

Energy-Aware Distributed Edge ML for mHealth Applications With Strict Latency Requirements

Omar Hashash¹, Sanaa Sharafeddine², *Senior Member, IEEE*, Zaher Dawy³, *Senior Member, IEEE*, Amr Mohamed⁴, *Senior Member, IEEE*, and Elias Yaacoub⁵, *Senior Member, IEEE*

Abstract—Edge machine learning (Edge ML) is expected to serve as a key enabler for real-time mobile health (mHealth) applications. However, its reliability is governed by the limited energy and computing resources of user equipment (UE), along with the wireless channel variations and dynamic resource allocation at edge servers. In this letter, we incorporate both UE and edge server computing to satisfy the strict latency requirements of mHealth applications while efficiently utilizing the UE’s energy resources. Specifically, we separate the feature extraction and classification processes of Edge ML inference and formulate an optimization problem to distribute them between the UE and the edge server while determining the optimal UE transmit power. We demonstrate the effectiveness of the proposed approach using an mHealth case study for predicting epileptic seizures using data from wearable health devices.

Index Terms—Machine learning, mobile edge computing, neurological mHealth systems, seizure detection and prediction.

I. INTRODUCTION

REAL-TIME remote healthcare applications demand reliable solutions to satisfy their quality of service requirements. Emerging mobile health (mHealth) applications rely on data collected from various sources including wearable health devices and require utilization of effective machine learning (ML) algorithms for prediction and inference purposes.

In edge machine learning (Edge ML)-based mHealth applications, the processing can take place at the end-user level, such as IoT device or user equipment (UE) [1]. Example applications include epileptic seizure prediction using electroencephalogram (EEG) signals [2] and cardiac abnormalities detection using electrocardiogram (ECG) signals [3]. The main challenge with end-user processing is the limited energy and computing resources. Therefore, efforts have been focused on enabling real-time mHealth applications using advances in mobile edge computing (MEC). For example, MEC-based

epileptic seizure detection has been shown to achieve accurate results with low latency [4]. However, most mHealth applications that rely on MEC merely address the offloading aspects, such as assigning UE transmit power, enabling efficient resource allocation, and mitigating wireless channel variations. With the concept of distributed inference between UE and MEC evolving, an optimization problem is proposed in [5] to minimize the weighted-sum cost that includes delay and energy consumption of partitioning the lower and higher layers of a pre-trained convolutional neural network (CNN). Moreover, an inference offloading problem of deep neural network (DNN) partitioning is formulated in [6] to enhance energy efficiency of both UEs and base stations (BSs) while meeting stringent application delay. This concept could be further extended to reach out ML algorithms suitable for mHealth monitoring applications that allow for separable feature extraction and classification processes, e.g., logistic regression and SVM.

In this letter, we optimize the performance of real-time mHealth monitoring applications by efficiently distributing the feature extraction and classification processes of Edge ML between the UE and MEC server in response to wireless channel variations and dynamic resource allocation. The proposed approach demonstrates the value of integrating Edge ML with MEC, with the UE and edge server acting as cooperative agents to meet the application’s strict latency requirements while minimizing the UE’s energy consumption.

II. SYSTEM MODEL

Consider a UE operating a real-time mHealth monitoring application to monitor health data from wearable devices as shown in Fig. 1. The UE falls under the coverage of a network BS, which is equipped with an MEC server that is capable of computing the UE offloaded computations. The application relies on Edge ML to initiate real-time responses, where it is assumed that both UE and edge server have the same pre-trained ML model specific to run the mHealth application. Thus, we can characterize the feature extraction process of the ML application with a processing density of L_f CPU cycles/bit. Similarly, the classification process is characterized with a processing density of L_c CPU cycles/bit [7], [8].

Operating in an MEC framework, the system timeline can be divided into timeslots τ of duration T seconds (sec) as shown in Fig. 1. At the beginning of each timeslot τ , the UE periodically receives a data vector of size A_f (bits) for feature extraction. After extracting the features, the resultant feature vector of size A_c (bits) is used for model classification. Hence, the resulting classification vector I_c (bits) that contains

Manuscript received August 31, 2021; accepted September 28, 2021. Date of publication October 5, 2021; date of current version December 9, 2021. This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) under Grant NPRP12S-0305-190231. The findings achieved herein are solely the responsibility of the authors. The associate editor coordinating the review of this article and approving it for publication was J. Tang. (*Corresponding author: Zaher Dawy.*)

Omar Hashash and Zaher Dawy are with the Department of Electrical and Computer Engineering, American University of Beirut, Beirut 1107 2020, Lebanon (e-mail: onh02@mail.aub.edu; zd03@aub.edu.lb).

Sanaa Sharafeddine is with the Department of Computer Science and Mathematics, Lebanese American University, Beirut 1102 2801, Lebanon (e-mail: sanaa.sharafeddine@lau.edu.lb).

Amr Mohamed and Elias Yaacoub are with the Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar (e-mail: amrm@qu.edu.qa; eliasy@ieee.org).

Digital Object Identifier 10.1109/LWC.2021.3117876

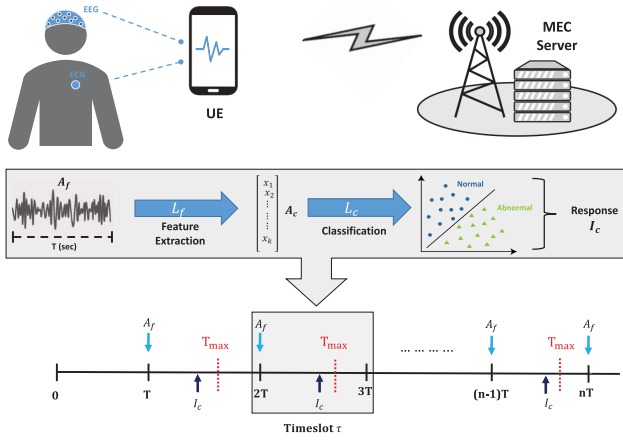


Fig. 1. Neurological mHealth monitoring system over the edge with the associated timeline processes.

the response should be used for real-time interpretation at the UE. This real-time process requires maintaining the computation time of the feature extraction and classification processes below maximum time limit T_{\max} , where $T_{\max} \leq T$. Hence, it is essential to consider each individual timeslot of this monitoring application in an independent manner.

Before the beginning of each timeslot, the MEC server notifies the UE about the computing resources available for the application. The assigned computing resources are represented in terms of the CPU frequency $f_{\text{MEC}}(\tau)$ (Hz) constant over timeslot τ . Meanwhile, the UE assigns its available computing resources $f_{\text{UE}}(\tau)$ (Hz) that is constant over each timeslot τ .

To enable distributed Edge ML, the feature extraction and classification processes are partitioned between the UE and MEC server, leading to four possible cases as shown in Fig. 2. To illustrate, case 1 refers to feature extraction and classification executed locally at the UE. On the other hand, case 4 refers to remote feature extraction and classification execution by offloading to the MEC server. Furthermore, cases 2 and 3 refer to distributed feature extraction and classification execution. Specifically, feature extraction is executed on UE and followed by classification at the MEC server in case 2, while feature extraction is performed at the MEC server and followed by classification at the UE in case 3. With the different vector sizes that can be offloaded while taking into consideration UE and MEC resource allocation, we can manage to choose the most energy efficient case that reliably guarantees meeting strict latency requirements for each τ .

Assuming that the response I_c has a minimal size in comparison with other vectors [6], [8], the latency $T_i(\tau)$ of each case $i \in \{1, 2, 3, 4\}$ shown in Fig. 2 is stated as:

$$T_1(\tau) = \frac{A_f L_f}{f_{\text{UE}}(\tau)} + \frac{A_c L_c}{f_{\text{UE}}(\tau)}, \quad (1)$$

$$T_2(\tau) = \frac{A_f L_f}{f_{\text{UE}}(\tau)} + \frac{A_c}{R_u(\tau)} + \frac{A_c L_c}{f_{\text{MEC}}(\tau)}, \quad (2)$$

$$T_3(\tau) = \frac{A_f}{R_u(\tau)} + \frac{A_f L_f}{f_{\text{MEC}}(\tau)} + \frac{A_c}{R_d(\tau)} + \frac{A_c L_c}{f_{\text{UE}}(\tau)}, \quad (3)$$

$$T_4(\tau) = \frac{A_f}{R_u(\tau)} + \frac{A_f L_f}{f_{\text{MEC}}(\tau)} + \frac{A_c L_c}{f_{\text{MEC}}(\tau)}, \quad (4)$$

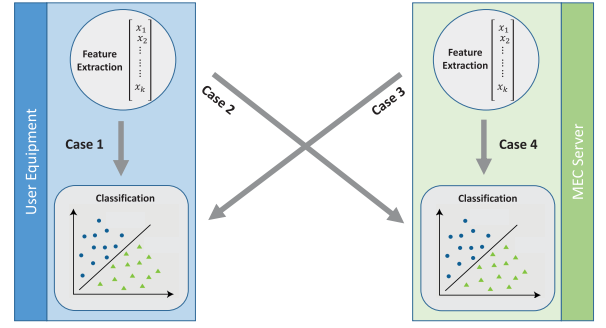


Fig. 2. Distributed Edge ML cases for mHealth monitoring system.

where $R_d(\tau)$ is the downlink data rate from the MEC server, and $R_u(\tau) = W \log_2(1 + \frac{h(\tau)P_t(\tau)}{N_0 W})$ is the UE uplink data rate where $h(\tau)$ is the wireless channel gain, $P_t(\tau)$ is the UE uplink transmit power for offloading, N_0 is the noise spectral density, and W is the uplink channel bandwidth.

To determine the consumed UE energy, we define the local computing power as $k[f_{\text{UE}}(\tau)]^3$, where k is a parameter directly related to the CPU hardware implementation [7]. Moreover, we define P_t^{\max} as the maximum UE transmit power and assume the consumed UE energy in the downlink is negligible. The corresponding UE energy for each case is the sum of local computing energy and energy needed for offloading over a timeslot. The UE energy consumed $E_i(\tau)$ for each case i is denoted as:

$$E_1(\tau) = k[f_{\text{UE}}(\tau)]^3 \times \left(\frac{A_f L_f}{f_{\text{UE}}(\tau)} + \frac{A_c L_c}{f_{\text{UE}}(\tau)} \right), \quad (5)$$

$$E_2(\tau) = k[f_{\text{UE}}(\tau)]^3 \times \frac{A_f L_f}{f_{\text{UE}}(\tau)} + P_t(\tau) \times \frac{A_c}{R_u(\tau)}, \quad (6)$$

$$E_3(\tau) = k[f_{\text{UE}}(\tau)]^3 \times \frac{A_c L_c}{f_{\text{UE}}(\tau)} + P_t(\tau) \times \frac{A_f}{R_u(\tau)}, \quad (7)$$

$$E_4(\tau) = P_t(\tau) \times \frac{A_f}{R_u(\tau)}. \quad (8)$$

It can be seen that the alternation between the cases can generate a mitigation process from elevated delays that might occur due to the dynamic resource allocation $f_{\text{UE}}(\tau)$ and $f_{\text{MEC}}(\tau)$, and the wireless conditions modeled in the channel gain $h(\tau)$. In addition, to design an energy efficient scheme while ensuring that latency requirements are met, the optimal value of $P_t(\tau)$ should be specified as it plays a vital role in determining latency and energy.

III. PROBLEM FORMULATION

To model the alternation in response to dynamic resource allocations and wireless channel conditions, it is required to formulate an optimization problem to determine the optimal case with the corresponding transmit power P_t . As the UE receives the channel gain $h(\tau)$ and the allocated resources $f_{\text{MEC}}(\tau)$ through control signaling, and has access to its own computing resources $f_{\text{UE}}(\tau)$, a decision making process can be initiated by the UE to determine the optimal case.

We define two binary variables a and b to denote whether feature extraction and classification, respectively, are performed locally at the UE or remotely at the MEC server as

TABLE I
LOCAL AND REMOTE EXECUTION

Case	a	b	Feature Extraction	Classification
1	1	1	Local	Local
2	1	0	Local	Remote
3	0	1	Remote	Local
4	0	0	Remote	Remote

shown in Table I. Furthermore, we omit the index τ from the variables for ease of notations. Then, we formulate the general latency T_o and energy E_o as a function of the binary variables as follows:

$$E_o = akA_fL_f[f_{UE}]^2 + bkA_cL_c[f_{UE}]^2 + (A_f + (A_c - A_f)a - abA_c) \frac{P_t}{W \log_2(1 + \frac{hP_t}{N_0W})}, \quad (9)$$

$$T_o = \left(\frac{A_fL_f}{f_{UE}} - \frac{A_fL_f}{f_{MEC}} \right) a + \left(\frac{A_cL_c}{f_{UE}} - \frac{A_cL_c}{f_{MEC}} \right) b + \frac{A_fL_f + A_cL_c}{f_{MEC}} + \frac{A_f + a(A_c - A_f) - abA_c}{W \log_2(1 + \frac{hP_t}{N_0W})} + \frac{b(1-a)A_c}{R_d}. \quad (10)$$

The optimization problem can then be formulated with UE energy minimization objective function as follows:

$$\begin{aligned} & \underset{a,b,P_t}{\text{minimize}} \quad E_o \\ & \text{subject to} \quad T_o \leq T_{\max}, \\ & \quad P_t \geq 0, \\ & \quad P_t - P_t^{\max}(1 - ab) \leq 0, \\ & \quad a, b \in \{0, 1\}. \end{aligned} \quad (11)$$

This is a mixed integer non-linear programming (MINLP) optimization problem. This problem is subject to the following constraints: i) the latency T_o in (10) should be maintained less than T_{\max} , ii) the transmit power P_t is a non-zero variable for all the cases except that of local feature extraction and classification and having an upper bound of P_t^{\max} , and iii) a and b are binary decision variables referring to the optimal case from Table I.

To transform the problem to a mixed integer linear programming (MILP) problem, the non-linearity introduced by the log function is removed by discretizing P_t into N levels such that

$$P_t = P_i = \frac{iP_t^{\max}}{N}, \quad (12)$$

where $i \in \{0, 1, 2, \dots, N\}$.

Then, the variables X and Y are introduced in (9) and (10) respectively, such that

$$X = \frac{P_t}{\log_2(1 + \frac{hP_t}{N_0W})}, \quad Y = \frac{1}{\log_2(1 + \frac{hP_t}{N_0W})} \quad (13)$$

Since N discrete values of P_t are defined, then we have N values for X and Y , respectively. Thus, X and Y are formulated as a summation of individual P_t values as follows:

$$X = \sum_{i=1}^N k_i x_i = \sum_{i=1}^N \frac{k_i P_i}{\log_2(1 + \frac{hP_i}{N_0W})}, \quad (14)$$

TABLE II
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
k	10^{-27}	N_0	-174 dBm/Hz
A_f	7.127 Mbit	P_t^{\max}	1 W
A_c	1920 bit	T	60 sec
L_f	2339 cycle/bit	T_{\max}	46 sec
L_c	8250 cycle/bit	R_d	1 Mbps
W	1 MHz	N	4 levels

$$Y = \sum_{i=1}^N k_i y_i = \sum_{i=1}^N \frac{k_i}{\log_2(1 + \frac{hP_i}{N_0W})}, \quad (15)$$

where $k_i \in \{0, 1\} \quad \forall i \in \{1, 2, \dots, N\}$.

To further handle the non-linearity in (11), common linearization techniques are utilized, where the product of the binary variables a and b is replaced by an additional binary variable c . In addition, the product of binary and continuous variables obtained from the multiplication of X and Y with the binary variables is also replaced. The resulting problem is a MILP and is solved using MATLAB.

IV. NEUROLOGICAL mHEALTH CASE STUDY: PERFORMANCE RESULTS AND ANALYSIS

As an application to the presented problem, we target an mHealth application that demands real-time inference to predict epileptic seizures occurrence. The system is based on the EEG feature extraction algorithm in [9]. To simulate this system, we consider a UE at a distance of $d = 50$ m from the BS and an operating frequency $f = 5.8$ GHz. We assume that the channels are subject to Rayleigh flat fading of unity variance, and the coherence time is considered greater than T_{\max} . The rest of the parameters are found in Table II [8].

A. System Performance Over a Single Timeslot

In this section, we aim to verify the optimal decision making by recording how the system responds to simulation conditions that can be encountered for a single timeslot. Hence, we consider one timeslot having a gain $h = -140$ dBm and assigned $f_{UE} = 0.5$ GHz. Then, according to the value of f_{MEC} that has been assigned for this timeslot, the optimal decision to ensure that the latency requirement is not violated while maintaining energy efficiency can be determined. Thus, we consider f_{MEC} to have a uniform distribution ranging below 1 GHz, and show the proposed decision making with the corresponding P_t in Fig. 3.

As shown in Fig. 3, when f_{MEC} is relatively minimal, the optimal decision leads to completing the feature extraction and classification process on the UE with no transmit power. However, if f_{MEC} is assigned slightly beyond 0.02 GHz, distributed computing is favored by executing feature extraction on the UE and followed by classification at the MEC server with P_t reaching 25% of max power P_t^{\max} . This scenario remains favored for assigned f_{MEC} values reaching 0.58 GHz, after which the decision is shifted towards remote MEC with P_t reaching 75% of the max power for f_{MEC} slightly beyond 0.58 GHz. The increase in P_t is verified as to explicitly increase the data rate to compensate computing delay resulting from the limited share of f_{MEC} . When f_{MEC} is assigned

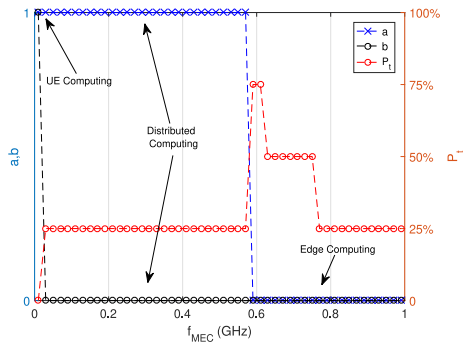


Fig. 3. Decision making with optimal transmit power over one timeslot according to f_{MEC} (GHz) assuming $h = -140$ dB and $f_{UE} = 0.5$ GHz.

beyond this critical inflection stage, the increased data rate is no longer needed to preserve the delay below T_{max} , and P_t can be gradually decreased to 50% of the max power. For f_{MEC} beyond 0.75 GHz, P_t can be decreased to 25% of the max transmit power afterwards to facilitate offloading. With the advantage of limiting P_t below P_t^{max} while considering the optimal decision making over the range of f_{MEC} , we can facilitate higher UE energy efficiency.

B. System Performance Over Multiple Timeslots

In this section, we consider decision making over multiple timeslots and record the performance in terms of the cumulative UE energy consumed and the computing latency. For this purpose, we simulate the system over a series of 30 timeslots, where both f_{UE} and f_{MEC} are assumed to have a uniform distribution such that $0.3 \text{ GHz} \leq f_{UE} \leq 0.8 \text{ GHz}$ and $0.15 \text{ GHz} \leq f_{MEC} \leq 1.2 \text{ GHz}$. Over each timeslot, the optimal decision is determined to choose the most energy efficient case without violating the latency T_{max} , with the corresponding optimal P_t . Moreover, we compare the effectiveness of our approach, with a static method which operates for one of the four cases explicitly with constant transmit power P_t^{max} , and with a baseline that represents the global optimal solution found through exhaustive search.

As shown in Fig. 4(a), the cumulative UE energy consumed for 30 timeslots has shown variations for different cases. Hence, cases 1 and 2 reach around 159 Joules (J) in comparison to 24.73 J in cases 3 and 4. Nonetheless, with the optimal decision making we reach a cumulative energy of 42.43 J, which is close to the range of cases 3 and 4. As the optimal decision making process allows for energy reduction, this also offers an advantage in terms of latency restriction over all timeslots as compared to all other alternative cases. As shown in Fig. 4(b), the latency over each timeslot changes reaching values beyond the strict latency T_{max} . Comparing the static technique to the dynamic decision making process in terms of latency, a latency mitigation process arises specifically for situations having latency above T_{max} , where the cases have recorded values over timeslots reaching up to 100 sec. Remarkably, it can be seen that optimal decision making records latency values below T_{max} for the 30 timeslots, and performs close to the global optimal solution.

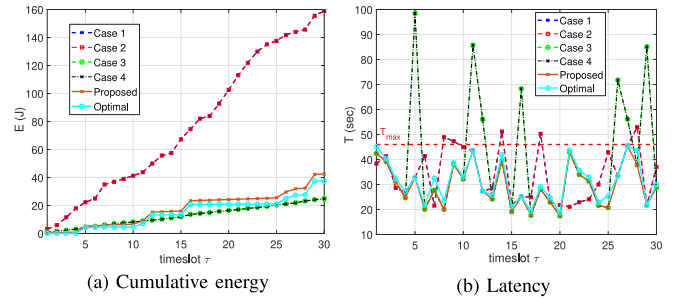


Fig. 4. a) Cumulative energy and b) latency over 30 timeslots where $0.3 \text{ GHz} \leq f_{UE} \leq 0.8 \text{ GHz}$ and $0.15 \text{ GHz} \leq f_{MEC} \leq 1.2 \text{ GHz}$.

V. CONCLUSION

In this letter, we present a distributed Edge ML system for mHealth monitoring applications with strict latency requirements. We partition the feature extraction and classification processes of the real-time inference application between UE and MEC server. This aims to mitigate elevated latency values due to the dynamic wireless conditions and computing resource allocation of the UE and MEC server, whilst ensuring UE energy efficiency. We formulate an optimization problem to model the decision making process and determine the optimal transmit power. We include an mHealth seizure prediction case study with our proposed method and demonstrate performance gains in terms of reduced energy and latency. An interesting extension is to leverage deep learning as an efficient approach to solve the optimization problem [10].

REFERENCES

- [1] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.
- [2] F. Samie, S. Paul, L. Bauer, and J. Henkel, "Highly efficient and accurate seizure prediction on constrained IoT devices," in *Proc. Design Autom. Test Eur. Conf. Exhibit.*, Mar. 2018, pp. 955–960.
- [3] S. Raj, "An efficient IoT-based platform for remote real-time cardiac activity monitoring," *IEEE Trans. Consum. Electron.*, vol. 66, no. 2, pp. 106–114, May 2020.
- [4] Z. S. Ali, N. Subramanian, and A. Erbad, "Smart health monitoring for seizure detection using mobile edge computing," in *Proc. Int. Wireless Commun. Mobile Comput.*, Jun. 2020, pp. 1903–1908.
- [5] B. Yang, X. Cao, C. Yuen, and L. Qian, "Offloading optimization in edge computing for deep learning enabled target tracking by Internet-of-UAVs," 2020. [Online]. Available: arXiv:2008.08001.
- [6] Z. Xu *et al.*, "Energy-aware inference offloading for DNN-driven applications in mobile edge clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 4, pp. 799–814, Apr. 2021.
- [7] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.
- [8] O. Hashash, S. Sharafeddine, and Z. Dawy, "MEC-based energy-aware distributed feature extraction for mHealth applications with strict latency requirements," in *Proc. IEEE ICC 4th Int. Workshop IoT Enabling Technol. Healthcare*, Montreal, QC, Canada, Jun. 2021, pp. 1–6.
- [9] M. Nassrallah, M. Haidar, H. Alawieh, A. El Hajj, and Z. Dawy, "Patient-aware EEG-based feature and classifier selection for e-health epileptic seizure prediction," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.
- [10] B. Yang, X. Cao, J. Bassey, X. Li, and L. Qian, "Computation offloading in multi-access edge computing: A multi-task learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 9, pp. 2745–2762, Sep. 2021.