Fall 11-11-2020

Incentivization in Mobile Edge Computing using a Full Bayesian Approach

Sean Lebien
University of New Mexico - Main Campus

Sean Lebien

Candidate

Electrical and Computer Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

E.Tsiropoulou Eirini Eleni Tsiropoulou, Chairperson

Jim Plusquellic

Michael Devetsikiotis

Incentivization in Mobile Edge Computing using a Full Bayesian Approach

by

Sean Lebien

B.S, John Brown University, 2018

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2020

Dedication

To my wife, Lauren, and my parents, Kim and Steve, who have inspired me to pursue a higher degree all while supporting, loving, and encouraging me along the way.

Acknowledgments

I want to thank Professor Eirini-Eleni Tsiropoulou for the support and guidance she has given me throughout my time at the university. I also want to thank both Dr. Plusquellic and Dr. Devetsikiotis for being apart of my thesis committee.

I would also like to thank and acknowledge Georgios Fragkos for being my research partner and for the technical support he provided when working on this project.

And finally, I would like to thank my loving wife for supporting me throughout these past few years. Thank you.

Incentivization in Mobile Edge Computing using a Full Bayesian Approach

by

Sean Lebien

B.S, John Brown University, 2018

M.S., Computer Engineering, University of New Mexico, 2020

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 thesis, 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.

This work has been published in:

G. Fragkos, S. Lebien and E. E. Tsiropoulou, "Artificial Intelligent Multi-Access Edge Computing Servers Management," in IEEE Access, vol. 8, pp. 171292-171304, 2020, doi: 10.1109/ACCESS.2020.3025047.

Contents

List of Figures

List of Tables

Glossary

IoT	Internet of Things
MEC	Multi-access Edge Computing
QoS	Quality of Service
QoE	Quality of Experience
BTS	Bayesian Truth Serum
$\lvert S \rvert$	Environment consisting of MEC Servers, $S = \{1, \ldots, s, \ldots, \lvert S \rvert\}$
$\lvert U \rvert$	Environment consisting of MEC Users, $U = \{1, \ldots, s, \ldots, \lvert U \rvert\}$
T	Set of timeslots where T $= \{1, ..., t, ..., T\}$
$\mu_s^{(t)}$	MEC Server's Reputation
$F_s^{(t)}(\frac{CPUcycles}{sec})$	MEC Server's Computation ability
$T_u^{(t)}$	Offloaded Tasks from the MEC Users
$I_u^{(t)}$	The computation total input bits
$C_u^{(t)}$	Total Number of CPU Cycles required to accomplish the computation task $T_u^{(t)}$

w_i^{acc}	Weight - Accuracy parameter
$\phi_u^{(t)}$	The computation complexity of the task requested by the User
$a_s^{(t)}$	The Final decsion of the MES server s at time slot t
$\mathbf{x_u}^{(t,\mathcal{S}^{(t)})}$	The personal answer of user u which can be yes or no
$\mathbf{y_u}^{(t,\mathcal{S}^{(t)})}$	The prediction regarding the fraction of the users whose answer is yes or no
$\overline{x}_i^{(t,\mathcal{S}^{(t)})}$	Population Endorsement frequencies
$\overline{y}_i^{(t,\mathcal{S}^{(t)})}$	Geometric mean of the users' population predictions
$sc^{(t)}_{BTS,u}(\mathbf{x_u}^{(t,\mathcal{S}^{(t)})},\mathbf{y_u}^{(t,\mathcal{S}^{(t)})})$	BTS Score of each user u
$x^{(t)}_{BTS}$	The most truthful answer regarding the activated MEC servers' performance
$Q_s^{(t)}$	The number of times the MEC server offered its services to the user, and was evaluated as helpful
$F_s^{(t)}$	The number of times the MEC server offered its services to the user, and was evaluated as not helpful
$R_u^{(t)}$	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
$F_s^{(t)}$	The number of times the MEC server offered its services to the user, and was evaluated as not helpful
$F_s^{(t)}$	The number of times the MEC server offered its services to the user, and was evaluated as not helpful

Chapter 1

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 [2] and video-based applications [3, 4], online gaming, and others, request services with various stringent delay, energy, and processing constraints [5, 6]. Such mobile applications are typically resource-hungry in terms of computation demand and energy consumption [7, 8], thus, the mobile devices cannot support them locally and they migrate them to other available computing resources [9, 10]. The current end-to-end computing continuum consists of multi-access edge, fog, and cloud computing that jointly contribute to the processing of the end-users' offloaded computation tasks [11].

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 [12, 13]. MEC, as compared to cloud computing, can significantly improve the data offloading delay [14], the corresponding mobile devices' energy consumption [15], as well as the security levels [16], 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 [17] or distributed approaches based on optimization [18], game theoretic [19], or learning methods [20, 21] 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 (QoE) prerequisites [22, 23], or from the service providers' perspective (the ones that own the MEC servers), i.e., maximize their techno-economics benefit or profit. Consequently, in this paper [24], the problem of distributed and autonomous MEC servers' management is studied based on an artificial intelligence-enabled framework that is jointly driven by the end users' QoE satisfaction and the computing service providers' reputation.

1.1 Related Work

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 [25]. 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. Their numerical results demonstrated the activation characteristics had a significant effect on the system's performance and that resource sharing through the studied threshold techniques lead to significant cost savings. In [26], 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. They did however run into several practical concerns, such as a detailed problem model and simulation inconsistencies with the real-world. An energy saving-driven approach is proposed in [27], where the authors formulated a minimization problem of the MEC servers' total energy consumption under the constraints of users' QoS prerequisites (e.g., transmission power, price, rate [28, 29, 30]) 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. Their proposed algorithm significantly reduced the energy savings, achieving up to 34% during peak hours and 16% during morning hours.

In [31], 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. This could be achieved all while minimizing the energy consumption and signal overhead for latency required devices. Their results demonstrated that they were able to satisfy latency requirements of different services while also saving energy for IoT devices. The latter challenge is also addressed in [32], 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 [17], 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 [33] and reinforcement learning approach, respectively. A similar approach is proposed in [34], 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, the computing demands of their offloading tasks, as well as their wireless communication characteristics (i.e., cell selection [35]). Their offloading strategy combined a policy gradient DRL-based approach with a differential neural computer (DNC), which allowed for optimal results far more efficiently than other results.

A centralized approach for the MEC servers' management problem is introduced in [36], 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. The MEC servers' were observed to significantly improve the processing capacity if they are allowed to learn from its previous interactions with other servers, allowing it to learn about its overall environmental state. Focusing on the energy efficient operation of the MEC servers, the authors in [37] use a Lyapunov optimization approach to determine the MEC servers' energy harvesting policy from the environment [38, 39], jointly with the task offloading scheduling. They found they were able to significantly increase the overall system utility through their chosen optimizations. Following the philosophy of MEC servers' energy efficient operation, a dynamic power management scheme is introduced in [40] 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 considering a multi-service applications environment [41]. The efficient use of power could significantly improve the use of power resources in data centers. A scalability analysis of the MEC system is discussed in [42] for increasing number of users and offloading tasks. The authors adopt a Particle Swarm Optimization method in order to decide on the number of

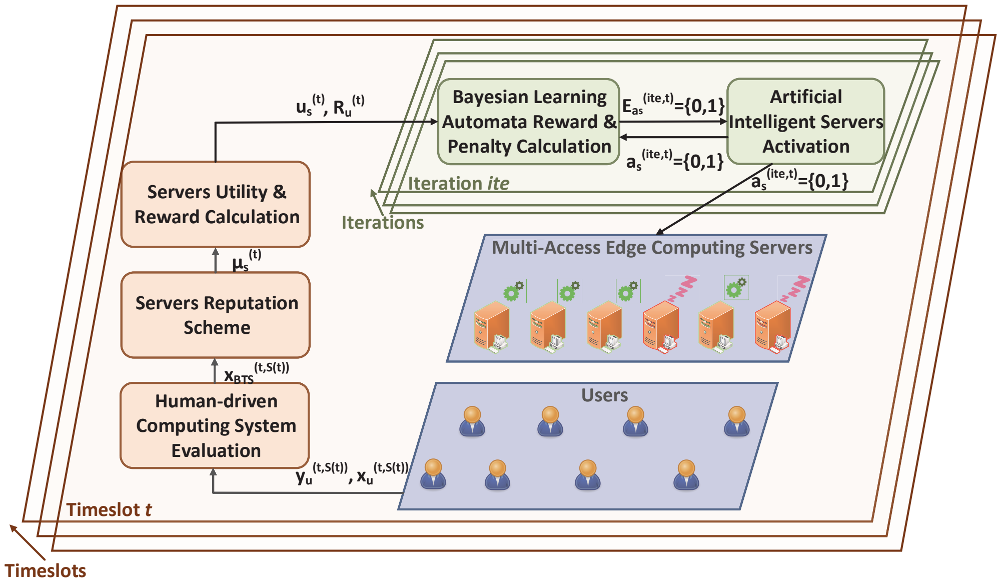
$u_s^{(t)}$, $R_u^{(t)}$
Bayesian Learning Automata Reward & Penalty Calculation
$E_{as}^{(ite,t)}$={0,1}
Artificial Intelligent Servers Activation
$a_s^{(ite,t)}$={0,1}
Iteration *ite*
$a_s^{(ite,t)}$={0,1}
Servers Utility & Reward Calculation
Iterations
Multi-Access Edge Computing Servers
$\mu_s^{(t)}$
Servers Reputation Scheme
$x_{BTS}^{(t,S(t))}$
Users
Human-driven Computing System Evaluation
$y_u^{(t,S(t))}$, $x_u^{(t,S(t))}$
Timeslot *t*
Timeslots

Figure 1.1: Overall architecture of the artificial intelligent multi-access edge computing servers management

activated MEC servers, while lowering the users' serving delay and maximizing the MEC system's cost effectiveness. Cloudlets of server clusters would theoretically reconfigure to maximize scalability, and would provide the same high-quality results for various types of requirements.

A minority game theoretic approach is introduced in [43, 44] 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 [45] via considering the problem of users' association to the MEC servers and introducing a distributed reinforcement learning-based decision making process. This approach can then be applied to smart cities, public safety networks, as well as mobile edge computational environments.

1.2 Contributions & Outline

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 [46, 47]. The proposed approach supports the extraction of an objective evaluation from subjective data provided by the users.

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 2, while the users' evaluation mechanism of the MEC servers is discussed in Section 3. Section 4 introduces the MEC servers' reputation model and the artificial intelligence-based MEC servers' activation is analyzed in Section 5. Simulation results are investigated in Section 6, while Section 7 concludes the paper.

Chapter 2

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 $F_s^{(t)}[\frac{CPUcycles}{sec}]$ 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 4.

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 $\phi_u^{(t)}[\frac{CPUcycles}{bits}]$ 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 [48, 49] 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 [50, 51].

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., $a_s^{(t)} \in \{0, 1\}$, where $a_s^{(t)}$ denotes the final decision of MEC server s at time slot t. For $a_s^{(t)} = 1$, the MEC server s is activated in time slot t, while for $a_s^{(t)} = 0$, the MEC server remains in sleep mode in time slot t. We consider $\mathcal{S}^{(t)} = \{s \in S : a_s^{(t)} = 1\}$ 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. 1.1.

Chapter 3

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 $T_u^{(t)}$ 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 $|U| \to \infty$ [52]. Specifically, at time slot t, user u answers a binary question, i.e., ”Are you satisfied from the activated MEC servers' $\mathcal{S}^{(t)}$ provided service?”, by providing the following two evaluation reports:

- The information report $\mathbf{x_u}^{(t,\mathcal{S}^{(t)})} = (x_{u,0}^{(t,\mathcal{S}^{(t)})}, x_{u,1}^{(t,\mathcal{S}^{(t)})})$, which denotes the personal answer of user u, where $\sum\limits_{i \in \{0,1\}} x_{u,i}^{(t,\mathcal{S}^{(t)})} = 1$, and i denotes the possible answers, i.e., "YES" ($i = 1$) or "NO" ($i = 0$) with $x_{u,i}^{(t,\mathcal{S}^{(t)})} \in \{0,1\}, \forall i \in \{0,1\}$.

- The prediction report $\mathbf{y_u}^{(t,\mathcal{S}^{(t)})} = (y_{u,0}^{(t,\mathcal{S}^{(t)})}, y_{u,1}^{(t,\mathcal{S}^{(t)})})$, where $y_{u,1}^{(t,\mathcal{S}^{(t)})}$ denotes the prediction regarding the fraction of the users whose answer is $x_{u,1}^{(t,\mathcal{S}^{(t)})} = 1$, i.e., "YES", and $y_{u,0}^{(t,\mathcal{S}^{(t)})}$ is the prediction for the fraction of the users whose answer is $x_{u,0}^{(t,\mathcal{S}^{(t)})} = 1$, i.e., "NO", and $\sum\limits_{i\in\{0,1\}} y_{u,i}^{(t,\mathcal{S}^{(t)})} = 1$.

Based on the users' information and prediction reports, the population endorsement frequencies $\overline{x}_i^{(t,\mathcal{S}^{(t)})}$ (Eq. 3.1) and the geometric mean of the users' population predictions $\overline{y}_i^{(t,\mathcal{S}^{(t)})}$ (Eq. 3.2) for each of the answers $i, i \in \{0,1\}$ are calculated as follows:

$$\overline{x}_i^{(t,\mathcal{S}^{(t)})} = \frac{1}{|U|} \cdot \sum_{u=1}^{|U|} x_{u,i}^{(t,\mathcal{S}^{(t)})} \tag{3.1}$$

and

$$log(\overline{y}_i^{(t,\mathcal{S}^{(t)})}) = \frac{1}{|U|} \cdot \sum_{u=1}^{|U|} \log(y_{u,i}^{(t,\mathcal{S}^{(t)})}) \tag{3.2}$$

Consequently, the BTS score $sc_{BTS,u}^{(t)}(\mathbf{x_u}^{(t,\mathcal{S}^{(t)})}, \mathbf{y_u}^{(t,\mathcal{S}^{(t)})})$ of each user u, which depicts how truthful its personal answer was, is calculated based on the formula:

$$\begin{aligned} sc_{BTS,u}^{(t)}(\mathbf{x_u}^{(t,\mathcal{S}^{(t)})}, \mathbf{y_u}^{(t,\mathcal{S}^{(t)})}) = &\sum_{i\in\{0,1\}} x_{u,i}^{(t,\mathcal{S}^{(t)})} \cdot \log(\frac{\overline{x}_i^{(t,\mathcal{S}^{(t)})}}{\overline{y}_i^{(t,\mathcal{S}^{(t)})}}) \\ &+ \alpha \cdot \sum_{i\in\{0,1\}} \overline{x}_i^{(t,\mathcal{S}^{(t)})} \cdot \log(\frac{y_{u,i}^{(t,\mathcal{S}^{(t)})}}{\overline{x}_i^{(t,\mathcal{S}^{(t)})}}) \end{aligned} \tag{3.3}$$

where the first part of Eq.3.3 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 $\overline{x}_i^{(t,\mathcal{S}^{(t)})}$ is higher than the corresponding mean prediction $\overline{y}_i^{(t,\mathcal{S}^{(t)})}$. 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 [53], the users that are truthful regarding their personal information reports $\mathbf{x_u}^{(t,\mathcal{S}^{(t)})}$, 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. 3.3) acts as a penalty proportional to the Kullback-Leibler divergence [54] between the actual population endorsement frequencies of the answers $\overline{x}_i^{(t,\mathcal{S}^{(t)})}$ and the respective user's population predictions' geometric means $y_{u,i}^{(t,\mathcal{S}^{(t)})}$. 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., $y_{u,i}^{(t,\mathcal{S}^{(t)})} = \overline{x}_i^{(t,\mathcal{S}^{(t)})}$. 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. 3.3 controls the effect of the prediction score (penalty) in the total BTS score, i.e., the higher is the value of parameter α, 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 α 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 $\overline{u}_i^{(t,\mathcal{S}^{(t)})}$ for each answer $i, i \in \{0,1\}$ for $\overline{x}_i \neq 0$, is calculated as follows, while $\overline{u}_i^{(t,\mathcal{S}^{(t)})} = 0$, if $\overline{x}_i = 0$.

$$\overline{u}_i^{(t,\mathcal{S}^{(t)})} = \frac{1}{|U| \cdot \overline{x}_i} \cdot \sum_{u=1}^{|U|} x_{u,i}^{(t,\mathcal{S}^{(t)})} \cdot sc_{BTS,u}^{(t)}(\mathbf{x_u}^{(t,\mathcal{S}^{(t)})}, \mathbf{y_u}^{(t,\mathcal{S}^{(t)})}) \tag{3.4}$$

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.,

$$x_{BTS}^{(t,\mathcal{S}^{(t)})} = argmax_{i \in \{0,1\}} \overline{u}_i^{(t,\mathcal{S}^{(t)})} \tag{3.5}$$

Since all the activated MEC servers contribute to the users' tasks execution in a distributed manner, the outcome of Eq. 3.5 determines if the experienced service provided by $\mathcal{S}^{(t)}$ activated MEC servers, satisfied ($x_{BTS}^{(t,\mathcal{S}^{(t)})} = 1$) or not ($x_{BTS}^{(t,\mathcal{S}^{(t)})} = 0$) the users' QoE prerequisites, based on the corresponding users' subjective evaluation.

Chapter 4

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 [55].

Specifically, all users share the same prior belief distribution $\mu_0^{(s,t)} = \mu_0, \forall s \in S, \forall t$, 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 3), i.e., if the users are overall satisfied $x_{BTS}^{(t,\mathcal{S}^{(t)})} = 1$ and if not $x_{BTS}^{(t,\mathcal{S}^{(t)})} = 0$ by the activated MEC servers set $\mathcal{S}^{(t)}$, creating thus a history for each MEC server s. We denote as $Q_s^{(t)}$, the number of times until time slot t that the MEC server s offered its services to the users and it was evaluated as helpful ($x_{BTS}^{(t,\mathcal{S}^{(t)})} = 1$), and as $\mathcal{F}_s^{(t)}$ we denote the number of times that the MEC server's s service was evaluated as not helpful ($x_{BTS}^{(t,\mathcal{S}^{(t)})} = 0$). Thus, each MEC server's s posterior distribution of reputation is given as follows [55].

$$\mu_s^{(t)} = \frac{\mu_0 \cdot a_H^{(Q_s^{(t)})} \cdot (1 - a_H)^{(\mathcal{F}_s^{(t)})}}{\mu_0 \cdot a_H^{(Q_s^{(t)})} \cdot (1 - a_H)^{(\mathcal{F}_s^{(t)})} + (1 - \mu_0) \cdot a_L^{(Q_s^{(t)})} \cdot (1 - a_L)^{(\mathcal{F}_s^{(t)})}} \tag{4.1}$$

Based on Eq.4.1, we observe that there is a correlation between the MEC server's s history of the users' positive and negative evaluations, i.e., the $Q_s^{(t)}$ 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., $Q_s^{(t)}$, and decreases with respect to the negative evaluations, i.e., $\mathcal{F}_s^{(t)}$.

4.1 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 sever 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.

$$u_s^{(t)}(a_s^{(t)}, a_{-\mathbf{s}}^{(\mathbf{t})}) = (\frac{\mu_s^{(t)} \cdot F_s^{(t)}}{\sum\limits_{s' \in S} (\mu_{s'}^{(t)} \cdot F_{s'}^{(t)} \cdot a_{s'}^{(t)})} \cdot \sum_{u \in U} R_u^{(t)} - F_s^{(t)}) \cdot a_s^{(t)} \tag{4.2}$$

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.4.2 expresses the portion of the total reward $\sum\limits_{u \in U} R_u^{(t)}$ 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 $s', s' \in \mathcal{S}^{(t)}$, 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 [55] based on the reputation $\mu_s^{(t)}$, as formulated in Eq. 4.1. 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 $s, s \in \mathcal{S}^{(t)}$, which offer their computational services to the users at time slot t, as follows.

$$\overline{\mu}^{(t)} = \frac{\sum\limits_{s=1}^{|S|} \mu_s^{(t)} \cdot a_s^{(t)}}{\sum\limits_{s=1}^{|S|} a_s^{(t)}} \tag{4.3}$$

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

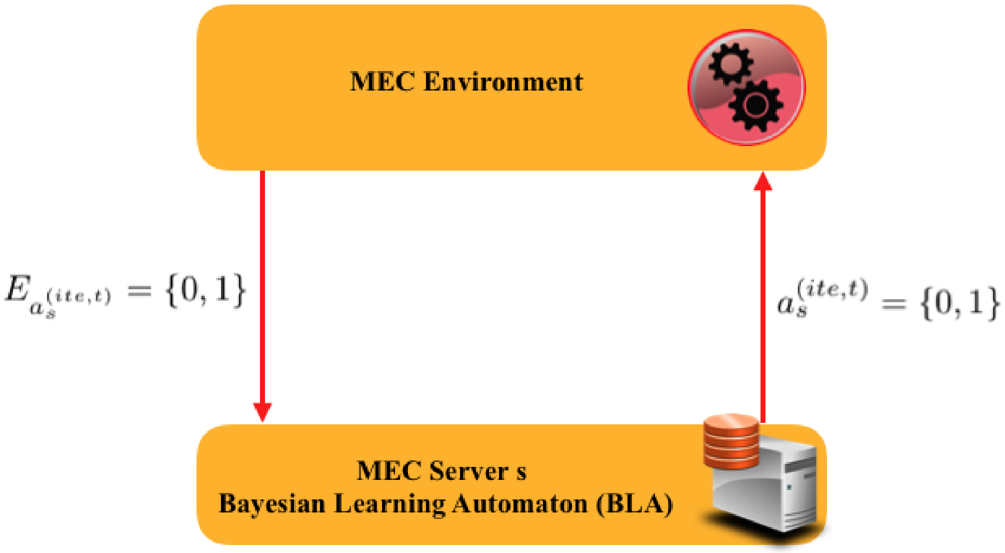
MEC Environment
$E_{a_s^{(ite,t)}} = \{0,1\}$
$a_s^{(ite,t)} = \{0,1\}$
MEC Server s
Bayesian Learning Automaton (BLA)

Figure 4.1: Bayesian Learning Automata (BLA) Learning Process

with the following reward:

$$R_u^{(t)} = \overline{\mu}^{(t)} \cdot a_H + (1 - \overline{\mu}^{(t)}) \cdot a_L \tag{4.4}$$

It is observed that the reward $R_u^{(t)}$ is an increasing function with respect to the average reputation $\overline{\mu}^{(t)}$, 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.

Chapter 5

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) [56] 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.4.1. Fig.1.1 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

Algorithm 1 BLA Rewards & Penalties

```
Input/Initialization: $S, a_{\mathbf{s}}^{(ite,t)}, \max_{ite}\{\mathbf{u_s}^{(ite-1,t)}\}$
Output: Reward or Penalty $\mathbf{E}_{a\mathbf{s}}^{(ite,t)}$
for $s = 1$ to $|S|$ do
    MEC Server $s$ calculates its utility $u_s^{(ite,t)}(a_s^{(ite,t)}, a_{-\mathbf{s}}^{(ite,t)})$
    if $(u_s^{(ite,t)}(a_s^{(ite,t)}, a_{-\mathbf{s}}^{(ite,t)}) \geq max_{ite}\{u_s^{(ite-1,t)}\})$ then
        Decision $a_s^{(ite,t)}$ leads to reward $E_{a_s^{(ite,t)}} = 1$
        Update the maximum experienced utility: $max_{ite}\{u_s^{(ite,t)}\} = u_s^{(ite,t)}(a_s^{(ite,t)}, a_{-\mathbf{s}}^{(ite,t)})$
    else
        Decision $a_s^{(ite,t)}$ leads to penalty $E_{a_s^{(ite,t)}} = 0$
    end if
end for
```

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.

5.1 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.4.1) 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 $\sum\limits_{u \in U} R_u^{(ite,t)} = \sum\limits_{u \in U} (\overline{\mu}^{(t)} \cdot a_H + (1 - \overline{\mu}^{(t)}) \cdot a_L)$ 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 every decision $a_s^{(ite,t)}$ is evaluated according to that value. The BLA rewards and penalties calculation algorithm is executed in every iteration of a time slot of the decision-making learning process and is presented in Algorithm 1.

The complexity of the BLA rewards and penalties calculation algorithm is $\Theta(|S|)$, since every MEC server s should determine if it will be rewarded or penalized by performing algebraic calculations, and the complexity of the latter is constant, i.e., $\Theta(1)$.

5.2 Bayesian Learning Automata Decision Making

The BLA-based learning method, which operates under the philosophy of the Bayesian reasoning framework [57], 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 computation need of the corresponding algorithm [58]. Every MEC server s that acts as a BLA preserves two hyper-parameters $\mathrm{a}_{s,i}$ and $\beta_{s,i}$ which correspond to the number of times that s received a reward and a penalty, respectively, by choosing action $i \in \{0,1\}$. Those hyper-parameters form the Beta probability density function (PDF) as follows:

$$f(x; \mathrm{a}_{s,i}, \beta_{s,i}) = \frac{x^{\mathrm{a}_{s,i}-1} \cdot (1-x)^{\beta_{s,i}-1}}{\int_0^1 u^{\mathrm{a}_{s,i}-1} \cdot (1-u)^{\beta_{s,i}-1} du} \tag{5.1}$$

thus leading to the respective cumulative Beta distribution function:

$$F(x; \mathrm{a}_{s,i}, \beta_{s,i}) = \frac{\int_0^1 v^{\mathrm{a}_{s,i}-1} \cdot (1-v)^{\beta_{s,i}-1} dv}{\int_0^1 u^{\mathrm{a}_{s,i}-1} \cdot (1-u)^{\beta_{s,i}-1} du} \tag{5.2}$$

where $x \in [0,1]$.

In every iteration *ite* at a specific time slot t, if the decision $a_s^{(ite,t)}$ of MEC server s leads to a reward ($E_{a_s^{(ite,t)}} = 1$), then the corresponding hyper-parameter $\mathrm{a}_{s,i}$ is increased, whereas if it leads to a penalty the corresponding hyper-parameter $\beta_{s,i}$ is increased. Thus, the cumulative Beta distribution function (Eq.5.1) serves as a Bayesian metric of the reward probabilities of each of the available actions $a_s^{(ite,t)} \in \{0,1\}$. The steps of the BLA-based decision making process are presented in Algorithm 2, where *convergence* is our termination criterion and it becomes true when each MEC server s that experiences a reward, i.e., $E_{a_s^{(ite,t)}} = 1$, chooses the

User u	Personal Answer $\mathbf{x_u}^{(t,\mathcal{S}^{(t)})}$	YES Prediction $y_{u,1}^{(t,\mathcal{S}^{(t)})}$	NO Prediction $y_{u,0}^{(t,\mathcal{S}^{(t)})}$	Information Score first term of Eq. 3.3	Prediction Score second term of Eq. 3.3	BTS Score $x_{BTS}^{(t,\mathcal{S}^{(t)})}$
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

Table 5.1: Bayesian Truth Serum (BTS) outcome

same action $a_s^{(ite,t)}$ for a specific number of K iterations, implying that a near-optimal solution has been determined with high probability.

Regarding the BLA-based Decision Making algorithm's complexity, for a specific time slot t, in a specific iteration ite, each MEC server s determines its action $a_s^{(ite,t)}$, which involves only algebraic calculations needing constant time $\Theta(1)$ to be executed, thus, concluding to a total execution time of $\Theta(|S|)$. Afterwards, Algorithm 1 is executed in order all the rewards and penalties to be determined with a corresponding complexity of $\Theta(|S|)$ (Section 5.1) and the hyper-parameters of all the MEC servers are updated concluding to a complexity of $\Theta(|S|)$. Thus, the total complexity of Algorithm 2 for time slot t is $\mathcal{O}(|S| \cdot Ite^{(t)})$, where $Ite^{(t)}$ is the total number of iterations for convergence in that time slot. The overall complexity of Algorithm 2 for all time slots T that the MEC system is examined, is $\mathcal{O}(T \cdot (|S| \cdot max_{t \in [1,T]}\{Ite^{(t)}\}))$.

Algorithm 2 BLA-based Decision Making

```
Input/Initialization: $ite = 0, convergence = false, \mathrm{a}_{s,1} = \beta_{s,1} = \mathrm{a}_{s,0} = \beta_{s,0} = 1$
    $\forall s, s \in S$
Output: Final Decision made by the MEC servers $a_{\mathbf{s}}^{(t)}$, Final Values of the hyper-
    parameters $\mathrm{a}_{s,i}$ and $\beta_{s,i} \quad \forall i, i \in \{0, 1\}$
while $(convergence = false)$ do
    $ite = ite + 1$
    for $s = 1$ to $|S|$ do
        Pick $x_{s,0}$ and $x_{s,1}$ from
        $F(x_{s,i}; \mathrm{a}_{s,i}, \beta_{s,i}) = \frac{\int_0^{x_{s,j}} v^{\mathrm{a}_{s,i}-1} \cdot (1-v)^{\beta_{s,i}-1} dv}{\int_0^1 u^{\mathrm{a}_{s,i}-1} \cdot (1-u)^{\beta_{s,i}-1} du}$
        if $(x_{s,0} \leq x_{s,1})$ then
            Participate: $a_s^{(ite,t)} = 1$
        else
            Don't Participate: $a_s^{(ite,t)} = 0$
        end if
    end for
    Based on the $a_{\mathbf{s}}^{(ite,t)}$ calculate the respective Rewards and Penalties by executing
    Algorithm 1.
    for $s = 1$ to $|S|$ do
        if $(E_{a_s^{(ite,t)}} = 1)$ then
            if $(a_s^{(ite,t)} = 0)$ then
                $\mathrm{a}_{s,0} = \mathrm{a}_{s,0} + 1$
            else
                $\mathrm{a}_{s,1} = \mathrm{a}_{s,1} + 1$
            end if
        else
            if $(a_s^{(ite,t)} = 0)$ then
                $\beta_{s,0} = \beta_{s,0} + 1$
            else
                $\beta_{s,1} = \beta_{s,1} + 1$
            end if
        end if
    end for
end while
```

Chapter 6

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 6.1). Then, the operation of the Bayesian Learning Automata (BLA) and the reputation system is demonstrated (Section 6.2), and a detailed comparative evlauation to other approaches is presented (Section 6.3).

The proposed framework's evaluation was conducted in a MacBook Pro Laptop, 2.5GHz Intel Core i7, with 16GB LPDDR3 available RAM. We consider $|S| = 20$ MEC servers, $K = 3$, $T = 100$ time slots, $\mu_0 = 0.2$, $a_H = 0.51$, $a_L = 0.49$, and $F_s^{(t)} \in [10 \cdot 10^9, 12 \cdot 10^9] \frac{CPU cycles}{sec}$, which is clipped into a smaller range of computing power units, i.e., $F_s^{(t)} \in [10, 12]$, 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 $|U| = 500$ users, otherwise explicitly stated.

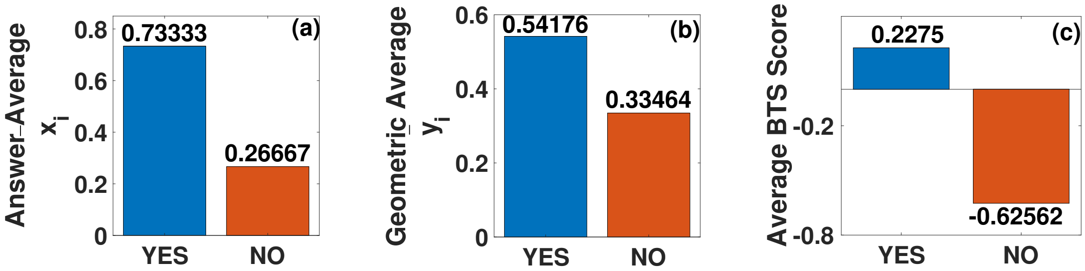

Figure 6.1: Human-driven computing system evaluation

6.1 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 $|U| = 15$ users for the evaluation of the BTS component at a specific time slot t.

Fig. 6.1a presents the average population endorsement frequencies $\overline{x}_i^{(t,\mathcal{S}^{(t)})}$ (Eq.3.1) 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. 6.1b shows the mean prediction $\overline{y}_i^{(t,\mathcal{S}^{(t)})}$ (Eq. 3.2) 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. 6.1a, 6.1b, we conclude to the average BTS scores $\overline{u}_i^{(t)}$ (Eq. 3.4) for the aforementioned answers, which are 0.2275 and -0.62562, respectively. Therefore, based on Eq.3.5, 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 3) 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. 6.1a-6.1c. 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 $\overline{x}_i^{(t,\mathcal{S}^{(t)})}$ and the corresponding mean predictions $\overline{y}_i^{(t,\mathcal{S}^{(t)})}$. Specifically, the information score (first part of Eq. 3.3) 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" $\overline{x}_1^{(t,\mathcal{S}^{(t)})}$ (73.33%, 11 out of 15 users) is higher than the corresponding mean prediction $\overline{y}_1^{(t,\mathcal{S}^{(t)})}$ (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 $\mathbf{y_u}^{(t,\mathcal{S}^{(t)})}$ to the mean of the actual personal answers $\overline{x}_i^{(t,\mathcal{S}^{(t)})}, i \in \{0,1\}$, 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 prediction $\mathbf{y_4}^{(t,\mathcal{S}^{(t)})}$ was the most accurate among all the users' predictions, and in combination with the positive information score of its surprisingly common answer, it is the most truthful answer in the examined system, receiving the highest BTS score (0.2558).

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 12^{th} 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

User u	Information Score	Prediction Score	RBTS Score x_{RBTS}
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	0.4375	0.7975
9	0	0.6400	0.64
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

Table 6.1: Robust Bayesian Truth Serum (RBTS) Outcome

"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 $\mathbf{y_4}^{(t,\mathcal{S}^{(t)})} = (0.5, 0.5)$, 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 $|U| \geq 3$ [59], 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 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 u provides the following two reports:

- The information report ${x'_u}^{(t,\mathcal{S}^{(t)})} \in \{0,1\}$, which is its personal answer and 1 denotes the answer "YES", while 0 denotes the answer "NO".
- The prediction report ${y'_u}^{(t,\mathcal{S}^{(t)})} \in [0,1]$, which is the user's prediction regarding the fraction of the users that answered "YES", i.e., ${x'_u}^{(t,\mathcal{S}^{(t)})} = 1$.

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:

$$y_u^{''(t,\mathcal{S}^{(t)})} = \begin{cases} {y'_{r_u}}^{(t,\mathcal{S}^{(t)})} + \delta & \text{,if } {x'_u}^{(t,\mathcal{S}^{(t)})} = 1 \\ {y'_{r_u}}^{(t,\mathcal{S}^{(t)})} - \delta & \text{,if } {x'_u}^{(t,\mathcal{S}^{(t)})} = 0 \end{cases} \tag{6.1}$$

where $\delta = min\{{y'_{r_u}}^{(t,\mathcal{S}^{(t)})}, 1 - {y'_{r_u}}^{(t,\mathcal{S}^{(t)})}\}$. Consequently, the RBTS score of each user u, which depicts how truthful its information report was, is calculated based on the formula [59]:

$$\begin{aligned} sc_u^{(t)}({x'_u}^{(t,\mathcal{S}^{(t)})}) = & R_{sps}(y_u^{''(t,\mathcal{S}^{(t)})}, {x'_{p_u}}^{(t,\mathcal{S}^{(t)})}) \\ & + R_{sps}({y'_u}^{(t,\mathcal{S}^{(t)})}, {x'_{p_u}}^{(t,\mathcal{S}^{(t)})}) \end{aligned} \tag{6.2}$$

where $R_{sps}(y_u^{'(t)}, x_{p_u}^{(t)})$ and $R_{sps}(y_u^{(t)}, x_{p_u}^{(t)})$ denote the information and the prediction scores respectively, while R_{sps} is a strictly proper scoring rule, defined as follows [59].

$$\begin{aligned} R_{sps}(y, x = 1) &= 2 \cdot y - y^2 \\ R_{sps}(y, x = 0) &= 1 - y^2 \end{aligned} \tag{6.3}$$

Finally, the truthful answer $x_{RBTS}^{(t)}$ regarding the MEC servers' service at time slot t is the answer that has the highest average RBTS score, and is determined as follows:

$$x_{RBTS}^{(t,\mathcal{S}^{(t)})} = max\{\frac{\sum\limits_{u\in U} sc_u^{(t)}({x'_u}^{(t,\mathcal{S}^{(t)})}=1)}{\sum\limits_{u\in U} {x'_u}^{(t,\mathcal{S}^{(t)})}}, \frac{\sum\limits_{u\in U} sc_u^{(t)}({x'_u}^{(t,\mathcal{S}^{(t)})}=0)}{|U| - \sum\limits_{u\in U} {x'_u}^{(t,\mathcal{S}^{(t)})}}\} \tag{6.4}$$

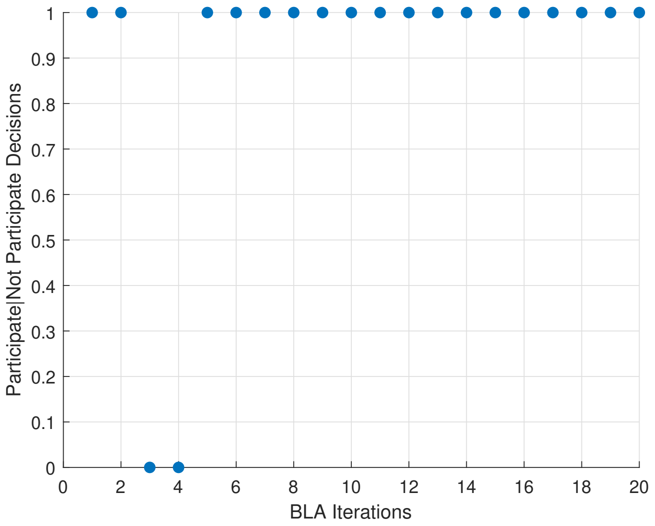

Figure 6.2: Bayesian Learning Automata-based MEC servers activation decision making

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., $x_{RBTS}^{(t)} = 1$ (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 inherent drawback of the RBTS mechanism, which determines the truthfulness of the users based only on the

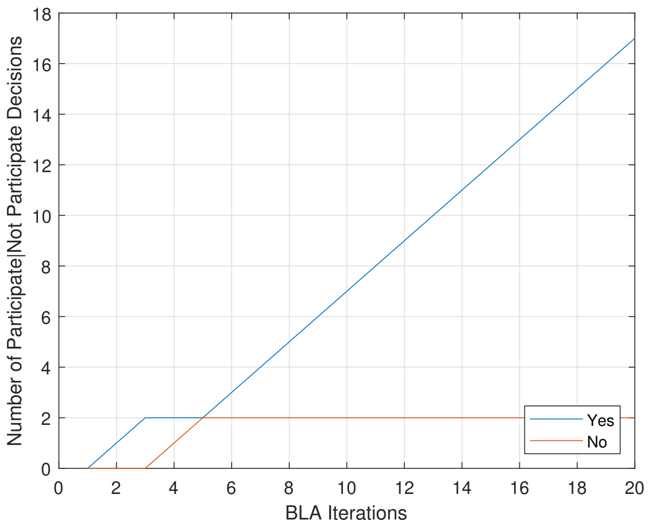

Figure 6.3: Bayesian Learning Automata-based MEC servers decision incrimination

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 $\mathbf{y_4^{(t)}} = (0.98, 0.02)$, 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

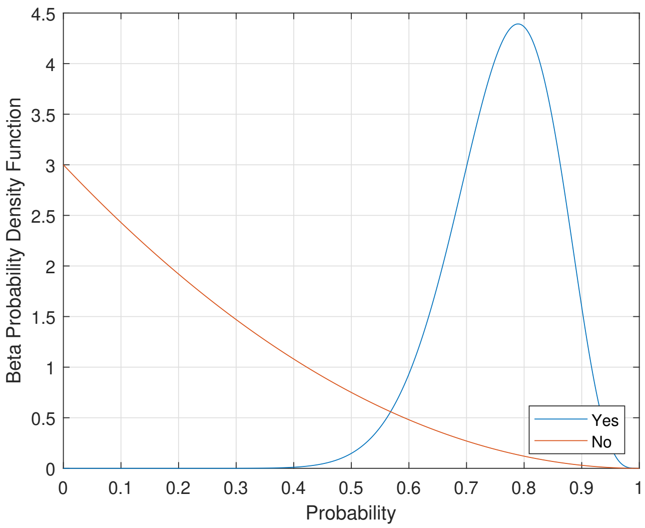

Figure 6.4: Bayesian Learning Automata-based MEC servers Beta distribution based on decision

more stable as shown in the previous discussion.

6.2 Bayesian Learning Automata & Reputation Evaluation

In this section, we study and analyze the behavior of the servers reputation scheme, as well as the artificial intelligent servers activation mechanism following the Bayesian Learning Automata approach, as they were introduced in the Sections 4 and 5, respectively. Our goal is to show the performance of the proposed framework towards determining the MEC servers' optimal decision of being activated or remaining in the sleep mode.

Fig.6.2 illustrates the decisions $a_s^{(ite,t)}$ of a single MEC server s throughout the BLA iterations (Algorithm 2) at a specific time slot t. It is observed that in the first

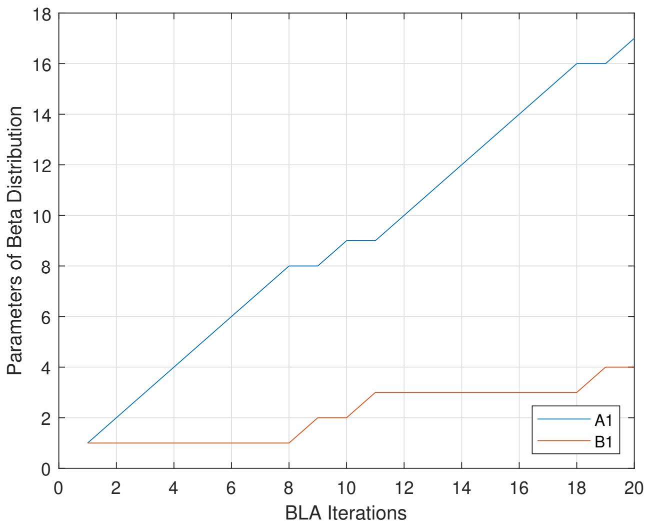

Figure 6.5: BLA-based MEC servers Parameters for Beta Distribution

4 iterations, the MEC server s explores both actions, i.e., being activated ($a_s^{(ite,t)} = 1$) or remaining at the sleep mode ($a_s^{(ite,t)} = 0$), 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., $a_s^{(t)} = 1$. The aforementioned behavior is also indicated in Fig.6.3, where the cumulative number of server's s decisions $a_s^{(ite,t)}$ is presented as a function of the BLA iterations. Specifically, we observe that the number of both the $a_s^{(ite,t)} = 1$ and $a_s^{(ite,t)} = 0$ decisions were increasing until the 4^{th} iteration, and then only the counter of the decision of being activated ($a_s^{(ite,t)} = 1$) 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.6.4. Specifically, the results illustrate that the probability of the server s to be activated ($a_s^{(t)} = 1$) 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.6.5, where the hyper-parameter $\mathrm{a}_{s,1}$ is monotonically increasing, while the hyper-parameter $\beta_{s,1}$ takes small values. The latter observation means that the server's decision of being

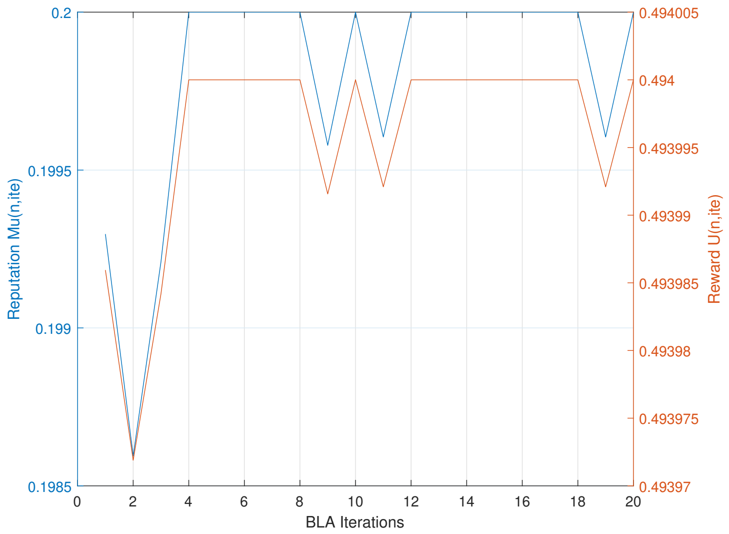

Figure 6.6: MEC servers average reputation and reward

activated ($a_s^{(ite,t)} = 1$), led most of the times to a reward ($E_{a_s^{(ite,t)}} = 1$), rather than a penalty.

The results presented in Fig. 6.6 and Fig. 6.7 illustrate the operation of the proposed servers reputation scheme. Specifically, Fig.6.6 captures the average reputation $\overline{\mu}^{(ite,t)}$ (Eq.8) and the total users' reward $\sum_{u \in U} R_u^{(ite,t)}$ (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., $a_s^{(ite,t)}$, 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 $\overline{\mu}^{(ite,t)}$ is, the more reward $\sum_{u \in U} R_u^{(ite,t)}$ 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.6.7 illustrates a Monte Carlo evaluation (10,000 executions) of the MEC

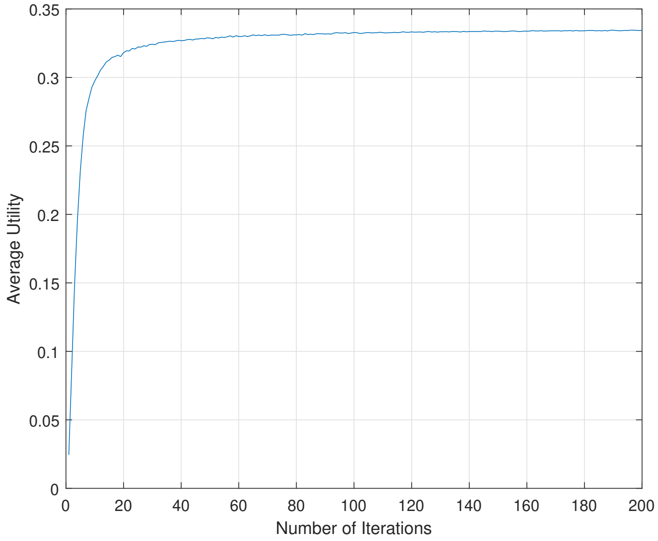

Figure 6.7: MEC servers average utility

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 $a_s^{(ite,t)}$ regarding its activation (or not), i.e., each MEC server considers if its overall reputation $\mu_s^{(t)}$ and computation capability $F_s^{(t)}$ are sufficient to receive a satisfying portion of the total reward $\sum\limits_{u \in U} R_u^{(ite,t)}$ compared to all the other MEC servers.

Fig. 6.8 through Fig. 6.12 shows the operation of the artificial intelligent servers activation mechanism for two indicative MEC servers. Specifically, Fig. 6.8 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. 6.9, where the total number of the positive users' evaluations for the first server, i.e., $Q_1^{(t)}$, is always higher than the corresponding $Q_2^{(t)}$

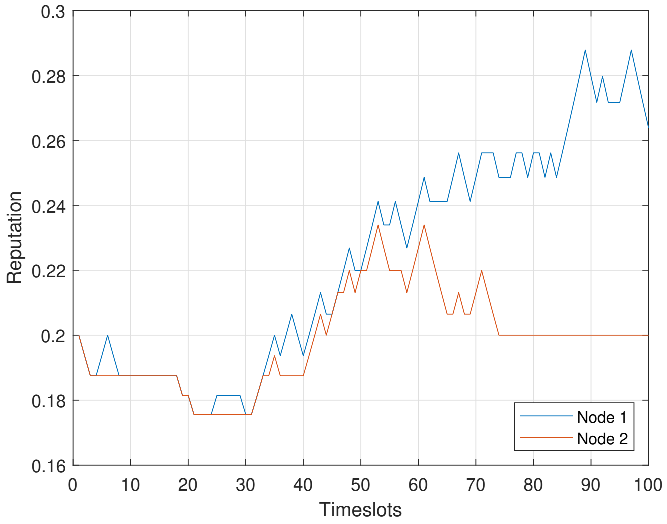

Figure 6.8: Reputation of two MEC servers over 100 timeslots

of the second server. This indicates that the users were more satisfied when server 1 contributed to their tasks' execution ($Q_1^{(t)} > \mathcal{F}_1^{(t)}$ and $Q_1^{(t)}$ increases faster than $\mathcal{F}_1^{(t)}$), 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 ($Q_2^{(t)} = F_2^{(t)}$). 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. 6.10). The latter observation is also confirmed from the results presented in Fig. 6.11, 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. 6.12).

Fig.6.13 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

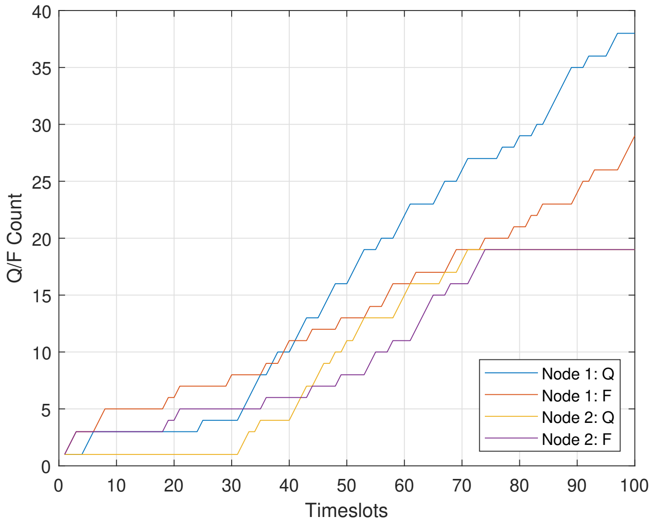

Figure 6.9: Q/F Count for two MEC servers

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.

6.3 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:

1. All the MEC servers are active at a certain time slot.

2. The MEC servers decide randomly to become active at a certain time slot.

3. Only half of the available MEC servers with the highest reputation are active at a certain time slot.

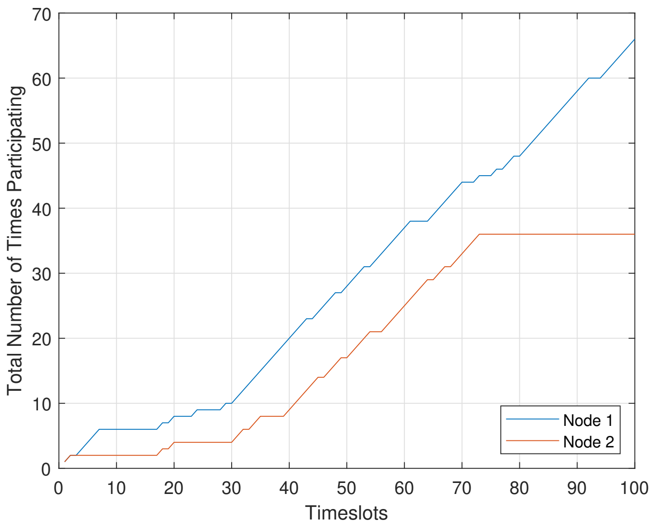

Figure 6.10: Number of participation events for two MEC servers

4. Only half of the MEC servers with the highest computation capability are active at a certain time slot.

Fig.6.14 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.

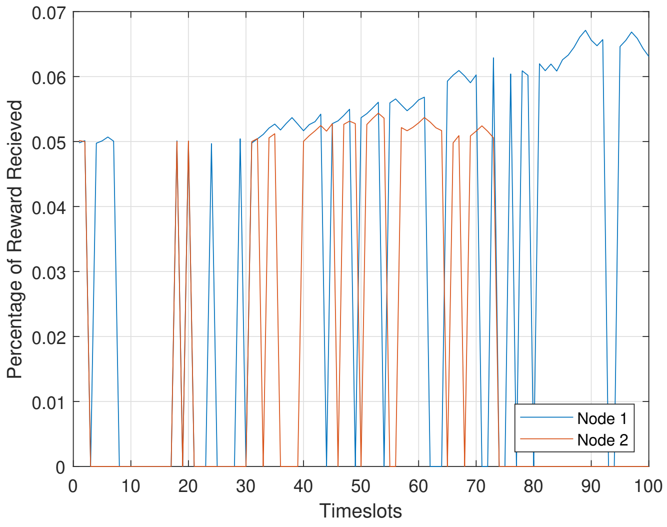

Figure 6.11: Percentage of reward for two MEC servers

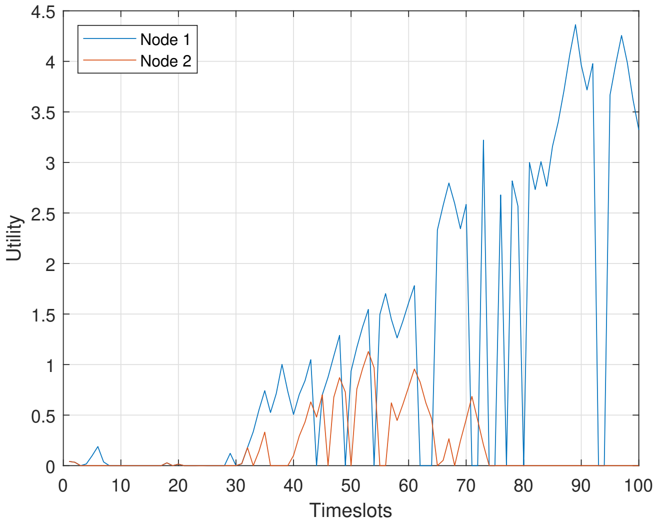

Figure 6.12: Utility of two MEC servers

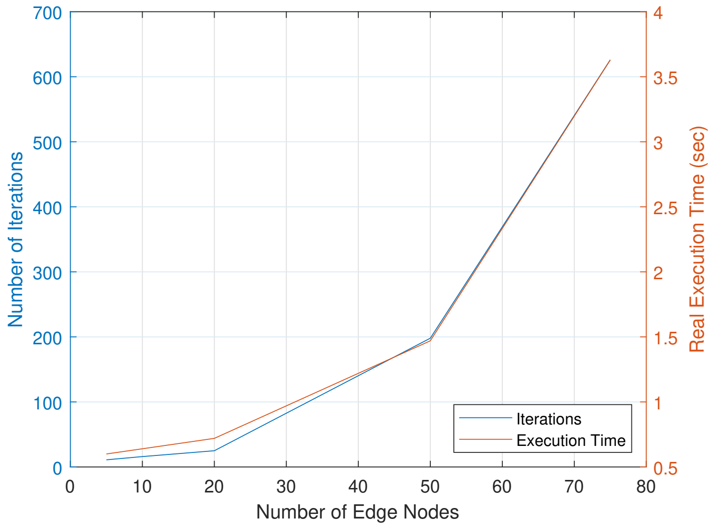

Figure 6.13: Scalability Evaluation

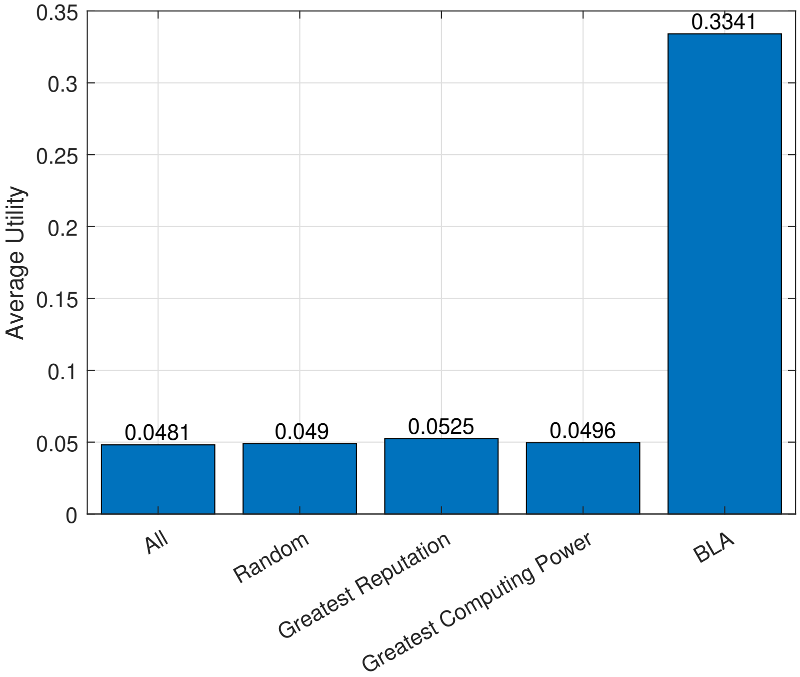

Figure 6.14: Comparative Evaluation

Chapter 7

Conclusion

The establishment of the multi-access edge computing (MEC) and its widespread adoption in 5G networks require the refinement of its management in an intelligent and cost-efficient manner. In this paper, a novel approach towards determining the MEC servers' activation, as well as the users' data processing, is introduced, based on the principles of artificial intelligence and specifically of reinforcement learning and Bayesian reasoning. The MEC servers operate as autonomous decision-makers towards maximizing their utility function, considering the offered reward from the users, which is modeled following the economic sale price model as well as the respective computing costs experienced from the aforementioned processing. Accordingly, the users follow a human-driven evaluation of the edge computing system based on the Bayesian Truth Serum concept regarding the satisfaction of their QoE prerequisites. Finally, this evaluation leads to a reputation scheme which characterizes the efficiency of the MEC servers in the edge computing system. A low complexity distributed algorithm which maximizes the utility function of the MEC servers is introduced, while detailed numerical results that clearly demonstrate our framework's operation and superiority are presented.

Our current and future work is focused on the extension of this model under the principles of Deep Reinforcement Learning, where the MEC servers will explore the continuous space of their available computing power resources in order to provide the optimal amount of it for the users' data processing. Moreover, we examine the prospect of incorporating Contract Theory into our framework, where each user acts as an "employer" offering rewards to the the MEC servers, which act as "employees" providing their computing resources to the users, in order the latter to process their data.

References

[1] "Ericsson mobility report," https://www.ericsson.com/49da93/assets/local/mobility-report/documents/2020/june2020-ericsson-mobility-report.pdf, 2020.

[2] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware social cloud computing based on serverless computing model," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[3] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, and R. Zimmermann, "Dynamic urban surveillance video stream processing using fog computing," in *2016 IEEE second international conference on multimedia big data (BigMM)*. IEEE, 2016, pp. 105–112.

[4] P. A. Apostolopoulos, M. Torres, and E. E. Tsiropoulou, "Satisfaction-aware data offloading in surveillance systems," in *Proceedings of the 14th Workshop on Challenged Networks*, 2019, pp. 21–26.

[5] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[6] E. E. Tsiropoulou, T. Kastrinogiannis, and S. Papavassiliou, "Uplink power control in qos-aware multi-service cdma wireless networks," *Journal of Communications*, vol. 4, no. 9, pp. 654–668, 2009.

[7] C. Singhal and S. De, *Resource allocation in next-generation broadband wireless access networks*. IGI Global, 2017.

[8] O. B. K. Pentikousis, R. A. Calvo, and S. Papavassiliou, *Mobile networks and management*. Springer, 2010.

[9] 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 Transactions on Cloud Computing*, 2020.

[10] G. Fragkos, N. Kemp, E. E. Tsiropoulou, and S. Papavassiliou, "Artificial intelligence empowered uavs data offloading in mobile edge computing," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.

[11] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *arXiv preprint arXiv:1701.01090*, 2017.

[12] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," *arXiv preprint arXiv:1502.01815*, 2015.

[13] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[14] D. Liu, L. Khoukhi, and A. Hafid, "Decentralized data offloading for mobile cloud computing based on game theory," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2017, pp. 20–24.

[15] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Energy efficient uplink joint resource allocation non-cooperative game with pricing," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2012, pp. 2352–2356.

[16] Q. Zhu and T. Başar, "Decision and game theory for security," 2013.

[17] 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, 2019.

[18] J. H. Anajemba, T. Yue, C. Iwendi, M. Alenezi, and M. Mittal, "Optimal cooperative offloading scheme for energy efficient multi-access edge computation," *IEEE Access*, vol. 8, pp. 53 931–53 941, 2020.

[19] G. Fragkos, E. E. Tsiropoulou, and S. Papavassiliou, "Artificial intelligence enabled distributed edge computing for internet of things applications," in *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE Computer Society, 2020, pp. 450–457.

[20] N. Maurice, Q.-V. Pham, and W.-J. Hwang, "Online computation offloading in noma-based multi-access edge computing: A deep reinforcement learning approach," *IEEE Access*, 2020.

[21] X.-L. Huang, X. Ma, and F. Hu, "Machine learning and intelligent communications," *Mobile Networks and Applications*, vol. 23, no. 1, pp. 68–70, 2018.

[22] R. Zhu, L. Liu, M. Ma, and H. Li, “Cognitive-inspired computing: Advances and novel applications,” 2020.

[23] M. Fasoulakis, E. E. Tsiropoulou, and S. Papavassiliou, “Satisfy instead of maximize: Improving operation efficiency in wireless communication networks,” *Computer Networks*, vol. 159, pp. 135–146, 2019.

[24] G. Fragkos, S. Lebien, and E. E. Tsiropoulou, “Artificial intelligent multi-access edge computing servers management,” *IEEE Access*, vol. 8, pp. 171 292–171 304, 2020.

[25] C.-F. Chou, L. Golubchik, and J. C.-S. Lui, “Multiclass multiserver threshold-based systems: A study of noninstantaneous server activation,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 1, pp. 96–110, 2006.

[26] 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 Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 200–211, 2012.

[27] S. Thananjeyan, C. A. Chan, E. Wong, and A. Nirmalathas, “Energy-efficient mobile edge hosts for mobile edge computing system,” in *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*. IEEE, 2018, pp. 1–6.

[28] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, “Joint customized price and power control for energy-efficient multi-service wireless networks via s-modular theory,” *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 1, pp. 17–28, 2017.

[29] R. Mahmud, S. N. Srirama, K. Ramamohanarao, and R. Buyya, “Quality of experience (qoe)-aware placement of applications in fog computing environments,” *Journal of Parallel and Distributed Computing*, vol. 132, pp. 190–203, 2019.

[30] E. E. Tsiropoulou, P. Vamvakas, G. K. Katsinis, and S. Papavassiliou, “Combined power and rate allocation in self-optimized multi-service two-tier femtocell networks,” *Computer Communications*, vol. 72, pp. 38–48, 2015.

[31] 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 Network*, vol. 32, no. 1, pp. 54–60, 2018.

[32] 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,” *Computer Networks*, vol. 125, pp. 64–75, 2017.

[33] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware resource control with flexible 5g access technology interfaces," in *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM).* IEEE, 2019, pp. 1–9.

[34] Y. Zhan, S. Guo, P. Li, and J. Zhang, "A deep reinforcement learning based offloading game in edge computing," *IEEE Transactions on Computers*, vol. 69, no. 6, pp. 883–893, 2020.

[35] E. E. Tsiropoulou, G. K. Katsinis, A. Filios, and S. Papavassiliou, "On the problem of optimal cell selection and uplink power control in open access multi-service two-tier femtocell networks," in *International Conference on Ad-Hoc Networks and Wireless.* Springer, 2014, pp. 114–127.

[36] Y. Xiao and M. Krunz, "Dynamic network slicing for scalable fog computing systems with energy harvesting," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2640–2654, 2018.

[37] W. Chen, D. Wang, and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 726–738, 2018.

[38] P. Vamvakas, E. E. Tsiropoulou, M. Vomvas, and S. Papavassiliou, "Adaptive power management in wireless powered communication networks: A user-centric approach," in *2017 IEEE 38th Sarnoff Symposium.* IEEE, 2017, pp. 1–6.

[39] E. Boshkovska, D. W. K. Ng, N. Zlatanov, and R. Schober, "Practical non-linear energy harvesting model and resource allocation for swipt systems," *IEEE Communications Letters*, vol. 19, no. 12, pp. 2082–2085, 2015.

[40] A. L. Gouhar and A. Ali, "Power savings in servers infrastructure under backup and availability constraints," in *2018 International Conference on Advancements in Computational Sciences (ICACS).* IEEE, 2018, pp. 1–8.

[41] T. Kastrinogiannis, E.-E. Tsiropoulou, and S. Papavassiliou, "Utility-based uplink power control in cdma wireless networks with real-time services," in *International Conference on Ad-Hoc Networks and Wireless.* Springer, 2008, pp. 307–320.

[42] T. 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 Transactions on Computers*, vol. 67, no. 9, pp. 1287–1300, 2018.

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

[44] ——, “Mobile edge computation offloading using game theory and reinforcement learning,” *arXiv preprint arXiv:1711.09012*, 2017.

[45] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, “Game-theoretic learning-based qos satisfaction in autonomous mobile edge computing,” in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2018, pp. 1–5.

[46] J. Cvitanić, D. Prelec, S. Radas, and H. Šikić, “Game of duels: Information-theoretic axiomatization of scoring rules,” *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 530–537, 2019.

[47] Y. Cai, G. Fragkos, E. E. Tsiropoulou, and A. Veneris, “A truth-inducing sybil resistant decentralized blockchain oracle,” in *2020 2nd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*, 2020, pp. 128–135.

[48] 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. 55 736–55 749, 2020.

[49] ——, “Risk-aware data offloading in multi-server multi-access edge computing environment,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1405–1418, 2020.

[50] M. Huang, W. Liu, T. Wang, A. Liu, and S. Zhang, “A cloud-mec collaborative task offloading scheme with service orchestration,” *IEEE Internet of Things Journal*, 2019.

[51] L. Zanzi, F. Giust, and V. Sciancalepore, “M 2 ec: A multi-tenant resource orchestration in multi-access edge computing systems,” in *2018 IEEE wireless communications and networking conference (wcnc)*. IEEE, 2018, pp. 1–6.

[52] D. Prelec, “A bayesian truth serum for subjective data,” *science*, vol. 306, no. 5695, pp. 462–466, 2004.

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

[54] T. Van Erven and P. Harremos, “Rényi divergence and kullback-leibler divergence,” *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.

[55] L. M. Cabral, “The economics of trust and reputation: A primer,” *New York University and CEPR*, vol. 2005, 2005.

[56] O.-C. Granmo, “A bayesian learning automaton for solving two-armed bernoulli bandit problems,” in *2008 Seventh International Conference on Machine Learning and Applications.* IEEE, 2008, pp. 23–30.

[57] D. Barber, *Bayesian reasoning and machine learning.* Cambridge University Press, 2012.

[58] 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, 2017.

[59] J. Witkowski and D. C. Parkes, “A robust bayesian truth serum for small populations,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.