IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 1, 2020, accepted November 9, 2020, date of publication November 16, 2020, date of current version November 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3038249

A Mobile Edge-Based CrowdSensing Framework for Heterogeneous IoT

HANANE LAMAAZI, RABEB MIZOUNI, SHAKTI SINGH, (Member, IEEE), AND HADI OTROK, (Senior Member, IEEE)
Center for Cyber-Physical Systems, EECS Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Corresponding author: Hanane Lamaazi (hanane.lamaazi@ku.ac.ae)

This work was supported by the Center for Cyber Physical System, Khalifa University, through the Internal Research Fund, under Grant 8474000137.

ABSTRACT In this article, we consider the problem of distributed offloading in mobile crowdsensing (MCS) by the means of mobile edge computing(MEC). Deploying MEC in MCS can help address many challenges the centralized MCS solutions are facing such as delays in answering real-time requirements due to the centralized nature of the solution, discovering and selecting non-connected devices in the Area of Interest (AoI), and dealing with the complexity of data computation. Specifically, we propose to improve the selection of crowdsourced workers by opting for a distributed mechanism, where the selection is partially offloaded to the Local Edge Nodes (LENs). The proposed framework, OffSEC, relies on a) a Mobile edge computing architecture that defines the Main Edge Node (MEN) and LENs responsible for selecting the local workers available in the AoI and b) a two-layer selection mechanism that helps to offload the selection of crowdsourced workers to the identified LENs. To do this, nodes in the area of interest are first clustered based on their locations, and then for each cluster, one LEN is identified based on the closeness metrics. MEN is then nominated based on a greedy selection. Finally, LENs discover the available nodes in their cluster, including heterogeneous IoT nodes and workers that are not necessarily connected to the Edge server, and select the final list of workers that maximize the quality of service (QoS). The process of selection is dynamic as it is updated according to the requested task. The proposed OffSEC is evaluated using a real dataset and is compared to a centralized approach. The results show that OffSEC outperforms the benchmark by maximizing the QoS of the sensing activities and improving the quality of the collected data readings (QoDR).

INDEX TERMS Edge computing, crowdsensing, distributed architecture, heteregenous IoT.

I. INTRODUCTION

As a revolutionary technology, the Internet of Things (IoT) covers all domains of our lives by connecting everyday objects to the Internet [1].The deployment of heterogeneous IoT devices (e.g sensors, dash cameras, and smartphones) and heterogeneous communication technologies (e.g WiFi, Bluetooth…) on a large-scale requires consideration for real-time applications such as energy efficiency, latency, network capability, and Quality of Service(QoS). As the volume of generated data increases, providing an adequate data processing solution becomes a necessity. Recently, Mobile Crowdsensing (MCS) has been identified as an enabling technology for IoT [2], since it provides an optimal solution for collecting large amounts of data. It relies on interactions between humans and/or devices to extend the IoT services. Generally, MCS applications are deployed on personal devices, like smartphones or wearable devices [3],which makes them benefit from their sensing capabilities to extract and share collected information about the physical environment for a specific phenomenon of interest [2], [4]. These devices generate a large amount of heterogeneous data that needs to be processed and stored efficiently. Usually, the collected data through mobile devices and sensors are uploaded into a centralized platform to be accurately processed where sufficient resources are available [5]. The centralized model is efficient for applications that have energy and delay tolerance. The deployment of a centralized architecture provides

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

a set of limitations regarding storage capability, data reading consistency, energy consumption, devices capability, and latency [5].

Additionally, these centralized crowdsensing platforms are responsible to perform all the crowdsensing related activities including worker's recruitment/selection, task allocation, worker's payment, data fusion, and processing. However, these MCS solutions suffer from a set of well-identified limitations regarding a) how to reach the workers in the area of interest (AoI), that are not connected to the server?, b) how to choose a good placement of the main edge node (MEN) and local edge nodes (LENs) to be close to the task and workers?, c) how to optimize the budget to select workers that maximize the QoS while answering the task requirements?, and d) how to improve the data readings processing to reduce the load of data computation at the ES?

Edge computing was designed to overcome the limitations of a centralized platform and improves its deployment in a MCS environment. The existing works on edge computing mainly focus on local computation [6], resource optimization [7], edge deployment [8], and task allocation/offloading [9].

This new paradigm can be a powerful solution to fulfill complex tasks and delay-sensitive applications by providing real-time computation. The computation can be done locally [10], [11] or offloaded, from an edge node to the cloud server [12] or from user to edge servers [13], according to the task assignment schemes adopted [14], [15]. The edge servers/nodes can receive tasks through different communication technologies like Wi-Fi/4G or Bluetooth, where each communication technology has specific characteristics such as transmission range, quantity of transmitted data, and resource consumption. Combining more than one technology constitutes a challenge for edge computing. Another important challenge of edge computing is to reduce the significant distance between the service provider and the workers. A good placement of the edge node to be close to the workers allows avoiding high end-to-end delay and reduce traffic congestion and increase the quality of service.

In this work, it is argued that edge computing nodes can play a vital role in improving the performance of MCS by improving the mechanism of workers' selection to execute the sensing tasks. The main research question this work is addressing: how to design a distributed strategy for worker selection that; a) considers EN placement close to the workers, b) extends the discovering process of connected and non-connected devices in the AoI, c) builds a mechanism that calculates the optimal payment for workers that deploy EN and sensing devices, d) offloads the selection process to EN to select workers that maximize QoS while responding to the task requirements.

To this end, a framework (OffSEC) for mobile crowdsensing based on mobile edge computing is proposed. OffSEC relies on a) a Mobile edge computing architecture that defines the MEN and LENs responsible for selecting the local workers available in the area of interest and b) a two-layer selection mechanism that helps to offload the selection of crowdsourced workers to the identified LENs This work exploits the benefit of edge computing in crowdsensing where the edge server (ES) and Edge nodes (EN) are deployed in the AoI. Using ES and EN improves worker selection for allocated tasks, by discovering more potential workers that are not discoverable by the ES. Two types of edge nodes (ENs) are defined: *MEN* that plays the role of a task sensing leader which manages all the collected data and sends it back to the ES; and the *LEN* that selects workers and locally collects the data, from its discoverable neighboring nodes. In the proposed framework, the ES is deployed as Edge-as-a-Service (EaaS) that uses homogeneous edge resources (eg. smartphones). The connected ENs are clustered by the ES using the centroid-based k-means algorithm where the closet EN to the centroid is defined as LEN. From candidate LENs, one MEN is selected based on its computational capability and closeness to the sensing task. Then, the worker selection is offloaded to the LENs themselves. In a second phase, each LEN locally discovers its neighboring nodes within the same cluster and the nodes that ES cannot reach due to the use of short-range Bluetooth technology. The proposed architecture deploys heterogeneous IoT devices that can be smartphones, surveillance cameras, dash cameras, or sensors, using Wi-Fi/4G (connected) and Bluetooth (non-connected) technologies. The sensing tasks are fully offloaded to the LENs that collect the data from the workers and then return the sensing outcome to the MEN. The MEN in turn sends the aggregated data to the ES for further processing.

The main contribution of the work is summarized as follows:

- Propose a distributed edge-based architecture for crowdsensing that allows overcoming the limitations of the centralized architecture in terms of worker selection.
- Propose a novel two-layers selection mechanism that a) enables the discovery of non-connected nodes (workers or IoT sensors) in the AoI. and b) offloads the ES by attributing the selection of heterogeneous workers to the LENs.
- Achieve a higher QoS by selecting better participants with the respect to the computed sensing budget.
- Validate the proposed approach by evaluating the reliability of submitted data collected by the selected sets of workers.

The proposed approach is validated using real-life datasets and compared to a centralized platform. Different scenarios are considered to assess the proposed approach where the results show that the proposed distributed selection outperforms the centralized one in terms of quality of service and the consistency of data readings.

II. RELATED WORKS

MEC is endorsed as an enabling technology for advanced IoT applications [16] such as MCS [17]. MEC deployment resolves a set of centralized MCS problems as significant load, high traffic and, high latency of data transmission.

A set of studies interested to improve its deployment by addressing: *MEC placement* as proposed in [8] where the authors deploy the MEC in the context of smart cities to resolve load balancing and delay problem, *Local computation* to address the problem of data storage [6], [11], *offloading* computational task by considering resources allocation and data transmission [18], [19] by using different techniques as process, component, application, and virtual machine migrations [20], *energy consumption* [21] where the selection of cluster heads is based on devices capabilities, which improves the communication sustainability while preserving the energy consumed by ENs, and the *makespan* or processing time minimization [22].

Many IoT applications have adopted MEC to address the heterogeneity issue of IoT devices. This diversity makes resource allocation very difficult especially when it is based on the task's weight. Authors in [23] propose a new intelligent model called "DRL+FL", based on the resource allocation algorithm, DDQN-RA, to optimally allocate computing and network resources according to the change of MEC environment. Authors in [24] propose a learning-based channel selection framework (SEB-UCB) to overcome the limitations of Edge Computing in terms of spectrum resources, battery capacity, and context unawareness. To optimize the problem, the authors used three techniques including matching preference learning, Lyapunov optimization, and matching theory. SEB-UCB allows improving the throughput as well as the backlog queue and the energy. Similarly, instruction translation and offloading for mobile devices (SIMDOM) framework in the cloud and edge environments was proposed in [25]. The framework allows reducing the execution overhead of migrated vectorized multimedia applications by adopting vector-to-vector instruction mappings, which leads to improving the energy, execution time, and performance efficiency.

In line with this, MEC has been proposed recently for MCS applications to improve worker selection and network performance. For instance, the authors in [26] propose a new social-driven edge computing architecture that is based on the attitude of workers to execute tasks. This proposed architecture demonstrates that using centrality measurement selection provides more benefits than cooperativeness scores, which positively impacts latency and requests' satisfaction. Authors in [27] propose a new architecture that combines both edge computing quality and MCS features. this new architecture allows to analyze huge collected data while it reduces the overhead due to the transmissions and it can support applications that require low latency. Similarly, the authors in [17] propose a new trust evaluation mechanism that combines both the Intelligence of MEC and the benefit of Crowdsourcing. The selection of ENs has been made through Trustworthy and Quality-Aware Incentive mechanisms, which provide more accurate trust evaluation. The authors in [28] propose a new selection of clients with resource constraints using an extension Federated Learning (FL) framework. The main goal of the proposed FedCS is to request, a random set of a heterogeneous client, to download a trainable model provided by the ES, and update it according to their data. The selection is made based on limited computational resource and wireless constraints, which allows to manage client and complete the training process in a short time.

Similarly, the authors in [29] consider the heterogeneity of user equipment (UE) and resource limitations for MEC service providers (SPs) selection. A multi-MEC and multi-UE scenario were adopted to study the matching problem, between MEC SPs and UE, using Auction theory. Another MEC selection was proposed in [30] where a new QoS-aware service selection for MEC systems was designed. The proposed event-driven framework was formulated by a dynamic optimization problem using ordinal optimization (OO) techniques. It provides a good selection of users by considering their dynamicity, which improves the convergence rate. A peer-to-peer enabled selection of a mobile ES was proposed in [31]. The ESs update their information, in a decentralized way, to be selected by the nearest mobile devices. The proposed approach is cost-effective and deadline-aware offloading, which makes it more efficient, less expensive and, faster for completing computational tasks even under different traffic congestion.

Table 1 summarizes all related works discussed above, and also includes our proposed Approach. The existing works related to the MEC deployment present a set of shortcomings. Some work interest to minimize the energy consumption, while choosing the closet placement of ES to the end devices with the lack of ES characteristics consideration. Also, some studies focus on offloading the tasks and local computation with no consideration of task requirements such as location, required sensors, and budget. In contrast, works that deploy MEC in MCS propose a selection on upward where only the end devices select the optimal ES to offload the data, while the ES can also be used to select and recruit workers during the sensing process. Our proposed approach differs from the others discussed above in the following aspects:

- ***Resource Discovery***: When Initiating the communication, EN discover more neighbor devices hidden from the ES through Bluetooth transmission. This improves the quality of selection and avoids transmission failures due to poor connectivity or high consumption of the device's resources.
- ***Task Allocation***: The selection is relative to the task constraints and the EN decision, where it can initiate an outgoing task request (eg, event detection) or accept or refuse an incoming one (e.g, whether measurement).
- ***Context-Aware***: where the distribution of LENs is based on their location in the AoI. Also, the selection is related to the data type that needs to be collected and which is based on the required sensors deployed for the sensing process.
- ***Heterogeneous Worker Selection***: Two types of workers are deployed based on their transmission technology. ENs that play the roles of LENs and MEN and use the

TABLE 1. Taxonomy of the existing works.

Ref		[17]	[26]	[27]	[28]	[29]	[30]	[31]	Our
Scheduling	Proactive						✓	✓	✓
	Reactive	✓	✓	✓	✓	✓			✓
Selection	plateform-centric								✓
	user-centric	✓	✓		✓	✓	✓	✓	✓
Infrastructure	LTE-direct/ cellular		✓	✓	✓	✓		✓	
	Wi-Fi/4G						✓		✓
	Bluetooth								✓
Acquisition	Homogeneous		✓	✓			✓		
	Heterogeneous	✓			✓		✓		✓
Assignment	Centralized		✓						
	Decentralized	✓		✓	✓	✓	✓	✓	✓
Execution	Single-Task	✓	✓	✓	✓			✓	✓
	Multi-Tasks					✓	✓		
Responsibility	Local	✓			✓			✓	✓
	Centralized		✓	✓		✓	✓		✓

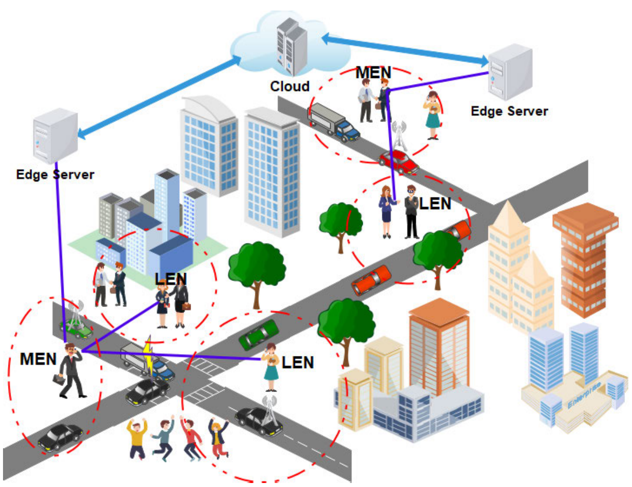


FIGURE 1. System illustration.

Wi-Fi/4G transmission, and workers that are discovered by LENs through Bluetooth transmission.

- ***Budget-Constraint***: The process of selection takes into consideration the available budget provided by the task. For each cluster, there is a sensing budget used for the selection. The selection is breaking off when this sensing budget is reached.

III. SYSTEM MODELLING

A. MOTIVATIONAL SCENARIO

Figure 1 illustrates the proposed approach for edge-based MCS solution. The initiation of a sensing task can be done either by the ES as a response to a recently published task, or by the ES as a notification of an ongoing event. Based on the requirement of the published task, the data is collected from mobile phones, dash cameras, and sensors available in the AoI.

The collected data is then transferred via an EN to the ES to be processed and fused in real-time. The ENs play an important role, especially for real-time applications, as they can be close to the end-user and they can also discover nodes that are not discoverable to the ES.

The decision making of the EN can attain a high level of effectiveness when it tries to discover its neighbor, compete to play the role of MEN, and form a set of workers according to their location and device capabilities. The proposed architecture can be deployed in a complex scenario where various types of devices can be used.

In this work, two kinds of devices are used for collecting data:

- Devices that play the role of EN that have a Wi-Fi/4G connection and are visible by the ES. The EN serves as a relay between the edge ES and sensing devices. Each set of devices follows one LEN which is the central EN within a cluster, and the LENs follows the MEN which is a relay between LENs and ES. The LEN transfers the collected data provided by the workers to the MEN. The MEN then combines the data received from different LENs and then sent the sensing outcome to the ES for additional processing.
- The sensing devices that are selected for sensing, based on their capabilities. The sensing devices can have Wi-Fi/4G and/or Bluetooth communication technologies. The devices that possess Bluetooth connection are discoverable only by the LENs. These hidden devices are useful during the selection process since they can help improve the quality of service by providing diverse choices for the LENs in selecting the best sensing devices.

B. SELECTION MODEL FORMULATION

The proposed model adopted for the selection process considers two main sets of parameters: infrastructure-related and task-related (see figure 2). The infrastructure- related parameters include the AoI coordinates, devices capabilities, and transmission technology of both EN and sensing devices, whereas the task-related parameters include the task requirements namely the task coordinates, required reputation, required sensors, and budget. The proposed selection model adopts the two-level of communication between server-to-edge nodes and edge-to-sensing devices.

The initiation of the tasks can be done from the ES to the ENs (e.g survey) or from the EN to the ES, as is the case in event detection. When the communication is initiated between the ES and ENs, the ES formulates the first set of participants by clustering them based on their locations as shown in figure 2.

For each cluster, there is a LEN defined based on its strategic position geographically close to the center of the cluster. The location of the LEN allows it to be near to most of the workers inside the cluster. The main advantage of this choice is to minimize the distance between the LEN and the workers when requesting the task and receiving collected data.

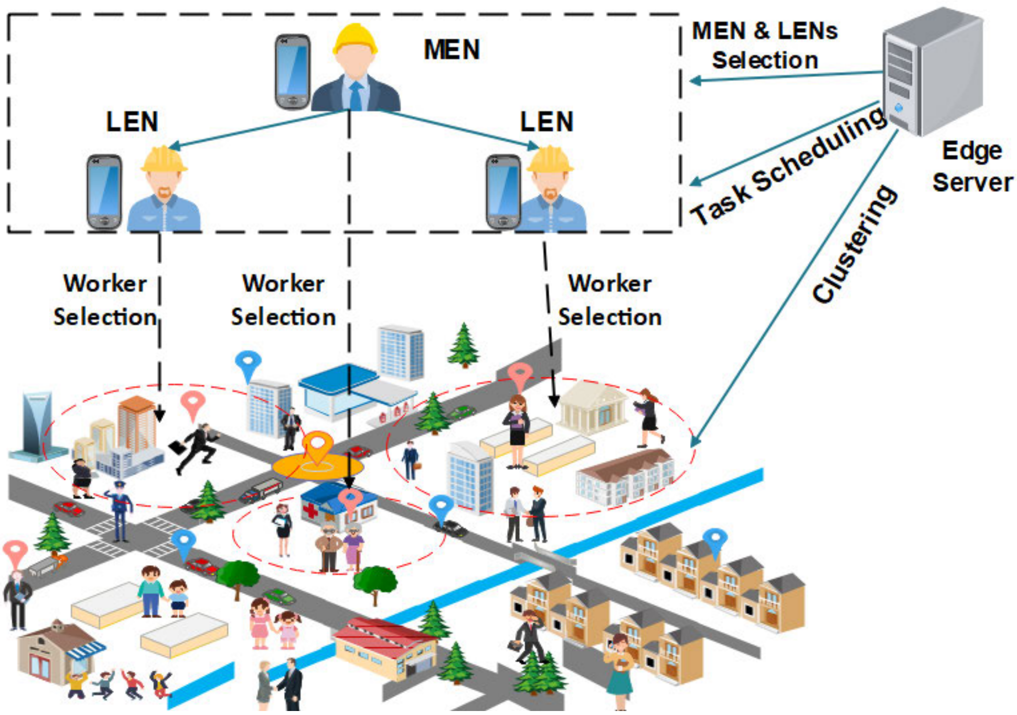


FIGURE 2. Conceptual architecture.

The MEN is then defined, amongst the LENs, in a greedy manner based on the computed objective function. This objective function considers EN characteristics: energy consumption, CPU, reputation of the worker, and closeness to the task. The LEN with the best objective function is considered as MEN. Each task has some specific requirements that the ENs need to consider during the selection process such as the required reputation, sensors, and budget. The distributed architecture proposed in this article is shown in Figure 2.

Each requested task requires a set of workers to perform the sensing task, according to a set of requirements and constraints. In the proposed model, each task t_j is defined as $t_j = (L_j^T, S_j^T, R_j^T, B_j^T)$ as presented in Table 2. Furthermore, each worker, W_i, is equipped with smart devices that are characterized by the location, sensor availability, residual energy, reputation, payment and connectivity type, where it is defined as $W_i = (L_i^w, SA_i^w, RE_i^w, R_i^w, P_i^w, Connect_i^w)$, as described in Table 2. $Connect_i^w$ is the technology used for connecting the device of the worker i. In this work, we consider two types of connectivity: Bluetooth and Wi-Fi/4G.

1) SENSOR AVAILABILITY (SA)

For a requested task, a set of sensors is required to be available in the workers' devices [32], [33]. The main criterion that is considered to evaluate the sensor availability is that all members should have at least one of the required sensors.

The number of sensors available in the worker's device SA_{ij}^w and the set of sensors relative to the participants' devices within a cluster SA_S^{Cr} are evaluated as follows:

$$SA_{ij}^w = \frac{NS_i^w}{S_j^T} \tag{1}$$

$$SA_S^{Cr} = \prod_{i}^{S} \frac{SA_{ij}^w}{N_S^{Cr}} \tag{2}$$

where

$$SA_S^{Cr} \in [0, 1]$$

2) RESIDUAL ENERGY (RE)

Due to the distribution of participants and their distance to the LEN and the task, some devices may not complete the sensing task or transfer the collected data [19], due to battery depletion. For this reason, it is important to consider the residual energy when selecting participants. To assess the residual energy of a set of devices within a cluster, the arithmetic mean and standard deviation are used to reflect the central tendency and the dispersion of the distribution [34]. The Residual Energy of a set of participants (S) is calculated as:

$$RE_S^{Cr} = \left(\frac{1}{N_S^{Cr}} \times \sum_{i \in S} RE_i^w\right) \times e^{-\sigma(RE_i^w)} \tag{3}$$

where RE_i^w is the residual energy for each worker i within-cluster Cr and, S is a set of selected participants.

3) REPUTATION (REP)

The reputation of a participant measures his commitment to perform and complete the sensing task [32]. As an important parameter for selection, it is used during the selection of MEN and workers, and for the final set of selected participants' assessment. The lowest reputation value in the set of selected participants within a cluster is considered to be the set's reputation. Hence, R_S^{Cr} is calculated as follows:

$$R_S^{Cr} = min\left(Rep_{i\epsilon Cr}^{w}\right) \tag{4}$$

where;

$$R_S^{Cr} \,\epsilon\, [0, 1]$$

4) OBJECTIVE FUNCTION

To select the best EN that can play the role of a relay between the ES and the other clusters when reporting the data, an objective function should be defined based on a set of metrics. The composition of these metrics is done using the weighted sum method [1] where the weights $w_1 - w_4$ reflect the metric's importance, thus making the objective function adaptable according to the application requirements. These weights should satisfy the following conditions:

$$0 < w_1, w_2, w_3, and\ w_4 < 1, and; \sum_{L=1}^{4} w_L = 1$$

The objective function is then calculated as follows:

$$Objective\_function = w_1 \times RE_i^W + w_2 \times U_i^W + w_3 \times R_i^W + w_4 \times C_{ij}^W \tag{5}$$

where C_{ij}^W is the LEN's closeness to the requested task, which can be calculated as:

$$C_{ij}^W = \frac{1}{\sqrt{\left(X_i^w - X_j^T\right)^2 + \left(Y_i^w - Y_j^T\right)^2}} \tag{6}$$

where:

X_i^w and Y_i^w are the coordinates of the worker's Li^w
X_j^T and X_j^T are the coordinates of the Required Task Lj^T

5) QUALITY OF SERVICE (QoS)

The QoS is an important parameter for the MEC deployment, as it measures the expected performance of the selected participants. For each task, each worker provides the QoS_i^{Cr} to the cluster that is calculated as described in Eq (7). The QoS_i^{Cr} defines the ability of the worker to complete the task. Worker's reputation, in addition to workers' device characteristics, namely residual energy and sensor availability are part of QoS of the cluster. The higher the QoS_i^{Cr}, the better is the selected worker.

$$QoS_i^{Cr} = w_1 \times SA_i^w + w_2 \times RE_i^w + w_3 \times R_i^w; \sum_{i=1}^{3} w_i = 1 \tag{7}$$

where $w_1 - w_3$ are weights reflecting the parameters importance.

To evaluate the QoS_S^{Cr}, The model used in [32] for group-based recruitment system for MCS, is adopted and adapted to the ENs needs. When defining the final list of selected workers to perform the task, the collective QoS_S^{Cr} provided by them to the cluster is calculated as:

$$QoS_S^{Cr} = w_1 \times SA_S^{Cr} + w_2 \times RE_S^{Cr} + w_3 \times R_S^{Cr}, \sum_{i=1}^{3} w_i = 1 \tag{8}$$

TABLE 2. List of variables used.

Symbole	Definition
L_j^T	Location of task j
S_j^T	Number of sensors required by task j
R_j^T	Min reputation required by task j
B_h^T	Available Budget of task j (for sensing and EN deployment)
W_i	the i-th worker
MEN_{ij}	Worker i selected to be Main EN for task j
LEN_ij	Worker i selected to be Local EN for task j
L_i^w	Location of the worker i
NS_i^w	Number of available sensors on the device of worker i
R_i^w	Reputation of the worker i
RE_i^w	Residual Energy of the device of worker i
$Connect_i^w$	connectivity type of device of worker i where 0 for Bluetooth and 1 for Wi-Fi/4G
U_i^W	CPU processing of the device of worker i
C_{ij}^W	Closeness of the worker i to the task j
P_i^w	Payment requested by worker i
P_i^m	Payment provided by the ES to the worker i playing the role of MEN
P_i^l	Payment provided by the ES to the worker i playing the role of LEN
QoS_i^{Cr}	QoS provided by worker i relative to cluster Cr
N_S^{Cr}	Number of Workers within Cluster Cr
CSS^{Cr}	Cumulative sum of payment of workers i relative to cluster Cr
SP_S^{Cr}	Sensing payment for selecting worker i relative to cluster Cr
QoS_S^{Cr}	QoS achieved by set of participants S relative to cluster Cr
SA_S^{Cr}	Set of available sensors achieved by set of participants S relative to cluster Cr
RE_S^{Cr}	Residual Energy achieved by set of participants S relative to cluster Cr
R_S^{Cr}	Minimum reputation achieved by set of participants S relative to cluster Cr
$QoDR_S^{Cr}$	QoDR achieved by set of participants S relative to cluster Cr
$QoDR_S^{Tot}$	Total QoDR achieved by all sets of participants relative to all clusters Cr

IV. PROPOSED APPROACH

The following section describes the proposed two-layer selection approach in edge-based crowd-sensing architecture, OffSEC. This novel approach improves the worker selection for allocated tasks, by discovering more potential workers that are not discoverable by the server. The first layer uses a centroid-based k-means algorithm, where the ES distributes the participants into clusters according to their locations in the AoI and then determines the LENs and the MEN. The second layer is the offloading selection process to the ENs, where the best workers maximizing the QoS are selected.

A. FIRST LAYER: MEN AND LENs SELECTION

1) SERVICE INITIATION

In the proposed framework, the ES is deployed as Edge-as-a-Service (EaaS) that uses homogeneous edge resources, which are the LENs. By adopting EaaS, the overhead for launching and terminating the services is reduced due to the integration of a lightweight handshaking protocol for EN discovery. The advantage of adopting an EaaS is that it keeps the task request and the data transfers and processing closer to ES and ENs, and makes the information, related to ENs such as location, connectivity, and worker device, up-to-date. The handshaking protocol follows three distinguishing steps, as illustrated in figure 3, to elaborate on the communication between ES and ENs: the request, the acceptance, and the confirmation extraction.

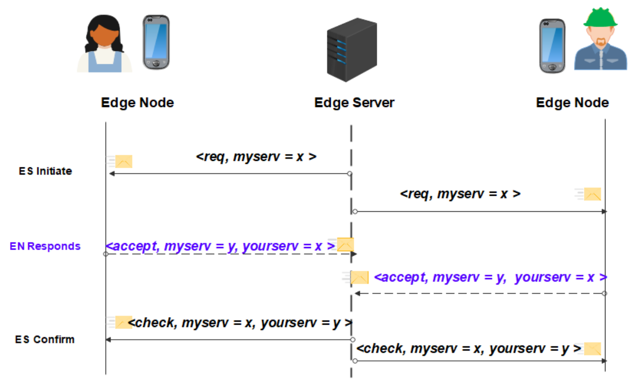


FIGURE 3. Edge server communication initiation.

The communication used between sensing devices and LENs is device-to-device (D2D). The LENs communicate with each other using homogeneous devices (smartphones) and use WiFi/4G as a transmission technology. However, the communication between LENs and other participants within the same cluster takes place using heterogeneous IoT devices (smartphones, dash cameras, surveillance cameras, sensors, etc) and using different transmission technologies (WiFi/4G and Bluetooth). The devices that use Bluetooth are discovered by LENs locally, while they are hidden from the ES.

2) LENs SELECTION

The distance between EN and workers is an important parameter that should be considered during the selection process. Thus, the position of LEN should be close to all workers inside a cluster. The selection of MEN and LENs are dynamic and can change according to the requested task, the number of discovered devices in the AoI, and the best objective function provided. The objective function is designed to consider the main parameters that make the MEN able to combine the data received from the set of participants within his cluster and the LENs. The MEN then transfers the final results to the ES. The closeness parameter (see (6)) has a big impact on preserving energy and hence improving the QoS. The overall proposed algorithm 1 is detailed as bellow.

As inputs, the dataset of the participants and the tasks are used, whereas the number of clusters, the number of LENs, and the preselected set of Workers are calculated as output. Initially, when the communication is established, the workers that do not satisfy the reputation required by the task are dropped (**line 1-5**). The silhouette mechanism is then deployed to optimally cluster the ENs in the AoI (**line 6**). The K-means algorithm is used to distribute the workers according to the number of clusters provided by the silhouette mechanism and to define the centroids (**line 7-8**). Then, the workers that are close to the center of the cluster are defined as LEN, else they are classified as workers (**line 9-15**). Being in the center of the cluster means that the LEN is the nearest EN to all devices in the same transmission range.

Algorithm 1 LEN Selection

Input : Participants W, where for each $w_i \in W$ $Connect_{\,i} = 4G$ or wifi or Bluetooth
Input : Set of tasks T_j
Output: Cr, LENs, Workers
1 initialization
2 **if** $R_i^w < R_j^T$ **then**
3 | *drop* W_i *from the list of preselected workers*
4 **end**
5 $best\_size = silhouette\_evaluation(L_i^W,$ '*Euclidean_dist*')
6 $Cr\_id = kmeans(L_i^W, best\_size)$
7 $Centroids = kmeans.Cr\_centers\_$
8 **if** $W_i = closet\_to\_centers$ **then**
9 | W_i *is LEN*
10 | **else**
11 | | W_i *is worker*
12 | **end**
13 **end**
14

3) MEN SELECTION

Algorithm 2 presents the process of MEN selection.

Algorithm 2 MEN Selection

Input : LEN's RE_i^W
Input : LEN's U_i^W
Input : LEN's R_i^W
Input : LEN's L_i^W
Input : L_j^T
Output: MEN
1 initialization;
2 calculate C_{ij}^W according to eq((16));
3 calculate objective function according to eq((15));
4 **if** $LEN_{ij} = best\_objective\ function$ **then**
5 | LEN_{ij} is MEN_{ij};
6 **end**

The location of the workers, the characteristics of their devices, and the location of the task are used as inputs. After the LENs are specified, the closeness of each LEN to the task is calculated using eq (6) (**line 2**). Then, the objective function for each LEN is calculated as described in section III-B4 (**line 3**). The MEN is chosen in a greedy manner, where from candidate LENs, the LEN that provides the best objective function is selected as MEN (**line 4-6**).

4) PAYMENT MECHANISM (SensPay)

In MCS, rewarding workers is important to motivate them to perform the tasks. In the proposed framework, offloading selection to the EN requires an incentive to be paid for MEN, LENs, and workers. Three different payments are defined P_i^m, P_i^l, and SP_S^{Cr}, where P_i^m is the payment for the MEN, P_i^l is the payment for LENs and SP_S^{Cr} is the payment for selected workers.

Being budget-constrained, the proposed approach defines each task with an independent budget B_j^T. Accordingly, workers are selected by LENs based on the sensing budget SP_S^{Cr} (Eq (9)). This sensing budget is distributed equitably between the LENs to recruit workers that perform the sensing task. The final set of selected participants depend on the amount provided to each LEN and also on the requested payment by each worker.

$$SP_S^{Cr} = \frac{(B_j^T - (P_i^m + P_i^l))}{Number\ of\ Cluster} \tag{9}$$

Algorithm 3 describes the process of calculating the sensing payment SP_S^{Cr} for worker selection. The payment of MEN, LENs, task budgets, and the number of clusters are used as input, whereas the sensing payment is calculated as the output. First, the sum of LEN's payments is calculated (**line 2**). Then, the sensing payment is calculated using Eq (9) and distributed equally among LENs (**line 3-4**).This sensing budget is used after, in the second layer, to select the final set of participants within a cluster.

Algorithm 3 Sensing Payment Calculation

Input : P_i^m, P_i^l, B_j^T
Input : Cr
Output: SP_S^{Cr}
1 initialization;
2 Calculate $\sum P_i^l$ of LENs;
3 Calculate SP_S^{Cr} according to eq ((9)) ;
4 Share the sensing payment to the LENs;

B. SECOND LAYER: WORKERS SELECTION

1) WORKERS SELECTION

To deeply improve the selection process based on MEC deployment, the second layer of selection is described in Algorithm 4. Instead of directly selecting workers through ES, as is the case in centralized architecture, in the proposed framework each LEN is responsible to select its workers' from the available set of participants within the cluster and the discovered devices.

The selection is based on the QoS_i^{Cr} (Eq (7)) that each worker provides and that responds to the task requirements. For each cluster, the number of available sensors (NS_i^w) and the QoS_i^{Cr}) are calculated (**line 2-6**). Then, the QoS_i^{Cr} is sorted in ascending way where a greedy selection is used to select workers with the best provided QoS_i^{Cr} (**line 7**). Subsequently, a cumulative sum (CSS_S^{Cr}) of the requested payment by workers is calculated. Based on this CSS_S^{Cr}, the selection is processed until reaching the provided sensing payment SP_S^{Cr} (**line 8-9**). Finally, the final list of workers that will perform the tasks is defined(**line 10**) and the device's characteristics of all workers within the list are calculated (**line 11-14**).

Algorithm 4 Worker Selection

Input : SA_i^w, RE_i^w, R_i^w
Input : S_j^T
Output: SA_S^{Cr}, RE_S^{Cr}, R_S^{Cr}, QoS_S^{Cr}, $QoDR_S^{Cr}$, $QoDR_S^{Tot}$
1 initialization;
2 **for** *each Cr* **do**
3   **for** *each* W_i **do**
4   Calculate NS_i^w according eq((1));
5   Calculate QoS_i^w according eq((7));
6   **end**
7   Sort QoS_i^w in ascending order;
8   Calculate CSS^{Cr} of P_i^{Cr};
9   Select W_i^{Cr} until $CSS^{Cr} <= SP_S^{Cr}$;
10   Define Final list of W_i^{Cr};
11   Calculate SA_S^{Cr} according eq((2));
12   Calculate RE_S^{Cr} according eq((3));
13   Calculate R_S^{Cr} according eq((4));
14   Calculate QoS_S^{Cr} according eq((8));
15   Calculate $QoDR_i^{Cr}(Avg)$ according eq((10));
16   Calculate $QoDR_i^{Cr}(Std)$ according eq((11));
17 **end**
18 **for** *all LENs* **do**
19   Calculate $QoDR_S^{Cr}(Avg)$ according eq((12));
20   Calculate $QoDR_S^{Cr}(Std)$ according eq((13));
21 **end**
22 Calculate N_S^{Tot} according eq((14));
23 Calculate $QoDR_S^{Tot}(Avg)$ according eq((15));
24 Calculate $QoDR_S^{Tot}(Std)$ according eq((16));

2) DATA COMPUTATION

To validate the proposed approach, the quality of data readings is calculated for the set of selected participants within each cluster, for each LEN, and at the MEN level as defined in Algorithm 4. The different equations used to calculate the combined QoDR are described in section V-B. After the selection process, each worker reports the data reading related to the requested task to the LEN related to his cluster C_r (**line 15-17**). Each LEN then calculates the combined $QoDR_S^{Cr}$, according to Eq (12) and Eq (13), and transmits it to the MEN (**line 18-22**). The MEN then calculates the provided data from all LENs using Eq (15) and Eq (16) and calculates the final results $QoDR_S^{Tot}$ and transmits it to the ES (**line 23-24**).

V. PERFORMANCE EVALUATION

In this section, a validation of the proposed OffSEC framework is presented. The dataset and task parameters used in this work are described in section V-A and the quality of data readings calculation is explained in V-B. OffSEC is compared to a centralized platform using two different scenarios: varying the budgets, and varying the number of participants.

A. SIMULATION SETUP

The simulations carried out in this work is based on real datasets. The Sarwat Foursquare dataset [35] for social

TABLE 3. Evaluation parameters.

Dataset Parameters	
Number of workers	100, 200, 300, 400, 500
Area(Lat, Long)	([31, …..43], [129, …..144])
Sensors Parameters	Residual Energy, CPU
Sensors Availability	GPS, Humidity, Temperature, Camera, Microphone
Workers characteristics	Reputation, Cost
Connectivity	Wi-Fi/4G, Bluetooth
MEN Payment	([12, …..15])
LEN Payment	([11, …..13])
Workers Payment	([1, …..12])
Task Parameters	
Number of Tasks	30
Location (Lat, Long)	([31, …..43], [129, …..144])
Required Sensors	([2, …..5])
Required Reputation	([0,3…..0,6])
Task Budget	([150, 300, 450, 600, 750])

networking applications is employed, which includes data about the user's device such as energy, sensor availability, worker payment [36]. The connectivity type, the task required SA, the task required reputation, the task available budget, and the MEN/LEN payments are randomly generated and assigned to the nodes following a uniform distribution. The Stack Exchange Data Dump dataset [37] is used for the user's reputation. To simulate data readings, a publicly available dataset [38] is used to evaluate the model.

A set of independent tasks equal to 30 is generated with different locations, required sensors, and reputation while the available budget is set according to the adopted scenario. For each task, multiple iterations are run to collect the obtained results. Each iteration is a complete run of the two-stage selection, while the second layer the selection is performed for multiple clusters obtained through the silhouette mechanism.

Table 3 summarize the evaluation parameters, such as the area of interest used, user characteristics, payments, and task parameters. To evaluate the proposed framework, two scenarios are adopted, a) when the available budget, required for sensing, is varied and, b) when the number of participants is increased.

Centralized architecture is chosen as the benchmark to compare the performances of OffSEC. By design, the selection in centralized architecture is done directly by the ES in a greedy way, where it selects only the visible devices that use the services and cannot discover those that use Bluetooth transmissions. For the benchmark, first, the participants that do not satisfy the reputation (R_j^T) required by the task are dropped. Each worker provides its QoS_i^{Cr}, calculated as in Eq (7), which is then sorted in ascending order. Finally, the workers are selected according to the available budget B_j^T and their requested payment P_i^w.

B. EVALUATION METRICS

To validate the proposed approach, a set of metrics are considered for evaluation. First, the approaches are assessed using RE, SA, Reputation, and QoS. Then, the number of selected participants and the spent budget are considered. Finally, the reported data by selected workers are evaluated using the Quality of Data Reading (QoDRg), as described next. All the figures present an average of 5 simulations per task. The reported data by workers can have different quality levels due to the participant devices' capabilities such as energy level, CPU, and communication technologies. In this work, the quality of data readings, from different devices and at different locations, is evaluated. To compare the quality of data readings from all clusters $QoDR_T^{Cr}$, the mean difference and the standard deviation of QoDR are calculated as described in [39]. The model follows three levels as illustrated next:

Level 1: For a set of participants in each cluster

Using the average $QoDR_S^{Cr}(Avg)$ and the standard deviation $(QoDR_S^{Cr})^2(Std)$, given as

$$QoDR_S^{Cr}(Avg) = \frac{\sum_{i=1}^{S} SubmittedData}{Number\ of\ Submission} \times N_S^{Cr} \tag{10}$$

$$QoDR_S^{Cr}(Std) = \sigma^2(N_S^{Cr} - 1) + (\frac{(QoDR_S^{Cr}(Avg))^2}{N_S^{Cr}}) \tag{11}$$

where: N_S^{Cr} is the number of participants per cluster and σ is the standard deviation of submitted data within a cluster.

Level 2: For all LENs in all clusters

The combined value of data reading that is calculated for all LENs is calculated as follows:

The average $QoDR_S^{Cr}(Avg)$ and standard deviation provided by all LENs is given as:

$$QoDR_S^{LEN}(Avg) = \sum_{k=1}^{S}(QoDR_S^{Cr}(Avg)) \tag{12}$$

$$QoDR_S^{LEN}(Std) = \sum_{k=1}^{S}(QoDR_S^{Cr}(Std))) \tag{13}$$

Level 3: At the MEN stage

The combined value of data reading that is calculated at the MEN is as follows:

Total number of selected workers for the whole selection process is:

$$N_S^{Tot} = \sum_{k=1}^{N}(N_S^{Cr}) \tag{14}$$

where N is the number of cluster in the AoI.

The total average of data readings $QoDR_S^{Tot}(Avg)$ and the total standard deviation $QoDR_S^{Tot}(Std)$ is given as:

$$QoDR_S^{Tot}(Avg) = \frac{QoDR_S^{Cr}(Avg)}{N_S^{Tot}} \tag{15}$$

$$QoDR_S^{Tot}(Std) = \frac{sqrt((\frac{QoDR_S^{Cr}(Std) - (QoDR_S^{Cr}(Avg))^2}{N_S^{Tot}})}{(N_S^{Tot} - 1)} \tag{16}$$

C. SCENARIO 1: IMPACT OF BUDGET VARIATION ON THE SELECTION PROCESS

As mentioned previously, LENs and MEN will be paid for the computation/the role they are playing in the selection.

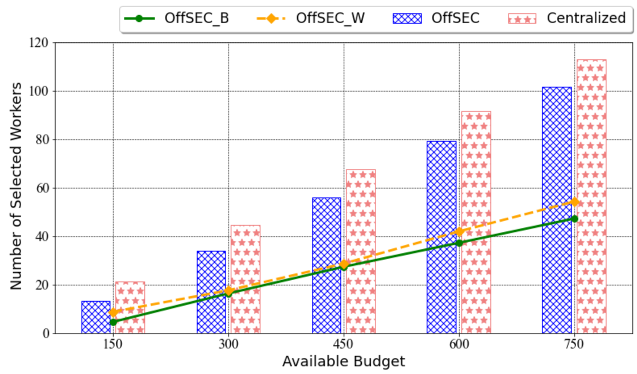


FIGURE 4. **Number of selected workers for OffSEC and centralized under different budgets.**

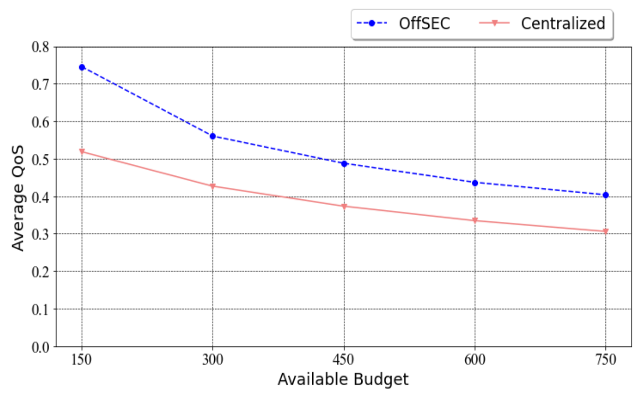


FIGURE 5. **Average QoS for OffSEC and centralized under different budgets.**

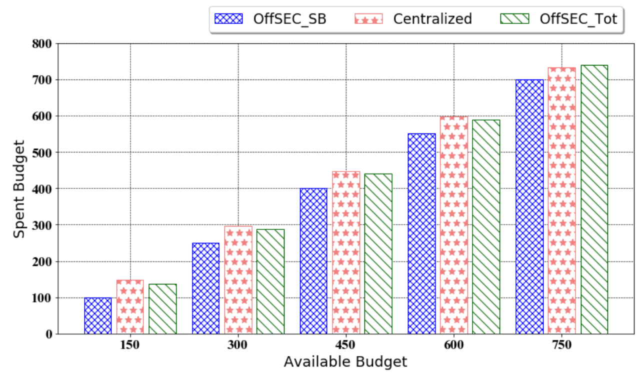


FIGURE 6. **Spent budget for OffSEC and centralized under different budgets.**

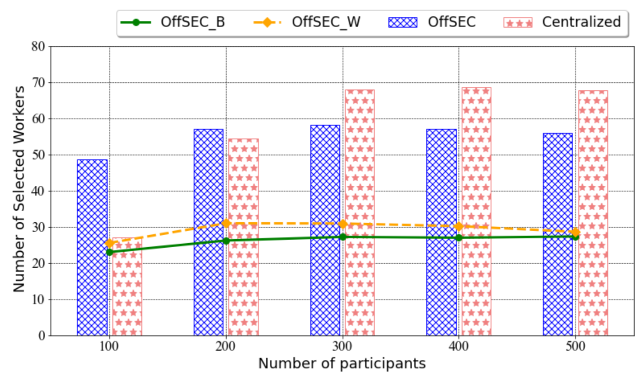


FIGURE 7. **Number of selected workers for OffSEC and centralized under the number of available participants in AoI.**

Consequently, for each task, the provided budget is used to pay MEN and LENs, in addition to the workers performing the sensing activity. To evaluate the impact of the budget on the selection, OffSEC and the benchmark are evaluated under varying budgets, from 150 to 750, for the same number of participants, i.e 500. The objective is to evaluate the impact of an increasing budget on the selection process.

Figures 4 and 6 show the comparison of the selected participants and the spent budget for the two approaches. Compared to OffSEC, the centralized approach recruits a higher number of participants, as illustrated in Figure 4, by up to 37.3 %. Besides, OffSEC selects approximately up to 52 % of the participants from Bluetooth devices (*OffSEC_B*) and up to 48 % devices that use Wi-Fi 4G (*OffSEC_W*). Overall, OffSEC consistently selects less workers than the centralized approach, though achieving higher QoS as shown in Figure 5.

As shown in Figure 6, the budget spent in sensing, *OffSEC_SB* is lower than the benchmark by 48.98 % and 4.7% for low and high budgets respectively, while *OffSEC_Tot* is lower than the benchmark by 7.34 % and 1.35 % for low and high budgets respectively.

This difference is due to the budget allocated to hire LENs and MEN. Only an average of 10 % to 20 % from the available budget is dedicated to the payment of LENs and MEN while the centralized approach uses the whole budget for sensing. However, this allocation of budget for LENs and MEN does not impact the QoS where OffSEC still provides between 23 % to 30 % of improvement in the achieved quality as shown in Figure 5.

D. SCENARIO 2: NUMBER OF AVAILABLE PARTICIPANTS IN THE AoI

In this scenario, the proposed OffSEC is compared to the centralized approach in terms of the available participants in the AoI, where it includes connected and IoT devices. Both the number of participants and their locations in the AoI is distributed randomly. The number of generated tasks is equal to 30 and the available budget is fixed for the different densities. Figure 7 shows that for a small number of participants (100 to 200), OffSEC selects more participants than the benchmark. While it maintains the heterogeneity by selecting approximately 54 % of participants that use Wi-Fi/4G (*OffSEC_W*) and 46 % of the discovered participants via Bluetooth (*OffSEC_B*). In contrast, when the number of participants is more than 300, OffSEC selects less workers than the centralized approach. This can be justified by the impact of the discovered workers by the LENs where the best participants that maximize the QoS are selected as shown in Figure 8.

For a set of participants of 100, OffSEC achieves approximately similar QoS as the centralized approach while this

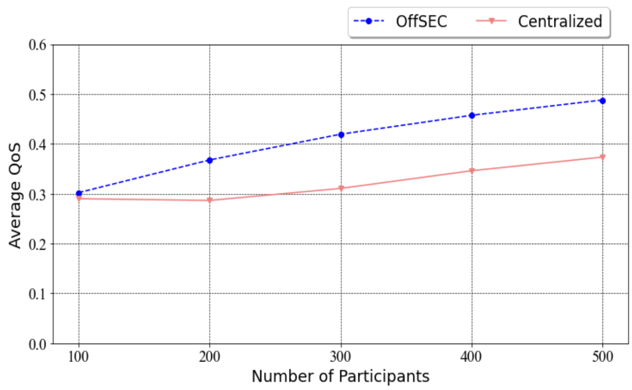


FIGURE 8. Average QoS for OffSEC and centralized under the number of available participants in AoI.

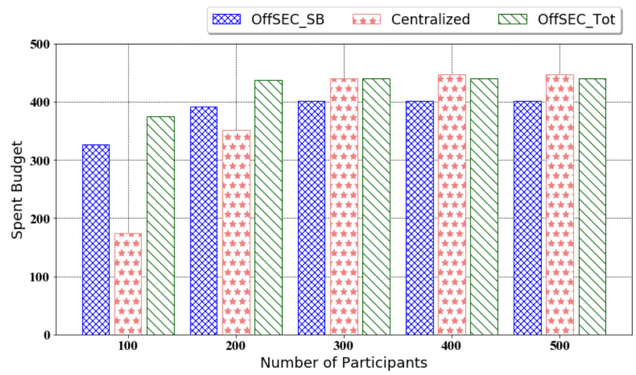


FIGURE 9. Spent budget for OffSEC and centralized under the number of available participants in AoI.

result increases, when the number of participants exceeds 100, by up to 22 %. Overall, OffSEC selects the best workers that provide high device capabilities and improves performances.

It is evident that selecting more workers leads to spending more budget. Figure 9 shows the spent budget for all approaches. For a small number of participants (100 to 200) OffSEC spends more budget for sensing *OffSEC_SB* and the total spent budget *OffSEC_Tot* than the benchmark, while it's lower when the number of participants becomes higher. Overall the proposed approach is an adequate solution for a crowded area where it selects the best workers with less budget and higher QoS.

E. APPROACH VALIDATION USING DATA READINGS

In this section, the quality of the sensing outcome, based on the collected readings, is presented for the two scenarios. For all figures, two parameters are illustrated; the average of the readings(mean) and their standard deviation. The standard deviation measures how much the data is dispersed, where low standard deviation indicates that the data readings are close to the mean, while high standard deviation implies that the data readings are spread out.

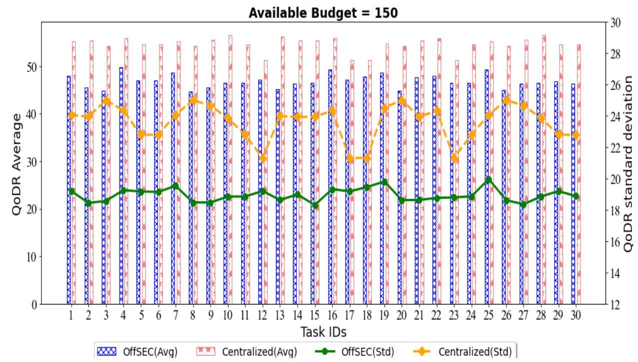


FIGURE 10. OffSEC compared to centralized approach in terms QoDR with limited budget.

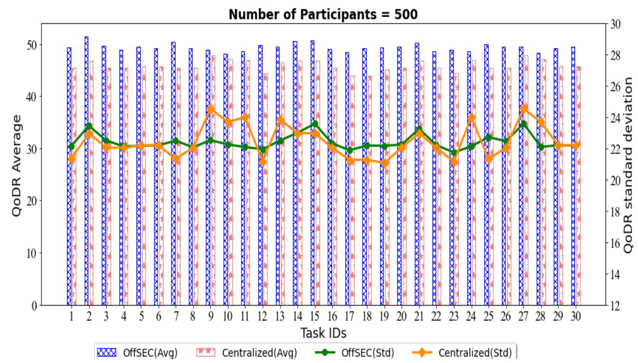


FIGURE 11. OffSEC compared to centralized approach in terms QoDR with 500 participants.

1) SCENARIO 1

Figure 10 presents $QoDR_S^{Tot}$ using limited budgets. Based on $QoDR_S^{Tot}(Std)$, OffSEC has a low standard deviation than the benchmark. By adopting the distributed selection strategy, the quality of data collected from workers is more consistent for all performed tasks. With a budget of 150, as in Figure 10, the benchmark has a standard deviation between 21.30 and 25.01, while this metric does not exceed 19.95 for OffSEC. On average, the benchmark has higher QoDR compared to OffSEC. This QoDR, unlike OffSEC, is relatively varying among tasks. This result is due to the heterogeneity that OffSEC adopts when selecting participants. Also, this selection strategy maximizes the QoS, as well as provides less disperse sensing data.

2) SCENARIO 2

Figure 11 shows the evaluation of the QoDR of a set of 500 participants. As depicted in the figure, the benchmark has lower quality of data reading with a higher variation in terms of standard deviation among the tasks. For tasks 9, 11, 13, 24, 27, and 28, the benchmark displays higher standard deviation which indicates that the data reading provided by workers is spread out over the collected data values. Overall, OffSEC has better QoDR compared to the benchmark. This result is due to non-connected devices discovered via Bluetooth that positively contribute to the data readings.

VI. CONCLUSION

To overcome the limitations of centralized architecture and to benefit from the deployment of MEC in the MCS environment, a two-stage Offloading Selection for MEC in MCS, OffSEC is proposed. In the first layer, two deployments of EN are defined; MEN and LENs. The MEN plays the role of the central point that reports the final collected data to the ES and can be accessible for all LENs. The second layer is the worker's selection, where each LEN selects from the EN around and the devices discovered via Bluetooth, the LEN then selects the final list of workers based on their provided QoS. The selection is processing until the available budget is reached. The proposed approach is evaluated and compared to the benchmark, which is a centralized approach. The simulation results show that OffSEC has better QoS compared to the centralized one by up 30% and 22% for the two different scenarios, respectively. Besides, the QoDR provided by OffSEC is also higher, which is an indication that the collected data is less dispersed. In all simulated scenarios, OffSEC preserves its efficacy in terms of quality of service and quality of data readings. As future work, more parameters can be evaluated such as transmission and computational delay, to address the local computation of data in MEC using the OffSEC framework.

REFERENCES

[1] H. Lamaazi and N. Benamar, "A comprehensive survey on enhancements and limitations of the RPL protocol: A focus on the objective function," *Ad Hoc Netw.*, vol. 96, Jan. 2020, Art. no. 102001.

[2] J. Liu, H. Shen, H. S. Narman, W. Chung, and Z. Lin, "A survey of mobile crowdsensing techniques: A critical component for the Internet of Things," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 3, pp. 1–26, 2018.

[3] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2419–2465, 3rd Quart., 2019.

[4] L. Kong, Z. Wu, G. Chen, M. Qiu, S. Mumtaz, and J. J. P. C. Rodrigues, "Crowdsensing based cross-operator switch in rail transit systems," *IEEE Trans. Commun.*, early access, Aug. 26, 2020, doi: 10.1109/TCOMM.2020.3019527.

[5] J. Portilla, G. Mujica, J.-S. Lee, and T. Riesgo, "The extreme edge at the bottom of the Internet of Things: A review," *IEEE Sensors J.*, vol. 19, no. 9, pp. 3179–3190, May 2019.

[6] C. Li, J. Bai, and J. Tang, "Joint optimization of data placement and scheduling for improving user experience in edge computing," *J. Parallel Distrib. Comput.*, vol. 125, pp. 93–105, Mar. 2019.

[7] C. Li, J. Tang, H. Tang, and Y. Luo, "Collaborative cache allocation and task scheduling for data-intensive applications in edge computing environment," *Future Gener. Comput. Syst.*, vol. 95, pp. 249–264, Jun. 2019.

[8] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 160–168, May 2019.

[9] L. Gu, J. Cai, D. Zeng, Y. Zhang, H. Jin, and W. Dai, "Energy efficient task allocation and energy scheduling in green energy powered edge computing," *Future Gener. Comput. Syst.*, vol. 95, pp. 89–99, Jun. 2019.

[10] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.

[11] C. Li, W. Chen, H. Tang, Y. Xin, and Y. Luo, "Stochastic computation resource allocation for mobile edge computing powered by wireless energy transfer," *Ad Hoc Netw.*, vol. 93, Oct. 2019, Art. no. 101897.

[12] C. Wu, Y. Zhang, and Y. Deng, "Toward fast and distributed computation migration system for edge computing in IoT," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10041–10052, Dec. 2019.

[13] J. Xue, Z. Wang, Y. Zhang, and L. Wang, "Task allocation optimization scheme based on queuing theory for mobile edge computing in 5G heterogeneous networks," *Mobile Inf. Syst.*, vol. 2020, pp. 1–12, May 2020.

[14] F. Yucel, M. Yuksel, and E. Bulut, "Qos-based budget constrained stable task assignment in mobile crowdsensing," *IEEE Trans. Mobile Comput.*, early access, May 25, 2020, doi: 10.1109/TMC.2020.2997280.

[15] X. Gao, H. Huang, C. Liu, F. Wu, and G. Chen, "Quality inference based task assignment in mobile crowdsensing," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 21, 2020, doi: 10.1109/TKDE.2020.2965932.

[16] J. Jang, J. Jung, and J. Hong, "K-LZF : An efficient and fair scheduling for edge computing servers," *Future Gener. Comput. Syst.*, vol. 98, pp. 44–53, Sep. 2019.

[17] T. Wang, H. Luo, X. Zheng, and M. Xie, "Crowdsourcing mechanism for trust evaluation in CPCS based on intelligent mobile edge computing," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, pp. 1–19, Dec. 2019.

[18] C. Li, J. Tang, Y. Zhang, X. Yan, and Y. Luo, "Energy efficient computation offloading for nonorthogonal multiple access assisted mobile edge computing with energy harvesting devices," *Comput. Netw.*, vol. 164, Dec. 2019, Art. no. 106890.

[19] X. Xu, Y. Li, T. Huang, Y. Xue, K. Peng, L. Qi, and W. Dou, "An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks," *J. Netw. Comput. Appl.*, vol. 133, pp. 75–85, May 2019.

[20] A. U. R. Khan, M. Othman, A. N. Khan, J. Shuja, and S. Mustafa, "Computation offloading cost estimation in mobile cloud application models," *Wireless Pers. Commun.*, vol. 97, no. 3, pp. 4897–4920, Dec. 2017.

[21] T. Han, L. Zhang, S. Pirbhulal, W. Wu, and V. H. C. de Albuquerque, "A novel cluster head selection technique for edge-computing based IoMT systems," *Comput. Netw.*, vol. 158, pp. 114–122, Jul. 2019.

[22] S. Li, W. Chen, Y. Chen, C. Chen, and Z. Zheng, "Makespan-minimized computation offloading for smart toys in edge-cloud computing," *Electron. Commerce Res. Appl.*, vol. 37, Sep. 2019, Art. no. 100884.

[23] N. Shan, X. Cui, and Z. Gao, "DRL+FL: An intelligent resource allocation model based on deep reinforcement learning for mobile edge computing," *Comput. Commun.*, vol. 160, pp. 14–24, Jul. 2020.

[24] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, "Learning-based context-aware resource allocation for Edge-Computing-Empowered industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4260–4277, May 2020.

[25] J. Shuja, A. Gani, K. Ko, K. So, S. Mustafa, S. A. Madani, and M. K. Khan, "SIMDOM: A framework for SIMD instruction translation and offloading in heterogeneous mobile architectures," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 4, p. e3174, Apr. 2018.

[26] P. Bellavista, D. Belli, S. Chessa, and L. Foschini, "A social-driven edge computing architecture for mobile crowd sensing management," *IEEE Commun. Mag.*, vol. 57, no. 4, pp. 68–73, Apr. 2019.

[27] M. Marjanovic, A. Antonic, and I. P. Zarko, "Edge computing architecture for mobile crowdsensing," *IEEE Access*, vol. 6, pp. 10662–10674, 2018.

[28] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[29] H. Zhang, F. Guo, H. Ji, and C. Zhu, "Combinational auction-based service provider selection in mobile edge computing networks," *IEEE Access*, vol. 5, pp. 13455–13464, 2017.

[30] J. Huang, Y. Lan, and M. Xu, "A simulation-based approach of qos-aware service selection in mobile edge computing," *Wireless Commun. Mobile Comput.*, vol. 2018, Nov. 2018, Art. no. 5485461.

[31] 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," *J. Syst. Archit.*, vol. 94, pp. 1–13, Mar. 2019.

[32] R. Azzam, R. Mizouni, H. Otrok, S. Singh, and A. Ouali, "A stability-based group recruitment system for continuous mobile crowd sensing," *Comput. Commun.*, vol. 119, pp. 1–14, Apr. 2018.

[33] R. Azzam, R. Mizouni, H. Otrok, A. Ouali, and S. Singh, "GRS: A group-based recruitment system for mobile crowd sensing," *J. Netw. Comput. Appl.*, vol. 72, pp. 38–50, Sep. 2016.

[34] A. Alagha, S. Singh, R. Mizouni, A. Ouali, and H. Otrok, "Data-driven dynamic active node selection for event localization in IoT applications–A case study of radiation localization," *IEEE Access*, vol. 7, pp. 16168–16183, 2019.

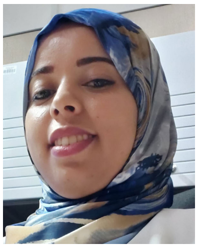
[35] M. Abououf, R. Mizouni, S. Singh, H. Otrok, and A. Ouali, "Multi-worker multi-task selection framework in mobile crowd sourcing," *J. Netw. Comput. Appl.*, vol. 130, pp. 52–62, Mar. 2019.

[36] M. Abououf, S. Singh, H. Otrok, R. Mizouni, and A. Ouali, "Gale-shapley matching game Selection—A framework for user satisfaction," *IEEE Access*, vol. 7, pp. 3694–3703, 2019.

[37] A. Alagha, S. Singh, H. Otrok, and R. Mizouni, "RFLS–Resilient fault-proof localization system in IoT and crowd-based sensing applications," *J. Netw. Comput. Appl.*, vol. 170, Nov. 2020, Art. no. 102783.

[38] M. Kadadha, R. Mizouni, S. Singh, H. Otrok, and A. Ouali, "ABCrowd an auction mechanism on blockchain for spatial crowdsourcing," *IEEE Access*, vol. 8, pp. 12745–12757, 2020.

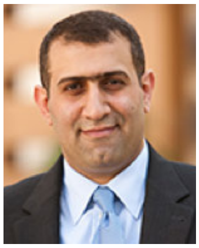
[39] A. Chang, "Statstodo: Combine means and SDS into one group program," Tech. Rep.



HANANE LAMAAZI received the M.Sc. degree in networks and telecommunications from the Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morocco, in 2013, and the Ph.D. degree in network and computer sciences from the Faculty of Sciences, Moulay Ismail University, Meknes, Morocco, in 2018. She is currently a Postdoctoral Fellow with the Electrical and Computer Engineering Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. Her research interests include the Internet of Things (IoT), RPL routing protocol, edge computing, and crowdsensing. She is also a TPC member in different conferences (SRC 2020, 3ICT, EAI AFRICOMM, PDCTA, and CRYPIS). She is acting as a Reviewer of several journals, such as *Computer Networks*, *Ad Hoc Networks*, IEEE SENSORS, IEEE ACCESS, and IJCS.

RABEB MIZOUNI received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Concordia University, Montreal, Canada, in 2002 and 2007, respectively. She is currently an Associate Professor in electrical and computer engineering with the Khalifa University of Science and Technology. Her research interests include the deployment of context aware mobile applications, crowd sensing, software product line, and cloud computing.

SHAKTI SINGH (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, USA. He is currently an Assistant Professor with the Electrical and Computer Engineering Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. His research interests include semiconductor devices and integrated circuits, sensors, sensing technologies, crowd sourcing, crowd sensing, and the development of the IoT and wireless sensor networks.



HADI OTROK (Senior Member, IEEE) received the Ph.D. degree in ECE from Concordia University, Montreal, Canada. He holds an Associate Professor position with the Department of ECE, Khalifa University of Science and Technology, an Affiliate Associate Professor with the Concordia Institute for Information Systems Engineering, Concordia University, and an Affiliate Associate Professor with the Electrical Department, École de Technologie Supérieure (ETS), Montreal. His research interests include the domain of computer and network security, web services, ad hoc networks, application of game theory, and cloud security. He has co-chaired several committees at various IEEE conferences. He is an Associate Editor of *Ad Hoc Networks* (Elsevier), the IEEE COMMUNICATIONS LETTERS, *Wireless Communications*, and *Mobile Computing* (Wiley).