} (\frac{2\kappa_k}{\beta_{k,n}})^{1/3}$ and we substitute $t_{k,n}^{\mathrm{LOC}}$ into the inequality $\Rightarrow \frac{2\; 3^6 \kappa_k c_k^3 (d_n^x)^3 (d_{k,1}^y)^3}{(2^{n-1})^3 (t_n)^3} < \beta_{k,n} < 2\kappa_k (\nu_k^{\max})^3$.
- $t_{k,n}^{\mathrm{LOC}} = t_n \Rightarrow \gamma_{k,n}^{\textcircled{1}} = 0, \gamma_{k,n}^{\textcircled{2}} > 0$: We substitute $t_{k,n}^{\mathrm{LOC}}$ into (51) $\Rightarrow \gamma_{k,n}^{\textcircled{2}} = \frac{2\; 3^6 \kappa_k c_k^3 (d_n^x)^3 (d_{k,1}^y)^3}{(2^{n-1})^3 (t_n)^3} - \beta_{k,n} > 0 \Rightarrow \beta_{k,n} < \frac{2\; 3^6 \kappa_k c_k^3 (d_n^x)^3 (d_{k,1}^y)^3}{(2^{n-1})^3 (t_n)^3}$.

Finally, we collect these results and obtain (34).

B. EXHANGE COMMUNICATION SOLUTION OF PROBLEM (49)

The Lagrangian is

$$\begin{aligned}\mathcal{L}_{k,n}^{\textcircled{2}} = {} & E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}} + t_{k,n}^{\mathrm{EXC_b}} \left(P_k^c + \beta_{k,n}\right) \\ & + \mu_{k,n}^{\textcircled{1}} (d_n^x - t_{k,n}^{\mathrm{EXC_b}} r_{k,n}^{\mathrm{EXC_b}}) - \mu_{k,n}^{\textcircled{2}} E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}} \\ & + \mu_{k,n}^{\textcircled{3}} (E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}} - t_{k,n}^{\mathrm{EXC_b}} p_k^{\max}) + \mu_{k,n}^{\textcircled{4}} (t_{k,n}^{\mathrm{EXC_b}} - t_n)\end{aligned} \tag{54}$$

where $\mu_{k,n}^{\textcircled{1}}, \mu_{k,n}^{\textcircled{2}}, \mu_{k,n}^{\textcircled{3}}, \mu_{k,n}^{\textcircled{4}} \geq 0$ are the Lagrange multipliers and we have the following KKT conditions

$$\frac{\partial \mathcal{L}_{k,n}^{\textcircled{2}}}{\partial E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}} = 1 - \mu_{k,n}^{\textcircled{2}} + \mu_{k,n}^{\textcircled{3}} - \mu_{k,n}^{\textcircled{1}} \frac{h_{k,n}^{\mathrm{EXC_b}}}{N_0 + \frac{h_{k,n}^{\mathrm{EXC_b}}}{B} \frac{E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}}{t_{k,n}^{\mathrm{EXC_b}}}} = 0 \tag{55}$$

and

$$\begin{aligned}\frac{\partial \mathcal{L}_{k,n}^{\textcircled{2}}}{\partial t_{k,n}^{\mathrm{EXC_b}}} = {} & P_k^c + \beta_{k,n} - \mu_{k,n}^{\textcircled{3}} p_k^{\max} + \mu_{k,n}^{\textcircled{4}} \\ & - \mu_{k,n}^{\textcircled{1}} r_{k,n}^{\mathrm{EXC_b}} + \mu_{k,n}^{\textcircled{1}} \frac{\frac{h_{k,n}^{\mathrm{EXC_b}}}{N_0} \frac{E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}}{t_{k,n}^{\mathrm{EXC_b}}}}{1 + \frac{h_{k,n}^{\mathrm{EXC_b}}}{B N_0} \frac{E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}}{t_{k,n}^{\mathrm{EXC_b}}}} = 0\end{aligned} \tag{56}$$

where the complementary slackness conditions are

$$\mu_{k,n}^{\textcircled{1}} (d_n^x - t_{k,n}^{\mathrm{EXC_b}} r_{k,n}^{\mathrm{EXC_b}}) = 0 \tag{57}$$

$$\mu_{k,n}^{\textcircled{2}} E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}} = 0 \tag{58}$$

$$\mu_{k,n}^{\textcircled{3}} (E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}} - t_{k,n}^{\mathrm{EXC_b}} p_k^{\max}) = 0 \tag{59}$$

$$\mu_{k,n}^{\textcircled{4}} (t_{k,n}^{\mathrm{EXC_b}} - t_n) = 0 \tag{60}$$

We can find the optimal values of $E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}$ and $t_{k,n}^{\mathrm{EXC_b}}$ as

- $E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}} = 0, 0 < t_{k,n}^{\mathrm{EXC_b}} < t_n \Rightarrow \mu_{k,n}^{\textcircled{2}} > 0, \mu_{k,n}^{\textcircled{3}} = 0$ and $\mu_{k,n}^{\textcircled{4}} = 0$:

  By using (55), $\mu_{k,n}^{\textcircled{2}} = 1 - \mu_{k,n}^{\textcircled{1}} \frac{h_{k,n}^{\mathrm{EXC_b}}}{N_0} > 0 \Rightarrow \mu_{k,n}^{\textcircled{1}} < \frac{N_0}{h_{k,n}^{\mathrm{EXC_b}}}$.
- $0 < E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}} < t_{k,n}^{\mathrm{EXC_b}} p_k^{\max} \Rightarrow \mu_{k,n}^{\textcircled{2}} = \mu_{k,n}^{\textcircled{3}} = 0$:

  From (55), $\mu_{k,n}^{\textcircled{1}} = \frac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} + \frac{E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}}{B t_{k,n}^{\mathrm{EXC_b}}} > 0$ and we obtain the following relation

$$\frac{E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}}{t_{k,n}^{\mathrm{EXC_b}}} = B(\mu_{k,n}^{\textcircled{1}} - \frac{N_0}{h_{k,n}^{\mathrm{EXC_b}}}) \tag{61}$$

  Also, because the multiplier $\mu_{k,n}^{\textcircled{1}}$ is greater than zero, i.e. $\mu_{k,n}^{\textcircled{1}} > 0$, the related constraint should be satisfied with equality to obey the complementary slackness condition in (57) so that the following equation occurs

$$d_n^x = t_{k,n}^{\mathrm{EXC_b}} B \ln(1 + \frac{h_{k,n}^{\mathrm{EXC_b}}}{B N_0} \frac{E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}}{t_{k,n}^{\mathrm{EXC_b}}}) \tag{62}$$

  which means that the exchange communication is performed at the achievable information rate. By substituting (61) into (62), we can find an expression for $t_{k,n}^{\mathrm{EXC_b}}$ as

$$t_{k,n}^{\mathrm{EXC_b}} = \frac{d_n^x}{B \ln(\frac{h_{k,n}^{\mathrm{EXC_b}}}{N_0} \mu_{k,n}^{\textcircled{1}})} \tag{63}$$

  where because we also have the constraint $t_{k,n}^{\mathrm{EXC_b}} \leq t_n$, the multiplier $\mu_{k,n}^{\textcircled{1}}$ should satisfy

$$\mu_{k,n}^{\textcircled{1}} \geq \frac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} e^{\frac{d_n^x}{B t_n}} \tag{64}$$

  In addition, from the definition $0 < \frac{E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}}}{t_{k,n}^{\mathrm{EXC_b}}} = B(\mu_{k,n}^{\textcircled{1}} - \frac{N_0}{h_{k,n}^{\mathrm{EXC_b}}}) < p_k^{\max}$, we have

$$\frac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} < \mu_{k,n}^{\textcircled{1}} < \frac{N_0}{h_{k,n}^{\mathrm{EXC_b}}} + \frac{p_k^{\max}}{B} \tag{65}$$

- $E_{k,n}^{\mathrm{EXC}_{\mathrm{RF_b}}} = t_{k,n}^{\mathrm{EXC_b}} p_k^{\max} \Rightarrow \mu_{k,n}^{\textcircled{2}} = 0, \mu_{k,n}^{\textcircled{3}} > 0$:

  From (55), we have

$$\mu_{k,n}^{\textcircled{3}} = \mu_{k,n}^{\textcircled{1}} \frac{h_{k,n}^{\mathrm{EXC_b}}}{N_0 + \frac{h_{k,n}^{\mathrm{EXC_b}}}{B} p_k^{\max}} - 1 > 0 \tag{66}$$

  where if $\mu_{k,n}^{\textcircled{1}} = 0$, the multiplier $\mu_{k,n}^{\textcircled{3}}$ is equal to -1, which is not possible. Therefore, the multiplier $\mu_{k,n}^{\textcircled{1}}$

should satisfy $\mu^{\textcircled{1}}_{k,n} > 0$ and to obey the complementary slackness condition in (57), we have the following equation

$$d^x_n = t^{\mathrm{EXC_b}}_{k,n} B \ln(1 + \frac{h^{\mathrm{EXC_b}}_{k,n}}{BN_0} p^{\max}_k) \tag{67}$$

and therefore, we have the following expression for $t^{\mathrm{EXC_b}}_{k,n}$

$$t^{\mathrm{EXC_b}}_{k,n} = \frac{d^x_n}{B \ln(1 + \frac{h^{\mathrm{EXC_b}}_{k,n}}{BN_0} p^{\max}_k)} = \frac{d^x_n}{r^{\mathrm{EXC_b}}_{k,n}\left(p^{\max}_k\right)} \tag{68}$$

Also, from (66),

$$\mu^{\textcircled{1}}_{k,n} > \frac{N_0}{h^{\mathrm{EXC_b}}_{k,n}} + \frac{p^{\max}_k}{B} \tag{69}$$

Finally, the optimal values for the optimization parameters can be found as (35) and (36).

APPENDIX B
SOLUTION OF EXTERNAL OPTIMIZATION PROBLEM

We have the following partial Lagrangian

$$\begin{aligned}
&\mathcal{L}(y_{k,n}, z_{k,n}) \\
&= \sum_{\substack{k=1\\k\neq\tilde{k}}}^{K} (E^{\mathrm{FRD_{RF}}}_k + t^{\mathrm{FRD}}_k P^c_{\tilde{k}}) \\
&\quad + \sum_{k=1}^{K}\sum_{n=1}^{N^{\mathrm{LAY}}} [cons^{\textcircled{1}}_{k,n}(d^y_{k,1})^3 - \beta_{k,n}t_n + \gamma^{\textcircled{1}}_{k,n} cons^{\textcircled{2}}_{k,n} d^y_{k,1}] \\
&\quad + \lambda(d^y - \sum_{k=1}^{K} d^y_{k,1}) + \phi(\sum_{\substack{k=1\\k\neq\tilde{k}}}^{K} t^{\mathrm{FRD}}_k + \sum_{n=1}^{N^{\mathrm{LAY}}} t_n - \tau) \\
&\quad + \sum_{\substack{k=1\\k\neq\tilde{k}}}^{K} \theta_k [d^x(d^y_{k,1} + 1 + I^{\mathrm{a}}_{k,1} I^{\mathrm{b}}_{k,1}) - t^{\mathrm{FRD}}_k r^{\mathrm{FRD}}_k]
\end{aligned} \tag{70}$$

where $y_{k,n} = \{E^{\mathrm{FRD_{RF}}}_k, t^{\mathrm{FRD}}_k, d^y_{k,1}, t_n\}$ and $z_{k,n} = \{\beta_{k,n}, \gamma^{\textcircled{1}}_{k,n}, \theta_k, \lambda, \phi\}$ and the constant terms in the Lagrangian can be expressed as

$$cons^{\textcircled{1}}_{k,n} = \frac{3^6 \kappa_k c^3_k (d^x_n)^3}{(2^{n-1})^3((t^{\mathrm{LOC}}_{k,n})^*)^2}, \quad cons^{\textcircled{2}}_{k,n} = \frac{9 c_k d^x_n}{2^{n-1} \nu^{\max}_k} \tag{71}$$

Also, $\phi, \theta_k \geq 0$ and $\lambda \in \Re$ are the Lagrange multipliers. The dual function is given as

$$\begin{aligned}
g(z_{k,n}) = \min_{y_{k,n}}.\ & \mathcal{L}(y_{k,n}, z_{k,n}) \\
\text{s.t. } & 0 \leq E^{\mathrm{FRD_{RF}}}_k \leq t^{\mathrm{FRD}}_k p^{\max}_{\tilde{k}} \\
& 0 \leq t^{\mathrm{FRD}}_k, t_n \leq \tau \\
& 0 \leq d^y_{k,1} \leq d^y
\end{aligned} \tag{72}$$

where the problem can be decomposed into three subproblems. The first problem is

$$\begin{aligned}
\min_{t_n}.\ & t_n(\phi - \sum_{k=1}^{K} \beta_{k,n}) \\
\text{s.t. } & 0 \leq t_n \leq \tau,
\end{aligned} \tag{73}$$

the second problem is

$$\begin{aligned}
\min_{d^y_{k,1}}.\ & \sum_{k=1}^{K} (d^y_{k,1})^3 \sum_{n=1}^{N^{\mathrm{LAY}}} cons^{\textcircled{1}}_{k,n} + \sum_{k=1}^{K} d^y_{k,1} \sum_{n=1}^{N^{\mathrm{LAY}}} \gamma^{\textcircled{1}}_{k,n} cons^{\textcircled{2}}_{k,n} \\
& - \sum_{k=1}^{K} \lambda\, d^y_{k,1} + \sum_{\substack{k=1\\k\neq\tilde{k}}}^{K} \theta_k d^x d^y_{k,1} \\
\text{s.t. } & 0 \leq d^y_{k,1} \leq d^y
\end{aligned} \tag{74}$$

and the last problem is

$$\begin{aligned}
\min_{E^{\mathrm{FRD_{RF}}}_k, t^{\mathrm{FRD}}_k}.\ & E^{\mathrm{FRD_{RF}}}_k + t^{\mathrm{FRD}}_k (P^c_{\tilde{k}} + \phi) - \theta_k t^{\mathrm{FRD}}_k r^{\mathrm{FRD}}_k \\
\text{s.t. } & 0 \leq E^{\mathrm{FRD_{RF}}}_k \leq t^{\mathrm{FRD}}_k p^{\max}_{\tilde{k}},\ \ 0 \leq t^{\mathrm{FRD}}_k \leq \tau
\end{aligned} \tag{75}$$

A. SOLUTION OF PROBLEM (73)

The Lagrangian is

$$\mathcal{L}^{\textcircled{3}}_n = t_n(\phi - \sum_{k=1}^{K} \beta_{k,n}) + \epsilon^{\textcircled{1}}_n (t_n - \tau) - \epsilon^{\textcircled{2}}_n t_n \tag{76}$$

where $\epsilon^{\textcircled{1}}_n, \epsilon^{\textcircled{2}}_n \geq 0$ are the Lagrange multipliers and we have the KKT conditions with respect to the variable t_n as

$$\frac{\partial \mathcal{L}^{\textcircled{3}}_n}{\partial t_n} = \phi - \sum_{k=1}^{K} \beta_{k,n} + \epsilon^{\textcircled{1}}_n - \epsilon^{\textcircled{2}}_n = 0 \tag{77}$$

where the complementary slackness conditions are

$$\epsilon^{\textcircled{1}}_n (t_n - \tau) = 0 \tag{78}$$

$$\epsilon^{\textcircled{2}}_n t_n = 0 \tag{79}$$

We can easily find the optimal value of t_n as in (37).

B. SOLUTION OF PROBLEM (74)

The problem (74) can be modified as

$$\begin{aligned}
\min_{d^y_{k,1}}.\ & (d^y_{k,1})^3 (\sum_{n=1}^{N^{\mathrm{LAY}}} cons^{\textcircled{1}}_{k,n}) - d^y_{k,1} \psi_k \\
\text{s.t. } & 0 \leq d^y_{k,1} \leq d^y
\end{aligned} \tag{80}$$

where

$$\psi_k = \begin{cases} \lambda - \sum_{n=1}^{N^{\mathrm{LAY}}} \gamma^{\textcircled{1}}_{k,n} cons^{\textcircled{2}}_{k,n} & k = \tilde{k} \\ \lambda - \sum_{n=1}^{N^{\mathrm{LAY}}} \gamma^{\textcircled{1}}_{k,n} cons^{\textcircled{2}}_{k,n} - \theta_k d^x & k \neq \tilde{k} \end{cases} \tag{81}$$

The Lagrangian for (80) is

$$\mathcal{L}_k^{\textcircled{4}} = (d_{k,1}^y)^3(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}}) - d_{k,1}^y \psi_k + \delta_k^{\textcircled{1}}(d_{k,1}^y - d^y) - \delta_k^{\textcircled{2}} d_{k,1}^y \tag{82}$$

where $\delta_k^{\textcircled{1}}, \delta_k^{\textcircled{2}} \geq 0$ are the Lagrange multipliers and we have the KKT conditions with respect to the variable $d_{k,1}^y$

$$\frac{\partial \mathcal{L}_k^{\textcircled{4}}}{\partial d_{k,1}^y} = 3(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}})(d_{k,1}^y)^2 - \psi_k + \delta_k^{\textcircled{1}} - \delta_k^{\textcircled{2}} = 0 \tag{83}$$

where the complementary slackness conditions are

$$\delta_k^{\textcircled{1}}(d_{k,1}^y - d^y) = 0 \tag{84}$$

$$\delta_k^{\textcircled{2}} d_{k,1}^y = 0 \tag{85}$$

We can find the optimal value of $d_{k,1}^y$ as

- $d_{k,1}^y = 0 \Rightarrow \delta_k^{\textcircled{1}} = 0, \delta_k^{\textcircled{2}} > 0$:
  From (83), $\delta_k^{\textcircled{2}} = -\psi_k > 0 \Rightarrow \psi_k < 0$
- $0 < d_{k,1}^y < d^y \Rightarrow \delta_k^{\textcircled{1}} = \delta_k^{\textcircled{2}} = 0$:
  From (83), $d_{k,1}^y = \sqrt{\frac{\psi_k}{3(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}})}}$ and we have the following inequality by substituting $d_{k,1}^y$ expression:
  $0 < \psi_k < 3(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}})(d^y)^2$
- $d_{k,1}^y = d^y \Rightarrow \delta_k^{\textcircled{1}} > 0, \delta_k^{\textcircled{2}} = 0$:
  From (83), $\delta_k^{\textcircled{1}} = \psi_k - 3(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}})(d^y)^2 > 0 \Rightarrow$
  $\psi_k > 3(\sum_{n=1}^{N^{\mathrm{LAY}}} cons_{k,n}^{\textcircled{1}})(d^y)^2$

We obtain the optimal $(d_{k,1}^y)^*$ as in (38).

C. SOLUTION OF PROBLEM (75)

This problem is similar to the problem that we have in the exchange communication in Appendix A-B with some exceptions. The Lagrangian is

$$\mathcal{L}_k^{\textcircled{5}} = E_k^{\mathrm{FRD_{RF}}} + t_k^{\mathrm{FRD}}(P_{\tilde{k}}^c + \phi) - \theta_k t_k^{\mathrm{FRD}} r_k^{\mathrm{FRD}} - \xi_k^{\textcircled{1}} E_k^{\mathrm{FRD_{RF}}} + \xi_k^{\textcircled{2}}(E_k^{\mathrm{FRD_{RF}}} - t_k^{\mathrm{FRD}} p_{\tilde{k}}^{\max}) - \xi_k^{\textcircled{3}} t_k^{\mathrm{FRD}} + \xi_k^{\textcircled{4}}(t_k^{\mathrm{FRD}} - \tau) \tag{86}$$

where $\xi_k^{\textcircled{1}}, \xi_k^{\textcircled{2}}, \xi_k^{\textcircled{3}}, \xi_k^{\textcircled{4}} \geq 0$ are the Lagrange multipliers and we have the KKT conditions with respect to the variables are

$$\frac{\partial \mathcal{L}_k^{\textcircled{5}}}{\partial E_k^{\mathrm{FRD_{RF}}}} = 1 - \xi_k^{\textcircled{1}} + \xi_k^{\textcircled{2}} - \theta_k \frac{h_k^{\mathrm{FRD}}}{N_0 + \frac{h_k^{\mathrm{FRD}}}{B} \frac{E_k^{\mathrm{FRD_{RF}}}}{t_k^{\mathrm{FRD}}}} = 0 \tag{87}$$

and

$$\frac{\partial \mathcal{L}_k^{\textcircled{5}}}{\partial t_k^{\mathrm{FRD}}} = P_{\tilde{k}}^c + \phi - \xi_k^{\textcircled{2}} p_{\tilde{k}}^{\max} - \xi_k^{\textcircled{3}} + \xi_k^{\textcircled{4}} - \theta_k r_k^{\mathrm{FRD}} + \theta_k \frac{\frac{h_k^{\mathrm{FRD}}}{N_0} \frac{E_k^{\mathrm{FRD_{RF}}}}{t_k^{\mathrm{FRD}}}}{1 + \frac{h_k^{\mathrm{FRD}}}{BN_0} \frac{E_k^{\mathrm{FRD_{RF}}}}{t_k^{\mathrm{FRD}}}} = 0 \tag{88}$$

where the complementary slackness conditions are

$$\xi_k^{\textcircled{1}} E_k^{\mathrm{FRD_{RF}}} = 0 \tag{89}$$

$$\xi_k^{\textcircled{2}}(E_k^{\mathrm{FRD_{RF}}} - t_k^{\mathrm{FRD}} p_{\tilde{k}}^{\max}) = 0 \tag{90}$$

$$\xi_k^{\textcircled{3}} t_k^{\mathrm{FRD}} = 0 \tag{91}$$

$$\xi_k^{\textcircled{4}}(t_k^{\mathrm{FRD}} - \tau) = 0 \tag{92}$$

By following the same path with the exchange communication solution in Appendix A-B, we can find the optimal value of $(p_k^{\mathrm{FRD}})^* = (E_k^{\mathrm{FRD_{RF}}})^*/(t_k^{\mathrm{FRD}})^*$ as in (39) where by using (88) and complementary slackness conditions, the optimal value of $(t_k^{\mathrm{FRD}})^*$ can be found as in (40).

REFERENCES

[1] C. Hardy, E. Le Merrer, and B. Sericola, "Distributed deep learning on edge-devices: Feasibility via adaptive compression," in *Proc. IEEE 16th Int. Symp. Netw. Comput. Appl. (NCA)*, Oct. 2017, pp. 1–8.

[2] M. Khayyat, I. A. Elgendy, A. Muthanna, A. S. Alshahrani, S. Alharbi, and A. Koucheryavy, "Advanced deep learning-based computational offloading for multilevel vehicular edge-cloud computing networks," *IEEE Access*, vol. 8, pp. 137052–137062, 2020.

[3] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, Jan. 2020.

[4] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[5] K. Yang, Y. Shi, and Z. Ding, "Data shuffling in wireless distributed computing via low-rank optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 12, pp. 3087–3099, Jun. 2019.

[6] Z. Zhao, K. M. Barijough, and A. Gerstlauer, "DeepThings: Distributed adaptive deep learning inference on resource-constrained IoT edge clusters," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2348–2359, Nov. 2018.

[7] R. Hadidi, J. Cao, M. S. Ryoo, and H. Kim, "Toward collaborative inferencing of deep neural networks on Internet-of-Things devices," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4950–4960, Jun. 2020.

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[9] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[10] L. Guan, X. Ke, M. Song, and J. Song, "A survey of research on mobile cloud computing," in *Proc. 10th IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, Sanya, China, May 2011, pp. 387–392.

[11] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4188–4200, Jun. 2019.

[12] Z. Ali, L. Jiao, T. Baker, G. Abbas, Z. H. Abbas, and S. Khaf, "A deep learning approach for energy efficient computational offloading in mobile edge computing," *IEEE Access*, vol. 7, pp. 149623–149633, 2019.

[13] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, "Collaborate edge and cloud computing with distributed deep learning for smart city Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8099–8110, Sep. 2020.

[14] E. El Haber, T. M. Nguyen, and C. Assi, "Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3407–3421, May 2019.

[15] J. Du, M. Shen, and Y. Du, "A distributed *in-situ* CNN inference system for IoT applications," in *Proc. IEEE 38th Int. Conf. Comput. Design (ICCD)*, Hartford, CT, USA, Oct. 2020, pp. 279–287.

[16] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge AI: Algorithms and systems," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2167–2191, 4th Quart., 2020.

[17] ETSI, Sophia Antipolis, France. (Sep. 2014). *Mobile-Edge Computing-Introductory Technical White Paper*. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf.

[18] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[19] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Coimbatore, India, Jan. 2016, pp. 1–8.

[20] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[21] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[22] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[23] *Multi-Access Edge Computing*. Accessed: Jun. 30, 2019. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

[24] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[25] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[26] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[27] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[28] C. You and K. Huang, "Exploiting non-causal CPU-state information for energy-efficient mobile cooperative computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4104–4117, Jun. 2018.

[29] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

[30] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[31] L. Zeng, X. Chen, Z. Zhou, L. Yang, and J. Zhang, "CoEdge: Cooperative DNN inference with adaptive workload partitioning over heterogeneous edge devices," *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 595–608, Apr. 2020.

[32] C.-C. Hsu, C.-K. Yang, J.-J. Kuo, W.-T. Chen, and J.-P. Sheu, "Cooperative convolutional neural network deployment over mobile networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–7.

[33] L. Zeng, E. Li, Z. Zhou, and X. Chen, "Boomerang: On-demand cooperative deep neural network inference for edge intelligence on the industrial Internet of Things," *IEEE Netw.*, vol. 33, no. 5, pp. 96–103, Sep. 2019.

[34] W. He, S. Guo, S. Guo, X. Qiu, and F. Qi, "Joint DNN partition deployment and resource allocation for delay-sensitive deep learning inference in IoT," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9241–9254, Oct. 2020.

[35] A. E. Eshratifar, M. S. Abrishami, and M. Pedram, "JointDNN: An efficient training and inference engine for intelligent mobile cloud computing services," 2018, *arXiv:1801.08618*. [Online]. Available: http://arxiv.org/abs/1801.08618

[36] C. Hu, W. Bao, D. Wang, and F. Liu, "Dynamic adaptive DNN surgery for inference acceleration on the edge," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 1423–1431.

[37] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Atlanta, GA, USA, Jun. 2017, pp. 328–339.

[38] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Comput. Archit. News*, vol. 45, no. 1, pp. 615–629, May 2017.

[39] F. M. Campos de Oliveira and E. Borin, "Partitioning convolutional neural networks for inference on constrained Internet-of-Things devices," in *Proc. 30th Int. Symp. Comput. Archit. High Perform. Comput. (SBAC-PAD)*, Lyon, France, Sep. 2018, pp. 266–273.

[40] T. Mohammed, C. Joe-Wong, R. Babbar, and M. D. Francesco, "Distributed inference acceleration with adaptive DNN partitioning and offloading," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Jul. 2020, pp. 854–863.

[41] J. Mao, X. Chen, K. W. Nixon, C. Krieger, and Y. Chen, "MoDNN: Local distributed mobile computing system for deep neural network," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 1396–1401.

[42] J. Mao, Z. Yang, W. Wen, C. Wu, L. Song, K. W. Nixon, X. Chen, H. Li, and Y. Chen, "MeDNN: A distributed mobile system with enhanced partition and deployment for large-scale DNNs," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Irvine, CA, USA, Nov. 2017, pp. 751–756.

[43] M. Grant and S. Boyd. (Sep. 2013). *CVX: MATLAB Software for Disciplined Convex Programming*. [Online]. Available: http://cvxr.com/cvx/

[44] T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze, "A method to estimate the energy consumption of deep neural networks," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2017, pp. 1916–1920.

[45] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[46] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[47] A. Paris, H. Mirghasemi, I. Stupia, and L. Vandendorpe, "Leveraging user-diversity in energy-efficient edge-facilitated collaborative fog computing," 2020, *arXiv:2004.00113*. [Online]. Available: http://arxiv.org/abs/2004.00113

[48] N. Janatian, I. Stupia, and L. Vandendorpe, "Optimal offloading strategy and resource allocation in SWIPT-based mobile-edge computing networks," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Lisbon, Portugal, Aug. 2018, pp. 1–6.

[49] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

EMRE KILCIOGLU received the B.Sc. and M.Sc. degrees in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Institute of Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM), Université Catholique de Louvain (UCLouvain), Ottignies-Louvain-La-Neuve, Belgium. He was a System Design Engineer with Aselsan Inc., Ankara, from July 2016 to January 2020. His research interests include massive MIMO, cooperative communication, fog computing, and deep learning applications for wireless communications.

HAMED MIRGHASEMI received the B.Sc. and M.Sc. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2006 and 2009, respectively, and the Ph.D. degree from Télécom ParisTech, Paris, France, in 2014. He is currently a Postdoctoral Researcher with the Université Catholique de Louvain (UCLouvain), Ottignies-Louvain-La-Neuve, Belgium. His research interests include information theory, stochastic optimization, and deep learning.

IVAN STUPIA received the Ph.D. degree from the University of Pisa, Italy, in 2009. In 2011, he joined with the Institute of Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM), Université Catholique de Louvain (UCLouvain), Belgium, where he is a Research Associate. He began his career with the University of Pisa. His academic experience is corroborated by more than 50 publications in international journals and proceedings of international conferences. He was involved in various European and national projects on wireless communications in different fields of application (cellular systems, wireless sensor networks, security, and aeronautical communications). His expertise and general interests span the areas of wireless communications and signal processing with special emphasis on application of advanced mathematical tools to the design of self-adaptive/self-organizing wireless networks, energy harvesting and wireless power transfer for the Internet of Things (IoT) services and green, and cost-effective design of wireless networks.

LUC VANDENDORPE (Fellow, IEEE) was born in Mouscron, Belgium, in 1962. He received the degree *(summa cum laude)* in electrical engineering and the Ph.D. degree in applied science from the Université Catholique de Louvain (UCLouvain), Ottignies-Louvain-La-Neuve, Belgium, in 1985 and 1991, respectively.

Since 1985, he has been with the Communications and Remote Sensing Laboratory, UCLouvain, where he first worked in the field of bit rate reduction techniques for video coding. In 1992, he was a Visiting Scientist and a Research Fellow with the Telecommunications and Traffic Control Systems Group, Delft Technical University, The Netherlands, where he worked on spread spectrum techniques for personal communications systems. From October 1992 to August 1997, he was a Senior Research Associate with Belgian NSF, UCLouvain, and an invited Assistant Professor. He is currently a Full Professor with the Institute for Information and Communication Technologies, Electronics, and Applied Mathematics, UCLouvain. His research interests include digital communication systems and more precisely resource allocation for OFDMA-based multicell systems, MIMO and distributed MIMO, sensor networks, UWB-based positioning, and wireless power transfer. He is or has been a TPC member for numerous IEEE conferences, such as VTC, GLOBECOM, SPAWC, ICC, PIMRC, and WCNC. He was an Elected Member of the Signal Processing for Communications Committee, from 2000 to 2005, and the Sensor Array and Multichannel Signal Processing Committee of the Signal Processing Society, from 2006 to 2008 and from 2009 to 2011. He was the Chair of the IEEE Benelux Joint Chapter on Communications and Vehicular Technology, from 1999 to 2003. He was a Co-Technical Chair of IEEE ICASSP 2006. He served as an Editor for Synchronization and Equalization of IEEE Transactions on Communications, from 2000 to 2002, and as an Associate Editor for IEEE Transactions on Wireless Communications, from 2003 to 2005, and IEEE Transactions on Signal Processing, from 2004 to 2006.