Distributed Task Allocation in Mobile Device Cloud Exploiting Federated Learning and Subjective Logic

Article *in* Journal of Systems Architecture · December 2020

DOI: 10.1016/j.sysarc.2020.101972

Palash Roy
University of Dhaka

Sujan Sarker
University of Dhaka

Abdur Razzaque
Green University of Bangladesh

Mamun Rashid
University of Dhaka

Distributed Task Allocation in Mobile Device Cloud Exploiting Federated Learning and Subjective Logic

Palash Roy[a], Sujan Sarker[b], Md. Abdur Razzaque[a,c], Md. Mamun-or-Rashid[a], Mohmmad Mehedi Hassan[d], Giancarlo Fortino[e]

[a] *Green Networking Research Group, Department of Computer Science and Engineering, University of Dhaka, Bangladesh*

[b] *Department of Robotics and Mechatronics Engineering, University of Dhaka, Bangladesh*

[c] *Department of Computer Science and Engineering, Green University of Bangladesh*

[d] *Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia*

[e] *Department of Informatics, Modeling, Electronics, and Systems, University of Calabria,Rende, Italy*

Abstract

Mobile Device Cloud (MDC) has become a promising and lucrative cloud environment that exploit nearby mobile devices' idle resources to improve compute-intensive applications. Computing code at nearby mobile devices rather than a distant master cloud helps improve real-time applications' performance. However, it is non-trivial to motivate the worker devices to participate voluntarily in sharing their unused resources. In this paper, we have provided a distributed mobile device cloud environment by which workers make their auction decisions distributively and parallelly. We also introduce the federated learning and multi-weight subjective logic-based reputation scheme to measure worker mobile devices' trustworthiness and reliability. Moreover, a novel utility function for the buyers is proposed considering the cost, Quality-of-Experience (QoE), and the workers' reputation by which buyers select the most suitable worker in a distributed way. We have also proved that our proposed system achieves the desirable properties of computational efficiency, individual rationality, truthfulness, and budget balance. Empirical evaluations have been carried out in MATLAB that demonstrate the significant performance improvement in terms of QoE and utility of the buyers compared to other state-of-the-art works.

Keywords: Distributed Mobile Device Cloud, Federated Learning, Incentive Mechanism, Worker Reputation, Quality-of-Experience

1. Introduction

The emergence of portable, wearable devices or other smart devices has increased the utilization of mobile devices such as smartphones and tablets in our daily life than in the last few years. A recent study shows that the number of smartphone users will exceed 4.4 billion by 2022 [1]. These mobile devices have various types of built-in sensors by which high compute-intensive mobile applications like pattern or gesture recognition, health monitoring

Figure 1: An example of distributed MDC environment

and diagnosis, mobile bio-metric, reality augmentation, video, and image processing can be run in our smart devices [2]. However, mobile devices face limitations of CPU power, memory size, battery lifetime, and storage shortage. Due to the resource limitation, one way is to offload the task in the resource-rich cloud [3, 4, 5, 6, 7]. Although the cloud provides unlimited resources capacity, executing code at a distant master cloud suffers from higher latency, intermittent connectivity, and inability to provide real-time response [8, 9, 10, 11].

Many studies have proposed to offload tasks to the edge cloud or cloudlet closer to the users [12, 13, 14, 15]. Although virtual resources on a cloudlet help the users to execute a task in a short amount of time, they also suffer from limited coverage and unable to provide service in real-time at the peak hour [16]. In this situation, computational offloading can also be carried out among the underutilized resources of the nearby mobile devices, which helps to achieve lucrative system performance in the form of Mobile Device Cloud (MDC) [8]. Almost everywhere, the users are surrounded by an abundance of mobile devices in stationary places (i.e., stadium, shopping center or movie theater) or traveling time by bus/train or air. Though these devices are mobile, collaboration among relatively stationary mobile devices may improve the performances of resource-hungry applications. As the task executors are closer to the task requesters in MDC, communication latency, i.e., response delay and network congestion reduce significantly than executing those tasks in clouds or cloudlets [17, 18]. Thus, the MDC facilitates the development of real-time applications which are essential for smart living, e.g., security, traffic control, education, and disaster and waste management.

Fig. 1 shows an example scenario of a typical MDC environment where buyers request the workers to execute a task. After getting the buyers' requests, resource-rich worker devices execute this requested task and send the results. However, it is non-trivial to encourage worker mobile devices to share their resources to help neighbor mobile devices for task completion. Therefore, buyer devices need to provide remunerations to the worker devices to incentivize them to share their resources. Building an auction model that minimizes the user's execution cost while providing reasonable payment to the task executor mobile devices is a significant challenge in the MDC environment. Besides these, in most of the typical MDC environment [19], [20], [21], there has a central server by which auction decision is done. However, in a large scale distributed environment, workers need to make their auction decisions distributively. Due to the jockeying nature of the workers, sometimes workers do the untruthful activities intentionally.

In the literature, a few research works have been carried out to address the auction mechanism in the MDC environment. In [21], A-Long Jin et al. have designed two auction models for the tasks' homogeneous nature. However, they limit their auction models in a one-to-one matching manner where resource-rich mobile devices can not provide multiple buyers service. The authors in [20] have proposed two incentive mechanisms for homogeneous and heterogeneous nature of tasks in which they have considered a centralized auctioneer that needs to hold the global knowledge of the whole system. For the first time, a distributed auction model has been proposed in [22], where the central auctioneer is not required, and buyers submit their bids and workers locally make auction decisions. However, none of these approaches have considered the workers' reputation when submitting their bids and worker selection process. In the traditional reputation scheme, only the workers' positive and negative opinions are taken into account. However, the uncertainty situations independent of the buyers' and workers' actions are not considered.

To mitigate these challenges in this paper, our main motivation is to design a reputation-based distributed auction model. In this paper, we have devised a **FE**derated learning and **S**ubjective logic-driven distributed **T**ask allocation system in MDC, namely **FEST**. The design philosophy of the FEST system is considered as follows: a) the tasks that we have considered may require a heterogeneous amount of resources; b) each resource-rich worker mobile device is capable of executing tasks until it has enough resources; c) due to selfishness, sometimes workers want a higher payment to increase their utility. Buyer devices submit their bids to the worker devices to complete their tasks, while each worker device acts as an auctioneer. Buyer devices collaboratively calculate the workers' reputation to discourage the workers' selfish activities using Federated learning (FL). FL is a distributed machine learning paradigm that allows buyer mobile devices to evaluate a worker device in a decentralized manner, collaboratively to address these challenges [23]. When selecting worker devices, buyers have considered QoE and reputation of worker devices, and workers with a surplus in offered quality get an incentive amount added to their regular payments. The main contributions of this paper is summarized as follows:

- We develop a distributed framework for the MDC environment where each worker device acts as an auctioneer while buyers recruit highly reputed workers exploiting multi-weight subjective logic (MWSL) and federated learning (FL).

- We prove that the proposed FEST-Auction ensures the desirable properties, such as computational efficiency, buyers and workers' truthfulness, individual rationality, and budget balance.

- The results of the simulation experiments carried out in MATLAB depict the efficiency of the proposed FEST system in terms of QoE and the buyer devices' utility.

The rest of the paper is organized as follows. Section 2 describes some state-of-the-art works in the field of MDC. Subsequently, in Section 3, we have presented the system model and assumption to describe our proposed distributed MDC system FEST. In section 4, we describe the detailed process of reputation calculation of workers using federated learning, and subjective logic, auction mechanism, worker selection, and provide theoretical proof of the truthfulness of the buyers and workers, followed by the performance evaluation is conducted in Section 5. Finally, we summarize the whole paper in Section 6.

2. Related Works

With the advancement of 5G network technology and increasing smartphones, many real-time applications are run on our mobile devices. Therefore, offloading modules of an application in the remote cloud or cloudlet fails to provide real-time mobile applications. For that reason, offloading tasks to the nearby mobile devices and utilizing their idle resources are considered in mobile device cloud (MDC), that are the future mobile computing [24]. MDC applications have been gained much popularity in many researchers [19, 25, 26, 27, 28, 29, 30, 31].

The concept of the MDC was first used in Serendipity [26], which is based on the collaboration of nearby mobile devices for allocating tasks that speed up the computing speed and conserve energy. The authors in [32] have proposed a Stackelberg game model to capture the interaction between the task owner and worker mobile devices. Proper allocation of the tasks and determining the worker devices' payment, the Stackelberg equilibrium can be achieved. The authors in [33] have shown that offloading tasks to the nearby mobile devices can save task execution time and energy up to 50% and 26%, respectively, compared to offloading a task in the distant master cloud. However, in [32], [33], the authors have not considered the heterogeneous nature of the worker devices. Tasks executed on different mobile devices may require different amounts of time and achieve various performances. Furthermore, they have assumed that all mobile devices have the same energy level, which is not a real-life scenario. In [19], Saha et al. have made a trade-off between the execution speedup and selection of reliable worker devices in the MDC environment. However, in their environment, worker devices need to participate in the MDC system spontaneously. Workers have not provided any payment for the execution of the tasks.

In [21], the authors have proposed a Truthful Incentive Mechanism (TIM) for studying the resource sharing problem in mobile cloud computing (MCC). This scheme guarantees the truthfulness of the buyers and performs well for a homogeneous environment. However, they design the system for a one-to-one trading model that does not consider resource-rich mobile devices to support multiple buyers. Circa [34] has offloaded the tasks to some mobile devices within the same vicinity. Their main target is to identify a group of mobile devices to do the same offloading task. However, they have not considered the energy level of those mobile devices. The authors in [20] proposed two auction mechanisms for the homogeneous and heterogeneous nature of the tasks. In this work, they have considered a cloudlet works as a centralized auctioneer. A centralized auctioneer needs to hold the global knowledge of the whole MDC system, and it also breaks down the privacy of mobile users. Due to the mobile users' dynamic nature, when any user moves from one place to another, it requires high cost to update the parameters. X. Zhang et. al. have proposed a distributed auction model in [22], where there has no centralized auctioneer or cloudlet. Buyers submit their bids to the workers where each worker acts as an auctioneer and locally makes their auction decisions. Each auctioneer is provided payment for their task execution. When the buyers submit their bids to the worker, they haven't considered the worker devices' reputation. Besides these, They also have not considered user Quality-of-Experience (QoE) and quality of the executed task.

None of these existing works has focused on reputation-based auction mechanisms in a distributed MDC environment. Most of the existing studies have focused on bidding price when selecting the worker for task allocation, and workers are not motivated by providing a reward for successful task execution along with regular payments. Minimizing the execution cost of the users' task and providing lucrative payment to the workers based on their execution task quality is a challenging task in a distributed environment. These observations have led us to design a reputation-based distributed auction model where workers make their auction decisions distributively, and buyers recruit the most suitable worker based on the workers' reputation, cost, and time-based utility. Workers are provided payments as well as incentives according to their task execution with the required quality.

3. System Model

We consider a distributed mobile device cloud (MDC) environment, which consists of two entities: i) worker devices (i.e., workers) and ii) buyer devices (i.e., buyers). The buyers having a compute-intensive task, recruit computational resource-enrich worker devices to accomplish these tasks. The communication between the worker and buyer devices is facilitated through Wi-Fi access points or Bluetooth.

Let, $\mathcal{B} = \{b_1, b_2, \cdots, b_m\}$ denotes the set of m buyer devices each having one compute-intensive task. In this paper, we use tasks and buyers interchangeably. Executing these tasks require higher computing power as well as different amount of resources, i.e., CPU cycles and memory due to the heterogeneous nature of those tasks. The computational task of each buyer, $b_i \in \mathcal{B}$ is attributed by its resource requirement r_i, task size λ, task delay deadline t_i^d and true valuation v_i representing in amount of money that can be paid to a worker and this value depends on computational complexity of the task.

We assume that there are n resource-enrich worker devices in the system whose set is denoted by $\mathcal{W} = \{w_1, w_2, \cdots, w_n\}$. Each worker device $w_j \in \mathcal{W}$ has R_j units of available resources with minimum unit resource cost A_j. Note that, for simplicity, we only consider the computational resources, i.e., CPU cycles required to complete the execution of a task. However, our system can also work with other types of resources or combinations of those. Due to mobility, worker devices can move to a new location dynamically. Therefore, workers' availability plays a significant role in task completion and affects the buyer's Quality of

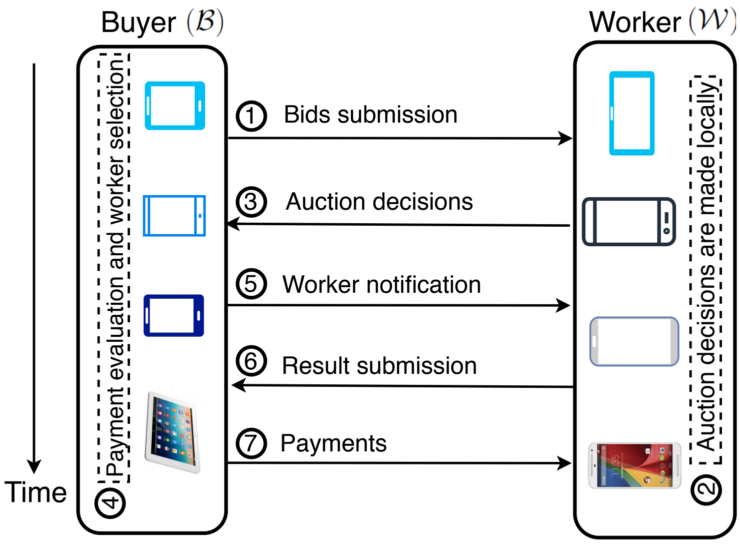


Figure 2: Interaction between the entities in distributed MDC

Service (QoS). Let T_j denotes the availability duration of the worker device $w_j \in \mathcal{W}$ in the system, which can be predicted accurately using the existing mobility models [35], [36] and each worker device has multi-tasking capability. Again, energy-constrained worker devices suffer from limited energy. Let, E_j be the available energy level of the worker device $w_j \in \mathcal{W}$ and energy expenditure for a computational task, $\mathcal{E}_{i,j}$ can be calculated from its size, λ and required CPU cycle N. A worker device, even with sufficient resources, may fail to deliver the computational results in time if it is overloaded with tasks, which in turn would degrade the reputation level of a worker to the buyer devices. Thus we limit the maximum number of workload by n_j^{max}.

In FEST, a worker device is paid according to the quality of its completed tasks. Upon successful completion of the assigned tasks, a worker gets its claimed price. In addition, if the worker completes the task with good quality, an incentive or reward is given to the user. Here, incentive denotes the monetary value added with the worker's claimed cost to yield its payment. Thus, payment is the summation of the workers' claimed cost and the incentive amount. On the other hand, a worker is penalized for providing low task quality. In that case, worker payment is less than its claimed price. In the FEST system, both the buyers and workers are non-cooperative as they try to maximize their utilities. However, the buyers only cooperate to calculate the reputation of the workers. Fig. 2 depicts the interactions between the buyer and worker devices in a distributed MDC environment. The detailed steps are listed below.

Step 1: Each buyer device $b_i \in \mathcal{B}$ first submits bids to the workers $w_j \in \mathcal{W}$, each denoted by $c_{i,j}$ along with its resource requirement and task delay deadline.

Step 2: After receiving the bids, each worker device (i.e., auctioneer) runs FEST-Auction to determine the candidate buyer devices and their claimed price for executing those tasks.

Table 1: Notations

Notation	Description
$\mathcal{B}$	The Set of all buyers in the system
$\mathcal{W}$	The Set of all workers in the system
$\mathcal{B}'_j$	The set of all winning buyers of worker $w_j \in \mathcal{W}$
r_i	The amount of resource required of buyer $b_i \in \mathcal{B}$
R_j	The amount of resource available at worker $w_j \in \mathcal{W}$
t_i^d	Task deadline of buyer $b_i \in \mathcal{B}$
t_i^c	Task completion time of buyer b_i from final worker
T_j	Availability time of worker $w_j \in \mathcal{W}$
p_i^b	Winning price of buyer $b_i \in \mathcal{B}$
$p_{i,j}^s$	Winning payment of worker $w_j \in \mathcal{W}$ from buyer $b_i \in \mathcal{B}$
$P_{i,j}$	Final price given by buyer $b_i \in \mathcal{B}$ to worker $w_j \in \mathcal{W}$
$c_{i,j}$	Unit resource bid submitted by buyer b_i to worker w_j
A_j	Minimum ask of worker $w_j \in \mathcal{W}$
$\Psi_{i,j}$	Priority factor of task b_i to worker w_j
$U_{i,j}^C, U_{i,j}^T, U_{i,j}^Q$	Buyer's utility due to cost, time and quality, respectively
$\mathcal{E}_{i,j}$	Energy required to execute task b_i by worker w_j
$\alpha_{i,j}, \beta_{i,j}$	Number of positive and negative interaction between buyer b_i and worker w_j
$\rho, \eta, \kappa, \zeta$	Weight of positive, negative, recent and past interactions, respectively
$\Omega_{i,j}^{fin}$	Final reputation value of buyer b_i for worker w_j

Step 3: The notifications are sent to the buyer devices about the winning bid candidates along with the claimed price.

Step 4: The buyer devices then distributively select the most suitable worker device based on its claimed price, task completion time, and reputation-based utility.

Step 5: After that, each buyer sends the selection notification to the worker along with the computational task.

Step 6: After completing the task execution, the worker sends back the computational result to the specific buyer device.

Step 7: The buyer then evaluates the computational results sent by the worker device, determines payment devising an incentive policy, namely FEST-Payment, and updates its reputation.

The major notations to describe the system are listed in Table 1.

4. Design of FEST-Auction System

In this section, we first develop a reputation scheme that relies on Federated Learning (FL) of worker reputation calculated using Multi-weight Subjective Logic (MWSL). Subsequently, we design an auction mechanism for workers, namely FEST-Auction, and finally, we propose a task allocation mechanism for buyers followed by a payment policy for the selected workers.

4.1. Computation of Worker Reputation

Worker reputation is one of the key design considerations of the FEST system. Due to the jockeying nature, sometimes workers with low quality may get selected hiding their true quality. To deal with this, the proposed FEST system relies on worker reputation as a highly reputed worker is expected to execute a computational task with higher reliability and trustworthiness. If reputation is not considered, there is a higher possibility of selecting low-quality workers having unreliable and inexperienced execution performance, reducing the system performance drastically. Unlike the existing reputation scheme, we use subjective logic-based reputation calculation, which represents the reputation of a worker through positive, negative, and uncertainty statements [23]. We exploit federated learning (FL) in distributed MDC for calculating worker reputation with high accuracy. In FL, a buyer calculates the reputation value of a worker by combining the worker's direct reputation (i.e., reputation calculated by itself) with the indirect reputations for that worker collected from other buyers.

4.1.1. Calculating Weighted Reputation

The opinion of a buyer device i to a worker device j is expressed by a vector $\chi_{i,j} = \{\Upsilon_{i,j}, d_{i,j}, u_{i,j}, a_{i,j}\}$, where, $\Upsilon_{i,j}, d_{i,j}, u_{i,j}$, and $a_{i,j}$ denote the belief, disbelief, uncertainty, and base-rate of an opinion, respectively. We calculate each of these parameters of an opinion based on subjective logic [37] as follows,

$$\begin{cases} \Upsilon_{i,j} = (1 - u_{i,j})\frac{\alpha_{i,j}}{\alpha_{i,j}+\beta i,j}, \\ d_{i,j} = (1 - u_{i,j})\frac{\beta_{i,j}}{\alpha_{i,j}+\beta i,j}, \\ u_{i,j} = (1 - \mathcal{Q}_{i,j}), \end{cases} \tag{1}$$

where, $\Upsilon_{i,j}, d_{i,j}, u_{i,j}, a_{i,j} \in [0, 1]$, and $\Upsilon_{i,j}+d_{i,j}+u_{i,j} = 1$. Here, $\alpha_{i,j}$ and $\beta_{i,j}$ denote the number of positive and negative interactions, respectively; $\mathcal{Q}_{i,j}$ denotes the success probability of the interaction between the mobile device b_i and worker device w_j. Now, a buyer device b_i calculates the reputation value of a worker device w_j as follows,

$$\Omega_{i,j} = \Upsilon_{i,j} + a_{i,j} \times u_{i,j}, \tag{2}$$

In subjective logic-based reputation scheme, the reputation (i.e., opinion) of a worker is affected by several factors [38]. In FEST, we identify four factors- interaction frequency, interaction effects, interaction similarity and interaction freshness affecting the reputation value of a worker device which are defined as below.

- **Interaction Frequency:** The higher value of the interaction frequency, $IF_{i,j}$ represents that the buyer device b_i has more prior knowledge about the worker device w_j and is calculated as,

$$IF_{i,j} = \frac{N_{i,j}}{\overline{N_i}}, \tag{3}$$

where, $\alpha_{i,j} + \beta_{i,j} = N_{i,j}$ and $\overline{N_i} = \sum_{w \in W} N_{i,w}$.

- **Interaction Effects:** In FEST, we consider both the positive and negative effects of the interaction between the buyer and worker devices. In case of positive interaction, the reputation value of a worker device is increased otherwise, it is decreased with the negative interaction. However, the negative interactions have higher effects than that of positive interactions to discourage the workers providing negative experiences. Thus FEST assigns weights to the positive and negative interactions denoted by ρ and η, respectively where, $\rho + \eta = 1$ and $\rho \leq \eta$.

- **Interaction Freshness:** To reflect the past interaction effects on the present interaction experience we assign different weights to the corresponding interactions between the buyer device b_i and worker device w_j. Let $\alpha^r_{i,j}$ and $\alpha^p_{i,j}$ denote the recent and past positive experiences, respectively. Note that, similar notations are used for the negative interaction experience, $\beta_{i,j}$. Now, we calculate the weighted effect for the interaction experiences as follows,

$$\begin{aligned} \alpha_{i,j} &= \kappa\rho\alpha^r_{i,j} + \zeta\rho\alpha^p_{i,j}, \\ \beta_{i,j} &= \kappa\eta\beta^r_{i,j} + \zeta\eta\beta^p_{i,j}, \end{aligned} \tag{4}$$

where, κ and ζ represent the weights of recent and past interactions and $\kappa + \zeta = 1$ and $\kappa \geq \zeta$. Now, the equation (1) can be rewritten as,

$$\begin{cases} \Upsilon_{i,j} = (1 - u_{i,j}) \frac{\rho(\kappa\alpha^r_{i,j} + \zeta\alpha^p_{i,j})}{\rho(\kappa\alpha^r_{i,j} + \zeta\alpha^p_{i,j}) + \eta(\kappa\beta^r_{i,j} + \zeta\beta^p_{i,j})}, \\ d_{i,j} = (1 - u_{i,j}) \frac{\eta(\kappa\beta^r_{i,j} + \zeta\beta^p_{i,j})}{\rho(\kappa\alpha^r_{i,j} + \zeta\alpha^p_{i,j}) + \eta(\kappa\beta^r_{i,j} + \zeta\beta^p_{i,j})}, \\ u_{i,j} = (1 - \mathcal{Q}_{i,j}). \end{cases} \tag{5}$$

Therefore, the Interaction Frequency, $IF_{i,j}$ is updated as follows,

$$IF_{i,j} = \frac{N_{i,j}}{\overline{N_i}} = \frac{\rho(\kappa\alpha^r_{i,j} + \zeta\alpha^p_{i,j}) + \eta(\kappa\beta^r_{i,j} + \zeta\beta^p_{i,j})}{\sum_{w \in W} N_{i,w}}. \tag{6}$$

- **Interaction Similarity:** Interaction similarity, $S^j_{i,k}$ measures the similarity of the interaction frequency of two buyer devices b_i and b_k while interacting with the worker devices w_j. We calculate the interaction similarity of two buyer devices as follows,

$$S^j_{i,k} = \frac{1}{1 + |IF_{i,j} - IF_{k,j}|} \tag{7}$$

4.1.2. Federated Learning on Worker Reputation

In FEST, each buyer device, $b_i \in \mathcal{B}$ calculates the reputation of a worker device w_j using Eq. 5 and Eq. 6. In a distributed MDC environment, all the buyer devices parallelly calculates the reputation values of all the workers present in the system. Therefore, to get the accurate reputation value of a worker, a buyer needs to combine the opinions of the other buyers (i.e., indirect reputation) with its own opinion (i.e., direct reputation). To facilitate this, FEST devises a federated learning-based reputation scheme where a buyer device leans the combined reputation of a worker device from its direct opinion and the indirect opinions of the other buyer devices. Let, $\chi'_{k,j} = \{\Upsilon'_{k,j},\ d'_{k,j},\ u'_{k,j}\}$ denotes the combined indirect opinions of a worker w_j from the buyer devices except b_i.

$$\begin{cases} \Upsilon'_{k,j} = \dfrac{1}{\sum\limits_{k \in \mathcal{B}\setminus\{b_i\}} \delta^j_{i,k}} \sum\limits_{k \in \mathcal{B}\setminus\{b_i\}} \delta^j_{i,k} \Upsilon_{k,j}, \\ d'_{k,j} = \dfrac{1}{\sum\limits_{k \in \mathcal{B}\setminus\{b_i\}} \delta^j_{i,k}} \sum\limits_{k \in \mathcal{B}\setminus\{b_i\}} \delta^j_{i,k} d_{k,j}, \\ u'_{k,j} = \dfrac{1}{\sum\limits_{k \in \mathcal{B}\setminus\{b_i\}} \delta^j_{i,k}} \sum\limits_{k \in \mathcal{B}\setminus\{b_i\}} \delta^j_{i,k} u_{k,j}. \end{cases} \tag{8}$$

Here, $\delta^j_{i,k}$ denotes the weight of the similarity from the recommender mobile device k which can be calculated as $\delta^j_{i,k} = \gamma \times S^j_{i,k}$; $\chi'_{k,j}$ represents the external knowledge of mobile device b_i about worker device w_j, which should be combined with the individual knowledge. The final reputation depends on not only the direct reputation opinion from buyer b_i, but also the reputation opinion from the recommenders to avoid cheating activity which is denoted by $\chi^{fin}_{i,j} = \{\Upsilon^{fin}_{i,j},\ d^{fin}_{i,j},\ u^{fin}_{i,j}\}$. We can use Cumulative Fusion Rule [23] to combine them and create a final reputation, $\chi^{fin}_{i,j} = \chi_{i,j} \oplus \chi'_{k,j}$, and the final belief, disbelief and uncertainty are computed as follows,

$$\begin{cases} \Upsilon^{fin}_{i,j} = \dfrac{\Upsilon_{i,j} u'_{k,j} + u_{i,j} \Upsilon'_{k,j}}{u'_{k,j} + u_{i,j} - u'_{k,j} u_{i,j}}, \\ d^{fin}_{i,j} = \dfrac{d_{i,j} u'_{k,j} + u_{i,j} d'_{k,j}}{u'_{k,j} + u_{i,j} - u'_{k,j} u_{i,j}}, \\ u^{fin}_{i,j} = \dfrac{u'_{k,j} u_{i,j}}{u'_{k,j} + u_{i,j} - u'_{k,j} u_{i,j}}. \end{cases} \tag{9}$$

The final reputation value can be calculated as Eq. (2) for worker device w_j of mobile device b_i is:

$$\Omega^{fin}_{i,j} = \Upsilon^{fin}_{i,j} + a_{i,j} \times u^{fin}_{i,j}. \tag{10}$$

4.2. FEST-Auction at Workers

In this section, a distributed auction mechanism, namely FEST-Auction is devised that is run at each worker device parallelly to select the winning bids submitted by the buyer devices followed by a payment determination process.

At the beginning of FEST-Auction, each worker device $w_j \in \mathcal{W}$ receives task requests from the buyer devices. Each request from a buyer device $b_i \in \mathcal{B}$ is characterized by the bid amount $c_{i,j}$, resource requirement r_i, and delay deadline, t_i^d of the task. Traditional auction mechanisms [21, 22], only consider tasks' bids while selecting the buyer devices. Unlike those mechanisms, FEST-Auction incorporates task preference, energy expenditure, and execution time of a task in a worker device along with the task bid. In FEST system, a worker device has different preferences for task execution from different buyer devices. This enables the worker devices to select tasks which best match with its expertise and interest level to maximize its own utility. We denote the preference of a worker device $w_j \in \mathcal{W}$ to the task $b_i \in \mathcal{B}$ by a metric, $\phi_{i,j} \in [0, 1]$. The higher value of $\phi_{i,j}$ reflects the higher preference.

To calculate the energy expenditure and task execution time, we assume that the computational task size is λ bits, where the processing of each bit requires N numbers of CPU cycles. Thus, the task consumes $\lambda \times N$ CPU cycles in total to be executed by the worker device. For a CPU cycle m, its frequency f_m can be calibrated using the dynamic voltage and frequency scaling technique proposed in [39]. Now each worker w_j calculates the energy expenditure and execution time of task b_i in the similar way as in [40] using Eq. 11 and Eq. 12, respectively as follows,

$$\mathcal{E}_{i,j} = \sum_{m=1}^{\lambda \times N} \vartheta f_m^2, \tag{11}$$

$$\mathcal{T}_{i,j} = \sum_{m=1}^{\lambda \times N} \frac{1}{f_m}, \tag{12}$$

where, ϑ indicates the capacitance coefficient of CPU chip architecture of the mobile devices. To incorporate the aforementioned parameters into the FEST-Action mechanism, we formulate an integrated metric based on task bid, execution time and preference to worker, namely priority factor denoted by $\Psi_{i,j}$ as follows,

$$\Psi_{i,j} = \frac{c_{i,j} \times \phi_{i,j}}{\mathcal{T}_{i,j}}. \tag{13}$$

Note that, the priority factor $\Psi_{i,j}$ has a direct influence in buyer selection decision and it increases with the higher values of $c_{i,j}$ and $\phi_{i,j}$ while decreases with the higher $\mathcal{T}_{i,j}$. Algorithm 1 summarizes the steps of FEST-Auction mechanism.

In Algorithm 1, we we prepare a short list of the buyer devices $\mathcal{B}_j \subseteq \mathcal{B}$ those satisfy the constraints: 1) *Budget Constraint:* bid must exceed the minimum recruitment cost of the worker, i.e., $c_{i,j} \geq A_j$; 2) *Resource Constraint:* required resources should be available at worker w_j, $r_i \leq R_j$; 3) *Deadline Constraint:* task deadline t_i^d of buyer b_i is less than the available time T_j of worker w_j; and 4) *Energy Constraint:* buyer b_i must also meet the energy requirement. Worker device then calculates the tasks' priority factor $\Psi_{i,j}$, execution time, and energy expenditure. Buyers in $\mathcal{B}_j$ are then sorted according to the descending order of $\Psi_{i,j}$. Finally, the worker greedily chooses the candidate buyers until it meets the resource, energy and maximum workload constraints. Next, we analyze the complexity of Algorithm

1. Short-listing buyer devices in statement 3 requires checking of m buyer devices in the worst case resulting in a computational complexity of $O(m)$. Similarly, the computational complexity of the statements 4 - 6 is $O(m)$. In statement 7, we use merge sort, whose computational complexity is $O(mlog(m))$ in worst case. Statements 8 - 16 are enclosed in a loop that iterates m times in the worst case having complexity of $O(m)$. Hence, the total time complexity of Algorithm 1 is $O(m + m + mlog(m) + m) \approx O(mlog(m))$.

After selecting candidate buyer devices using algorithm 1, the worker device determines its claimed cost for each of the buyer devices. To calculate the claimed cost, worker device adopts a critical payment [20] based policy where the cost is decided as the bid price of the next preferable buyer device. Algorithm 2 summarizes the steps of FEST-Auction payment algorithm. At first, the worker device determines the minimum bid price c_{min} from the winning bids, $\mathcal{C}_j$ (Line 1). In Line 2, we select the alternate buyers $\mathcal{B}''_j$ and their bids $\mathcal{C}'_j$ for worker w_j running Algorithm 1 with input $\mathcal{B} \setminus \mathcal{B}'_j$. Next we find the most priority bid c'_{min} of the buyer $b_k \in \mathcal{B}''_j$. Note that, c'_{min} is the critical bid as the buyer b_k has the highest priority of winning in absence of $\mathcal{B}'_j$. Finally, for each $b_i \in \mathcal{B}'_j$ worker sets its claimed cost $p^b_{i,j}$ using conditions in statements 5 - 10. Now we analyze the time complexity of Algorithm 2. Line 1 and 3 iterates at most m times individually having an worst-case complexity $O(m)$ for each. In Line 2, we run Algorithm 1 whose worst case complexity is $O(mlog(m))$. Statements 4 - 12 are enclosed in a loop that iterates at most m in the worst case wiht the time complexity of $O(m)$. Therefore, the overall time complexity of Algorithm 2 is

---

Algorithm 1 FEST-Auction at each worker $w_j \in \mathcal{W}$

Input: Set of the bids of all buyers $\mathcal{B}$
Output: Set of winning bid set $\mathcal{C}_j$, winning buyer set $\mathcal{B}'_j$, time required to execute task $\mathcal{T}_{i,j}$

1: $\mathcal{C}_j, \mathcal{B}'_j \leftarrow \emptyset$
2: $T, R, E, n_j \leftarrow 0,\ x'_{i,j} \leftarrow 0, \quad \forall i$
3: $\mathcal{B}_j = \{b_i \mid c_{i,j} \geq A_j,\ r_i \leq R_j,\ t^d_i \leq T_j,\ E_j + \Gamma \geq \mathcal{E}_i\ \forall b_i \in \mathcal{B}\}$
4: **for all** $b_i \in \mathcal{B}_j$ **do**
5:     Calculate $\mathcal{T}_{i,j}, \mathcal{E}_{i,j}$ and $\Psi_{i,j}$ using Eq. 12, 11 and 13, respectively
6: **end for**
7: Sort $\mathcal{B}_j$ in descending order of $\Psi_{i,j}$
8: **for all** $b_i \in \mathcal{B}_j$ **do**
9:     **if** $(r_i \leq (R_j - R)$ && $\mathcal{E}_{i,j} \leq (E_j - E)$ && $n_j < n^{max}_j$ **then**
10:         $R \leftarrow R + r_i,\ E \leftarrow E + \mathcal{E}_i,$
11:         $x'_{i,j} = 1,\ n_j \leftarrow n_j + 1$
12:         $\mathcal{C}_j \leftarrow \mathcal{C}_j \bigcup \{c_{i,j}\},\ \mathcal{B}'_j \leftarrow \mathcal{B}'_j \bigcup \{b_i\}$
13:     **else**
14:         Break
15:     **end if**
16: **end for**
17: **return** $\mathcal{C}_j, \mathcal{B}'_j, \mathcal{T}_{i,j}$

Algorithm 2 FEST-Auction Payment

Input: Winning buyers' set $\mathcal{B}'_j$ and bids' set $\mathcal{C}_j$

Output: Worker claimed cost $p^b_{i,j}$, $\forall i \in \mathcal{B}'_j$

$c_{min} \leftarrow \min_{c_{i,j} \in \mathcal{C}_j} c_{i,j}$

Find $\mathcal{B}''_j$ and $\mathcal{C}'_j$ using Algorithm 1 with input, $\mathcal{B} \setminus \mathcal{B}'_j$

$c'_{min} \leftarrow c_{k,j}, b_k = \arg\max_{b_i \in \mathcal{B}''_j} \Psi_{i,j}$

for all $b_i \in \mathcal{B}'_j$ **do**

    if $(c_{min} \leq c'_{min})$ **then**

        $p^b_{i,j} = c_{min}$

    else if $(c_{min} > c'_{min})$ **then**

        $p^b_{i,j} = c'_{min}$

    else if $(\mathcal{C}'_j == \emptyset)$ **then**

        $p^b_{i,j} = A_j$

    end if

end for

return $p^b_{i,j}$

$O(m + m + mlog(m) + m) \approx O(mlog(m))$.

4.3. Task Allocation to Workers

In FEST, each worker device $w_j \in \mathcal{W}$ selects the preferable buyer set, $\mathcal{B}'_j$ for those it is willing to provide computational services using the FEST-Auction described in section 4.2. The buyer devices are then notified about their winning bids. Note that a buyer device can win bids from multiple worker devices. However, in FEST, a buyer device recruits only one worker for a computation task. In the distributed FEST system, each buyer device $b_i \in \mathcal{B}$ greedily recruits a worker device to allocate the computational task that maximizes its utility. The selected worker device is paid after receiving its computational services using an incentive-driven payment policy, namely FEST-Payment.

To facilitate worker device recruitment and task allocation, each buyer device, $b_i \in \mathcal{B}$ defines the utility of a worker device, $w_j \in \mathcal{W}$ as an integrated metric, $U_{i,j}$ which is a weighted combination of three sub-metrics - $U^C_{i,j}$, $U^T_{i,j}$ and $\Omega^{fin}_{i,j}$ due to the worker recruitment cost, task completion time and worker device's reputation, respectively.

Calculation of $U^C_{i,j}$. Utility of a buyer b_i for a worker device w_j highly depends on its true valuation, $v_{i,j}$ of the unit resource required for the service and the price, $p^b_{i,j}$ charged by the worker for rendering that service. Note that the true cost $v_{i,j}$ is only known to the buyer itself. Now we calculate the cost-based utility of a buyer $b_i \in \mathcal{B}$ as follows,

$$U^C_{i.j} = 1 - \frac{p^b_{i,j}}{v_{i,j}}. \tag{14}$$

Here, $U^C_{i,j} \in [0,1]$, and from Eq. 14 it is clear that a worker charging less than the buyer's true valuation provides higher utility to the worker.

Calculation of $U^T_{i,j}$. Having a delay deadline, t^d_i for the computational task of the buyer device b_i, the utility of the buyer depends on the worker device's task completion time, $\mathcal{T}_{i,j}$. In FEST, a buyer device calculates this utility from the ratio of $\mathcal{T}_{i,j}$ and t^d_i as follows,

$$U^T_{i,j} = \begin{cases} e^{-(\frac{\mathcal{T}_{i.j}}{t^d_i})} & \mathcal{T}_{i,j} \leq t^d_i, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Here, $U^T_{i,j} \in [0,1]$. Note that, a lower the value of task execution time results in a higher utility for that worker. The exponent function in Eq. 15 decreases the utility value sharply when the execution time, $\mathcal{T}_{i.j}$ is close to t^d_i.

Our final metric which constitutes the utility is the worker device's reputation $\Omega^{fin}_{i,j}$. For the detailed calculation of $\Omega^{fin}_{i,j}$, refer to section 4.1. A worker device with a higher reputation value is expected to render the computational task with higher reliability and trustworthiness, thus it provides higher utility to the buyer. Now, the buyer device calculates a joint utility, $U^Q_{i,j}$ as the linear combination of $\Omega^{fin}_{i,j}$ and $U^T_{i,j}$ as follows,

$$U^Q_{i,j} = \mathcal{K} \times U^T_{i,j} + (1 - \mathcal{K}) \times \Omega^{fin}_{i,j}. \tag{16}$$

Here, $(0 \leq \mathcal{K} \leq 1)$ is a weight factor that is set locally according to the buyer's preference. Note that, $U^Q_{i,j}$ is an indicator metric for the task quality and Eq. 16 ensures its value to fall between 0 and 1. To bring a trade-off between the task quality and completion time, the buyer device calculates a final utility metric, $U_{i.j}$ as follows,

$$U_{i,j} = \omega \times U^C_{i,j} + (1 - \omega) \times U^Q_{i,j}, \tag{17}$$

where, $\omega \in [0,1]$ is a weight factor. By setting the value of ω to 1, the buyer device recruits the workers those render the computational service with minimum cost. On the other hand, setting $\omega = 0$ ensures the maximum task quality while recruiting the worker devices. The other values of ω makes a desired trade-off between the task quality and worker recruitment cost.

In Algorithm 3, we describe the detailed process of selecting the final worker from multiple workers. At first, each buyer calculates the utility $U_{i,j}$ from all of the candidate workers w_j from lines 2 - 4. Then from all of the candidate workers, we have selected the one as a final worker from which buyer b_i gets the largest utility and set the claimed cost of that worker as the winning price of buyer b_i. The complexity of Algorithm 3 is analyzed as follows: statement 1 and lines 2 - 4 are enclosed in a loop that iterates n times individually. For calculating $\sum_{w_j \in W} x'_{i,j}$ and $U_{i,j}$ from lines 5 - 9 require $O(n^2)$ time complexity. Therefore, the overall complexity of Algorithm 3 is $O(n + n + n^2) \approx O(n^2)$.

Algorithm 3 Task Allocation to Worker by Buyer $b_i \in \mathcal{B}$

Input: Price from all workers $p^b_{i,j}$
Output: Winning price of buyer per unit resource p^b_i
$p^b_i \leftarrow 0$, $x_{i,j} \leftarrow 0$, $\forall j$
for all $w_j \in \mathcal{W}$ **do**
  Calculate $U_{i,j}$ using Eq. 17
end for
if $\sum_{w_j \in W} x'_{i,j} \geq 1$ **then**
  $j* = \arg\max_{j \in W}\{U_{i,j} \mid x'_{i,j} = 1\}$
  set $x_{i,j*} = 1$ and $x_{i,j} = 0$ for others j where $j \neq j*$
  $p^b_i = p^b_{i,j*}$, where $x_{i,j*} = 1$
end if
return p^b_i

4.4. Worker Payment Policy

After getting the final recruitment notification from a buyer device $b_i \in \mathcal{B}$, a worker device $w_j \in \mathcal{W}$ executes the computational task and sends back the result to the buyer device. Then buyer device then access the task quality and pay the workers based on the assessment result. In FEST, we evaluate the task quality of the worker device using a metric, $q_{i,j}$ similar to [41] namely quality indicator as follows,

$$q_{i,j} = \Gamma_{i,j} - \Gamma_{i*}, \tag{18}$$

where, $\Gamma_{i,j}$ and Γ_{i*} denote the offered task quality and true task quality. Note that, the value of Γ_{i*} can be estimated using truth discovery method proposed in [42].

In FEST, the buyer devises a payment policy similar to [43] namely FEST-Payment which rewards a worker device for rendering good quality service whereas penalizes it for failing to provide the required task quality. Such a payment policy incentivizes the workers to join the system with good contribution. According to FEST-Payment policy, the payment, $P_{i.j}$ of a worker device, $w_j \in \mathcal{W}$ for performing the computational task of the buyer, b_i is calculated using the following equation,

$$P_{i,j} = \begin{cases} p^b_i + \varphi & q_{i,j} \leq q_{max} \,\&\&\, t^c_i < t^d_i, \\ \varepsilon \times p^b_i & q_{i,j} > q_{max}, \\ p^b_i & \text{otherwise}, \end{cases} \tag{19}$$

where, q_{max} denotes the maximum allowable deviation of the task quality and t^c_i indicates the task completion time. The value of q_{max} and t^c_i are dependent on the applications' nature which is set by the buyer itself. In case, $q_{i,j} \leq q_{max}$ and $t^c_i < t^d_i$, the worker w_j provides high task quality as well as high QoE to the buyer, b_i. Thus φ amount of incentive is added to the actual cost of worker. On the another case, $q_{i,j} > q_{max}$ indicates that worker w_j provides low data quality which ends up penalizing the worker by a factor of $\varepsilon \in [0, 1]$. Otherwise, the worker is provided with its actual cost. The incentive amount, φ can be calculated using the approach similar to [41] as follows,

$$\varphi = (v_{i,j} - p_i^b) \times r_i \times \{1 - e^{(q_{i,j} - q_{max}) \times \theta_1}\} \tag{20}$$

Again, we calculate the penalty factor ε using the following equation,

$$\varepsilon = e^{(q_{i,j} - q_{max}) \times \theta_2}, \tag{21}$$

where, θ_1 and θ_2 are the amplification factors that amplify the effect of the quality deviation $q_{i,j}$ of a task from the maximum allowable deviation. Here we use the exponential function in Eq. 20 and 21 to create a sharp change in the reward and penalty, respectively, for a higher quality deviation. The higher values of θ_2 and θ_1 provide a higher penalty and lower reward, respectively, on the workers' claimed price. Finally, buyers also update the reputation of the workers considering their interaction experience using Eq. 4 to 6.

4.5. An Illustrative Example of FEST

In this section, we provide an example to illustrate the procedure of the proposed FEST-Auction mechanism. Let consider a small snapshot of the MDC environment consisting of five workers and four buyer devices denoted by the sets $\{w_1, w_2, \cdots, w_5\}$ and $\{b_1, b_2, \cdots, b_4\}$, respectively. The workers' available resources, asking prices and maximum workload, are denoted by the vectors [4, 5, 6, 8, 6], [4, 3, 4, 3, 4], and [2, 1, 3, 2, 2], respectively. The buyers' bids, along with the resource requirement, task priority, and task execution time, are listed in Table 2. For simplicity, we assume that each worker's availability time is higher than the corresponding tasks' delay deadline as well as each worker device has sufficient energy to complete its workload. We assume that all workers are available for 6 seconds each and tasks' delay deadlines are 3, 4, 5, and 3 seconds, respectively.

FEST-Auction at each worker $w_j \in \mathcal{W}$. the FEST-Auction mechanism at each worker acts as follows:

- w_1: $\mathcal{B}_1 = \{b_1, b_3, b_4\}$. After sorting according to $\Psi_{i,j}$, $\mathcal{B}_1 = \{b_3, b_1, b_4\}$. After considering resource requirement R_1 and maximum number of winning bids n_1^{max}, $\mathcal{B}'_1 = \{b_3\}$, $\mathcal{C}_1 = \{5\}$ and $x'_{3,1} = 1$. Here, $c_{min} = 5$ and $c'_{min} = 4$. Then we get $p^b_{3,1} = 4$ and $\mathcal{T}_{3,1} = 2.5$.
- w_2: After sorting, $\mathcal{B}_2 = \{b_1, b_4\}$. Then after considering R_2 and n_2^{max}, $\mathcal{B}'_2 = \{b_1\}$, $\mathcal{C}_2 = \{6\}$ and $x'_{1,2} = 1$, Here, $c_{min} = 6$ and $c'_{min} = 5$. Then we get $p^b_{1,2} = 5$ and $\mathcal{T}_{1,2} = 2.5$.
- w_3: After sorting, $\mathcal{B}_3 = \{b_3, b_2\}$. Then after considering R_3 and n_3^{max}, $\mathcal{B}'_3 = \{b_3, b_2\}$, $\mathcal{C}_3 = \{4, 5\}$ and $x'_{3,3} = x'_{2,3} = 1$. Here, $c_{min} = 4$ and $c'_{min} = 5$. As $c_{min} < c'_{min}$. Therefore, we get $p^b_{3,3} = 4$, $p^b_{2,3} = 4$ and $\mathcal{T}_{3,3} = 1.8, \mathcal{T}_{2,3} = 2.3$.
- w_4: After sorting, $\mathcal{B}_4 = \{b_1, b_4, b_2\}$. Then after considering R_4 and n_4^{max}, $\mathcal{B}'_4 = \{b_1, b_4\}$, $\mathcal{C}_4 = \{5, 5\}$ and $x'_{1,4} = x'_{4,4} = 1$. Here, $c_{min} = 5$ and $c'_{min} = 4$. Therefore, we get $p^b_{1,4} = 4$, $p^b_{4,4} = 4$ and $\mathcal{T}_{1,4} = 2.5$, $\mathcal{T}_{4,4} = 2.5$.

Table 2: An illustrative example of auction mechanism

	Required Resource	w_1 Bid	w_1 Priority	w_1 Time	w_2 Bid	w_2 Priority	w_2 Time	w_3 Bid	w_3 Priority	w_3 Time	w_4 Bid	w_4 Priority	w_4 Time	w_5 Bid	w_5 Priority	w_5 Time
b_1	2	4	0.7	2	6	0.8	2.5	0	0	0	5	0.8	2.5	0	0	0
b_2	3	0	0	0	0	0	0	5	0.7	2.3	4	0.6	3	4	0.8	2.5
b_3	3	5	0.9	2.5	0	0	0	4	0.9	1.8	0	0	0	5	0.9	2
b_4	2	5	0.6	4	5	0.4	3.5	5	0.6	3.5	5	0.5	2.5	3	0.3	2

- w_5: After sorting, $\mathcal{B}_5 = \{b_3, b_2\}$. Then after considering R_5 and n_5^{max}, $\mathcal{B}'_5 = \{b_3, b_2\}$, $\mathcal{C}_5 = \{5, 4\}$ and $x'_{3,5} = x'_{2,5} = 1$. Here, $c_{min} = 4$ and $\mathcal{C}'_5 = \emptyset$. Therefore, we get $p^b_{3,5} = 4, p^b_{2,5} = 4$ and $\mathcal{T}_{3,5} = 2, \mathcal{T}_{2,5} = 2.5$.

Table 3: Worker's Reputation

	w_1	w_2	w_3	w_4	w_5
b_1	0.85	0.8	0.7	0.5	0.52
b_2	0.71	0.6	0.4	0.85	0.85
b_3	0.75	0.65	0.45	0.75	0.55
b_4	0.6	0.55	0.8	0.65	0.7

Task allocation to worker by each buyer $b_i \in \mathcal{B}$. Each buyer $b_i \in \mathcal{B}$ also determine final worker locally. We have used reputation value $\Omega^{fin}_{i,j}$ according to the Table 3.

- b_1: As, $\sum_{w_j \in W} x'_{1,j} > 1$, we should calculate the utility of b_1 from all possible workers $\{w_2,\, w_4\}$ using $U^C_{i,j}$ and $U^Q_{i,j}$. Let b_1 selects the value of weight parameter $\mathcal{K}$ and ω is 0.3 and 0.7. Then we find $U_{1,2} = 0.33$ and $U_{1,4} = 0.28$. As b_1 gets highest utility from worker w_2, therefore, w_2 is selected as the final worker by b_1. Hence $x_{1,2} = 1$ and $x_{1,j} = 0$, $\forall j \neq 2$.

- b_2: It is noted that $\sum_{w_j \in W} x'_{2,j} > 1$ and b_2 has winning buyer candidates $\{w_3,\, w_5\}$. Hence, we should calculate the utility of b_2 from all possible workers using $U^C_{i,j}$ and $U^Q_{i,j}$. Let b_2 selects the value of weight parameter $\mathcal{K}$ and ω is 0.6 and 0.4. Then we find $U_{2,3} = 0.38$ and $U_{2,5} = 0.41$. As b_2 achieves best utility from worker w_5, therefore, w_5 is the final worker selected by b_2. Hence $x_{2,5} = 1$ and $x_{2,j} = 0$, $\forall j \neq 5$.

- b_3: It is noted that $\sum_{w_j \in W} x'_{3,j} > 1$ and b_3 has winning buyer candidates $\{w_1, w_3,\, w_5\}$. Hence, we should calculate the utility of b_3 using $U^C_{i,j}$ and $U^Q_{i,j}$. We have considered b_3 set $\mathcal{K} = 0.4$ and $\omega = 0.4$. Then we find $U_{3,1} = 0.49$, $U_{3,3} = 0.33$ and $U_{5,3} = 0.44$. As

b_3 achieves more utility from worker w_1 than w_3 and w_5, therefore, w_1 is chosen as the final worker of b_3. Hence $x_{3,1} = 1$ and $x_{3,j} = 0$, $\forall j \neq 1$.

- b_4: It is noted that $\sum_{w_j \in W} x'_{4,j} = 1$ and w_4 is the only winning buyer candidate of b_4. Hence, we should calculate the utility of b_3 using $U^C_{i,j}$. We find $U_{4,4} = 0.2$ and $x_{4,4} = 1$. While $x_{4,j} = 0$, for $\forall j \neq 4$.

Therefore, using our proposed mechanism the successful trading pairs are (b_1, w_2), (b_2, w_5), (b_3, w_1) and (b_4, w_4) with unit resource price 5, 4, 4 and 4, respectively.

4.6. Theoretical Analysis

In this section, we theoretically analyze the FEST-Auction's desirable properties: computationally efficiency, individually rationality, truthfulness, and budget balance.

Theorem 1. *Proposed FEST-Auction is computationally efficient.*

Proof. The proof of this theorem requires the analysis of the time complexity of Algorithm 1 to 3. The worst-case time complexity of Algorithm 1 and 2 is $O(mlog(m))$ as analyzed in Section 4.2. Again, the time complexity of Algorithm 3 is $O(n^2)$ in worst case as detailed in Section 4.3. Therefore, the proposed FEST-Auction is solvable in polynomial time and thus completes the proof. □

Theorem 2. *FEST-Auction is individually rational.*

Proof. In FEST, we determine the payment of FEST-Auction winners payment using the critical payment policy as detailed in Algorithm 2. According to this policy, a winning buyer's payment is determined as its critical payment, i.e., the minimum of c_{min} and c'_{min} in case of $\mathcal{C}'_j \neq \emptyset$, otherwise its asking price is paid. We have considered the payment gained by worker w_j is equal to the claimed cost i.e., $p^b_{i,j} = p^s_{i,j}$. Thus according to the used payment policy, three cases arise while determining the payment of a winning buyer device:

- *Case 1* $(c_{min} \leq c'_{min})$: In this case, c_{min} is selected as the winning price/payment of the buyer/worker, where $c_{i,j} \geq c_{min} = p^b_i$ and $A_j \leq c_{min} = p^s_{i,j}$.
- *Case 2* $(c_{min} > c'_{min})$: In this case, c'_{min} is set as the winning price that need to be paid by the winning buyers of worker w_j, where $c_{i,j} \geq c'_{min} = p^b_i$ and $A_j \leq c'_{min} = p^s_{i,j}$.
- *Case 3* : In this case, there is no alternate winning bid i.e., $\mathcal{C}'_j = \emptyset$, thus its payment is kept equal to its asking price A_j, where $c_{i,j} \geq A_j = p^b_i = p^s_{i,j}$.

From the above discussion it is clear that a buyer device, b_i is never charged a higher price than its bid amount. Similarly, a worker w_j is never provided a lower payment than its asking price A_j. Hence, the proposed FEST-Auction is individually rational to both the buyer and worker devices. □

Theorem 3. *FEST-Auction is truthful to the buyer and worker devices.*

The proof of the Theorem 3 requires the satisfaction of the Lemma 1 and 2 for the buyer and worker devices, respectively.

Lemma 1. *The FEST-Auction is truthful for the buyer devices.*

Proof. To proof this Lemma, we divide the buyer set, $\mathcal{B}$ into two subsets: $\mathcal{B}'_j \neq \emptyset$ and $\mathcal{B} \setminus \mathcal{B}'_j$.

Case 1 ($b_i \in \bigcup_{w_j \in \mathcal{W}} \mathcal{B}'_j$). : Buyer $b_i \in \mathcal{B}$ has at least one winning bid. We assume that buyer b_i wins its bid truthfully at worker w_j. We now consider the cases where the buyer, b_i submits its bids untruthfully. Let, $\overline{U}_{i,j}$ and $\overline{p}^b_i$ denote the buyer's utility and charged price for untruthful bidding.

- Buyer, b_i loses at worker w_j. Therefore, it has $\overline{U}_{i,j} = 0$ and $\overline{U}_{i.j} \leq U_{i,j}$.
- Buyer, b_i still wins at worker w_j. According to Algorithm 2, the buyer's payment is determined as an amount equal to c_{min} or, c'_{min} or A_j, which is independent of its submitted bids. In that case, $\overline{U}_{i.j} = U_{i,j}$.
- Now, we take into consideration that buyer b_i also wins its bid at different worker $w_{j'}$, where $j \neq j'$. Therefore the following cases may arise:
  - In the first case, $x'_{i,j'} = 1$ and b_i submits truthful bids, i.e., buyer b_i also gets winning notification from the worker w'_j. However, the final worker is w_j selected by b_i instead of w'_j because $U_{i,j} \geq \overline{U}_{i,j'}$. The cost based utility $\overline{U}^C_{i,j'}$ can be higher than $U^C_{i,j}$, however, the overall utility $U_{i,j}$ is the highest due to the task quality and reputation of the worker w_j.
  - In the second case, $x'_{i,j'} = 0$, which means b_i can't get winning bid notification from any of the other workers. Thus, we need to consider the following cases.
    * In this case $b_i \notin \mathcal{B}_{j'}$ i.e., $c_{i,j'} = v_{i,j'} < A_{j'}$. However, when b_i wins at worker $w_{j'}$ with untruthful bids, it needs to pay $\overline{p}^b_i$, where the minimum value of $\overline{p}^b_i$ is $A_{j'}$. Then we get the cost based utility $\overline{U}^C_{i,j'} = (v_{i,j'} - A_{j'}) \times r_i \leq 0$. As $\overline{U}_{i,j'}$ depends on $\overline{U}^C_{i,j'}$, we can conclude that $\overline{U}_{i,j'} \leq U_{i,j}$.
    * In the another case, $b_i \in \mathcal{B}_{j'}$, but, $b_i \notin \mathcal{B}'_{j'}$. This case happens when the value of $\Psi_{i,j'}$ is very low. Nevertheless, when b_i wins at worker $w_{j'}$ with untruthful bids, it needs to pay c_{min} or c'_{min}. It is possible that $v_{i,j'} \geq c_{min}$ or $v_{i,j'} \geq c'_{min}$ which indicates that $\overline{U}^C_{i,j'} \geq 0$ or $\overline{U}^C_{i,j'} \geq U^C_{i,j}$. However, the total utility due to untruthful bid $\overline{U}_{i,j'}$ is less than $U_{i,j}$ i.e., $\overline{U}_{i,j'} \leq U_{i,j}$.

Case 2 ($b_i \notin \bigcup_{w_j \in \mathcal{W}} \mathcal{B}'_j$). : Buyer b_i can't win at any of the worker with truthful bid, i.e., $U_{i,j} = 0$, $\forall w_j \in \mathcal{W}$. We now consider the cases when b_i submit bids untruthfully to worker w_j.

- Buyer b_i still loses at worker w_j. Therefore, $\overline{U}_{i,j} = U_{i,j} = 0$.

- Buyer b_i wins at worker w_j by submitting untruthful bids. The main reason to lose at worker w_j with truthful bids.
  - $v_{i,j} < A_j$: In that case, b_i obviously can't win at worker w_j. However, if buyer b_i wants to win at w_j with untruthful bids $c_{i,j}$, it must submit bid higher than A_j. In that case, the value of $\overline{U}^C_{i,j} \leq 0$ which also effects the total utility $\overline{U}_{i,j}$, where $\overline{U}_{i,j} \leq U_{i,j}$.
  - $v_{i,j} \geq A_j$: In that case $b_i \in \mathcal{B}_j$ but, $b_i \notin \mathcal{B}'_j$. The main reason to lose at worker w_j is the low priority of the task. If buyer b_i wants to win at w_j, it needs to bid untruthfully. It may increase the value of $\overline{U}^C_{i,j}$, however, the overall utility $\overline{U}_{i,j}$ is less than $U_{i,j}$ due to the time based utility i.e., $\overline{U}_{i.j} \leq U_{i,j}$.

In brief, submitting untruthful bids can't enhance the utility of the buyers which implies that a buyer device can only maximize its utility by truthful bidding. □

Lemma 2. *FEST-auction is truthful for workers.*

Proof. For simplicity, we consider $\mathcal{W}'_i$ be the worker to those the buyer b_i has a winning bid. We now divide the worker set $\mathcal{W}$ into two sub-sets: $\mathcal{W}'_i$ and $\mathcal{W} \setminus \mathcal{W}'_i$. To facilitate this proof, we also define the worker utility, $U^w_{i,j}$ as the difference between the worker payment and its true ask price. Let, $\overline{U}^w_{i,j}$ and $\overline{A}_j$ denote the utility and minimum asking price in case of untruthful information.

1. For worker $w_j \in \bigcup_{b_i \in \mathcal{B}} \mathcal{W}'_i$: When worker w_j asks untruthfully, there are two cases that need to be considered.
   - All buyers $b_i \in \mathcal{B}$ loss at worker w_j. Therefore, $\overline{U}^w_{i,j} = U^w_{i,j} = 0$.
   - $\mathcal{B}'_j \neq \emptyset$, i.e, some buyers $b_i \in \mathcal{B}$ wins at worker w_j with untruthful ask $\overline{A}_j$, where $\overline{A}_j > A_j$. However, This worker can't be selected as the final worker of any buyer due to the higher ask of worker w_j.
2. For buyer $w_j \notin \bigcup_{b_i \in \mathcal{B}} \mathcal{W}'_i$: In this case, worker w_j is not selected by any of the buyer b_i as the final worker by submitted truthful bids. There are two cases that need to be considered, when worker w_j provides untruthful asks.
   - All buyers still loses at worker w_j. Hence, we have $\overline{U}^w_{i,j} = U^w_{i,j} = 0$.
   - Assume, worker w_j decreases its minimum ask to win by some buyers. However, due to the true cost is higher than the minimum ask of w_j, worker w_j gets the utility $\overline{U}^w_{i,j} \leq 0$.

Therefore, it is shown that with only truthful ask, a worker gets maximum utility. □

Theorem 4. *FEST-auction is budget balanced.*

Proof. As shown in Algorithm 2, $x_{i,j} = 1$ if the buyer device $b_i \in \mathcal{B}$ wins the FEST-Auction ran at the worker device $w_j \in \mathcal{W}$. Then the unit resource cost, p_i^b paid by the buyer is equal to the payment, p_i^b received by the worker. Therefore, for a system consisting of m buyers and n worker devices it can easily be shown that,

$$\sum_{i=1}^{m} x_{i,j} \times r_i \times p_i^b = \sum_{j=1}^{n} x_{i,j} \times r_i \times p_{i,j}^s \tag{22}$$

Hence, we can say the proposed FEST-Auction is budget balanced for both of the buyers and workers.

□

4.7. Discussion

In this section, we discuss on the chosen values of the weight parameters ρ, η, κ, ζ and γ. In this work, we mainly focus on the design of MWSL and FL based auction model in the distributed MDC environment. Formulating the exact value of the weight parameters in a dynamic manner, helps the buyers to calculate the reputation of the workers more precisely. In FEST, we have gone through numerous simulation experiments and identify that for the following value, FEST system provides us better result. Therefore, we set $\rho = 0.3$, $\eta = 0.7$, $\kappa = 0.7$, $\zeta = 0.3$ and $\gamma = 1$ for all of the experiments. However, finding the optimal value of these parameters require extensive analysis and mathematical modeling which depends on the size of network, number of the buyers and workers, which is kept as a future work.

5. Performance Evaluation

In this section, we have implemented our proposed FEST system using MATLAB and compare the performances with two state-of-art-works: TIM [21] and DTAM [22].

5.1. Simulation Environment

We assume that buyers and workers are scattered using random distribution in a 100×100 m^2 area. The arrivals of buyers and workers' devices are generated using poisson distribution. Buyers and workers are connected through a wireless local area network (WLAN). Bidding price per unit resource of each buyer is selected using uniform random distribution from 3 to 10 units. We consider the compute-intensive applications like image processing, face recognition, and text translation where each application's size requires 3 - 10 GHz CPU cycles. As we have considered real-time applications; therefore, we consider that the deadline of tasks is $500 - 1500$ ms [44]. To calculate the workers' availability time in an area, we have used the smooth random mobility model [35]. Each experiment is done for 1000 sec, and each graph data point is plotted from the average results of 50 simulation runs with different random seed values. Table 4 shows all of the performance parameters and their ranges that are used in various experiments. All of the simulation experiments are run on a machine having 2.8 GHz CPU and 8 GB memory.

Table 4: Simulation Parameters

Parameter	**Value**
Simulation area	$100 \times 100m^2$
Number of buyers	10 - 60
Number of workers	10 - 60
Resource required by each buyer	3 - 10 GHz
Available resource at each worker	20 - 40 GHz
buyer's bid per unit resource	3 - 10 units
Worker's unit resource cost	3 - 10 units
Task deadline of buyers	500 - 1500 ms
Task completion time	300 - 2000 ms
Simulation time	1000 sec
Weight parameters	$\rho = 0.3$, $\eta = 0.7$, $\kappa = 0.7$, $\zeta = 0.3$

5.2. Performance Metrics

To analyze the performance of our proposed FEST system, we have used the following performance metrics:

- *Quality-of-Experience (QoE)* of the buyers is calculated as the inverse ratio of task completion time and deadline of the task. It helps us to measure how fast a buyer gets service from a worker.

- *Average utility of buyers* indicates the utility achieved by a worker from a reliable worker for the successful completion of a task. A higher value indicates that the buyer is more satisfied with the received result of the task.

- *Task completion Ratio* denotes the number of successful trades of the buyers. It is calculated as the ratio of the total number of winning buyers and the total number of buyers in the system. A higher value indicates that more number of tasks is allocated to some worker mobile devices.

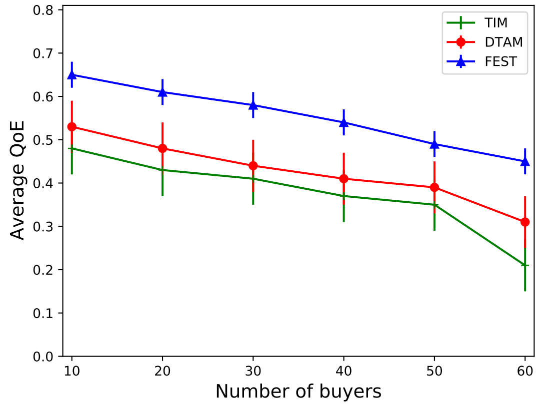
- *Average payment of workers* denotes the amount of payment that a worker gets for the successful execution of a task.

5.3. Simulation Result

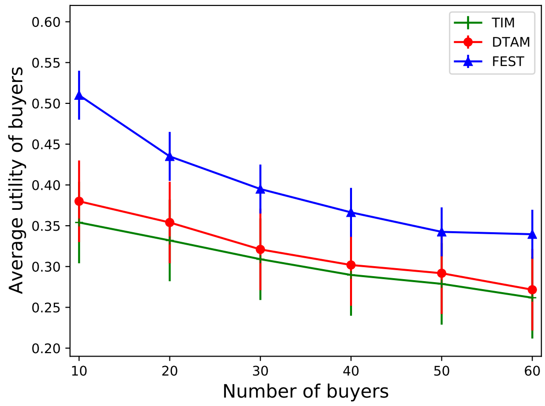
In this section, we have discussed the results of the performance of our proposed FEST model by varying the number or buyers and workers. We also investigate the performance of the proposed FEST-system on the individual rationality and truthfulness.

5.3.1. Impact of Varying the Number of Buyers

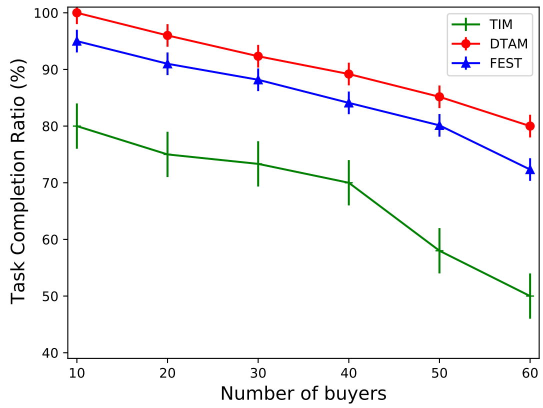
In this experiment, we vary the total number of buyers of the system by keeping the number of workers fixed at 30.



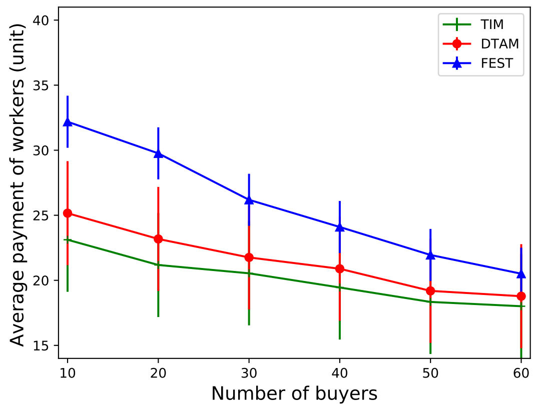
(a) Quality-of-Experience (QoE) of buyers



(b) Average utility of buyers



(c) Task completion ratio

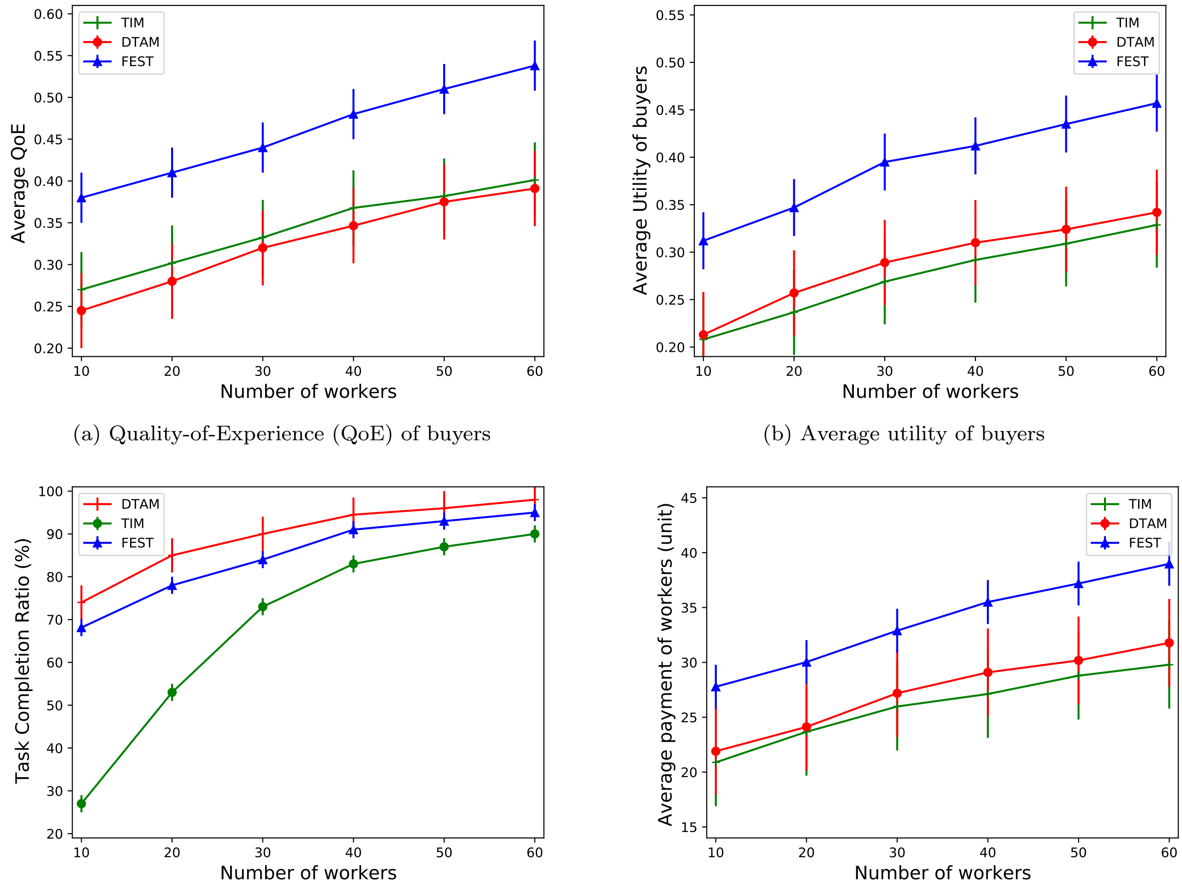


(d) Average payment of workers

Figure 3: Impacts of varying the number of buyers

Fig. 3(a) shows that increasing the number of buyers, QoE of each buyer, is decreased. The rationale behind it is that buyers get service slowly due to a limited number of workers with the increment number of buyers because some workers need to provide service to multiple buyers. However, FEST performs better than TIM and DTAM. Because they haven't considered QoE of the buyers, they have allocated a task to a worker by considering only the resource availability. In contrast, our system FEST has selected the most suitable buyer by considering the QoE and reputation of the workers, which develops the knowledge using MWSL and federated learning. For that reason, FEST performs better in terms of QoE.

Increasing the number of buyers, the average utility of each buyer is decreased that is shown in Fig. 3(b). This is because many buyers fail to select the reliable workers from which a buyer get achieves higher utility because of the limited number of workers in the system. However, FEST performs better because TIM and TDAM don't consider workers' reputation and QoE during the final worker selection. Sometimes buyers select the unreliable workers, which decreases the utility of the buyers. FEST incorporates worker reputation using the knowledge of MWSL and FL as well as QoE into the utility while selecting the final worker.



(a) Quality-of-Experience (QoE) of buyers (b) Average utility of buyers

(c) Task completion ratio (d) Average payment of workers

Figure 4: Impacts of varying the number of workers

Therefore, our proposed FEST outperforms in this case.

From Fig. 3(c), we observe that increasing the number of workers, the task completion ratio is decreased. Due to resource and energy constraints and the limited number of workers in the system, some tasks are failed to allocated to the appropriate worker mobile devices for a higher number of buyers in the system. DTAM performs better in that case because it has not considered the energy constraint and worker's preference when allocating the tasks. On the other hand, TIM is only applicable to the homogeneous system and discusses one-to-one trading between buyers and workers. For that reason, the task completion ratio in TIM is abysmal.

Average payment of workers for successful task completion by increasing the number of buyers for all studied systems is depicted in Fig. 3(d). Here we observe that average payment is decreased with the increasing number of buyers. However, the payment is higher in our proposed FEST system than TIM and DTAM. The reason behind the fact is that the FEST system provides not only actual payment but also rewards to the buyers to provide higher data quality and penalize the workers for poor sensing quality.

5.3.2. Impact of Varying the Number of Workers

In this experiment, we vary the number of workers of the system by keeping the number of buyers fixed at 30. Increasing the number of workers allows the buyers to select the more reliable and suitable worker, resulting in better performance depicted in Fig. 4.

With the increasing number of workers, QoE of the buyers is also increased, as shown in Fig. 4(a). The reason behind this shape is articulated by the fact that is having more available workers give the buyers more option to select the more reliable workers from which a buyer can get service within a short amount of time which in turn increase the QoE. TIM and DTAM allocate the task to a worker by considering only the resource capacity and bidding cost. They haven't considered the QoE of the buyers. Therefore, FEST outperforms in terms of QoE. For a similar reason, the average utility of buyers is also enhanced with the increasing number of workers in the system depicted in Fig. 4(b). FEST performs better because we have considered reputation and QoE when calculating utility. Our reputation scheme is more sensitive due to the knowledge from MWSL and FL, which helps the buyers select the most reliable worker and increase the utility.

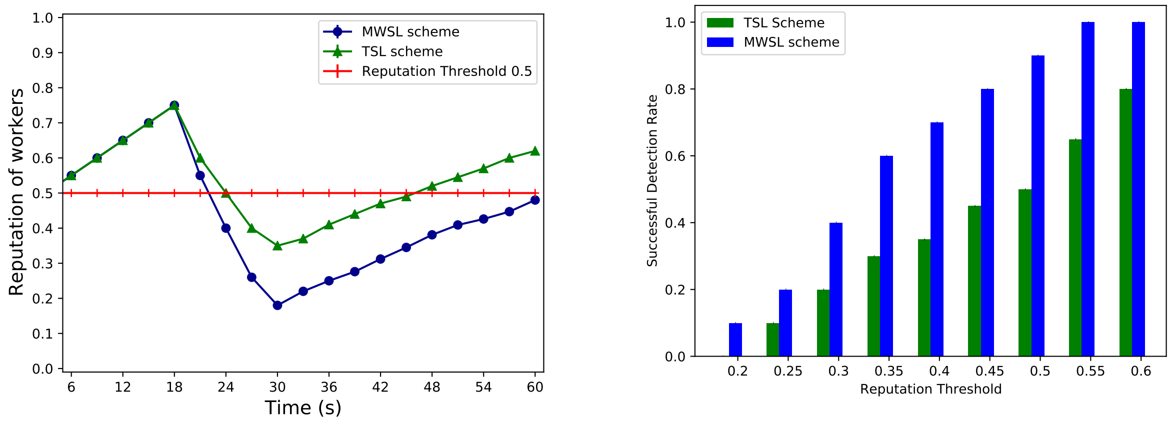
From Fig. 4(c) we observe that with the increasing number of workers, the task completion ratio is increased. The reason behind the shape is that more workers mean more available resources in the system, which helps more buyers be allocated to some workers. TIM performs worst in that case because they restrict that each worker can provide service to only one buyer, while DTAM and our proposed FEST system, workers can assist as long as they have enough resources. DTAM performs best in this case because they allocate the buyers without considering the energy level and task preference of the worker devices.

The average payment of workers is improved with the increasing number of workers shown in Fig. 4(d). Due to the limited number of buyers, the workers can provide services to buyers with task delay deadline. The probability of selecting more reliable workers is also enhanced in that case. It helps the buyers get high-quality tasks and workers to get rewards along with winning price due to successful task execution. TIM and DTAM do not provide any reward to the workers for providing higher QoE and successful task execution. Therefore, the FEST system outperforms in terms of the average payment of workers.

5.3.3. Impact of Worker Reputation

In this experiment, we have analyzed the effects of the reputation of the workers in FEST using the Multi-weight Subjective Logic (MWSL) scheme and Traditional Subjective Logic (TSL) scheme.

In Fig. 5(a), we have shown the effects of the reputation of the worker in the FEST system for 60 seconds. Here, we have set the reputation threshold 0.5 and the initial reputation value of the workers 0.5. We assume that at first unreliable workers pretend to behave well intentionally to increase the reputation value. After that, these unreliable workers start to provide data to the buyers with poor quality, which decreases the reputation value below the threshold value. In the MWSL scheme, this reputation is falling faster than TSL. If these workers try to behave well intentionally again, it takes a higher time to increase the reputation above the threshold value in the MWSL scheme than the TSL scheme. This is because of the weights of interaction effects, freshness, frequency, and federated learning,



(a) Reputation value of workers

(b) Detection rate of untruthful workers under different threshold

Figure 5: Impacts of Reputation in MWSL scheme

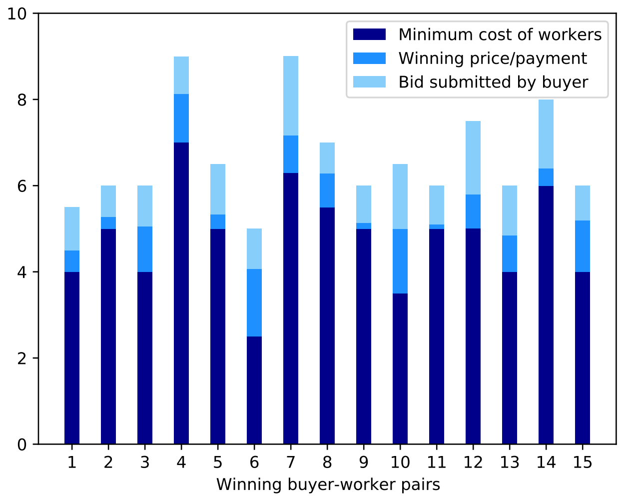
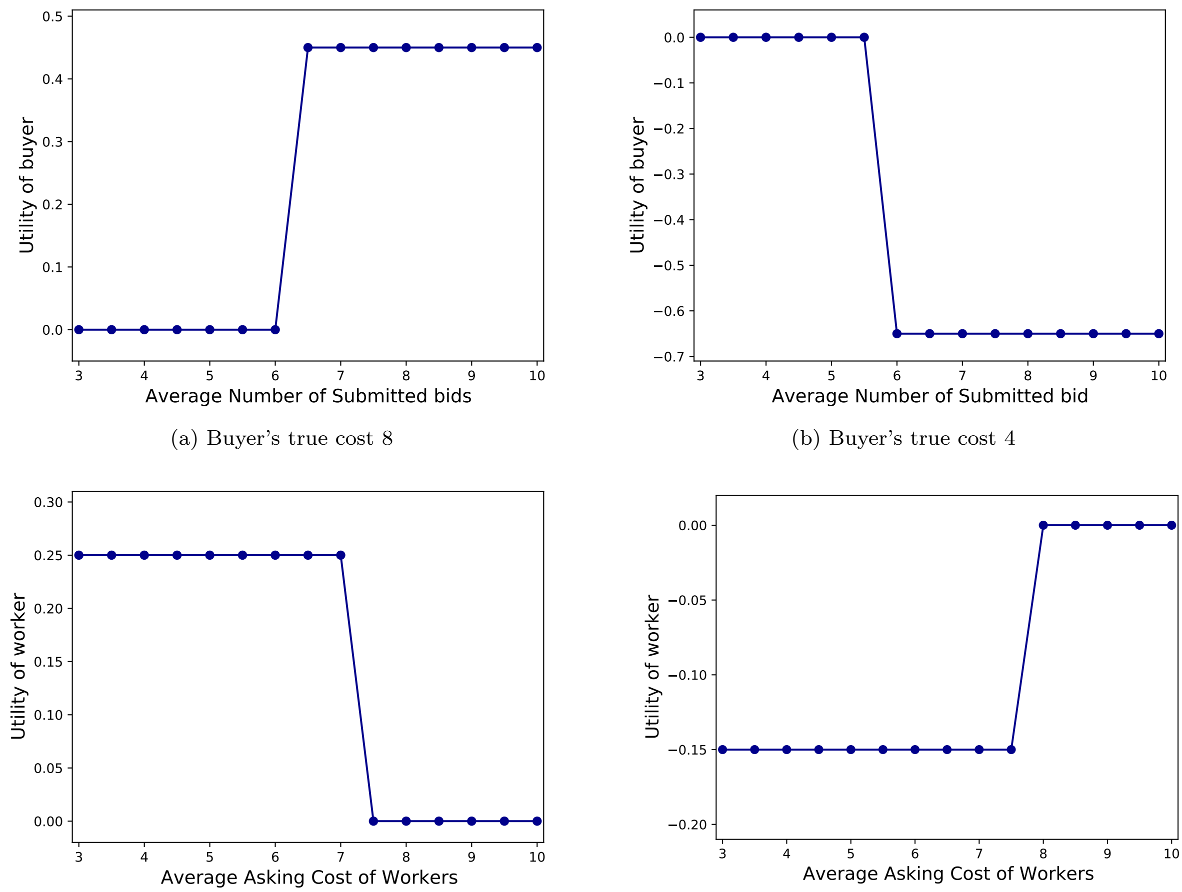


Figure 6: Performance of individual rationality

which discourage the worker from doing compromised works. Therefore, our reputation scheme in FEST is more accurate and sensible for reputation calculation.

From Fig. 5(b), we observe that in MWSL, the successful detection rate of the unreliable workers is much higher than in the TSL scheme. We have seen that when the reputation threshold is 0.5, the detection rate is almost 90% higher than the TSL scheme. This is due to the weight of the several parameters that help the FEST system prevent the workers' unreliable activities more accurately. Also, note that, in FEST-auction, computational resources are allocated to the worker devices based on their reputation as a worker with a higher reputation value is expected to complete the assigned task with higher reliability. Thus, MWSL and federated learning (FL) based reputation scheme ensure the utilization



(a) Buyer's true cost 8

(b) Buyer's true cost 4

(c) Worker's true cost 6

(d) Worker's true cost 8

Figure 7: Impacts of truthfulness of buyers and workers

of reliable workers' resources to increase the task quality.

5.3.4. Impact on Individual Rationality

In this experiment, we have analyzed the performance of our proposed system on individual rationality. Here we have set the number of buyers and workers 30 and 30, respectively. Among the winning buyers and their corresponding workers, we choose 15 pairs of them. For each pair, we have shown the bid submitted by the buyer, the minimum cost to execute a worker's task, the winning price that is needed to be given to a worker by a buyer is shown in Fig. 6. From that graph, we observe that the final price paid by a buyer is no more than the submitted bid by the buyer as well as the payment earned by each worker is no less than the minimum asking cost of each worker. Therefore, individual rationality is achieved by both buyers and workers by FEST system.

5.3.5. Impact of Truthfulness of Buyers and Workers

In this experiment, we have randomly selected the several buyers and workers pairs and observe how their utilities change by varying bids of the buyers and minimum ask of the workers.

In Fig. 7(a), we have set the value of the true cost $v_{i,j}$ is 8 units. However, buyer b_i can submit many untruthful bid that is shown on the x-axis. If buyer b_i submits a very low bid than the true cost, it can't win at worker w_j due to the minimum asking price of that worker. For that reason, the utility of buyer b_i at worker w_j is 0. By submitting truthful bid buyer b_i achieves utility $U_{i,j} = 0.45$. From this graph, we also observe that submitting untruthful bids can't increase buyer's utility b_i than submitting truthful bids. In Fig. 7(b), we have shown the case, when b_i loses at worker w_j with the true cost 4 units. From that graph, we observe that by submitting an untruthful bid higher than four units buyer b_i can win at worker w_j, however, the utility obtained at worker w_j is less than the losing at worker w_j.

Fig. 7(c) shows that, worker w_j achieves maximum utility when its true minimum ask 6 units. However, worker w_j may set many untruthful ask that is shown in the x-axis. From this graph, we also observe that increasing the ask of worker w_j untruthfully achieves no utility by worker w_j, which indicates that any buyer can't win at w_j after increasing the worker's ask. In Fig. 7(d), we have shown the case when worker w_j can't be selected as the final worker by any buyer when w_j sets the true minimum cost eight units. When worker w_j decrease the true ask for selecting as the final worker from any buyer, it gets negative utility. Therefore, from those graphs, we can say that bidding or asking untruthfully by buyers or workers, the utility can't be maximized then the truthful case in the FEST system.

6. CONCLUSION

In this paper, we proposed a distributed task allocation framework namely FEST for the mobile device cloud applications. A distributed auction mechanism was devised for the workers to select candidate buyer devices who make the final decision on computational task allocation. Federated learning on subjective logic-based worker reputation proved effective and resulted in higher utility gain of the buyer devices. Simulation results revealed that the FEST system can achieve performance improvement as high as 30% and 25% in terms of QoE and utility of the buyer devices, respectively, compared to the other state-of-the-art works. Both theoretical and experimental analyses showed that FEST-Action mechanism was computationally efficient, individually rational, truthful to the buyers and workers, and budget balance.

Dynamic pricing updates of different buyers and workers following the demand supply theory of economics might make the task allocation problem more interesting. Learning the behavior of the buyers and workers using deep learning approaches and incorporating those in different decision making steps might help us to further increase their utilities and we have kept it in the list of our future works.

Acknowledgment

The authors are grateful to King Saud University, Riyadh, Saudi Arabia for funding this work through Researchers Supporting Project number RSP-2020/18.

References

[1] M. usage overview. https://www.statista.com/topics/1002/mobile-app-usage/. [Online; accessed 20-July-2020].

[2] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using mapreduce," Carnegie-mellon univ Pittsburgh PA school of computer science, Tech. Rep., 2009.

[3] Z. Cheng, P. Li, J. Wang, and S. Guo, "Just-in-time code offloading for wearable computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 74–83, 2015.

[4] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2014, pp. 352–357.

[5] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.

[6] P. Roy, A. Tahsin, S. Sarker, T. Adhikary, M. A. Razzaque, and M. M. Hassan, "User mobility and quality-of-experience aware placement of virtual network functions in 5g," *Computer Communications*, vol. 150, pp. 367 – 377, 2020.

[7] C. T. Joseph and K. Chandrasekaran, "Intma: Dynamic interaction-aware resource allocation for containerized microservices in cloud environments," *Journal of Systems Architecture*, vol. 111, p. 101785, 2020.

[8] A. Mtibaa, K. A. Harras, and A. Fahim, "Towards computational offloading in mobile device clouds," in *Proceedings of the 2013 IEEE International Conference on Cloud Computing Technology and Science - Volume 01*, 2013, p. 331–338.

[9] W. Tang, X. Zhao, W. Rafique, L. Qi, W. Dou, and Q. Ni, "An offloading method using decentralized p2p-enabled mobile edge servers in edge computing," *Journal of Systems Architecture*, vol. 94, pp. 1 – 13, 2019.

[10] S. Shekhar, A. Chhokra, H. Sun, A. Gokhale, A. Dubey, X. Koutsoukos, and G. Karsai, "Urmila: Dynamically trading-off fog and edge resources for performance and mobility-aware iot services," *Journal of Systems Architecture*, vol. 107, p. 101710, 2020.

[11] H. Dai, X. Zeng, Z. Yu, and T. Wang, "A scheduling algorithm for autonomous driving tasks on mobile edge computing servers," *Journal of Systems Architecture*, vol. 94, pp. 14 – 23, 2019.

[12] S. H. Marakkalage, S. Sarica, B. P. L. Lau, S. K. Viswanath, T. Balasubramaniam, C. Yuen, B. Yuen, J. Luo, and R. Nayak, "Understanding the lifestyle of older population: Mobile crowdsensing approach," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 1, pp. 82–95, 2019.

[13] X. Wang, W. Tushar, C. Yuen, and X. Zhang, "Promoting users' participation in mobile crowdsourcing: A distributed truthful incentive mechanism (dtim) approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5570–5582, 2020.

[14] P. Roy, S. Sarker, M. A. Razzaque, M. M. Hassan, S. A. AlQahtani, G. Aloi, and G. Fortino, "Ai-enabled mobile multimedia service instance placement scheme in mobile edge computing," *Computer Networks*, vol. 182, p. 107573, 2020.

[15] J. Luo, X. Deng, H. Zhang, and H. Qi, "Qoe-driven computation offloading for edge computing," *Journal of Systems Architecture*, vol. 97, pp. 34 – 39, 2019.

[16] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[17] C. Shi, M. H. Ammar, E. W. Zegura, and M. Naik, "Computing in cirrus clouds: the challenge of intermittent connectivity," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 23–28.

[18] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[19] S. Saha, M. A. Habib, T. Adhikary, M. A. Razzaque, and M. M. Rahman, "Tradeoff between execution speedup and reliability for compute-intensive code offloading in mobile device cloud," *Multimedia Systems*, vol. 25, no. 5, pp. 577–589, 2019.

[20] X. Wang, X. Chen, and W. Wu, "Towards truthful auction mechanisms for task assignment in mobile device clouds," in *IEEE INFOCOM - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[21] A. Jin, W. Song, P. Wang, D. Niyato, and P. Ju, "Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing," *IEEE Transactions on Services Computing*, vol. 9, no. 6, pp. 895–909, 2016.

[22] X. Wang, Y. Sui, J. Wang, C. Yuen, and W. Wu, "A distributed truthful auction mechanism for task allocation in mobile cloud computing," *IEEE Transactions on Services Computing*, pp. 1–1, 2018.

[23] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[24] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, ser. MCS '10. New York, NY, USA: Association for Computing Machinery, 2010.

[25] S. Al Noor, R. Hasan, and M. M. Haque, "Cellcloud: A novel cost effective formation of mobile cloud based on bidding incentives," in *2014 IEEE 7th International Conference on Cloud Computing*. IEEE, 2014, pp. 200–207.

[26] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 145–154.

[27] F. Lu, L. Gu, L. T. Yang, L. Shao, and H. Jin, "Mildip: An energy efficient code offloading framework in mobile cloudlets," *Information Sciences*, vol. 513, pp. 84 – 97, 2020.

[28] W. Tushar, C. Yuen, D. B. Smith, and H. V. Poor, "Price discrimination for energy trading in smart grid: A game theoretic approach," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1790–1801, 2017.

[29] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou, "Agent-oriented cooperative smart objects: From iot system design to implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 11, pp. 1939–1956, 2018.

[30] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, and G. Fortino, "Agent-based internet of things: State-of-the-art and research challenges," *Future Generation Computer Systems*, vol. 102, pp. 1038–1053, 2020.

[31] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli, "Modelling and simulation of opportunistic iot services with aggregate computing," *Future Generation Computer Systems*, vol. 91, pp. 252–262, 2019.

[32] X. Wang, X. Chen, W. Wu, N. An, and L. Wang, "Cooperative application execution in mobile cloud computing: A stackelberg game approach," *IEEE Communications Letters*, vol. 20, no. 5, pp. 946–949, 2016.

[33] A. Mtibaa, K. A. Harras, K. Habak, M. Ammar, and E. W. Zegura, "Towards mobile opportunistic computing," in *Proceedings of the 2015 IEEE 8th International Conference on Cloud Computing*, ser. CLOUD '15. USA: IEEE Computer Society, 2015, p. 1111–1114.

[34] X. Lin, J. Jiang, C. H. Y. Li, B. Li, and B. Li, "Circa: collaborative code offloading among multiple mobile devices," *Wireless Networks*, vol. 26, no. 2, pp. 823–841, 2020.

[35] C. Bettstetter, "Smooth is better than sharp: a random mobility model for simulation of wireless networks," in *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, 2001, pp. 19–27.

[36] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic properties of the random waypoint mobility model," *Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[37] X. Huang, R. Yu, J. Kang, Z. Xia, and Y. Zhang, "Software defined networking for energy harvesting internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1389–1399, 2018.

[38] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE*

Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2906–2920, 2019.

[39] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268–4282, 2016.

[40] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for iot devices with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1930–1941, 2019.

[41] S. Sarker, M. A. Razzaque, M. M. Hassan, A. Almogren, G. Fortino, and M. Zhou, "Optimal selection of crowdsourcing workers balancing their utilities and platform profit," *IEEE Internet of Things Journal*, 2019.

[42] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han, "Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 1986–1999, 2016.

[43] J. Ren, Y. Zhang, K. Zhang, and X. S. Shen, "Sacrm: Social aware crowdsourcing with reputation management in mobile sensing," *Computer Communications*, vol. 65, pp. 55 – 65, 2015, mobile Ubiquitous Sensing from Social Network Viewpoint.

[44] S. Sundar and B. Liang, "Offloading dependent tasks with communication delay and deadline constraint," in *IEEE INFOCOM -IEEE Conference on Computer Communications*. IEEE, 2018, pp. 37–45.