ABSTRACT

The advances of multi-access edge computing (MEC) have paved the way for the integration of the MEC servers, as intelligent entities into the Internet of Things (IoT) environment as well as into the 5G radio access networks. In this paper, a novel artificial intelligence-based MEC servers’ activation mechanism is proposed, by adopting the principles of Reinforcement Learning (RL) and Bayesian Reasoning. The considered problem enables the MEC servers’ activation decision-making, aiming at enhancing the reputation of the overall MEC system, as well as considering the total computing costs to serve efficiently the users’ computing demands, guaranteeing at the same time their Quality of Experience (QoE) prerequisites satisfaction. Each MEC server decides in an autonomous manner whether it will be activated or remain in sleep mode by utilizing the theory of Bayesian Learning Automata (BLA). A human-driven peer-review-based evaluation of the edge computing system’s provided services is also introduced based on the concept of Bayesian Truth Serum (BTS), which supports the development of a reputation mechanism regarding the MEC servers’ provided services. The intelligent MEC servers’ autonomous decisions’ satisfaction is captured via a holistic utility function, which they aim to maximize in a distributed manner. Finally, detailed numerical results obtained via modeling and simulation, highlight the key operation features and superiority of the proposed framework.

INDEX TERMS

Multi-access edge computing (MEC), artificial intelligence, decision making, Bayesian Truth Serum, Bayesian Learning Automata, Bayesian Belief.

I. INTRODUCTION

With the advent of Internet of Things (IoT) and the 5G networks, the number of mobile devices, such as smartphones, wearable devices, tablet computers, and others, has dramatically increased, resulting in 8.2 billion mobile subscriptions in 2020 [1]. Also, the mobile applications running on the mobile devices, such as social networking and video-based applications, online gaming, and others, request services with various stringent delay, energy, and processing constraints. Such mobile applications are typically resource-hungry in terms of computation demand and energy consumption, thus, the mobile devices cannot support them locally and they migrate them to other available computing resources [2]. The current end-to-end computing continuum consists of multi-access edge, fog, and cloud computing.

The novel concept of multi-access edge computing (MEC) brings the computing capability closer to the end users by deploying modest-size MEC servers at the edge of the radio access networks. MEC, as compared to cloud computing, can significantly improve the data offloading delay, the corresponding mobile devices’ energy consumption, as well as the security levels, given that the users’ offloading tasks travel shorter distances. A large part of the recent literature has focused on the problem of users’ optimal data offloading (or equivalently tasks offloading) by proposing either centralized approaches exploiting the softwarization of the data offloading decision-making process [3] or distributed approaches based on optimization [4], game theoretic [5], or learning methods [6] to support the users’ autonomous decision-making.

However, limited research efforts have been devoted to the problem of activating the sufficient number of MEC servers, either from the users’ perspective, i.e., satisfy the users’ Quality of Service (QoS) and Quality of Experience...
In this section, we will give a brief overview of some related research works in regards to MEC servers’ management in terms of their activation and allocation of their computing capability to the end users. The problem of managing the computing capabilities of the servers has attracted the researchers’ interest early enough in the literature. In 2007, the problem of noninstantaneous server activation (i.e., the non-zero time to replicate and activate a server) is studied in [8]. The authors introduce a dynamic approach in order to share the servers’ resources among multiple users, while considering the hysteresis control to reduce the servers’ activation and deactivation cost when momentary fluctuations of the workload occur. In [9], the authors introduce a two-time-scale approach, where the servers’ activation decision occurs in slow time scale, while the servers’ computing capacity allocation to the users’ offloaded tasks is made at a faster time scale based on the servers’ power scaling criterion. The introduced two-time-scale joint job scheduling and servers’ management problem is addressed via stochastic optimization. An energy saving-driven approach is proposed in [10], where the authors formulated a minimization problem of the MEC servers’ total energy consumption under the constraints of users’ QoS prerequisites and mobility patterns. The solution of this problem concluded to the selection of a set of MEC servers to be activated based on the users’ offloading tasks request profiles.

In [11], a three-layer computing architecture is introduced consisting of the user, edge computing, and the cloud computing layers. Focusing on the edge computing plane, the authors propose a hierarchical structure of the geo-distributed activated MEC servers to aggregate the users’ offloaded tasks, efficiently exploit the MEC servers’ resources and manage the workloads during peak hours. The latter challenge is also addressed in [12], where virtualized network functions are strategically allocated to the available servers in order to minimize the energy consumption of the computing and networking infrastructure, as well as improve the protection level against resource demand uncertainty. In [3], a software defined networking approach is introduced to jointly manage the users’ tasks offloading and the MEC servers’ resources’ exploitation based on a game theoretic and reinforcement learning approach, respectively. A similar approach is proposed in [13], by adopting a deep reinforcement learning model, while considering that the users may be reluctant in revealing personal information about the MEC servers’ selection preferences and the computing demands of their offloading tasks.

A centralized approach for the MEC servers’ management problem is introduced in [14], where a regional orchestrator manages the servers’ activation and distribution of workloads towards guaranteeing the users’ QoS prerequisites via formulating the problem as a stochastic overlapping coalition formation game. Focusing on the energy efficient operation of the MEC servers, the authors in [15] use a Lyapunov optimization approach to determine the MEC servers’ energy harvesting policy from the environment, jointly with the task offloading scheduling. Following the philosophy of MEC servers’ energy efficient operation, a dynamic power management scheme is introduced in [16] towards deciding when and for how long each MEC server should be in sleep mode in order to minimize the system’s energy consumption without compromising the users’ QoS prerequisites satisfaction. A scalability analysis of the MEC system is discussed in [17] for increasing number of users and offloading tasks. The authors adopt a Particle Swarm Optimization method in order to decide on the number of activated MEC servers, while lowering the users’ serving delay and maximizing the MEC system’s cost effectiveness.

A minority game theoretic approach is introduced in [18], [19] via considering the total number of MEC servers that should be activated to serve the users and allowing the MEC servers to autonomously decide their activation in order to respect the aforementioned constraint by following the theory of minority games. This work has been further extended in [20] via considering the problem of users’ association to the MEC servers and introducing a distributed reinforcement learning-based decision making process.

Despite the efforts made in the previous works, in regards to the MEC servers’ management, how to incorporate the end users’ personalized Quality of Experience satisfaction in the MEC servers’ activation decision making still remains to be an open issue. Moreover, to facilitate the infrastructure/service providers’ penetration in the computing market, how to capture the MEC servers’ reputation is even challenging. In this work, we strive to tackle these issues. In detail, the design goal is to capture the end users’ evaluation of the edge computing system and of the activated MEC servers that process their offloaded computing tasks in terms of satisfying their QoE prerequisites, create a reputation model for the MEC servers, and devise a distributed and autonomous MEC servers’ activation mechanism based on the principles of artificial intelligence. The main contributions of this work that differentiate it from the rest of the literature, are summarized below:

1) A human-driven evaluation of the edge computing system (i.e., activated MEC servers) is introduced in regards to the services that they provide to the end users, based on the theory of Bayesian Truth Serum. The proposed
2) A reputation scheme is proposed based on a novel Bayesian model to quantify the reputation of each activated MEC server in the computing market based on the Quality of Service that they provide to the end users and based on the latter ones subjective QoE satisfaction.

3) An artificial intelligence-based MEC servers’ activation mechanism is devised based on the theory of Bayesian Learning Automata. The MEC servers decide their activation or not in a distributed and autonomous manner towards improving their reputation in the overall edge computing system, while considering their cost to serve the users computing demands. A distributed and low-complexity algorithm is also designed that converges to the Bayesian Nash equilibrium, which is a stable operation point for the whole edge computing system.

4) A series of experiments are performed to evaluate the performance of the overall MEC servers’ management scheme, in terms of the extraction of the objective evaluation of the MEC servers’ provided computing services based on the users’ subjective data, the reputation scheme for the MEC servers, and their autonomous decision making regarding their activation or not. Also, a detailed comparative evaluation with alternative MEC servers’ management schemes demonstrates our proposed framework’s superiority and benefits.

The rest of the paper is organized as follows. The system model is introduced in Section II, while the users’ evaluation mechanism of the MEC servers is discussed in Section III. Section IV introduces the MEC servers’ reputation model and the artificial intelligence-based MEC servers’ activation is analyzed in Section V. Simulation results are investigated in Section VI, while Section VII concludes the paper.

II. SYSTEM MODEL

A heterogeneous MEC environment is considered consisting of $|S|$ MEC servers and $|U|$ users, where their sets are denoted as $S = \{1, \ldots, s, \ldots, |S|\}$ and $U = \{1, \ldots, u, \ldots, |U|\}$, respectively. At each time slot $t$, each MEC server $s \in S$ has a corresponding reputation $\mu_s(t)$ and a computation capability $F_s(t)$ [CPUcycles] that is allocated to the users’ offloaded computing tasks in order to process them. The MEC server’s reputation is gained based on its capability to serve the users and satisfy their Quality of Experience prerequisites. More information regarding the MEC servers’ reputation scheme is provided in Section IV.

Each user $u \in U$ offloads a task $T_u(t) = (I_u(t), C_u(t), \phi_u(t))$ to the set of activated MEC servers, where $I_u(t)$ [bits] denotes the computation total input bits and $C_u(t)$ [CPUcycles] is the total number of CPU cycles required to accomplish the computation task $T_u(t)$. The parameter $\phi_u(t)$ [CPUcycles/bits] expresses the computation complexity of the task requested by the user, and its value depends on the nature of the application, i.e., a larger value of $\phi_u(t)$ expresses a more computationally intensive task. The problem of users’ association to the available MEC servers is not addressed in this paper, while a similar approach is adopted from our previous works [21], [22] considering the users’ computation tasks and personal characteristics in order to select a MEC server. Thus, in the rest of the analysis, we consider that each user’s tasks are offloaded to the edge computing system and distributed among all the activated MEC servers following an intelligent software-defined orchestration mechanism [23], [24].
Eventually, each MEC server $s \in S$ decides in an autonomous and distributed manner whether it will offer its computation capability to the users’ tasks execution or not, i.e., $a^{(t)}_s \in \{0, 1\}$, where $a^{(t)}_s$ denotes the final decision of MEC server $s$ at time slot $t$. For $a^{(t)}_s = 1$, the MEC server $s$ is activated in time slot $t$, while for $a^{(t)}_s = 0$, the MEC server remains in sleep mode in time slot $t$. We consider $S^{(t)} = \{s \in S : a^{(t)}_s = 1\}$ to be the set of the activated MEC servers at a specific time slot $t$. The adopted system model, as well as the overall architecture of the proposed artificial intelligent multi-access edge computing servers management system is presented in Fig. 1.

III. HUMAN-DRIVEN COMPUTING SYSTEM EVALUATION

In this section, a human-driven peer-review-based evaluation of the edge computing system and the corresponding MEC servers is introduced in regards to the services that they provide to the end users, based on the theory of Bayesian Truth Serum. At the end of each time slot $t$, each user $u \in U$ evaluates the perceived QoE from the activated MEC servers, which are all responsible for the execution of its task $T^{(t)}_u$ at time slot $t$, in a distributed manner. In order to assess how truthful were the aforementioned users’ evaluations, we adopt the concept of Bayesian Truth Serum (BTS), which allows us to elicit an overall objective evaluation from subjective data when the ground truth is unknown and being at the same time strict Bayes-Nash incentive compatible for $|U| \to \infty$ [25]. Specifically, at time slot $t$, user $u$ answers a binary question, i.e., “Are you satisfied from the activated MEC servers’ $S^{(t)}$ provided service?” by providing the following two evaluation reports:

- The information report $x^{(t, S^{(t)})}_u = (x^{(t, S^{(t)})}_{u,0}, x^{(t, S^{(t)})}_{u,1})$, which denotes the personal answer of user $u$, where
  \[ \sum_{i \in \{0,1\}} x^{(t, S^{(t)})}_{u,i} = 1, \]  
  i.e., “YES” ($i = 1$) or “NO” ($i = 0$) with $x^{(t, S^{(t)})}_{u,i} \in \{0, 1\}, \forall i \in \{0, 1\}$.

- The prediction report $y^{(t, S^{(t)})}_u = (y^{(t, S^{(t)})}_{u,0}, y^{(t, S^{(t)})}_{u,1})$, where $y^{(t, S^{(t)})}_{u,1}$ denotes the prediction regarding the fraction of the users whose answer is $x^{(t, S^{(t)})}_{u,1} = 1$, i.e., “YES”, and $y^{(t, S^{(t)})}_{u,0}$ is the prediction for the fraction of the users whose answer is $x^{(t, S^{(t)})}_{u,0} = 1$, i.e., “NO”, and
  \[ \sum_{i \in \{0,1\}} y^{(t, S^{(t)})}_{u,i} = 1. \]

Based on the users’ information and prediction reports, the population endorsement frequencies $\bar{y}^{(t, S^{(t)})}_i$ (Eq. 1) and the geometric mean of the users’ population predictions $\hat{y}^{(t, S^{(t)})}_i$ (Eq. 2) for each of the answers $i, i \in \{0, 1\}$ are calculated as follows:

\[ \bar{y}^{(t, S^{(t)})}_i = \frac{1}{|U|} \sum_{u=1}^{|U|} x^{(t, S^{(t)})}_{u,i} \]  
(1)

and

\[ \log(y^{(t, S^{(t)})}_{i,u}) = \frac{1}{|U|} \sum_{u=1}^{|U|} \log(y^{(t, S^{(t)})}_{i,u}) \]  
(2)

Consequently, the BTS score $s_{BTS,u}^{(t)}(x^{(t, S^{(t)})}_u, y^{(t, S^{(t)})}_u)$ of each user $u$, which depicts how truthful its personal answer was, is calculated based on the formula:

\[ s_{BTS,u}^{(t)}(x^{(t, S^{(t)})}_u, y^{(t, S^{(t)})}_u) = \sum_{i \in \{0,1\}} x^{(t, S^{(t)})}_{u,i} \cdot \log(\frac{\bar{y}^{(t, S^{(t)})}_i}{\hat{y}^{(t, S^{(t)})}_i}) \]  
(3)

where the first part of Eq. 3 is the information score and the second part is the prediction score of the user. The information score increases if an answer $i, i \in \{0, 1\}$ is surprisingly common, i.e., if the mean endorsement population frequency $\bar{y}^{(t, S^{(t)})}_i$ is higher than the corresponding mean prediction $\hat{y}^{(t, S^{(t)})}_i$. The surprisingly common criterion is based on the Bayesian reasoning principle, which states that a user believes that the rest of the population will underestimate its personal opinion, reporting thus a higher prediction for that answer in order to further support it.

As a result, according to the Bayesian argument [26], the users that are truthful regarding their personal information reports $x^{(t, S^{(t)})}_u$, they should also provide higher predictions, leading thus to the conclusion that a truthful user’s opinion is more likely to be surprisingly common. Moreover, it should be noted that the prediction score (second term of Eq. 3) acts as a penalty proportional to the Kullback-Leibler divergence [27] between the actual population endorsement frequencies of the answers $\bar{y}^{(t, S^{(t)})}_i$ and the respective user’s population predictions’ geometric means $\hat{y}^{(t, S^{(t)})}_i$. The physical meaning of this observation is that the optimal prediction score is achieved when the user’s prediction is equal to the actual mean frequency of the answer $i$ (absolute accuracy), i.e., $y^{(t, S^{(t)})}_{i,u} = \bar{y}^{(t, S^{(t)})}_i$. In this case, the prediction score is equal to $0$ since the user will experience a zero penalty.

The parameter $\alpha, \alpha > 0$ in Eq. 3 controls the effect of the prediction score (penalty) in the total BTS score, i.e., the higher is the value of parameter $\alpha$, the higher is the contribution of the prediction score in the calculation of the BTS score and the lower will be the overall BTS score. The physical meaning of the parameter $\alpha$ is that it fine-tunes the importance of the prediction error, since a potential minority of the users who may be more distant from the activated MEC servers will experience a worse QoE satisfaction due to the increased latency than the majority of the users which is closer, thus, satisfied. As a consequence, the prediction of the unsatisfied users may not be representative with respect to the overall service provided by the activated MEC servers and thus, they should not be strictly penalized because of the
inaccurate prediction. As a next step, the average BTS score \( \overline{u}_i^{(t)}(S^{(t)}) \) for each answer \( i, i \in \{0, 1\} \) for \( \overline{x}_i \neq 0 \), is calculated as follows, while \( \overline{u}_i^{(t)}(S^{(t)}) = 0 \), if \( \overline{x}_i = 0 \).

\[
\overline{u}_i^{(t)}(S^{(t)}) = \frac{1}{|U|} \sum_{i=1}^{[U]} x_{u,i}^{(t)} \cdot S_{BTS,i}^{(t)} \cdot (x_u(t,S^{(t)}), y_u(t,S^{(t)}))
\]

(4)

Finally, the most truthful answer \( x_{BTS}^{(t)} \) regarding the activated MEC servers’ provided services at time slot \( t \) is the answer that has the highest average BTS score, i.e.:

\[
x_{BTS}^{(t)} = \arg \max_{i \in [0,1]} \overline{u}_i^{(t)}(S^{(t)})
\]

(5)

Since all the activated MEC servers contribute to the users’ tasks execution in a distributed manner, the outcome of Eq. 5 determines if the experienced service provided by \( S^{(t)} \) activated MEC servers, satisfied \( x_{BTS}^{(t)} = 1 \) or not \( x_{BTS}^{(t)} = 0 \) the users’ QoE prerequisites, based on the corresponding users’ subjective evaluation.

IV. SERVERS REPUTATION SCHEME

In this section, we exploit the human-driven evaluation of the edge computing system, as presented in the previous section, in order to create a reputation scheme for the MEC servers, capturing their capability to serve the users and satisfy their QoE constraints. Towards capturing the MEC servers capability to serve the users and the corresponding penetration of the infrastructure/service provider, who owns the MEC servers, into the computing market, we introduce the reputation \( \mu_s^{(t)} \) for each activated MEC server \( s \) at time slot \( t \). A Bayesian model is devised, which features adverse selection based on the Bayesian updating of belief of the users [28].

Specifically, all users share the same prior belief distribution \( \mu_0^{(t)} = \mu_0, \forall s \in S, \forall t \), regarding the QoE satisfaction that each MEC server \( s \) can provide to them. Each MEC server \( s \) can either offer a high or low QoE satisfaction, with probabilities \( a_H \) and \( a_L \), respectively, where \( 0 < a_L < a_H < 1 \). At every time slot \( t \), the activated MEC servers that offer their computation services to the users, are evaluated by the BTS mechanism (Section III), i.e., if the users are overall satisfied \( x_{BTS}^{(t)} = 1 \) and if not \( x_{BTS}^{(t)} = 0 \) by the activated MEC servers set \( S^{(t)} \), creating thus a history for each MEC server \( s \). We denote as \( \mathcal{Q}_s^{(t)} \), the number of times until time slot \( t \) the MEC server \( s \) offered its services to the users and it was evaluated as helpful \( x_{BTS}^{(t)} = 1 \), and as \( \mathcal{F}_s^{(t)} \) we denote the number of times that the MEC server’s service was evaluated as not helpful \( x_{BTS}^{(t)} = 0 \). Thus, each MEC server’s \( s \) posterior distribution of reputation is given as follows [28].

\[
\mu_s^{(t)} = \frac{\mu_0 \cdot a_H^{(t)} \cdot (1-a_H)^{(t)} \cdot \mathcal{Q}_s^{(t)}}{\mu_0 \cdot a_H^{(t)} \cdot (1-a_H)^{(t)} + (1-\mu_0) \cdot a_L^{(t)} \cdot (1-a_L)^{(t)} \cdot \mathcal{F}_s^{(t)}}
\]

(6)

Based on Eq.6, we observe that there is a correlation between the MEC server’s \( s \) history of the users’ positive and negative evaluations, i.e., the \( \mathcal{Q}_s^{(t)} \) and \( \mathcal{F}_s^{(t)} \), respectively, and its corresponding reputation. Specifically, the MEC server’s reputation \( \mu_s^{(t)} \) increases with respect to the positive received evaluations, i.e., \( \mathcal{Q}_s^{(t)} \), and decreases with respect to the negative evaluations, i.e., \( \mathcal{F}_s^{(t)} \).

A. MEC SERVERS UTILITY FUNCTION

Following the presented reputation scheme for the activated MEC servers at each time slot \( t \), in this section we formulate each MEC server’s utility function, which captures the benefit of the MEC server by participating or not in the computing system and offering its computing capability to the users. Each MEC server \( s \) acts as an artificial intelligent agent making decisions in an autonomous and distributed manner, thus determining whether it should provide its computation capability \( F_s^{(t)} \), i.e., \( a_s^{(t)} = 1 \) (the server is activated), or not, i.e., \( a_s^{(t)} = 0 \) (the server remains in sleep mode), for the users’ data processing at time slot \( t \). A holistic utility function is introduced for each MEC server to capture its perceived benefit in terms of rewards by processing the users’ offloaded data (users positive or negative provided evaluations). The MEC server’s \( s \), \( s \in S \), utility function is formulated as follows.

\[
u_u^{(t)}(a_s^{(t)}, a_{s-s}^{(t)}) = \sum_{s \in S} \frac{\mu_s^{(t)} \cdot F_s^{(t)} \cdot a_s^{(t)} - \sum_{s \neq s} \mu_s^{(t)} \cdot F_s^{(t)} \cdot a_{s-s}^{(t)}}{\sum_{s \in S} \mu_s^{(t)} \cdot F_s^{(t)} \cdot a_s^{(t)}}
\]

(7)

where \( R_u^{(t)} \) denotes the overall reward which is offered by the user \( u \in U \) at time slot \( t \) for the computational services of the activated MEC servers, i.e., \( \forall s \in S \), where \( a_s^{(t)} = 1 \). In case that \( a_s^{(t)} = 1 \), i.e., the MEC server \( s \) is activated at time slot \( t \), the first part of Eq.7 expresses the portion of the total reward \( \sum_{s \in U} R_u^{(t)} \) that MEC server \( s \) will receive, with respect to its reputation \( \mu_s^{(t)} \) and the provided computation power \( F_s^{(t)} \). Specifically, the higher are the reputation \( \mu_s^{(t)} \) and the computation capability \( F_s^{(t)} \) compared to the other activated servers \( s' \), \( s'' \in S^{(t)} \), the larger will be the portion of the reward that \( s \) will receive by the users. The second term of the utility function depicts the operating computing costs of the activated MEC server \( s \) in order to process parts of all the users’ tasks. In case that \( a_s^{(t)} = 0 \), i.e., the MEC server remains in sleep mode, then the experienced utility is 0, as the MEC server \( s \) will not receive reward from the users and it will not spend its computational resources. The introduced utility function drives the MEC servers to select in an autonomous and distributed manner whether they will serve the users or not at a certain time slot \( t \), by evaluating the potential economic gains over the operating costs.

In order to formulate the overall reward \( R_u^{(t)}, \forall u \in U \), that is provided by each user to the activated MEC servers, we adopt the concept of the economic sale price [28] based on the reputation \( \mu_s^{(t)} \), as formulated in Eq. 6. Specifically,
in order to depict in a holistic way the efficiency of the overall MEC environment in terms of serving the users’ computation demands, we consider the average Bayesian reputation of the activated MEC servers \(s, s \in S^{(t)}\), which offer their computational services to the users at time slot \(t\), as follows.

\[
\overline{\mu}^{(t)} = \frac{|S|}{\sum_{s=1}^{S} \mu_{s}^{(t)} \cdot a_{s}^{(t)}}
\]

(8)

It is noted that if no MEC server is activated, we have \(\overline{\mu}^{(t)} = 0\). Thus, based on the average holistic bayesian belief \(\overline{\mu}^{(t)}\), the users reward the activated MEC servers with the following reward:

\[
R_{a}^{(t)} = \overline{\mu}^{(t)} \cdot a_{H} + (1 - \overline{\mu}^{(t)}) \cdot a_{L}
\]

(9)

It is observed that the reward \(R_{a}^{(t)}\) is an increasing function with respect to the average posterior belief \(\overline{\mu}^{(t)}\), i.e., the better is the overall posterior belief of the users regarding the MEC servers provided QoE satisfaction to the users, the higher is the reward offered by the users.

**Algorithm 1 BLA Rewards & Penalties**

1. **Input/Initialization:** \(S, a_{s}^{(ite,t)} \max_{ite\in U} u_{s}^{(ite-1,t)}\)
2. **Output:** Reward or Penalty \(E_{a_{s}}^{(ite,t)}\)
3. for \(s = 1 \text{ to } |S| \) do
4. **MEC Server** \(s\) calculates its utility \(u_{s}^{(ite,t)}\) \((a_{s}^{(ite,t)}, a_{\sim s}^{(ite,t)})\)
5. if \(u_{s}^{(ite,t)}(a_{s}^{(ite,t)}, a_{\sim s}^{(ite,t)}) \geq \max_{ite} u_{s}^{(ite-1,t)})\) then
6. Decision \(a_{s}^{(ite,t)}\) leads to reward \(E_{a_{s}}^{(ite,t)} = 1\)
7. Update the maximum experienced utility: \(\max_{ite} u_{s}^{(ite,t)} = u_{s}^{(a_{s}^{(ite,t)}, a_{\sim s}^{(ite,t)})}\)
8. else
9. Decision \(a_{s}^{(ite,t)}\) leads to penalty \(E_{a_{s}}^{(ite,t)} = 0\)
10. end if
11. end for

**V. ARTIFICIAL INTELLIGENT SERVERS ACTIVATION**

Towards realizing an autonomous and distributed decision-making mechanism, we propose a Learning Automata-based learning algorithm, where each MEC server \(s\) operates as a Bayesian Learning Automaton (BLA) [29] in order to determine whether it should offer its computation capability \(F_{s}^{(t)}\) for the users’ offloaded data execution or not at a certain time slot \(t\). The goal of each MEC server is to maximize its utility function as presented in Eq.6. Fig.1 presents the overall procedure of the Bayesian Learning Automata, where the learning process unfolds throughout the iterations at time slot \(t\) and in every iteration \(ite\), each BLA (i.e., MEC server \(s\)), selects an action \(a_{s}^{(ite,t)} \in \{0, 1\}\), which is afterwards evaluated as a good \((E_{a_{s}}^{(ite,t)} = 1)\) or a bad \((E_{a_{s}}^{(ite,t)} = 0)\) decision by the MEC environment.

According to the feedback that the server \(s\) receives from the MEC environment at the \(ite\) iteration, it develops an intelligence, i.e., an action probability distribution, based on which it can make better decisions in the future, i.e., in the upcoming iterations during time slot \(t\), with respect to the perceived environment’s reward or penalty and the experienced utility. Eventually, each MEC server \(s\) aims at maximizing the probability of making decisions \(a_{s}^{(ite,t)}\) that will lead it to receive higher rewards during the learning process. In the following subsections, we present the distributed algorithms regarding the processes of the MEC environment determining whether to reward or penalize the MEC servers for their decisions and the Bayesian Learning Automata decision making process. It is noted that the traditional Learning Automata schemes need to have a predefined (constant) value of the learning parameter, which controls the trade-off between the learning speed and the learning accuracy. However, this assumption is unrealistic within the examined dynamic MEC environment. Thus, by adopting the Bayesian Learning Automata scheme, which does not require a constant learning parameter, we enable our framework to be dynamic and adaptable during the learning process.

**A. BAYESIAN LEARNING AUTOMATA REWARD AND PENALTY FORMULATION**

In this section, we present the process of determining if a MEC server \(s\), \(s \in S\), i.e., a Bayesian Learning Automaton, should be rewarded or penalized for its selected action \(a_{s}^{(ite,t)}\). The MEC server \(s\) must be rewarded or penalized depending on the utility \(u_{s}^{(ite,t)}\) (Eq.6) that it experiences in an iteration \(ite\). Specifically, if the MEC server decides to participate \((a_{s}^{(ite,t)} = 1)\) in the users’ offloaded data execution, but it provides a low computation capability \(F_{s}^{(t)}\) and has a low reputation \(\mu_{s}^{(t)}\) compared to the other participating MEC servers, then it will probabilistically receive a small portion of the total offered reward \(\sum_{u \in U} R_{a_{s}}^{(ite,t)} = \sum_{u \in U} (\overline{\mu}^{(t)} \cdot a_{H} + (1 - \overline{\mu}^{(t)}) \cdot a_{L})\) resulting in a negative utility \(u_{s}^{(ite,t)}\) and in a penalty, i.e., \(E_{a_{s}}^{(ite,t)} = 0\). Therefore, if in a previous iteration \(ite\), the MEC server \(s\) had selected to remain in sleep mode \((a_{s}^{(ite,t)} = 0)\), it would have experienced a higher utility, i.e., \(u_{s}^{(ite,t)} = 0\). For that reason, we store the maximum experienced utility \(\max_{ite} u_{s}^{(ite,t)}\) for every MEC server \(s\) and
B. BAYESIAN LEARNING AUTOMATA DECISION MAKING

The BLA-based learning method, which operates under the philosophy of the Bayesian reasoning framework [30], is computationally efficient, since it is independent of any learning parameter (in contrast to other Learning Automata-based schemes). Thus, it is self-adaptive to dynamic changes of the MEC environment. Furthermore, it follows an action Beta probability distribution instead of storing an action probability vector, which would increase the memory and computational need of the corresponding algorithm [31]. Every MEC server’s ability vector, which would increase the memory and computational complexity, for a specific time slot is increased. Thus, the cumulative Beta distribution function of the MEC environment. Furthermore, it follows an action Beta probability distribution instead of storing an action probability vector, which would increase the memory and computational need of the corresponding algorithm [31]. Every MEC server’s ability vector, which would increase the memory and computational complexity, for a specific time slot is increased. Thus, the cumulative Beta distribution function

The complexity of the BLA-based learning method, which operates under the philosophy of the Bayesian reasoning framework [30], is computationally efficient, since it is independent of any learning parameter (in contrast to other Learning Automata-based schemes). Thus, it is self-adaptive to dynamic changes of the MEC environment. Furthermore, it follows an action Beta probability distribution instead of storing an action probability vector, which would increase the memory and computational need of the corresponding algorithm [31]. Every MEC server’s ability vector, which would increase the memory and computational complexity, for a specific time slot is increased. Thus, the cumulative Beta distribution function of the MEC environment. Furthermore, it follows an action Beta probability distribution instead of storing an action probability vector, which would increase the memory and computational need of the corresponding algorithm [31]. Every MEC server’s ability vector, which would increase the memory and computational complexity, for a specific time slot is increased. Thus, the cumulative Beta distribution function

BLA-Based Decision Making

1. Input/Initialization: \( \text{ite} = 0 \), convergence = false, \( a_{s,1} = \beta_{s,1} = a_{s,0} = \beta_{s,0} = 1 \ \forall s, s \in S \)
2. Output: Final Decision made by the MEC servers \( a_s^{(i)} \), Final Values of the hyper-parameters \( a_s \) and \( \beta_s \) \( \forall i, i \in \{0, 1\} \)
3. while (convergence = false) do
   4. \( \text{ite} = \text{ite} + 1 \)
   5. for \( s = 1 \) to \( |S| \) do
      6. Pick \( x_{s,0} \) and \( x_{s,1} \) from
      7. if \( (x_{s,0} \leq x_{s,1}) \) then
         8. Participate: \( a_{s}^{(\text{ite},1)} = 1 \)
         9. else
         10. Don’t Participate: \( a_{s}^{(\text{ite},0)} = 0 \)
      11. end if
   12. end for
   13. Based on the \( a_{s}^{(\text{ite},i)} \) calculate the respective Rewards and Penalties by executing Algorithm 1.
   14. for \( s = 1 \) to \( |S| \) do
      15. if \( (E_{a_s^{(\text{ite},i)}} = 1) \) then
         16. if \( (a_{s}^{(\text{ite},i)} = 0) \) then
            17. \( a_{s,0} = a_{s,0} + 1 \)
            18. else
               19. \( a_{s,1} = a_{s,1} + 1 \)
            20. end if
         21. else
            22. if \( (a_{s}^{(\text{ite},i)} = 0) \) then
               23. \( \beta_{s,0} = \beta_{s,0} + 1 \)
               24. else
                  25. \( \beta_{s,1} = \beta_{s,1} + 1 \)
               26. end if
            27. end if
         28. end if
      29. end while

VI. NUMERICAL RESULTS

In this section, a detailed numerical evaluation is provided in terms of the overall proposed framework’s operation effectiveness, its scalability, and its efficiency compared to other alternatives. Specifically, the extraction of the users’ evaluation through the Bayesian Truth Serum approach is presented and compared to an alternative approach, named Robust Bayesian Truth Serum, to show the drawbacks and benefits of the proposed human-driven computing system evaluation (Section VI-A). Then, the operation of the Bayesian Learning...
Automata (BLA) and the reputation system is demonstrated (Section VI-B), and a detailed comparative evaluation to other approaches is presented (Section VI-C).

The proposed framework’s evaluation was conducted in a MacBook Pro Laptop, 2.5GHz Intel Core i7, with 16GB LPDDR3 available RAM. We consider \( |\mathcal{S}| = 20 \) MEC servers, \( K = 3 \), \( T = 100 \) time slots, \( \mu_0 = 0.2 \), \( a_H = 0.51 \), \( a_L = 0.49 \), and \( F_s^{(y)} \in [10 \cdot 10^9, 12 \cdot 10^9] \) CPU cycles, which is clipped into a smaller range of computing power units, i.e., \( F_s^{(y)} \in [10, 12] \), ensuring thus the same order of magnitude of the individual considered terms in the MEC servers’ utility function. In the following analysis, we consider a population of \( |\mathcal{U}| = 500 \) users, otherwise explicitly stated.

![FIGURE 3. Human-driven computing system evaluation.](image)

### A. EXTRACTION OF HUMAN-DRIVEN COMPUTING SYSTEM EVALUATION

In this section, the proposed human-driven computing system evaluation based on the theory of the Bayesian Truth Serum (BTS) is presented. In the following, we consider \( |\mathcal{U}| = 15 \) users for the evaluation of the BTS component at a specific time slot \( t \).

Fig. 3a presents the average population endorsement frequencies \( \bar{x}_s^{(t, S^{(y)})} \) (Eq.1) for both answers \( i \in \{0, 1\} \), i.e., the “YES” and “NO” answers. We observe that the majority of the users, i.e., 73.33% voted for “YES” since they were satisfied from the perceived QoE provided by the MEC servers at time slot \( t \). Fig. 3b shows the mean prediction \( y_{u,i}^{(t, S^{(y)})} \) (Eq. 2) of the users regarding the population proportion that voted for “YES” and for “NO”, i.e., 54.18% and 33.46%, respectively, which concludes to the outcome that “YES” is a surprisingly common answer. Based on the results presented in Fig. 3a, 3b, we conclude to the average BTS scores \( \bar{y}_{BTS}^{(t)} \) (Eq. 4) for the aforementioned answers, which are 0.2275 and \(-0.62562\), respectively. Therefore, based on Eq.5, the users that are overall satisfied from the experienced QoE are more truthful than the users with the opposite opinion (as it is shown by the higher average BTS score for the answer “YES”), thus, the final answer determined by the BTS framework (Section III) is “YES”, i.e., \( x_{RBTS}^{(t)} = 1 \).

Complementary to the above analysis, Table 1 presents all the users’ answers, predictions and individual scores, which led to the previously discussed results, presented in Fig. 3a-3c. The results reveal that the users, who provided the same answer, achieve the same information score, since the latter is only related to the users’ answers’ mean endorsement population frequencies \( \bar{x}_s^{(t, S^{(y)})} \) and the corresponding mean predictions \( y_{u,i}^{(t, S^{(y)})} \). Specifically, the information score (first part of Eq. 3) of the users that answered “YES” is 0.2566, i.e., positive, whereas the information score of the users who provided “NO” as an answer is \(-0.1755\), i.e., negative. The latter outcome is derived due to the fraction of the population that voted “YES” \( \bar{x}_s^{(t, S^{(y)})} \) (73.33%, 11 out of 15 users) is higher than the corresponding mean prediction \( y_{u,i}^{(t, S^{(y)})} \) (54.18%), which makes this answer surprisingly common, thus, a truthful answer, while on the other hand, the “NO” answer is unsurprisingly common. Moreover, we can clearly observe that the closer is a user’s \( u \) prediction \( y_{u,i}^{(t, S^{(y)})} \) to the mean of the actual personal answers \( \bar{x}_s^{(t, S^{(y)})} \), \( i \in \{0, 1\} \), the better is the prediction score, meaning that it is closer to the optimal value of 0. Thus, we observe that user 4, who answered “YES”, had the best prediction score since its

| User  | Personal Answer | YES Prediction \( y_{u,1} \) | NO Prediction \( y_{u,0} \) | Information Score \( \bar{y}_{BTS} \) | Prediction Score \( \bar{y}_{BTS} \) |
|-------|----------------|----------------|----------------|----------------|----------------|
| 1     | YES (1)        | 50%            | 50%            | 0.2566         | \(-0.1132\)     | 0.1433         |
| 2     | YES (1)        | 98%            | 2%             | 0.2566         | \(-0.4781\)     | \(-0.2215\)    |
| 3     | NO (0)         | 30%            | 70%            | \(-0.1755\)    | \(-0.3981\)     | \(-0.5736\)    |
| 4     | YES (1)        | 75%            | 25%            | 0.2566         | \(-7.3 \cdot 10\)\(-4\) | 0.2558         |
| 5     | YES (1)        | 56%            | 44%            | 0.2566         | \(-0.0642\)     | 0.1923         |
| 6     | YES (1)        | 63%            | 37%            | 0.2566         | \(-0.0240\)     | 0.2325         |
| 7     | YES (1)        | 66%            | 34%            | 0.2566         | \(-0.0125\)     | 0.2441         |
| 8     | NO (0)         | 25%            | 75%            | \(-0.1755\)    | \(-0.5134\)     | \(-0.6889\)    |
| 9     | YES (1)        | 60%            | 40%            | 0.2566         | \(-0.0390\)     | 0.2175         |
| 10    | YES (1)        | 55%            | 45%            | 0.2566         | \(-0.0714\)     | 0.1851         |
| 11    | NO (0)         | 31%            | 69%            | \(-0.1755\)    | \(-0.03779\)    | \(-0.5534\)    |
| 12    | NO (0)         | 70%            | 30%            | \(-0.1755\)    | \(-0.0027\)     | \(-0.1782\)    |
| 13    | YES (1)        | 79%            | 21%            | 0.2566         | \(-0.0091\)     | 0.2474         |
| 14    | YES (1)        | 70%            | 30%            | 0.2566         | \(-0.0027\)     | 0.2539         |
| 15    | YES (1)        | 80%            | 20%            | 0.2566         | \(-0.0129\)     | 0.2437         |
This approach is considered to be more stable for smaller populations than the BTS framework, leading to more truthful answers’ elicitation. Specifically, according to the RBTS each user provides the following two reports:

- The information report \( x_u(t, S^{(0)}) \) is the user’s personal answer and 1 denotes the answer “YES”, while 0 denotes the answer “NO”.
- The prediction report \( y_u(t, S^{(0)}) \) is the user’s prediction regarding the fraction of the users that answered “YES”, i.e., \( x_u(t, S^{(0)}) = 1 \).

Afterwards, a reference user \( r_u = u + 1 \mod |U| \) and a peer user \( p_u = u + 2 \mod |U| \) are selected for each user \( u \in U \) and the following is calculated:

\[
y_u(t, S^{(0)}) = \begin{cases} 
  y_u(t, S^{(0)}) + \delta, & \text{if } x_u(t, S^{(0)}) = 1 \\
  y_u(t, S^{(0)}) - \delta, & \text{if } x_u(t, S^{(0)}) = 0 
\end{cases}
\]

where \( \delta = \min(y_u(t, S^{(0)}), 1 - y_u(t, S^{(0)})) \). Consequently, the RBTS score of each user \( u \), which depicts how truthful its information report was, is calculated based on the formula [32]:

\[
sc_u(t)(x_u(t, S^{(0)})) = R_{\text{sps}}(y_u(t, S^{(0)}), x_u(t, S^{(0)})) + R_{\text{sps}}(y_u(t, S^{(0)}), x_u(t, S^{(0)}))
\]

Finally, the truthful answer \( x_{R_{\text{BTS}}}^{(t)}(\cdot) \) regarding the MEC servers’ service at time slot \( t \) is the answer that has the highest average RBTS score, and is determined as follows:

\[
x_{R_{\text{BTS}}}^{(t)}(\cdot) = \max_{u \in U} \sum_{u \in U} \frac{sc_u(t)(x_u(t, S^{(0)}))}{|U| - \sum_{u \in U} x_u(t, S^{(0)})}
\]

It is noted that in the presented experiment in this section, the outcome of the RBTS was identical to the one concluded by the BTS mechanism, i.e., \( x_{R_{\text{BTS}}}^{(t)} = 1 \) (i.e., YES) for fairness in the comparison of the two frameworks. Table 2 presents the users’ information, prediction, and RBTS scores, whose personal and prediction reports are the same as the aforementioned BTS instance. The results reveal that users 5 and 10 that provide the same answer, i.e., “YES”, and almost identical prediction reports, they achieve completely different RBTS scores (1.8064 and 1.3131, respectively), compared to the BTS approach where their scores are almost identical (0.1923 and 0.1851, respectively). This outcome reveals the

---

**Table 2. Robust Bayesian Truth Serum (RBTS) Outcome.**

| User u | Information Score | Prediction Score | RBTS Score X\(u_{R_{\text{BTS}}}^{(t)}\) |
|--------|-------------------|-----------------|--------------------------|
| 1      | 0                 | 0.75            | 0.75                     |
| 2      | 0.84              | 0.9996          | 1.8396                   |
| 3      | 0.75              | 0.51            | 1.26                     |
| 4      | 1                 | 0.9375          | 1.9375                   |
| 5      | 1                 | 0.8064          | 1.8064                   |
| 6      | 0                 | 0.6031          | 0.6031                   |
| 7      | 0.75              | 0.8844          | 1.6344                   |
| 8      | 0.36              | 0.4315          | 0.7975                   |
| 9      | 0                 | 0               | 0                        |
| 10     | 0.6156            | 0.6975          | 1.3131                   |
| 11     | 0.64              | 0.5239          | 1.1639                   |
| 12     | 0.8236            | 0.91            | 1.7336                   |
| 13     | 1                 | 0.9559          | 1.9559                   |
| 14     | 1                 | 0.91            | 1.91                     |
| 15     | 1                 | 0.96            | 1.96                     |

Another interesting observation is that the proposed BTS framework penalizes the extreme users’ beliefs. Specifically, we observe that user 2, who answered “YES”, is extremely biased since it predicted that 98% of the population will provide “YES” as a personal report and only the rest 2% will answer “NO”. However, because of that extreme prediction report, the user received the worst prediction score (−0.4781), since it did not approach the actual mean endorsement population frequencies and its overall BTS score was negative (0.2215), thus being ranked in the 12th place of all the BTS scores in the examined system. Furthermore, user 12 answers “NO”, but it gives a higher prediction for the “YES” answer (70%) compared to that of the “NO” answer (30%). In that case, we observe that even though this user receives a negative information score (because of its unsurprisingly common answer), it also receives the second best prediction score (−0.0027) and a better BTS score than the aforementioned extreme user 2, which makes user 12 more truthful compared to user 2. Finally, it is shown that the prediction score of user 1 (−0.1132), who is uncertain for its opinion since its prediction is \( y_1(t, S^{(0)}) = (0.5, 0.5) \), is close to the median of the prediction scores’ range. Thus, we conclude that the proposed BTS framework evaluates in a neutral manner the uncertain users, assigning them also a positive BTS score.

As a next step, we examine the descendant of the BTS, i.e., the Robust Bayesian Truth Serum (RBTS), which is strictly Bayes-Nash incentive compatible for \( |U| \geq 3 \) [32], in order to compare it with our BTS component. It is noted that the key difference of the RBTS model compared to the BTS approach is that the first one converges to a stable equilibrium solution for small number of users, i.e., \( |U| \geq 3 \), while the BTS approach requires \( |U| \rightarrow \infty \). Thus, the RBTS
inherent drawback of the RBTS mechanism, which determines the truthfulness of the users based only on the reference \( r_u \) and peer \( p_u \) users. Given that these users (i.e., reference and peer users) are different for each user, the final RBTS scores are fully dependent on the reports of only two other users, which can totally decrease the truthful reporting incentive.

Moreover, we observe that the user 2 with the extreme reporting beliefs, who gave as a prediction report \( y_4^{(1)} = (0.98, 0.02) \), has acquired the 5\(^{th} \) best RBTS score, while the BTS approach discouraged such an extreme behavior by ranking it in the 12\(^{th} \) place. However, if we invert the positions of users 2 and 10, the RBTS scores would be 0.6552 and 1.6375, which means that in that case user 2 would be ranked in the 12\(^{th} \) place among all the users. Thus, we conclude to the outcome that even though the RBTS converges to a Bayes-Nash Equilibrium for smaller populations (\(|U| \geq 3\)) than the BTS mechanism (\(|U| \rightarrow \infty \)), it is more unstable regarding the users’ individual truthfulness scores. The basic cause for this fundamental problem is that the RBTS score of each user depends on its position (i.e., index) inside the entire population. For that reason, we conclude that the RBTS mechanism provides a weak incentive to the users to be truthful, compared to the BTS which is a lot more stable as shown in the previous discussion.

For some iterations, the MEC server \( s \) explores both actions, i.e., being activated (\( d_s^{(ite,t)} = 1 \)) or remaining at the sleep mode (\( d_s^{(ite,t)} = 0 \)), while learning its available decisions and exploring the MEC environment. However, afterwards the server chooses to be activated and its decision converges in few iterations, i.e., \( d_s^{(ite,t)} = 1 \). The aforementioned behavior is also indicated in Fig.4b, where the cumulative number of server’s \( s \) decisions \( d_s^{(ite,t)} \) is presented as a function of the BLA iterations. Specifically, we observe that the number of both the \( d_s^{(ite,t)} = 1 \) and \( d_s^{(ite,t)} = 0 \) decisions were increasing until the 4\(^{th} \) iteration, and then only the counter of the decision of being activated (\( d_s^{(ite,t)} = 1 \)) keeps increasing until the convergence of the server to its final decision of being activated. The reason that led the MEC server to that decision can be derived from the Beta Probability Density Function (PDF) presented in Fig.4c. Specifically, the results illustrate that the probability of the server \( s \) to be activated (\( a_s^{(1)} = 1 \)) is higher (blue curve) compared to the probability of remaining in the sleep mode (red curve). This behavior is justified based on the results presented in Fig.4d, where the hyper-parameter \( \beta_{s,1} \) is monotonically increasing, while the hyper-parameter \( \beta_{s,0} \) takes small values. The latter observation means that the server’s decision of being activated \( (d_s^{(ite,t)} = 1) \) led most of the times to a reward \( (E_{a_s^{(ite,t)}} = 1) \) rather than a penalty. The results presented in Fig.5 illustrate the operation of the proposed servers reputation scheme. Specifically, Fig.5a captures the average reputation \( \overline{R}_{a_s^{(ite,t)}} \) (Eq.8) and the total users’ reward \( \sum_{u_{\in U}} R_{a_s^{(ite,t)}} \) (Eq.9) as a function of the BLA iterations at a certain time slot \( t \). Initially, we observe that both the average reputation and the users’ reward change over time, since in every iteration the decisions of every server \( s \), i.e., \( d_s^{(ite,t)} \), may be different, meaning that different MEC servers decide to participate until the BLA’s convergence point. Also, we conclude to the outcome that the greater the average reputation \( \overline{R}_{a_s^{(ite,t)}} \) is, the more reward \( \sum_{u_{\in U}} R_{a_s^{(ite,t)}} \) the users offer to the servers. The latter observation confirms our theoretical formulation (Eq.9) and secondly is expected according to the Bayesian belief, i.e., the greater the servers’ reputation is, the stronger is the users’ trust that they will experience a high QoE, thus, they provide a high reward to the MEC servers. Moreover, Fig.5b illustrates a Monte Carlo evaluation (10,000 executions) of the MEC servers’ achieved utility with respect to the BLA iterations at a certain time slot \( t \). The results indicate that less than 100 iterations are required for the MEC servers to converge to their final decisions (equivalent to 0.8 sec), while at the same time the MEC
servers’ average utility is maximized. The latter observation is derived due to the fact that the proposed framework drives each MEC server to make optimal decisions \( a^{(d,e,t)}_u \) regarding its activation (or not), i.e., each MEC server considers if its overall reputation \( \mu^{(t)}_u \) and computation capability \( F^{(t)}_u \) are sufficient to receive a satisfying portion of the total reward \( \sum_{u \in U} R^{(d,e,t)}_u \) compared to all the other MEC servers.

Fig. 6 shows the operation of the artificial intelligent servers activation mechanism for two indicative MEC servers. Specifically, Fig. 6a shows that MEC server 1 achieves a higher reputation than the MEC server’s 2 reputation. Thus, the users trust more server 1 that it will satisfy their QoE requirements. This fact can also be validated in Fig. 6b, where the total number of the positive users’ evaluations for the first server, i.e., \( Q^{(t)}_1 \), is always higher than the corresponding \( Q^{(t)}_2 \) of the second server. This indicates that the users were more satisfied when server 1 contributed to their tasks’ execution (\( Q^{(t)}_1 > F^{(t)}_1 \) and \( Q^{(t)}_1 \) increases faster than \( F^{(t)}_1 \)), than they were when server 2 was activated since its respective total number of the negative evaluations became equal with that of the positive evaluations after a certain point of time (\( Q^{(t)}_2 = F^{(t)}_2 \)). Thus, MEC server 2 remained in the sleep mode for the last time slots since it was non-profitable for it to serve the users. Moreover, the fact that the users were satisfied from the received QoE from MEC server 1, drove the latter to become active more times than the MEC server 2 (Fig. 6c). The latter observation is also confirmed from the results presented in Fig. 6d, which show that the first server received a higher fraction of the total users’ reward due to its higher reputation, which finally resulted to a higher experienced utility over the time horizon (Fig. 6e).

Fig. 7 depicts the BLA iterations and the corresponding real execution time required for convergence of the MEC servers activation decision making as a function of the increasing number of MEC servers. The results reveal that the proposed framework scales very well with respect to the increasing number of servers, resulting in realistic execution time. Thus, the proposed framework can be implemented in a realistic MEC environment in real or close to real time manner.

C. COMPARATIVE EVALUATION

In this section, a comparative analysis is presented among the proposed artificial intelligent MEC servers management framework and four other approaches, which are described as follows:

(a) All the MEC servers are active at a certain time slot.
(b) The MEC servers decide randomly to become active at a certain time slot.
(c) Only half of the available MEC servers with the highest reputation are active at a certain time slot.
(d) Only half of the MEC servers with the highest computation capability are active at a certain time slot.

Fig. 8 depicts the average utility per MEC server at a specific time slot, including all the aforementioned alternative strategies. The results reveal that our approach, i.e., (e), outperforms compared to all the alternatives since it jointly considers both the servers’ reputation and computation capability in a holistic and intelligent manner, confirming that it enables all the servers to take the most profitable decisions in an autonomous manner. The approach (c) illustrates the second best results in terms of MEC servers’ average achieved utility, given that it allows only the most trusted MEC servers to become active. The worst performance is exhibited by the first two alternative approaches (i.e., (a) and (b)), as they are the least sophisticated and they do not consider the system parameters in the MEC servers’ activation decision making.
The research of Dr. Tsiropoulou and Mr. Fragkos was conducted as part of the NSF CRII-1849739.

REFERENCES

[1] (2020). Ericsson Mobility Report. [Online]. Available: https://www.ericsson.com/mobility-report/documents/2020/june2020-ericsson-mobility-report.pdf

[2] M. Ali, A. Anjum, O. Rana, A. R. Zamani, D. Balouek-Thomert, and M. Parashar, “RES: Real-time video stream analytics using edge enhanced clouds,” IEEE Trans. Cloud Comput., early access, May 6, 2020, doi: 10.1109/TCC.2020.2991748.

[3] G. Mitsis, P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Intelligent dynamic data offloading in a competitive mobile edge computing market,” Future Internet, vol. 11, no. 5, p. 118, May 2019.

[4] J. H. Anaemjumba, T. Yue, C. Iwendi, M. Alenezi, and M. Mittal, “Optimal cooperative offloading scheme for energy efficient multi-access edge computing,” IEEE Access, vol. 8, pp. 53931–53941, 2020.

[5] G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, “Artificial intelligence enabled distributed edge computing for Internet of Things applications,” in Proc. 16th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS), May 2020, pp. 450–457.

[6] M. Nduwayezu, Q.-V. Pham, and W.-J. Hwang, “Online computation offloading in NOMA-based multi-access edge computing: A deep reinforcement learning approach,” IEEE Access, vol. 8, pp. 99098–99109, 2020.

[7] R. Zhu, L. Liu, M. Ma, and H. Li, “Cognitive-inspired computing: Advances and novel applications,” Future Gener. Comput. Syst., vol. 109, pp. 706–709, Aug. 2020.

[8] C.-F. Chou, L. Golubchik, and J. Lui, “Multiclass multiserver threshold-based systems: A study of noninstantaneous server activation,” IEEE Trans. Parallel Distrib. Syst., vol. 18, no. 1, pp. 96–110, Jan. 2007.

[9] Y. Yao, L. Huang, A. B. Sharma, L. Golubchik, and M. J. Neely, “Power cost reduction in distributed data centers: A two-time-scale approach for delay tolerant workloads,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 200–211, Jan. 2014.

[10] S. Thananjeyan, C. A. Chan, E. Wong, and A. Nirmalathas, “Energy-efficient mobile edge hosts for mobile edge computing system,” in Proc. IEEE Int. Conf. Inf. Automat. Sustainability (ICIAS), Dec. 2018, pp. 1–6.

[11] X. Lyu, H. Tian, L. Jiang, A. Vinel, S. Maharjan, S. Gjessing, and Y. Zhang, “Selective offloading in mobile edge computing for the green Internet of Things,” IEEE Netw., vol. 32, no. 1, pp. 54–60, Jan. 2018.

[12] A. Marotta, F. D’Andreagiovanni, A. Kassler, and E. Zola, “On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures,” Comput. Netw., vol. 125, pp. 64–75, Oct. 2017.

[13] Y. Zhan, S. Guo, P. Li, and J. Zhang, “A deep reinforcement learning based offloading game in edge computing,” IEEE Trans. Comput., vol. 69, no. 6, pp. 883–893, Jun. 2020.

[14] Y. Xiao and M. Krunz, “Dynamic network slicing for scalable fog computing systems with energy harvesting,” IEEE J. Sel. Areas Commun., vol. 36, no. 12, pp. 2640–2654, Dec. 2018.

[15] W. Chen, D. Wang, and K. Li, “Multi-user multi-task computation offloading in green mobile edge cloud computing,” IEEE Trans. Services Comput., vol. 12, no. 5, pp. 726–738, Sep. 2019.

[16] A. L. Gouhar and A. Ali, “Power savings in servers infrastructure under backup and availability constraints,” in Proc. Int. Conf. Advances Comput. Sci. (ICACS), Feb. 2018, pp. 1–8.

[17] P. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, “Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration,” IEEE Trans. Comput., vol. 67, no. 9, pp. 1287–1300, Sep. 2018.

[18] S. Ranadheera, S. Maghsudi, and E. Hossain, “Computation offloading and activation of mobile edge computing servers: A minority game,” IEEE Wireless Commun. Lett., vol. 7, no. 5, pp. 688–691, Oct. 2018.

[19] S. Ranadheera, S. Maghsudi, and E. Hossain, “Mobile edge computation offloading using game theory and reinforcement learning,” 2017, arXiv:1711.09012. [Online]. Available: http://arxiv.org/abs/1711.09012

[20] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Game-theoretic learning-based QoS satisfaction in autonomous mobile edge computing,” in Proc. Global Inf. Infrastruct. Netw. Symp. (GIIS), Oct. 2018, pp. 1–5.

[21] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Cognitive data offloading in mobile edge computing for Internet of Things,” IEEE Access, vol. 8, pp. 55736–55749, 2020.

[22] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Risk-aware data offloading in multi-server multi-access edge computing environment,” IEEE/ACM Trans. Netw., vol. 28, no. 3, pp. 1405–1418, Jun. 2020.

[23] M. Huang, W. Liu, T. Wang, A. Liu, and S. Zhang, “A cloud–MEC collaborative task offloading scheme with service orchestration,” IEEE Internet Things J., vol. 7, no. 7, pp. 5792–5805, Jul. 2020.

[24] L. Zanzi, F. Giust, and V. Sciancalepore, “M2EC: A multi-tenant resource orchestration in multi-access edge computing systems,” in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Apr. 2018, pp. 1–6.

[25] D. Peelee, “A Bayesian truth serum for subjective data.” Science, vol. 306, no. 5695, pp. 462–466, Oct. 2004.

VOLUME 8, 2020

171303
[26] I. Zukerman, R. McConachy, and K. B. Korb, “Bayesian reasoning in an abductive mechanism for argument generation and analysis,” in Proc. AAAI/IAAI, 1998, pp. 833–838.

[27] T. van Erven and P. Harremos, “Rényi divergence and Kullback-Leibler divergence,” IEEE Trans. Inf. Theory, vol. 60, no. 7, pp. 3797–3820, Jul. 2014.

[28] L. M. B. Cabral, “The economics of trust and reputation: A primer,” New York Univ. CEPR, 2005.

[29] O.-C. Granmo, “A Bayesian learning automaton for solving two-armed Bernoulli bandit problems,” in Proc. 7th Int. Conf. Mach. Learn. Appl., Dec. 2008, pp. 23–30.

[30] D. Barber, Bayesian Reasoning and Machine Learning, Cambridge, U.K.: Cambridge Univ. Press, 2012.

[31] R. Thapa, L. Jiao, B. J. Oommen, and A. Yazidi, “A learning automaton-based scheme for scheduling domestic shiftable loads in smart grids,” IEEE Access, vol. 6, pp. 5348–5361, 2018.

[32] J. Witkowski and D. C. Parkes, “A robust Bayesian truth serum for small populations,” in Proc. 26th AAAI Conf. Artif. Intell., 2012, pp. 1492–1498.

GEORGIOS FRAGKOS (Student Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of New Mexico. He is also a Research Assistant with the Department of Electrical and Computer Engineering, University of New Mexico. His main research interests include reinforcement learning, game theory, optimization, contract theory, and blockchain. He was selected by the IEEE Albuquerque Section to receive the IEEE Outstanding Graduate Engineering Student Award 2020 for outstanding academic performance in the field of computing and communications as well as by the University of New Mexico, School of Engineering, for the Outstanding Graduate Award 2020.

SEAN LEBIEN received the bachelor’s degree in electrical and computer engineering from John Brown University, Siloam Springs, AR, USA, in 2018. He is currently pursuing the master’s degree in electrical and computer engineering with the University of New Mexico. He is currently working as a Computer Engineer for Dynamic Systems and Research, focusing on embedded software, FPGA design, and hardware implementation.

EIRINI ELENI TSIROPOULOU (Member, IEEE) is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of New Mexico. Her main research interests include cyber-physical social systems and wireless heterogeneous networks, with an emphasis on network modeling and optimization, resource orchestration in interdependent systems, reinforcement learning, game theory, network economics, and the Internet of Things. Totally, four of her papers received the Best Paper Award at IEEE WCNC in 2012, ADHOCNETS in 2015, IEEE/IFIP WMNC 2019, and INFOCOM 2019 by the IEEE ComSoc Technical Committee on Communications Systems Integration and Modeling. She was selected by the IEEE Communication Society - N2Women - as one of the top ten Rising Stars of 2017 in the communications and networking field. She received the NSF CRII Award in 2019 and the Early Career Award by the IEEE Communications Society Internet Technical Committee in 2019.