NEMESIS: No-Regret E-health User Experience in Multi-Access Edge Computing Systems

Aisha B Rahman, Odyssefs Diamantopoulos Pantaleon, and Eirini Eleni Tsiropoulou
Email: {{arahman3, odiamantopoulospanta, eirini}@unm.edu}
Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA

***Abstract*—The rapid growth of data and computing needs in the Internet of Medical Things (IoMT) necessitates efficient mechanisms for optimizing the resource management in e-health applications. This paper presents the NEMESIS framework, which enables the users to determine their optimal Multi-Access Edge Computing (MEC) server selection and data offloading strategies by considering the reliability of the MEC servers based on individual interactions and shared user experiences. A comprehensive system model is introduced that defines the users' interactions, the data offloading processes, and the impact of various IoMT devices, along with a novel utility function that evaluates the tradeoffs in the MEC server selection and task offloading. Additionally, a reliability model is proposed that incorporates the direct user interactions and their peers evaluations of the MEC servers' computing services, while a regret learning mechanism is designed to optimize the users' strategies under varying information scenarios. The results demonstrate that the NEMESIS framework operates efficiently in real-time and outperforms state-of-the-art scheduling and offloading schemes in terms of latency and energy consumption.**

***Index Terms*—E-health, Regret Learning, Reliability, Edge Computing, Internet of Medical Things.**

I. Introduction

The Internet of Medical Things (IoMT) enables real-time communication between individuals and smart medical devices like fitness trackers and glucose monitors, enhancing accuracy and response times while supporting proactive care. However, the large volume of data generated requires high computing power for real-time processing. Multi-access Edge Computing (MEC) addresses this by placing servers closer to users, enabling faster data processing [1]. In this paper, the NEMESIS framework is introduced to enable the users to determine their optimal MEC servers selection and data offloading strategies while accounting for the MEC servers' reliability levels to process the offloaded data. The reliability levels are assessed based on both the individual user interactions and the collective experiences shared among the users. A regret learning approach is introduced to enable the users to determine their optimal server selection and data offloading strategies under scenarios of complete and incomplete information.

A. Related Work

Task offloading in edge computing has been recently explored in IoMT systems to support the substantial data processing requirements stemming from the large number of IoMT devices [2]. A task classification and scheduling scheme is proposed in [3] considering a multi-layered edge computing environment to optimize the latency for critical e-Health applications. A machine learning-based task offloading mechanism is designed in [4] to address the computational demands and energy consumption of the IoMT devices. A multi-agent soft-critic-discrete scheme for task offloading and resource allocation in IoMT is analyzed in [5] aiming at optimizing the throughput and power consumption while adhering to ultra-reliable and low-latency communication constraints using Lyapunov optimization and the extreme value theory. A similar method is proposed in [6] aiming at balancing the accuracy, performance, and energy consumption through the task offloading process. A decentralized federated framework that integrates MEC capabilities and software-defined networking is discussed in [7] to optimize the clinical decision support systems based on double deep Q-networks.

Additionally, the problem of *task scheduling and MEC server selection* complements the task offloading optimization to further improve the IoMT devices' experienced latency and energy consumption. A priority-based task scheduling and resource allocation mechanism for MEC-enabled IoMT systems is designed in [8] to optimize the task processing time, bandwidth use, and response to emergency conditions based on data from smart wearable devices. A Q-learning-based algorithm is proposed in [9] to optimize the unmanned aerial vehicles trajectories for real-time, energy-efficient, and delay-sensitive transmission of patient vital signs in IoMT systems. An adaptive resource allocation scheme for MEC-assisted IoMT systems is introduced in [10] utilizing demand forecasting and queuing theory to improve the resource utilization. A multi-objective meta-heuristic method for scheduling and offloading augmented reality applications in IoMT systems is discussed in [11] in order to address the privacy protection, mobility, latency, and energy consumption of the IoMT devices. A low-complexity task scheduling iterative algorithm is designed in [12] to minimize the average Peak Age of Information in a real-time e-Health IoMT system.

B. Contributions and Outline

Despite the significant advancements in task offloading and scheduling within MEC-assisted IoMT systems, most existing studies primarily target either the latency or the energy consumption without adequately exploring their interrelationship. Also, they frequently neglect the integration of user-

specific experiences and collective knowledge to improve the decision-making process. This paper presents the NEMESIS framework, which addresses these research gaps by enabling the users to optimize their MEC server selection and data offloading strategies. NEMESIS considers the reliability of the MEC servers based on individual interactions and shared user experiences and incorporates a regret learning approach to adapt to varying information scenarios.

The main contributions of this paper are summarized below.

1) The paper introduces a comprehensive system model for e-health applications utilizing MEC servers and defining the users' interactions, data offloading processes, and the impact of various IoMT devices on the data management. Also, the wireless communication characteristics among the users' IoMT devices and the MEC servers are thoroughly analyzed.
2) A novel utility function is introduced considering both the energy and time overheads, which helps in evaluating the trade-offs involved in the users' optimal MEC servers' selection and task offloading decisions.
3) A novel reliability model is proposed considering the users' direct interactions with the MEC servers to process their IoMT devices' data, as well as the experience shared from trusted peer evaluations. The proposed reliability model accounts for the users' experiences and reviews to the MEC servers for their provided computing services and is based on an intelligently aggregated feedback from the users' direct and indirect interactions with the MEC servers through their peers.
4) A regret learning mechanism is designed to determine the users' optimal MEC servers' selection and task offloading strategies under complete and incomplete information regarding their peers' decisions. The regret learning algorithm's convergence is shown to an ϵ-coarse correlated equilibrium point, enabling the stable operation of the MEC-assisted IoMT system.
5) Detailed numerical results show the efficient operation of the NEMESIS framework in a real-time manner, as well as its superiority compared to state-of-the-art task scheduling and offloading schemes in IoMT systems in terms of improved latency and energy consumption.

The remainder of this paper is organized as follows. Section II introduces the MEC-assisted e-health system model, while the MEC servers' reliability model is discussed in Section III. Section IV analyzes the regret learning-based MEC server selection and task offloading NEMESIS framework, while detailed numerical results are presented in Section V. Finally, Section VI concludes the paper.

II. MEC-assisted e-health System Model

An IoMT system is considered, consisting of a set of users $\mathcal{N} = \{1, \ldots, n, \ldots, |\mathcal{N}|\}$, where each user is equipped with an IoMT device, e.g., smartwatches, fitness trackers, glucose monitors, etc. A set of MEC servers $\mathcal{K} = \{1, \ldots, k, \ldots, |\mathcal{K}|\}$ resides in the users' vicinity, supporting their computing demands to offload and process their e-health data generated by their IoMT devices. The users' IoMT devices require substantial data processing to monitor the users' health metrics, such as movement and heart rate. To efficiently manage these processing tasks, the users can offload their data to multiple MEC servers. If a user n offloads data to the MEC server k at timeslot t, then $a^t_{n,k} = 1$ (otherwise $a^t_{n,k} = 0$). Let us denote by $b^t_{n,k}$ $[bits]$ the volume of data offloaded to MEC server k by the user n.

The users experience different levels of path loss during the task offloading process, which depends on their physical distance from the MEC servers and the surrounding environment and its fading characteristics. The path loss characteristics experienced by the users significantly impact their task offloading decisions, particularly regarding their experienced latency and energy consumption.

Based on the third generation partnership project (3GPP) [13], the Line of Sight (LoS) and non-LoS (NLoS) path losses experienced between node n and a MEC server k residing in 2D and 3D distances, denoted as $d^{2D}_{n,k}(t)$ [m] and $d^{3D}_{n,k}(t)$ [m], respectively, during timeslot t, are represented as $PL^{LoS}_{n,k}(t)$ [dB] and $PL^{NLoS}_{n,k}(t)$ [dB], respectively. The calculation for these losses is as follows:

$$PL^{LoS}_{n,k}(t) = \begin{cases} PL_1[dB], & \text{if } 10m \leq d^{2D}_{n,k}(t) \leq d_{BP} \\ PL_2[dB], & \text{if } d_{BP} < d^{2D}_{n,k}(t) \leq 5km \end{cases} \quad (1a)$$

$$PL^{NLoS}_{n,k}(t) = \max(PL^{LoS}_{n,k}(t), 13.54 + 39.08\log_{10}(d^{3D}_{n,k}(t)) + 20\log_{10}(f_c) - 0.6(h_n - 1.5)) \quad (1b)$$

where $PL_1 = 28 + 22\log_{10}(d^{3D}_{n,k}(t)) + 20\log_{10}(f_c)$ and $PL_2 = 28 + 40\log_{10}(d^{3D}_{n,k}(t)) + 20\log_{10}(f_c) - 9\log_{10}[d^2_{BP} + (h_k - h_n)^2]$. Here, h_k [m] is the height of the MEC server with an effective height of $h'_k = h_k - h_E$, h_n [m] is the height of the user n with an effective height of $h'_n = h_n - h_E$ with $h_E = 1$ [m] if $h_n < 13$[m], and $d_{BP} = \frac{4h'_k h'_n f_c}{c}$, where c [m/s] denotes the speed of light. The resulting path loss is calculated as, $PL^t_{n,k} = Pr^{LoS}_{n,k}(t)PL^{LoS}_{n,k}(t) + (1 - Pr^{LoS}_{n,k}(t))PL^{NLoS}_{n,k}(t)$, where $Pr^{LoS}_{n,k}(t)$ is the probability of experiencing LoS communication during the timeslot t:

$$Pr^{LoS}_{n,k}(t) = \begin{cases} 1, & \text{if } d^{2D}_{n,k}(t) \leq 18m \\ [(\frac{18}{d^{2D}_{n,k}(t)} + e^{-\frac{d^{2D}_{n,k}(t)}{63}}(1 - \frac{18}{d^{2D}_{n,k}(t)}))(1 + C'(h_n) \\ \times \frac{5}{4}(\frac{d^{2D}_{n,k}(t)}{100})^3 e^{-\frac{d^{2D}_{n,k}(t)}{150}})], & \text{if } 18m < d^{2D}_{n,k}(t) \end{cases} \quad (2)$$

with $C'(h_n) = 0$ if $h_n \leq 13$ [m], or, $C'(h_n) = \frac{h_n - 13}{10}^{1.5}$ if $13 < h_n \leq 23$ [m]. The resulting channel gain between the user n and the MEC receiver k during timeslot t is $g^t_{n,k} = \frac{1}{10^{\frac{PL^t_{n,k}}{10}}}$.

Considering the users' channel gain characteristics and their transmission power $P_n[W], \forall n \in \mathcal{N}$, the user's corresponding data rate is given as follows:

$$r^t_{n,k} = \omega \log_2 \left(1 + \frac{P_n g^t_{n,k}}{I_0 + \sum\limits_{\forall n' \in \mathcal{K}_n} P_{n'} g^t_{n',k}}\right) \quad [bps] \quad (3)$$

where ω [Hz] denotes the available bandwidth in the communication links among the users and the MEC server, and $\mathcal{K}_n$ denotes the set of other users simultaneously offloading to the same server k selected by n. Thus, the transmission latency experienced by user n while offloading $b^t_{n,k}$ to the MEC server k in timeslot t is calculated as follows:

$$TL^t_{n,k} = \frac{a^t_{n,k} b^t_{n,k}}{r^t_{n,k}} \quad [s] \tag{4}$$

The latency experienced by a user in order for the MEC server to process the offloaded data is defined as the duration from its arrival to its completion. This total time consists of two parts, i.e., the time spent waiting and the time spent on computation. To represent the execution dynamics of a task on an edge server, we utilize an M/M/1 queue model, since both the intervals between incoming tasks and their respective processing times follow exponential distributions. Thus, the average serving time for a user n who offloads $b^t_{n,k}$ data to server k in timeslot t is determined as follows [14]:

$$SL^t_{n,k} = \frac{a^t_{n,k}}{\frac{\phi^t_n a^t_{n,k} b^t_{n,k} F_k}{\frac{\sum_{\forall n' \in \mathcal{K}_n \cup n} \phi^t_{n'} a^t_{n',k} b^t_{n',k}}{c_k} - \frac{n^t_k}{\Delta t}}} \quad [s] \tag{5}$$

where F_k $[\frac{CPU-Cycles}{s}]$ is the CPU frequency of the server k, ϕ^t_n is the computing intensity $[\frac{CPU-cycles}{bit}]$ requirement of user's n data, $c_k[CPU-Cycles]$ is the average number of CPU cycles assigned to tasks arriving, n^t_k is the number of tasks executed on k in time slot t such that $\frac{n^t_k}{\Delta t}$ represents the arriving rate of tasks at the MEC server k.

The user's utility is determined by the processing of a substantial volume of data on a reliable MEC server (first term of Eq. 6), while also considering low energy consumption for offloading the data to the MEC server (second term of Eq. 6), and minimal latency to facilitate the timely offloading and processing of the data (third term of Eq. 6). Thus, the user's utility from offloading and processing its data to a MEC server k is captured as follows.

$$\begin{aligned} U^t_{n,k} = \alpha_n \ln\left(1 + \sum_{\forall k \in \mathcal{K}} a^t_{n,k} \rho^t_{n,k} b^t_{n,k}\right) - \\ \beta_n \sum_{\forall k \in \mathcal{K}} TL^t_{n,k} P_n - \gamma_n \sum_{\forall k \in \mathcal{K}} (TL^t_{n,k} + SL^t_{n,k}) \end{aligned} \tag{6}$$

where $\alpha_n, \beta_n[\frac{1}{J}], \gamma_n[\frac{1}{s}] \in \mathbb{R}^+$ are controlling parameters of each term's impact on the user's utility, and $\rho^t_{n,k} \in \mathbb{R}^+$ represents the user's belief regarding the MEC server's reliability with respect to the provided computing services, as explained in detail in the next section.

III. MEC Server's Reliability

The MEC servers exhibit varying levels of reliability based on the computing services they offer to the users. Each user develops a unique perception of the reliability of the MEC servers while considering their personalized treatment during the data offloading and processing. This perception is influenced by the volume of data being offloaded, the actual computing capabilities of the MEC server, and the proximity of the server to the user. The user's individualized belief regarding the reliability of the MEC server is defined as follows:

$$\mathcal{B}^t_{n,k} = \frac{a^t_{n,k} b^t_{n,k}}{d^{3D}_{n,k}(t)} \text{Zipf}(x_k) \tag{7}$$

where $\text{Zipf}(x_k) = \frac{z_1}{(1/x_k)^{z_2}}$, $X = \{x_1, \ldots, x_k, \ldots X_K\}$, $x_k, z_1, z_2 \in \mathbb{R}^+$ and larger x denotes higher CPU frequency. If $\mathcal{B}^t_{n,k} > \mathcal{B}_{\text{thr}}$, where $\mathcal{B}_{\text{thr}} > 0$, then the user perceives a positive experience with the MEC server k resulting in increased confidence in the server's reliability. The exact opposite holds true if $\mathcal{B}_{\text{thr}} < 0$. In both cases, the user's perception of the server's reliability diminishes over time as previous experiences become less salient, replaced by more recent interactions. Based on this observation, the user's evaluation of both positive (Eq. 8) and negative services (Eq. 8) can be expressed as follows:

$$R^+_{n,k} = \sum_{\lambda=1}^{\lambda_{n,k}} \delta^\lambda_{n,k} \log_2\left(\frac{b}{T - t^\lambda_{n,k}} + 1\right) \tag{8}$$

$$R^-_{n,k} = \sum_{\lambda=1}^{\lambda_{n,k}} (1 - \delta^\lambda_{n,k}) \log_2\left(\frac{b}{T - t^\lambda_{n,k}} + 1\right) \tag{9}$$

where

$$\delta^{\lambda_{n,k}}_{n,k} = \begin{cases} 1, & \text{if } \mathcal{B}^t_{n,k} \geq \mathcal{B}_{\text{thr}} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

is a binary variable capturing if the MEC server's provided services satisfied the user's reliability threshold or not, $\lambda_{n,k}$ is the number of times that user n selected to offload $b^t_{n,k}$ bits to the MEC server k, T is the total time that we examine the data offloading process, $t^\lambda_{n,k}$ is the time instance that user n selected the MEC server k at the $\lambda_{n,k}$th interaction, and $b > 0$ captures the time decay factor. It is noted that small values of b indicate a faster rate of forgetting past interactions.

Based on the services experienced by the users from the MEC servers, each user n develops a personal reliability belief for each server k, which is derived as follows.

$$\mathcal{R}_{n,k} = \mathbb{E}\left(\text{beta}(R^+_{n,k} + 1, R^-_{n,k} + 1)\right) = \frac{R^+_{n,k} + 1}{R^+_{n,k} + R^-_{n,k} + 2} \tag{11}$$

In the MEC-assisted IoMT system, all the users share a common objective, i.e., to receive the fastest service from the MEC servers while efficiently transmitting and processing their data. Thus, the personal reliability belief of user n regarding server k is influenced by other users n' who are also served by server k and exhibit similar levels of personal reliability belief. Specifically, if the absolute difference between the reliability beliefs of users n and n' for server k falls within a predefined threshold R_{Thr} (i.e., $|\mathcal{R}_{n,k} - \mathcal{R}_{n',k}| \leq R_{Thr}$), then user n considers user n' as a trusted peer in evaluating server k. As a result, each user n will establish a set of trusted users, denoted as $\mathcal{N}^{\text{tr}}_n$, which collectively influence their trust in the server. The final expression for the personal reliability

belief of user n for the MEC server k, regarding the positive (Eq. 12) and negative (Eq. 13) offered computing services, is derived as follows.

$$\hat{R}^{+}_{n,k} = w_1 R^{+}_{n,k} + w_2 \sum_{n'=1}^{|\mathcal{N}^{\text{tr}}_n|} R^{+}_{n',k} \tag{12}$$

$$\hat{R}^{-}_{n,k} = w_1 R^{-}_{n,k} + w_2 \sum_{n'=1}^{|\mathcal{N}^{\text{tr}}_n|} R^{-}_{n',k} \tag{13}$$

Thus, the overall level of personal reliability belief calculated by user n for a MEC server k is formulated as follows.

$$\rho^{t}_{n,k} = \mathbb{E}\left(\text{beta}(\hat{R}^{+}_{n,k}+1, \hat{R}^{-}_{n,k}+1)\right) = \frac{\hat{R}^{+}_{n,k}+1}{\hat{R}^{+}_{n,k}+\hat{R}^{-}_{n,k}+2} \tag{14}$$

IV. NEMESIS Framework

In this section, the NEMESIS framework is introduced to enable the users to autonomously determine their optimal MEC server selection and data offloading strategies. To achieve this goal, a regret learning-motivated noncooperative game model is introduced. Formally, the game is defined as $G = [\mathcal{N}, \{\mathcal{S}_n\}_{\forall n \in \mathcal{N}}, \{U^{t}_{n,k}\}_{\forall n \in \mathcal{N}, k \in \mathcal{K}}]$, where $\mathcal{N}$ is the set of users (players), $\mathcal{S}_n$ is the user's strategy set with $(s_n = (a^{t}_{n,k}, b^{t}_{n,k}))$, and $U^{t}_{n,k}$ denotes the user's utility function (Eq. 6). In this work, we present a detailed examination of the NEMESIS framework within two distinct scenarios: (1) when complete information is available and (2) when complete information regarding the strategies of other users is lacking.

Complete Information: In this case, each user is aware of the strategies of the rest of the users and selects a strategy $s_n \in \mathcal{S}_n$ with a given probability. Given that $|\mathcal{S}_n| < +\infty$, the game G has at least one equilibrium in mixed strategies, which is defined as follows.

Definition 1: (**ϵ-coarse correlated equilibrium**) A mixed probability strategy profile $\mathsf{Pr}_n = (\mathsf{Pr}_{n,s^1_n}, \ldots, \mathsf{Pr}_{n,s^{|\mathcal{S}_n|}_n})$ is an ϵ-coarse correlated equilibrium, if $\forall n \in \mathcal{N}$ and $\forall s'_n \in \mathcal{S}_n$ the following property holds.

$$\sum_{\forall \mathbf{s}_{-n}} (U^{t}_{n,k}(s'_n, \mathbf{s}_{-n}) \mathsf{Pr}_{-n,\mathbf{s}_{-n}}) - \sum_{\forall s_n} (U^{t}_{n,k}(s_n, \mathbf{s}_{-n}) \mathsf{Pr}_{n,s_n}) \leq \epsilon$$

It is noted that $\mathsf{Pr}_{-n,\mathbf{s}_{-n}} = \sum_{\forall s_n \in \mathcal{S}_n} \mathsf{Pr}(s_n, \mathbf{s}_{-n})$ represents the marginal probability distribution concerning s_n and $\sum_{\forall s_n \in \mathcal{S}_n} \mathsf{Pr}_{n,s_n} = 1$. For every $n \in \mathcal{N}$, the regret for choosing a strategy $s^i_n, i \in \{1, 2, \ldots, \mathcal{S}_n\}$ during a particular round t is calculated as follows.

$$\begin{aligned} r_{n,s^i_n}(t) = \lambda \left[U^{t}_{n,k}(s^i_n, \mathbf{s}_{-n}) - U^{t}_{n,k}(s_n, \mathbf{s}_{-n})\right] + \\ (1-\lambda)\frac{1}{t-1}\sum_{j=1}^{t-1}\left[U^{j}_{n,k}(s^i_n, \mathbf{s}_{-n}) - U^{j}_{n,k}(s_n, \mathbf{s}_{-n})\right] \end{aligned} \tag{15}$$

where $\lambda \in \mathbb{R}^{+}$ discounts the influence of past regrets [15]. User's n regret vector is: $\mathbf{r}_n(t) = (r_{n,s^1_n}(t), \ldots, r_{n,s^{|\mathcal{S}_n|}_n}(t))$.

From the analysis of a particular regret value during a specific iteration, the subsequent insights can be derived. If $r_{n,s^i_n}(t) > 0$, the user n was better off choosing s^i_n previously, while if $r_{n,s^i_n}(t) \leq 0$, then user n has no regret in choosing s^i_n. In the following rounds of strategy selection, user n chooses the strategy that exhibits the greatest regret based on the mixed strategy probability distribution (Eq. 16), where $r^{\max}_{n,s^i_n}(t) = \max_{\forall s^i_n \in \mathcal{S}_n} \{0, r_{n,s^i_n}(t)\}$.

$$\mathsf{Pr}_{n,s^i_n}(t) = \frac{r^{\max}_{n,s^i_n}(t)}{\sum_{\forall\, s^i_n \in \mathcal{S}_n} r^{\max}_{n,s^{i'}_n}(t)} \tag{16}$$

Incomplete Information: In practical scenarios, the user n is typically unaware of the MEC server selection and data offloading strategies employed by other users. Thus, each user n must rely solely on their own information and environmental factors, such as interference, when executing the optimal strategy selection process [15]. As a result, the user's regret in the context of incomplete information regarding the strategies of other users can be reformulated as follows.

$$\begin{aligned} \hat{r}_{n,s^i_n}(t) = \lambda \left[\hat{U}^{t}_{n,k}(s^i_n, \mathbf{s}_{-n}) - \hat{U}^{t}_{n,k}(s_n, \mathbf{s}_{-n})\right] \\ +(1-\lambda)\frac{1}{t-1}\sum_{j=1}^{t-1}\left[U^{j}_{n,k}(s^i_n, \mathbf{s}_{-n}) - U^{j}_{n,k}(s_n, \mathbf{s}_{-n})\right] \end{aligned} \tag{17}$$

where $\hat{U}^{t}_{n,k} = \alpha_n \ln\left(1 + \sum_{\forall k \in \mathcal{K}} a^{t}_{n,k}\rho^{t}_{n,k}b^{t}_{n,k}\right) - \beta_n \sum_{\forall k \in \mathcal{K}} TL^{t}_{n,k}P_n$. Similarly to the Complete Information scenario, the user selects a strategy in the following rounds based on the mixed strategy probability distribution, with $\hat{r}^{\max}_{n,s^i_n}(t) = \max_{\forall s^i_n \in \mathcal{S}_n} \{0, \hat{r}_{n,s^i_n}(t)\}$.

$$\hat{\mathsf{Pr}}_{n,s^i_n}(t) = \frac{\hat{r}^{\max}_{n,s^i_n}(t)}{\sum_{\forall\, s^i_n \in \mathcal{S}_n} \hat{r}^{\max}_{n,s^{i'}_n}(t)} \tag{18}$$

Theorem 1: The regret learning algorithm under the complete and incomplete information scenarios converges to an ϵ-coarse correlated equilibrium.

Proof: To demonstrate the regret learning algorithm's convergence, we need to show the user's utility function, which feeds the regret values' calculation, is a Lipschitz function, while also considering that the strategy space of each user $\mathcal{S}_n$ is bounded [16]. In the following, we analyze each term of Eq. 6. For the first term, we have: $\frac{\partial}{\partial b^{t}_{n,k}}\left(\alpha_n \ln\left(1 + \sum_{\forall k \in \mathcal{K}} a^{t}_{n,k}\rho^{t}_{n,k}b^{t}_{n,k}\right)\right) = \frac{\alpha_n a^{t}_{n,k}\rho^{t}_{n,k}}{1+a^{t}_{n,k}\rho^{t}_{n,k}b^{t}_{n,k}}$ with $\lim_{b^{t}_{n,k}\to\infty} \frac{\alpha_n a^{t}_{n,k}\rho^{t}_{n,k}}{1+a^{t}_{n,k}\rho^{t}_{n,k}b^{t}_{n,k}} = 0$ and $\lim_{b^{t}_{n,k}\to 0} \frac{\alpha_n a^{t}_{n,k}\rho^{t}_{n,k}}{1+a^{t}_{n,k}\rho^{t}_{n,k}b^{t}_{n,k}} = \alpha_n a^{t}_{n,k}\rho^{t}_{n,k}$. Focusing on the second term, we have: $\frac{\partial}{\partial b^{t}_{n,k}}\left(\beta_n \sum_{\forall k \in \mathcal{K}} TL^{t}_{n,k}P_n\right) = \frac{\beta_n P_n a^{t}_{n,k}}{\omega \log_2(1+\frac{P_n g^{t}_{n,k}}{I_0 + \sum_{\forall n' \in \mathcal{K}_n} P_{n'} g^{t}_{n',k}})} \in \mathbb{R}$. Finally, for the third term of Eq. 6, we have: $\frac{\partial}{\partial b^{t}_{n,k}}\left(\gamma_n \sum_{\forall k \in \mathcal{K}} TL^{t}_{n,k} + SL^{t}_{n,k}\right) = \frac{\partial}{\partial b^{t}_{n,k}}\left(\gamma_n TL^{t}_{n,k}\right) - \frac{\gamma_n a^{t}_{n,k} c_k^2 \Delta t^2 \frac{a^{t}_{n,k}\phi^{t}_{n,k}}{\sum_{\forall n' \in \mathcal{K}_n \cup n} a^{t}_{n',k}\phi^{t}_{n',k}b^{t}_{n',k}} F_k}{(\frac{a^{t}_{n,k}\phi^{t}_{n,k}b^{t}_{n,k}}{\sum_{\forall n' \in \mathcal{K}_n \cup n} a^{t}_{n',k}\phi^{t}_{n',k}b^{t}_{n',k}} F_k \Delta t - c_k n^{t}_{k})^2}$. The first term of the

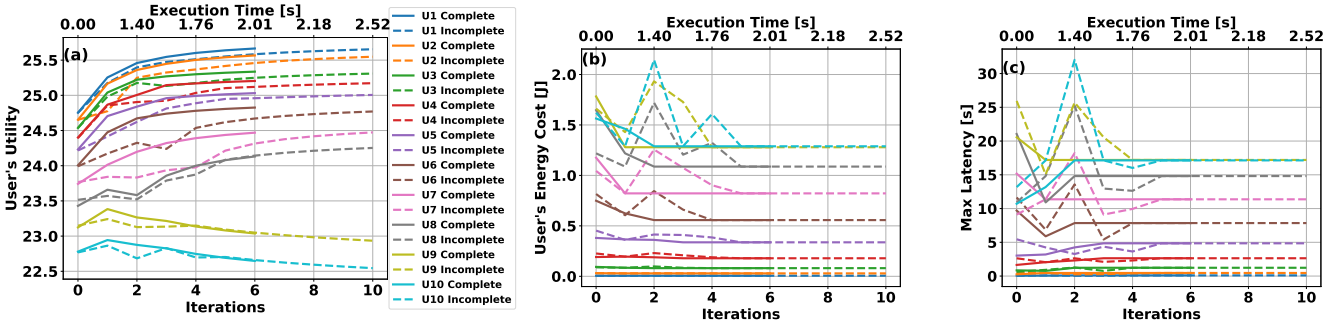

Fig. 1: NEMESIS framework's pure operation and performance from the users' perspective.

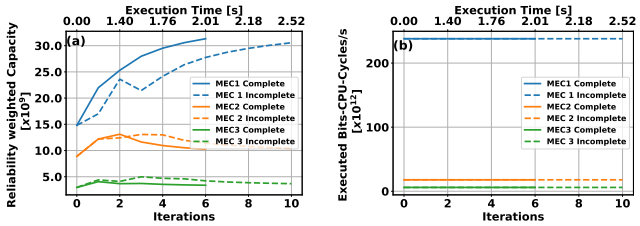

Fig. 2: NEMESIS framework's pure operation and performance from the MEC servers' perspective.

later derivative is bounded, while for the second term, we have: $\lim_{b^t_{n,k}\to\infty}\left(-\frac{\gamma_n a^t_{n,k}c_k^2\Delta t^2\frac{a^t_{n,k}\phi^t_{n,k}}{\sum_{\forall n'\in\mathcal{K}_n\cup n}a^t_{n',k}\phi^t_{n',k}b^t_{n',k}}F_k}{(\frac{a^t_{n,k}\phi^t_{n,k}b^t_{n,k}}{\sum_{\forall n'\in\mathcal{K}_n\cup n}a^t_{n',k}\phi^t_{n',k}b^t_{n',k}}F_k\Delta t-c_k n_k^t)^2}\right)=0$ and $\lim_{b^t_{n,k}\to 0}\left(-\frac{\gamma_n a^t_{n,k}c_k^2\Delta t^2\frac{a^t_{n,k}\phi^t_{n,k}}{\sum_{\forall n'\in\mathcal{K}_n\cup n}a^t_{n',k}\phi^t_{n',k}b^t_{n',k}}F_k}{(\frac{a^t_{n,k}\phi^t_{n,k}b^t_{n,k}}{\sum_{\forall n'\in\mathcal{K}_n\cup n}a^t_{n',k}\phi^t_{n',k}b^t_{n',k}}F_k\Delta t-c_k n_k^t)^2}\right)=-\frac{\gamma_n a^t_{n,k}c_k^2\Delta t^2}{c_k^2 n_k^{t\,2}}\in\mathbb{R}$. Thus, the derivative of the utility function is bounded, resulting in a Lipschitz function. ■

V. Numerical Results

In this section, a detailed simulation-based analysis is presented to demonstrate the pure operation and performance of the NEMESIS framework (Section V-A) and its scalability (Section V-B). Also, a real-world application of the NEMESIS is presented (Section V-C) along with a detailed comparative evaluation to other state-of-the-art task scheduling and offloading schemes in IoMT systems (Section V-D). The following parameters were used for the simulation, $F_k \in [1,5]\frac{GCPU-Cycles}{s}$, $n_k^t = 20$, $c_k = 1GCPU-Cycles$, $\Delta t = 1s$, $\omega = 5MHz$, $f_c = 952.6MHz$, $P_n = 50mW$, $b = 0.7$, $\lambda = 0.8$, $d^{3D}_{n,k}(t) \in [65,250]m$, $h_n = 3m$, $h_k = 25m$, unless otherwise stated. For demonstration purposes, we consider that users with higher ID are characterized by longer distances from the servers and lower amount of generated data. Similarly, MEC servers with higher ID reside at longer distances from the users and have lower CPU frequency F_k.

A. Pure Operation of NEMESIS framework

Fig. 1a – 1c present the users' utility (Eq. 6), energy cost (second term of Eq. 6), and maximum experienced latency by offloading to multiple selected servers as a function of the regret learning mechanism's iterations (lower horizontal axis) and its real execution time (upper horizontal axis) under the complete and incomplete information scenarios, respectively. The results show that users residing at further distances from the servers experience higher energy cost and latency, resulting in lower utility. Moreover, the results indicate that the proposed NEMESIS framework under the incomplete information scenario achieves very similar results as in the complete information, while experiencing longer convergence time (almost double), due to the longer exploration process.

Fig. 2a - 2b illustrates the MEC servers' weighted computing capacity based on the users' reliability (i.e., $\sum_{n\in\mathcal{N}}\rho^t_{n,k}\cdot F_k$) and their data processing load (i.e., $\sum_{n\in\mathcal{N}} b^t_{n,k}\cdot F_k$) as a function of the regret learning mechanism's iterations and real execution time, respectively. The results show that the MEC servers with higher computing capacity consistently achieve higher reliability among the users due to the superior computing services they provide. This leads to their higher weighted computing capacity (Fig. 2a), enabling them to handle a larger data processing load (Fig. 2b).

B. Scalability Analysis

In this section, a scalability analysis of the NEMESIS framework is performed for an increasing number of users to quantify its robustness in large-scale IoMT systems. Fig. 3a – 3b present the mean users' utility, latency, energy cost, and execution time of the NEMESIS framework as a function of an increasing percentage of users, respectively. The results show that a 100% increase in the number of users results in 5.84% decrease in the users' mean utility, and a relatively large increase in the latency, energy cost, and execution time of NEMESIS. However, it is noted that even with this large increase in the number of users, the absolute values of the latency, energy cost, and execution time remain reasonably small. Thus, we conclude that the NEMESIS framework can still be implemented in a near-real-time manner in large scale IoMT systems, while the users' experienced energy and latency cost increases exponentially. The later observation is an inherent characteristic of the IoMT systems, which can be resolved by more densely developing MEC servers to support the users' increasing computing demands.

C. Real-world Scenario of NEMESIS

In this section, we present a real-world scenario to illustrate the practical application of the NEMESIS framework in an

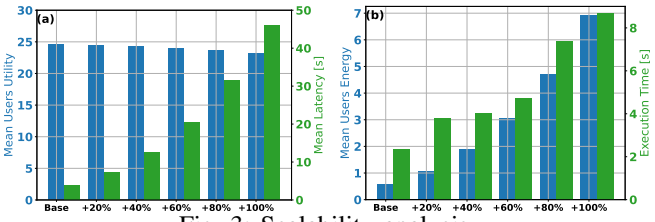

Fig. 3: Scalability analysis.

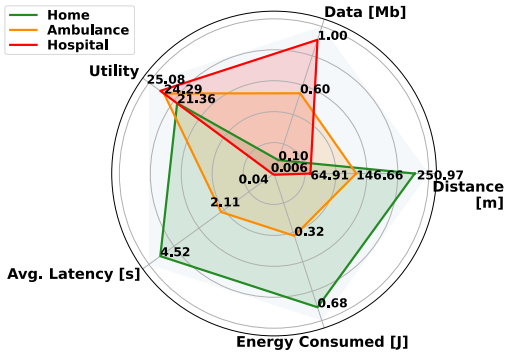

Fig. 4: Real-world scenario of implementing NEMESIS.

e-health context. The scenario involves a patient being transported from home to a hospital via ambulance. As the patient's condition becomes more critical during the journey, the volume of data generated by monitoring vital signs increases, necessitating more robust computational support. Fig. 4 depicts the patient's utility, data volume, distance from the hospital, energy cost, and latency across the three distinct stages of the journey. The results indicate that as the patient moves closer to the hospital, where the MEC server is assumed to be located, the utility improves significantly due to reductions in both latency and energy consumption. This demonstrates the NEMESIS framework's ability to adapt dynamically to emergency situations, offering optimized resource allocation and data offloading in real-time.

D. Comparative Evaluation

In this section, a comparative evaluation of the NEMESIS framework compared to three state-of-the-art scenarios is performed. The comparative scenarios are described as follows: (i) Uniform: Data is offloaded equally across all the servers. (ii) Proportional to Computing Capacity (PtCC): Data is offloaded to MEC servers in proportion to their computing capacity; and (iii) Proportional to Reliability (PtR): Data is distributed among MEC servers proportionally to their reliability levels. The results show that the NEMESIS framework has superior performance compared to these alternatives by achieving lower average latency and reduced total energy cost. This superior performance is achieved due to the NEMESIS's ability to dynamically account for both the reliability and computing capacity of each MEC server, rather than relying on static or simplified criteria.

VI. Conclusion

In conclusion, this paper introduces the NEMESIS framework, which effectively addresses existing research gaps in optimal MEC server selection and data offloading strategies

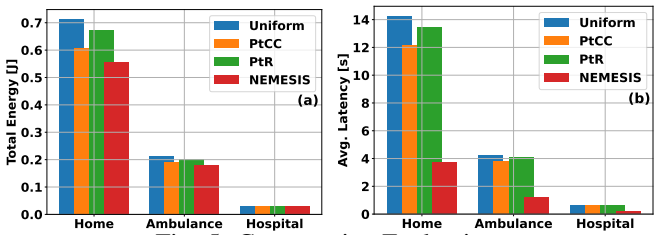

Fig. 5: Comparative Evaluation.

for e-health applications by introducing a reliability and regret learning-based solution. Part of our future work is the extension of NEMESIS to incorporate security guarantees during the users' data offloading and processing at the MEC servers.

References

[1] X. Yuan, H. Tian, W. Zhang, H. Zhao, Z. Zhao, and N. Zhang, "Capso: A combinatorial auction and improved particle swarm optimization based computation offloading approach for e-healthcare," in *IEEE Int. Conf. on Communications*, 2022, pp. 3850–3855.

[2] J. L. Sarkar, R. V, A. Majumder, B. Pati, C. R. Panigrahi, W. Wang, N. M. F. Qureshi, C. Su, and K. Dev, "I-health: Sdn-based fog architecture for iiot applications in healthcare," *IEEE/ACM Trans. on Computational Biology and Bioinf.*, vol. 21, no. 4, pp. 644–651, 2024.

[3] A. AlZailaa, H. R. Chi, A. Radwan, and R. Aguiar, "Low-latency task classification and scheduling in fog/cloud based critical e-health applications," in *IEEE Int. Conf. on Communications*, 2021, pp. 1–6.

[4] M. Aazam, S. Zeadally, and E. F. Flushing, "Task offloading in edge computing for machine learning-based smart healthcare," *Computer networks*, vol. 191, p. 108019, 2021.

[5] Y. Wang, H. Wu, R. H. Jhaveri, and Y. Djenouri, "Drl-based urllc-constraint and energy-efficient task offloading for internet of health things," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 6, pp. 3305–3316, 2024.

[6] V. Amini, M. Momtazpour, and M. Saheb Zamani, "An energy-efficient and accuracy-aware edge computing framework for heart arrhythmia detection: A joint model selection and task offloading approach," *The Journal of Supercomputing*, vol. 79, no. 8, pp. 8178–8204, 2023.

[7] Z. Xue, P. Zhou, Z. Xu, X. Wang, Y. Xie, X. Ding, and S. Wen, "A resource-constrained and privacy-preserving edge-computing-enabled clinical decision system: A federated reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9122–9138, 2021.

[8] Z. Sharif, L. T. Jung, M. Ayaz, M. Yahya, and S. Pitafi, "Priority-based task scheduling and resource allocation in edge computing for health monitoring system," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 2, pp. 544–559, 2023.

[9] Z. Askari, J. Abouei, M. Jaseemuddin, A. Anpalagan, and K. N. Plataniotis, "A q-learning approach for real-time noma scheduling of medical data in uav-aided wbans," *IEEE Access*, vol. 10, pp. 115 074–115 091, 2022.

[10] L. Zhang, X. Yuan, J. Luo, C. Feng, G. Yang, and N. Zhang, "An adaptive resource allocation approach based on user demand forecasting for e-healthcare systems," in *IEEE ICC Workshops*, 2022, pp. 349–354.

[11] K. Peng, P. Liu, M. Bilal, X. Xu, and E. Prezioso, "Mobility and privacy-aware offloading of ar applications for healthcare cyber-physical systems in edge computing," *IEEE Trans. on Network Science and Engineering*, vol. 10, no. 5, pp. 2662–2673, 2023.

[12] Z. Ling, F. Hu, H. Zhang, Z. Han, and H. V. Poor, "Distributionally robust optimization for peak age of information minimization in e-health iot," in *IEEE Int. Conf. on Communications*, 2021, pp. 1–6.

[13] 3GPP, "5G; Study on channel model for frequencies from 0.5 to 100 GHz, TR 38.901 (version 16.1.0) Release 16," 2020.

[14] T. Liu, S. Ni, X. Li, Y. Zhu, L. Kong, and Y. Yang, "Deep reinforcement learning based approach for online service placement and computation resource allocation in edge computing," *IEEE Trans. on Mobile Computing*, vol. 22, no. 7, pp. 3870–3881, 2023.

[15] Q. Li, X. Ma, A. Zhou, X. Luo, F. Yang, and S. Wang, "User-oriented edge node grouping in mobile edge computing," *IEEE Trans. on Mobile Computing*, vol. 22, no. 6, pp. 3691–3705, 2023.

[16] M. Benaïm, "Dynamics of stochastic approximation algorithms," in *Seminaire de probabilites XXXIII*. Springer, 2006, pp. 1–68.