Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 17 No. 16 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijim.v17i16.42685

PAPER

Recruitment Algorithm in Edge-Cloud Servers based on Mobile Crowd-Sensing in Smart Cities

Muhammad Alkirom Wildan[1](✉), Mahmudah Enny Widyaningrum[2], T. Padmapriya[3], Basant Sah[4], Niroj Kumar Pani[5]

[1]Department of Management, Faculty of Economics and Business, University of Trunojoyo Madura, Bangkalan, Jawa Timur, Indonesia

[2]Faculty of Economics and Business, Universitas Bhayangkara Surabaya, Surabaya, Indonesia

[3]Melange Publications, Puducherry, India

[4]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

[5]Department of Computer Science Engineering and Applications, Indira Gandhi Institute of Technology, Sarang, Odisha, India

wildan.alkirom69@trunojoyo.ac.id

ABSTRACT

As more and more mobile devices rely on cloud services since the introduction of cloud computing, data privacy has emerged as one of the most pressing security concerns. Users typically encrypt their important data before uploading it to cloud servers to safeguard data privacy, which makes data usage challenging. On the other side, this also increases the possibility of brand-new issues in cities. A clever, effective and efficient urban monitoring system is required to address possible challenges that may arise in urban settings. In the smart city concept, which makes use of sensors, one strategy that might be used in IoT and cloud computing is to monitor and gather data on problems that develop in cities in real-time. However, it will take a while and be rather expensive to install IoT and sensors throughout the city. The Mobile Crowd-Sensing (MCS) method is proposed to be used in this study to retrieve and gather data on issues that arise in metropolitan areas from citizen reports made using mobile devices. And we suggest a budget-constrained, reputation-based collaborative user recruitment (RCUR) procedure for a MCS system. To construct an edge-assisted MCS system in urban situations, we first integrate edge computing into MCS. We also examine how user reputation affects user recruitment. Finally, we create a collaborative sensing approach using the edge nodes' sensing capabilities.

KEYWORDS

collaborative sensing, user recruiting algorithm, mobile crowd sensing (MCS), edge cloud servers (ECSs), smart city

1 INTRODUCTION

Mobile devices worn by people increasingly include sensors, including accelerometers, gyroscopes, sensor technology and Global Positioning System (GPS) locators that can gather a variety of data. In order to gather sensing data and carry out various sensing functions, including navigation assistance, tracking traffic, indoor location and environmental monitoring, the MCS system can make use of the sensors built into mobile devices. The task requester publishes detecting tasks and gives

Wildan, M.A., Widyaningrum, M.E., Padmapriya, T., Sah, B., Pani, N.K. (2023). Recruitment Algorithm in Edge-Cloud Servers based on Mobile Crowd-Sensing in Smart Cities. *International Journal of Interactive Mobile Technologies (iJIM)*, 17(16), pp. 116–128. https://doi.org/10.3991/ijim.v17i16.42685

Article submitted 2023-05-06. Resubmitted 2023-06-18. Final acceptance 2023-06-25. Final version published as submitted by the authors.

rewards for sensing data. The MCS system, in general, consists of the task requester, sensing server, and participants [1]. The server gathers users for the sensing task, analyses their data and communicates the findings to the task publisher. The general structures of MCS and its participating entities are illustrated in Figure 1.

In the real-world MCS system, user-collected sensing data are occasionally unreliable because of a variety of challenges, such as subpar sensor performance, laziness and background noise. As a result, if we average the data supplied by each user, for example, the final result may not be correct. Truth finding is a significant issue of interest to both industry and academics. The primary tenet of the majority of truth discovery algorithms is to assign greater importance (i.e., dependability) to users whose data are more accurate than the ground truth. Additionally, if a user has a higher weight, their data will be counted more in the collection process [2]. Based on this fundamental concept, several truth-finding techniques have recently been put forth to determine user weight and aggregated outcomes. However, one drawback of these approaches is that they require users to be online to communicate with the web server. Otherwise, it can malfunction, and you will need to restart the MCS system. Therefore, the MCS system can become more resilient if we create a truth-finding mechanism that permits users to depart. The truth discovery process needs a sufficient number of operators and high-quality sensing data to operate well. The MCS system often uses an incentive system to encourage enough people to take part in detecting duties.

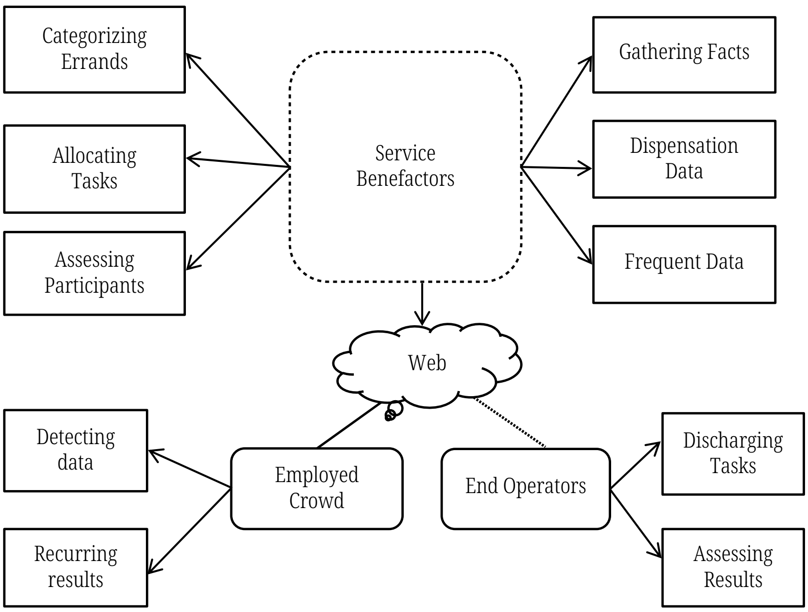

Fig. 1. Principal components and activities of a general MCS system in smart cities

The majority of current mobile crowd-sensing (MCS) systems rely on cloud services to gather and recognise cumulative records, distribute errands, evaluate veracity and motivate mobile managers. A few clear disadvantages of cloud-based MCS systems include their limited scalability for large-scale crowd-sensing due to computational and message load on cloud servers, their difficulty in identifying fake sensing data locations, and their high risk of information safety and user privacy contact. Lately, a MCS design based on edge computing has been proposed to address the aforementioned problems [3]. The following are the key advantages of an architecture based on edge computing for extensive mobile crowd-sensing:

- Reduce computational complexity by offloading processing from the cloud to numerous edge servers, which is possible with an edge computing-based mobile crowd-sensing system.
- Reduce the dormancy: There is little to no communication required between the cloud and mobile devices.
- The majority of MCS jobs depend on location. Edge computing resources include, for instance, base stations and access points that are typically found in particular places. Because edge servers only gather information from within the installation region, it is easy to determine the location of sensor data. For instance, when tracking traffic, noise, or people in particular places, the precision of the location information has a significant impact on the sensing data.
- Data processing with flexibility: Mobile crowd-sensing based on edge computing offers edge servers the capacity to process local data, including gathering, establishing the truth and drawing conclusions about humidity, noise level, transportation and cooling for specific locations. The edge cloud can be used for tasks that don't need to be carried out in the deep cloud, such as analysing local traffic footage or estimating the amount of noise in the area.
- Reduced privacy threats thanks to the distribution of the sensing data over several edge servers. The dispersed storage of sensor information across the presence of many edge servers enhances data security while reducing threats to user confidentiality. For instance, crowd-sensing data concerning a person's home surroundings or images is private information and is best handled by edge computers.

The remainder of this paper is structured as follows: We begin by discussing the associated works in Section 2. The MCS platform's structure and key design concept are then introduced in Section 3. In Section 4, we present the platform's implementation details and the simulations used to evaluate the effectiveness of our suggested algorithm. Section 5 is where we wrap up this paper.

2 RELATED WORKS

According to Jiang, W., Chen, P., Zhang, et al. [4], the work done by academics in the area of MCS task allocation can be primarily comprised of two features: using an appropriate incentive system to encourage users to provide information to actively gather high-quality data while working within financial constraints to choose users who completed activities with significant perceived worth. In terms of user hiring, we investigated, suggested and provided multitask strategy option as well as a new user-selecting method that empowers users to participate in a variety of perceptual duties. The best user keeps the study's overall distance to a minimum, which lowers the platform's expense.

Liwang, M., Gao, Z., Hosseinalipour, et al. [5] have conducted studies on MCSC, focusing on the development of both online and offline trading techniques for hiring workers. When doing online decision-making (such as worker-task allocation), online MCSC takes into account the most recent information about workers and tasks during each actual trade (for example, information connected with a network picture). Online trading encounters significant challenges in adjusting to the unpredictable and dynamic character of mobile networks, even though it can capture the present job, worker, and network circumstances and produce quite accurate conclusions.

Ayu, V et al. [6] have highlighted the importance of having comprehensive and complete knowledge in MCS which can be achieved by obtaining a substantial amount of data. However, recruiting enough participants to carry out the sensing work can be challenging. Data quality, sensing costs and privacy assurance are some factors in the data collection procedure that are important to consider. Due to the comprehensive sensing coverage, a great number of participants are helpful in the data collection process; nevertheless, not all of the obtained data are of the same quality, despite the vast volume of data that is produced. The quantity of recruited individuals and the calibre of sensed data must therefore be optimised.

Zhao, Y., Li, Z., Chen, et al. [7], focused on the assignment of tasks, which is a crucial issue in crowd sensing. Due to the use of various algorithms, there are significant discrepancies in the price of the crowd-sensing system and the effectiveness of task completion. As a result, choosing an appropriate work allocation mechanism is crucial. They used task- and people-centred methods to choose participants for multitasking. To solve the issue of task allocation, these techniques offer the lowest detection quality standards that are task-specific, but they do not consider how to decide when many different tasks must be dispersed.

Gad-Elrab, A., and Noaman, A. Y. et al. [8] investigated the impact of sensing information regarding various privacy protections for MCS systems and developed two perturbation procedures for two distinct viewpoints. They presented a system that, from the viewpoint of the guardian, employs the Bayesian network's concept of the probabilistic significance of the sensing data, and the scale value is calculated using the widely accepted definition of privacy differential. To characterise the correlation of the information, the adversary, from their perspective, presented a technique that evaluates the relevance of the most linked group using the Gaussian correlation theory to compute the Bayesian distinguishing leakage of private information.

Girolami, M., Belli, D., et al. [9] discussed the typical MCS design, where users must regularly receive tasks from distant servers. Users may participate in tasks directly (by offering comments or responding to a survey, for example), or tasks may be carried out automatically without human involvement. In this regard, it is important to point out that certain jobs, such as gathering GPS location, sampling the surroundings, evaluating noise power, or evaluating the signal strength of Wi-Fi connections, permit obtaining sensory information from a gadget. In both situations, scheduling the appropriate job at the appropriate time will maximise the likelihood of effectively obtaining the desired data from the intended users. We also take into account the potential for delivering consumers' so-called geo-referenced assignments.

Concone, F., Lo Re, G., and Morana et al. [10] Due to its adaptability for a variety of use scenarios, the literature has devoted a lot of time to sensor-based crowd-sensing. By utilising pre-existing software libraries like the iOS Core Motion framework and the Google Activity Recognition APIs, simple mobile interactive sensing apps can be created. However, several works emphasised the need for systems that allow different devices, including mobile phones with limited resources, to cooperate when performing more complicated tasks, like real-time human activity recognition. The cloud computing concept can be applied in this case to develop efficient systems that can swiftly and locally analyse enormous volumes of information.

3 METHODS AND MATERIALS

The edge system that we propose in this paper is made up of a collection of ECSs and several MDs, as the main deployment technique for MEC [11]. We presume that an MD selects a set (R = 1, 2,...,r) of ECSs to which it will outsource its calculations, representing the complete ECS computation for all R represents the calculation loaded from the MD to the ECSs, [0, 1, 2].

3.1 A mobile device

Using the assumption that the calculation's offloaded profile is x = (1,2...r), $\varnothing(y)$ is the local residual unfolded calculation provided by

$$\varnothing(y) = \sum_{j=1}^{R}(Y_j - Y_i) \tag{1}$$

And calculations made entirely locally without loading are provided by

$$\varnothing(1) = \sum_{j=1}^{R}(Y_j) \tag{2}$$

The price of mobile device computing locally is provided by

$$E(Y) = f^{\partial(\varnothing(y))} \tag{3}$$

The modelling parameter's location. The payment scheme is n = (1,2,...r). NY is the price for loading determined by provided n and m

$$N(Y) = \sum_{j=1}^{R}(n_j) \tag{4}$$

When the payment q_j is based on the cost of the loading unit and n_j is the cost of the calculation, that is $q_j.y_j$,

$$n_j = q_j.y_j \tag{5}$$

We count the time spent performing computational offloading as a component of the offloading expenses, together with the time it takes to compute any local remaining calculations and transfer any data that is required. $L(Y)$ is a measure of the MD's transmission capability as well as its computational capacity. The amount of time spent loading performance is represented by

$$L(Y) = \sum_{j=1}^{R}(L_j^{exe} - L_j^{off}) \tag{6}$$

Where L_j^{exe} denotes the time spent performing local remaining calculations and ∂ is determined by

$$L_j^{exe} = \partial.\frac{y_j}{L_d} \tag{7}$$

Where L_j^{off} is a measure of the computation's difficulty. $\cap$ Reflects the time required for transferring the data required for computational operations and $\frac{y_j}{L}$ is provided by

$$L_j^{off} = \cap . \frac{y_j}{L} \tag{8}$$

Where $D(Y)$ is the coefficient of the data needed for computations to be transferred. Therefore, $f^{\partial(\varnothing(y))}$ the time spent computing, transmitting information and $N(Y)$ paying for ECSs all contribute to the price of the MD for conducting calculating loading, i.e.

$$D(Y) = E(Y) + N(Y) + L(Y) = f^{\partial(\varnothing(y))} + \sum_{j=1}^{R}(n_j) + \sum_{j=1}^{R}(L_j^{exe} - L_j^{off}) \tag{9}$$

The local usefulness is defined as the price savings from conducting loading, i.e.

$$\complement = D(1) - D(Y) = D(1) - E(Y) - N(Y) - L(Y) \tag{10}$$

3.2 A summary of Mobile Crowd-Sensing in smart cities

We initially provide a general structure for MCS systems in this section. Then, we give the standard MCS classifications. We then conclude by outlining the essential characteristics, benefits and difficulties of MCS in smart cities.

The framework of mobile crowd-sensing systems. In this section, we outline the overall design and structural elements of MCS systems.

The architecture of mobile crowd-sensing systems. In the research, several MCS architectural models have been suggested. We introduce the most prevalent architectures in this section [12]. Several objects, including service breadwinners, end operators and employed groups, make up the architecture. A platform that manages chores and offers end users crowd-sourced services is typically referred to as a service provider. Customers in smart cities are often either linked cars that publish one or more tasks or pedestrians who are carrying mobile devices. The service provider distributes the jobs to the proper working groups. A worker can execute a task while utilising their mobile gadgets, such as computers, cameras, smartphones and smartwatches, as well as smart vehicles and other mobile devices.

Primary purposes. The primary roles of service providers, consumers in an MCS system, and people who work are depicted in Figure 2. After receiving an inquiry from a client, the service provider classifies the work into different categories. Then, it assigns modest jobs to willing workers and awaits the results of the tasks. The platform evaluates the data after receiving all the results and then provides the end users with the final results, completing the crowdsourcing assignment. The supplier of service will assess everyone involved after they have completed several tasks to determine whether or not their behaviour is honest. Additionally, platform users can contribute their ideas and assess the outcomes.

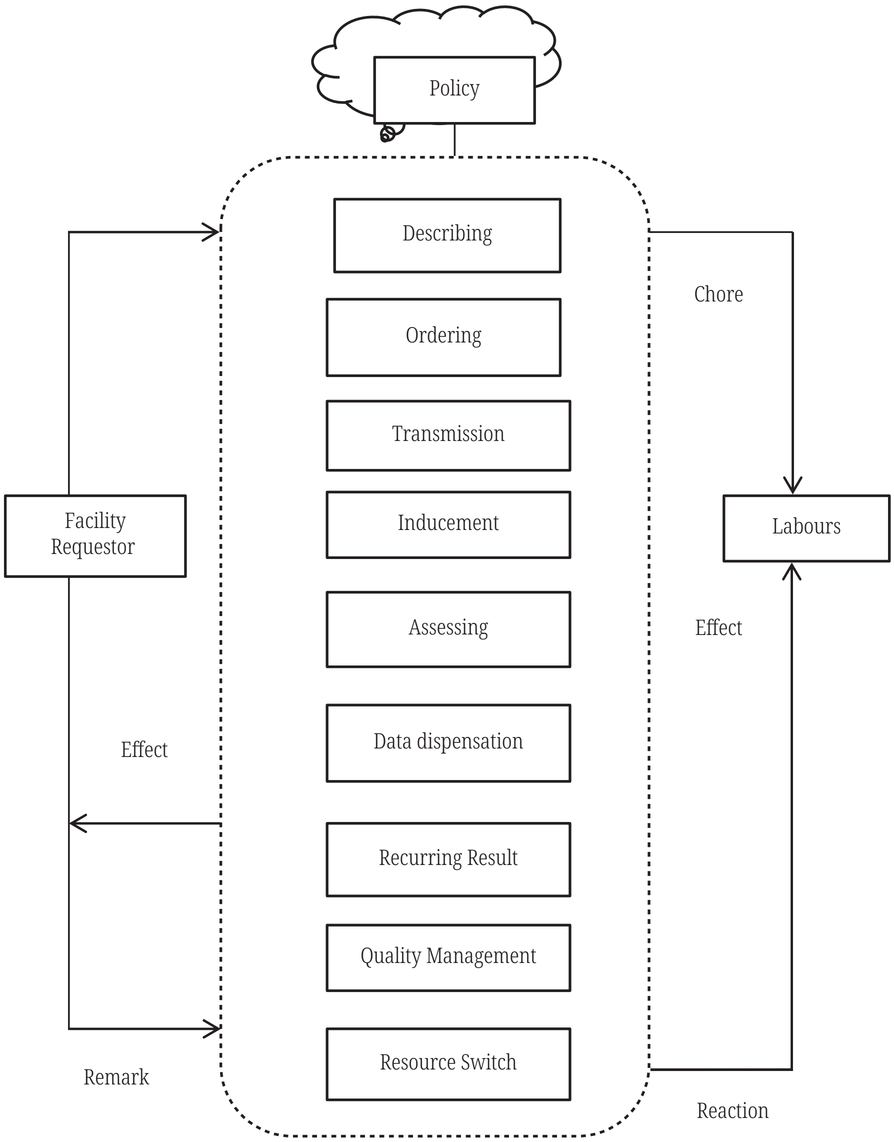

Fig. 2. The overall procedures for carrying out an MCS assignment

To do this, the platform evaluates whether the assets at its disposal can guarantee that the work will be completed with the requested level of service. Different kinds of storage, computing and electrical power could be required for a certain task. Therefore, the platform must assess whether its resources are adequate with the goal of ensuring excellence.

The platform seeks to identify the best solution for the assignment once its viability has been confirmed. It then evaluates the sufficiency of the funding offered by the task providers, and based on the earnings and expenses, a work might be accepted or rejected by the crowdsourcing network.

Problem creation and system modelling. There are still very few pre-deployed fixed nodes, especially in metropolitan environments that have powerful sensing, collecting, determining and interacting abilities [13, 14]. These nodes include smart posts, signal stations, smart cameras, and other devices that can also collect sensing data. Therefore, in addition to using mobile users to obtain pertinent data and execute activities, we also make use of these edges with nodes as a supplementary resource to create a new crowd-sensing platform. Our framework is a three-layer hierarchical network, as depicted in Figure 3.

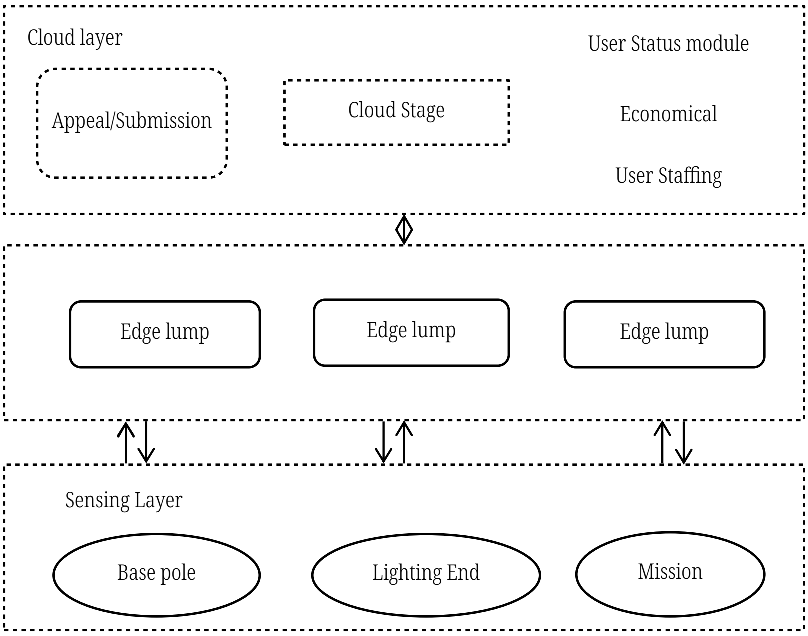

Fig. 3. MCS system with edge support

A cloud cover: The cloud layer handles job publication and also offers computational and data-gathering resources. It receives tasks from multiple users or consumers and then publishes them to the users in its capacity as a centralised controller. The cloud policy facilitates communication between the other two tiers and maintains user personal facts, such as histories, whereabouts, and reputations [17, 18]. The cloud platform in our system will set up a manipulator popularity component to choose the appropriate users based on their previous ability to carry out the activities effectively and efficiently. Additionally, the cloud layer serves as a data processing hub that will analyse and compile the detected data before packaging and delivering it to the apps and consumers.

Border layer. Between the cloud foundation and the mobile sensing end devices lies the edge layer, as depicted in Figure 3. The pre-deployed fixed nodes would be the ideal border nodes in the crowd-sensing system because they have strong computation and communication abilities, which can support and improve the efficiency of the mobile final devices. Smart static nodes like clever lampposts and headquarters can take the place of edge nodes in urban settings. The edge layer, which is located closer to the users and uses 5G, 6G, or WiFi to link heterogeneous handheld gadgets to the cloud service, can relieve the load on the cloud's processing power, decrease latency and expand data storage capacity.

Sensing level. Mobile users that use intelligent tablets with several integrated sensors are the primary component in the sensing layer of crowd-sensing technology and play a crucial role in detecting and gathering diverse data at specified times in a particular place. We presume that sensed data are reliable and that all users voluntarily participate in tasks. The flexibility of ubiquitous detection is substantially increased by the ad hoc mobility of mobile users, and the static nodes can guarantee the level of protection of the sensing networks. Mobile and static sensors can work together and interact to improve the efficiency and standard of sensing and data interpretation.

Algorithm 1: A cooperative user recruitment algorithm based on reputation
Input: **Output:** 1. *in V do for you* 2. *Duj, θj≤Dmax,tuj∈tθj* 3. *The applicant must set SCan=(u2,u3,...,ux)* 4. *Finish* 5. *for v in SCan do* 6. *Inform their ui(u) rating from the past* 7. *Determine the user's immediate popularity score (instui(t))* 8. *Obtain the user's popularity score using ui(v)* 9. *Compute fv of uSCan* 10. *Choice an uiSCan with the highest fu* 11. *End* 12. *if SCan=∅* 13. *Compute fSn of Sm.* 14. *Choose Sn to maximise fSn* 15. *Finish*

To achieve higher task completion rates and greater coverage, this will assist the cloud-based system in finding the person best suited to carry out the task efficiently and successfully. It is an algorithm that focuses on tasks to enhance the advantages of tasks and platforms in the cloud. The greedy method is then used to further aid in problem-solving. The RCUR algorithm's precise procedure is outlined below and displayed in Algorithm 1.

4 IMPLEMENTATION AND EXPERIMENTAL RESULTS

We perform in-depth simulations to validate the techniques and strategies we've suggested. We set up three ECSs as the starting standard parameters for the simulations we ran. The total number of ECSs computed is 100. We suppose that an area with a size of 1:1 is covered by the MD and the ECS. The MDs are distributed uniformly at random, but the ECSs are situated at grid locations around the area. We adjusted the modelling parameter in the experiments to 1.5 and the computing complexity parameter to 1.

Table 1. Exploratory setting of parameters

Signs	Values	Description
X(i)	0.05	Unity of MD
3 ECSs	0.06	Unity of ECS
Increases	0.03	Constraints
1km*1km	0.08	Accountable Time

The Repast platform, which is open source, is used for this simulation experiment [15, 16]. The laboratory setting and parameter settings for this study are identical to ensure objectivity and fairness. Details are shown in Table 1.

Edge cloud servers high utility is attributable to their willingness to perform computation for their users, which results in efficient computation loading. Surprisingly, the ECSs perform worse when the income is high enough, for it is well known that they are driven by profit to offer computation to their customers. However, for ECSs, high unit income might be a double-edged sword. Although edge servers are more ready to help their clients, this may result in a reduction in computing demand. The graph demonstrates how the value of ECSs starts to rise as the MD's payment rises and starts to fall after reaching a peak. The usefulness of ECSs first increases as a result of the higher payment, but since this raises the price of the MD, it won't continue to rise.

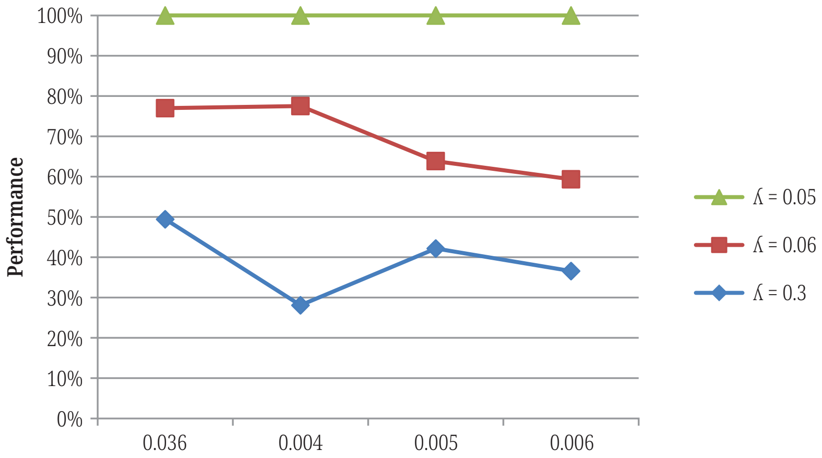

Fig. 4. Compensation of MD

We established three distinct unit revenues that were used in the simulation to examine how the unit revenue impacts the offloading computations. We set the values to 0.036, 0.004, and 0.006 correspondingly. Figure 4 illustrates how the patterns of the various groups are comparable and how the computation for loading rises as the payment does. In other words, for the three-unit incomes discussed above, the calculation grows as the MD's payment grows. Note that the ECSs will not do any loading if the unit revenue exceeds the MD's given compensation.

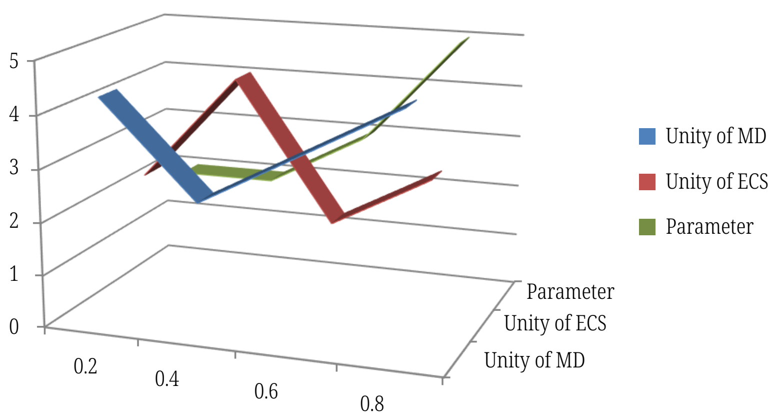

Fig. 5. Effect of computational precision

The utility's effect and response time. We are interested in the role of precision in the proposed method. As a result, we chose a range of values between 0.2 and 0.8. The results of the simulation are shown in Figure 5. As the value rises, our method becomes more accurate, MD becomes more useful, and ECSs perform better. Calculating the benefits of loading will be more advantageous to the MD and the edge cloud servers.

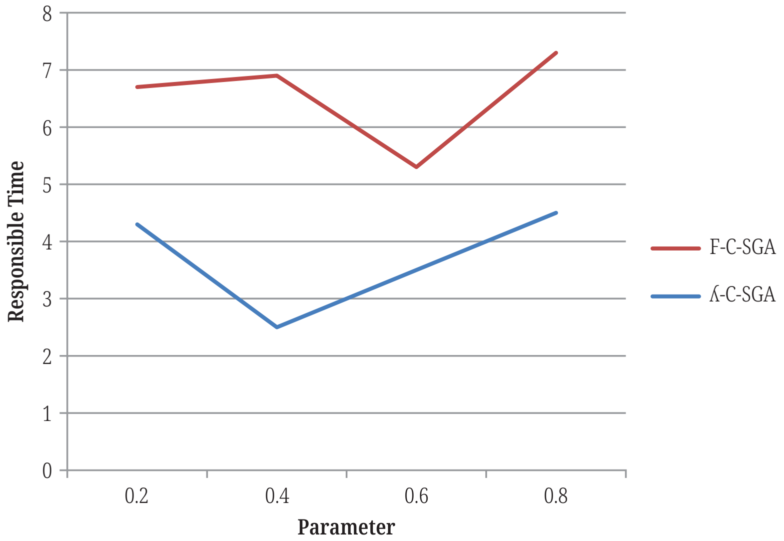

Fig. 6. Time to response for F-SGA and C-SGA

Figure 6 compares the F-SGA and C-SGA's response times under the same computationally loaded circumstances. The graph shows that the reaction time of C-SGA decreases as the parameter increases. The F-SGA's response time, however, has remained constant and is currently at a low level. This is because when the value of a parameter increases, the CSGA must perform more iterations and judgement rounds, increasing the reaction time. However, F-SGA may always decide swiftly, irrespective of the parameter's value.

5 CONCLUSION

Mobile crowd sensing is a practical platform for several applications thanks to the rise of IoT and the widespread use of sophisticated mobile devices. Edge computing's debut has, in turn, hastened the advancement of MCS. We focus on technology that makes MCS possible in smart cities, such as managing tasks, data collecting, incentive systems, monitoring, and cost-cutting tools. Finally, we outlined several unresolved issues and potential study directions. We hope that this article's contributions will broaden the scope of an ongoing study in this challenging area and serve as an inspiration to claim and system designers to produce interesting MCS solutions for smart cities.

One of our upcoming projects is an algorithm for dynamic user recruitment. In general, user recruitment always includes the recruitment of one person to carry out a certain duty. But how to create the best algorithm to choose a group of users when the jobs call for additional people to sense this data has been a significant issue in MCS. Another challenging issue is how the user's properties, such as cover capacity, reputation, and cost, can be used to dynamically compute the number of people required for the jobs. Moving forward, we intend to concentrate on this dynamic user onboarding in mobile crowd sensing.

6 REFERENCES

[1] X. Zhang, L. Liang, C. Luo, and L. Cheng, "Privacy-preserving incentive mechanisms for mobile crowdsensing," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 47–57, 2018. https://doi.org/10.1109/MPRV.2018.03367734

[2] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019. https://doi.org/10.1109/JIOT.2019.2921879

[3] J. Xu, S. Yang, W. Lu, L. Xu, and D. Yang, "Incentivizing for truth discovery in edge-assisted large-scale mobile crowdsensing," *Sensors*, vol. 20, no. 3, p. 805, 2020. https://doi.org/10.3390/s20030805

[4] W. Jiang, P. Chen, W. Zhang, Y, Sun, C. Junpeng, and Q. Wen, "User recruitment algorithm for maximizing quality under limited budget in mobile crowdsensing," *Discrete Dynamics in Nature and Society*, vol. 2022, pp. 1–13, 2022. https://doi.org/10.1155/2022/4804231

[5] M. Liwang, Z. Gao, S. Hosseinalipour, Z. Cheng, and X. Wang, "Long-term or temporary? Hybrid worker recruitment for mobile crowd sensing and computing," arXiv preprint arXiv:2206.04354, 2022.

[6] V. Ayu, "Revisiting mobile crowdsensing: An open challenge," In *Proceedings. CS & IT Conference*, vol. 11, no. 15, 2021. https://doi.org/10.5121/csit.2021.111509

[7] Y. Zhao, Z. Li, X. Chen, and Y. Chen, "Mobile crowd sensing service platform for social security incidents in edge computing," In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 568–574, 2018. https://doi.org/10.1109/Cybermatics_2018.2018.00118

[8] A. Gad-Elrab and A.Y. Noaman, "Clustering ant colony-based edge-server location strategy in mobile crowdsensing," *Applied Computational Intelligence and Soft Computing*, vol. 2022, pp. 1–17, 2022. https://doi.org/10.1155/2022/2998385

[9] M. Girolami, D. Belli, S. Chessa, and L. Foschini, "How mobility and sociality reshape the context: A decade of experience in mobile crowdsensing," *Sensors*, vol. 21, no. 19, p. 6397, 2021. https://doi.org/10.3390/s21196397

[10] F. Concone, G. Lo Re, and M. Morana, "SMCP: A secure mobile crowdsensing protocol for fog-based applications," *Human-Centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–23, 2020. https://doi.org/10.1186/s13673-020-00232-y

[11] M. Li, Q. Wu, J. Zhu, R. Zheng, and M. Zhang, "A computing offloading game for mobile devices and edge cloud servers," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11, 2018. https://doi.org/10.1155/2018/2179316

[12] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019. https://doi.org/10.1109/JIOT.2019.2921879

[13] Y. Liu, Y. Li, W. Cheng, W. Wang, and J. Yang, "A reputation-based collaborative user recruitment algorithm in edge-aided mobile crowdsensing," *Applied Sciences*, vol. 13, no. 10, p. 6040, 2023. https://doi.org/10.3390/app13106040

[14] S. Muruganandam, A. Anas, Salameh, Mohd Affendi Ahmad, S.V. Manikanthan, and T. Padmapriya, "Sensors and machine learning & AI operation-constrained process control method for sensor-aided industrial internet of things and smart factories," *Measurement: Sensors*, vol. 25, p. 100668, 2023. https://doi.org/10.1016/j.measen.2023.100668

[15] D. Belli, S. Chessa, L. Foschini, M. Girolami, "A probabilistic model for the deployment of human-enabled edge computing in massive sensing scenarios," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2421–2431, 2019. https://doi.org/10.1109/JIOT.2019.2957835

[16] J. Liu, X. Shen, W. Liu, Z. Lv, R. Liu, and D. Li, "Decision analysis under behavioral economics—Incentive mechanism for improving data quality in crowdsensing," *Mathematics*, vol. 11, no. 10, p. 2288, 2023. https://doi.org/10.3390/math11102288

[17] U.N. D, U. S, L. Tamilselvan, and S.N. J, "Client aware scalable cloudlet to augment edge computing with mobile cloud migration service," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 12, pp. 165–178, 2020. https://doi.org/10.3991/ijim.v14i12.14407

[18] S. Maftah, M. El Ghmary, H. El Bouabidi, M. Amnai, and A. Ouacha, "Optimal task processing and energy consumption using intelligent offloading in mobile edge computing," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 16, no. 20, pp. 130–142, 2022. https://doi.org/10.3991/ijim.v16i20.34373

7 AUTHORS

Prof. Dr. Muhammad Alkirom Wildan is an expert in the field of Human Resource Management, which synergizes with technological developments. Currently, he is Lecturer in the Faculty of Economics and Business, at Universitas Trunojoyo Madura. In addition, he is currently involved in research in Artificial Intelligence and Information Systems in the field of Human Resource management and, to his credit, has authored numerous books and journal publications (E-mail: wildan.alkirom69@trunojoyo.ac.id).

Prof. Dr. Mahmudah Enny Widyaningrum is a professor in the Department of Management at Universitas Bhayangkara Surabaya. She is active to contributing to the academic climate, including publishing numerous reputed and high-impact journals (E-mail: ennymahmudah@yahoo.co.id).

Dr. T. Padmapriya is a Managing Director of Melange Publications, Puducherry, India. She obtained her PhD at Pondicherry Engineering College, Puducherry, India. Her area of research interest is LTE and wireless networks. She has number of international publications to her credit and is a reviewer for various international journals (E-mail: padmapriyaa85@pec.edu).

Dr. Basant Sah received M.Tech (CS) degree from Birla Institute of Technology Mesra, Ranchi, Jharkhand (India) in 2008 and Ph.D. in engineering from same university in 2021. Currently he is working in KL University as an Associate Professor in CSE Department. He has published more than 20 research paper in national and international reputed Journals. His are of interest includes information security, security in cloud, network security etc. (E-mail: basantbitmtech2008@gmail.com).

Dr. Niroj Kumar Pani received M. Tech in Computer Science with a specialization in Information Security from the National Institute of Technology, Rourkela, India in 2009, and Ph.D. in Computer Science from Utkal University, Bhubaneswar, India in 2018. He had worked as an Assistant Professor at Indian Institute of Science and Information Technology, Bhubaneswar, India, and as a senior analyst in ProcessMAP Corporations, India. At present, he is working as an Assistant Professor in the Department of Computer Science Engineering and Applications at Indira Gandhi Institute of Technology, Sarang, Odisha, India. He has authored many research articles in reputed international journals and books. His research interests include network security, wireless ad hoc and sensor networks, cloud computing, IoT, machine learning, and artificial intelligence.