PAPER

Data-Quality Aware Incentive Mechanism Based on Stackelberg Game in Mobile Edge Computing

Shuyun LUO[†], **Wushuang WANG**[†], **Yifei LI**[†], **Jian HOU**[†a)], *Nonmembers*, *and* **Lu ZHANG**[†], *Member*

SUMMARY Crowdsourcing becomes a popular data-collection method to relieve the burden of high cost and latency for data-gathering. Since the involved users in crowdsourcing are volunteers, need incentives to encourage them to provide data. However, the current incentive mechanisms mostly pay attention to the data quantity, while ignoring the data quality. In this paper, we design a Data-quality awaRe IncentiVe mEchanism (DRIVE) for collaborative tasks based on the Stackelberg game to motivate users with high quality, the highlight of which is the dynamic reward allocation scheme based on the proposed data quality evaluation method. In order to guarantee the data quality evaluation response in real-time, we introduce the mobile edge computing framework. Finally, one case study is given and its real-data experiments demonstrate the superior performance of DRIVE.

key words: *mobile edge computing, crowdsourcing, incentive mechanism*

1. Introduction

In recent years, crowdsourcing [1] has emerged as an innovative and cost-effective method for data collection. Crowdsourcing uses the Internet to distribute tasks, discover more ideas or solve technical problems. This approach significantly reduces both cost and time needed to complete tasks. However, while crowdsourcing has garnered attention for its ability to collect vast amounts of data, the aspect of data quality is often overlooked. Since most crowdsourcing participants are volunteers, ensuring the reliability and accuracy of the collected data becomes challenging [2]. Data quality plays a significant role in determining the Quality of Service (QoS), particularly in collaborative tasks that involve a group of users working together [3]. Therefore, it becomes crucial to design an incentive mechanism that dynamically adjusts users' rewards based on the quality of the data they provide.

Although there have been studies on data quality evaluation and incentive mechanism design, they are often only applied for specific application scenarios and fails to have effective integration. Consequently, the design of incentive mechanisms based on data quality awareness still faces several challenges: 1) Data quality is influenced by various factors, such as user data collection capabilities and network transmission stability [4], [5]. How to access the data quality precisely? 2) Since the complexity of data quality evaluation may hinder real-time assessment [6], how to achieve timely and reliable data quality feedback? 3) How to classify users effectively based on their data quality's impact on task completion, which is vital for designing appropriate reward allocation schemes? 4) How to design an incentive mechanism such that it can encourage users to upload high-quality data while discouraging free-riding behaviors [7] among users submitting low-quality data?

To address the above challenges, we first present a data-quality evaluation method from both signal and semantic [8] perspectives to classify users into three categories: good users, malicious users and users under unstable networks. Moreover, we introduce a crowdsourcing architecture based on Mobile Edge Computing (MEC) [9], [10] to guarantee the timeliness of data collection, where the data quality can be evaluated by edge server rather than the remote cloud. The framework of quality aware data collection in MEC is shown in Fig. 1. Lastly, we design a Data-quality awaRe IncentiVe mEchanism (DRIVE) for collaborative tasks to motivate users to provide high-quality data.

Besides, we use the Taxis-Drive trajectory data [11] as the study case to evaluate the performance of quality evaluation method and DRIVE. Experimental results show that our incentive mechanism can effectively motivate users to upload high-quality data and make the MEC server obtain more utility. Specifically, the main intellectual contributions of this work are summarized as follows.

- We design a data-quality evaluation method from the signal and semantic perspectives to classify users into three categories.
- A dynamic reward allocation scheme is drawn up based on users' different categories and further present the DRIVE based on the Stackelberg game to enhance MEC server's utility.

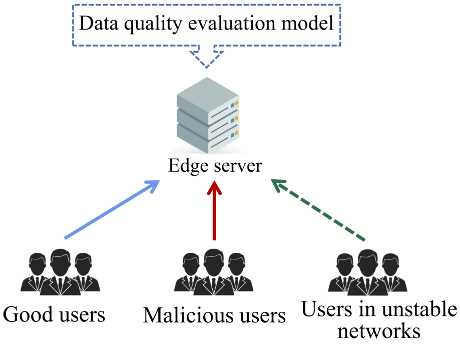


Fig. 1 Quality aware data collection in MEC.

Manuscript received February 16, 2023.
Manuscript revised August 7, 2023.
Manuscript publicized September 14, 2023.

[†]The authors are with the School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China.
a) E-mail: changeleap@163.com (Corresponding author)

DOI: 10.1587/transfun.2023EAP1022

- Finally, one realistic case study is given and the real-data experiments demonstrate the superiority of our proposed incentive mechanism.

The rest of this paper is organized as follows. In Sect. 2, we overview some previous works. Then we discuss the system model and data quality evaluation method in Sect. 3. In Sect. 4, we illustrate the design of the proposed incentive mechanism. The performance of our proposed method and mechanism is evaluated based on the real dataset in Sect. 5. Finally, we conclude the paper in Sect. 6.

2. Related Work

Data quality refers to the extent to which a set of inherent data attributes meets consumer requirements. The key attributes of data quality are integrity, freshness, and accuracy. In today's era of big data, we have the ability to predict crime, forecast consumption trends, and establish urban planning, among other things. However, to make the most of this potential, it is crucial to pay close attention to the quality of data during the collection process. Neglecting data quality can lead to the accumulation of poor-quality data, causing failures of applications.

The current related studies are listed in Table 1 from the perspectives of incentive mechanism design and data quality evaluation. At present, there are some studies to detect data quality from the signal perspective, such as the received signal strength, and the network stability. With the training data set (contextual information and data quality pairs), Liu et al. [12] build a context-data quality classifier, which is used to estimate the data quality in real-time. Peng et al. [13] extend the Expectation Maximization algorithm that combines maximum likelihood estimation and Bayesian inference to estimate the quality of sensing data. To ensure data integrity and real-time collection, Yang et al. [14] propose a data-collecting mechanism based on the greedy algorithm. However, the above research work only pays attention to the implication from the signal aspect while neglecting the significance of the semantic factors.

Other current work focused on the data quality from the semantic aspect while lacking a comprehensive analysis from the signal dimension. An et al. [16] propose a crowdsensing framework based on a lightweight blockchain-based model and exploit the expectation-maximization algorithm to evaluate the performance of participants. Their research group further [15] present an edge-computing-enabled crowdsensing framework based on the blockchain and introduced a data quality assessment mechanism based on the reinforcement learning algorithm. Wang et al. [17] first propose a trust evaluation mechanism using crowdsourcing and edge computing, then two incentive mechanisms are designed to motivate mobile edge users to conduct trust evaluation. In order to evaluate the quality of participants, a fairness-aware and privacy-ensuring scheme is presented in [18] by integrating the blockchain, trusted execution environment and machine learning. Zhang et al. [19] predict the data quality from different users through federated learning and develop a user recruitment algorithm based on the prediction of data quality.

Table 1 Comparison among data collection approaches.

Data collection approach	Incentive	Data quality	
		Signal detection	Semantic level
[14] [12]	✗	✓	✗
[13]	✓	✓	✗
[15] [19]	✗	✗	✓
[16] [17] [18]	✓	✗	✓
[2] [20] [22] [21]	✗	✓	✓
DIRVE(proposed in this paper)	✓	✓	✓

A few studies detected data quality from both signal and semantic perspectives. Lamaazi et al. [20] propose a two-stage data-driven decision-making mechanism using edge computing to select participants by their quality of outcomes. Gao et al. [21] propose quality bounded task assignment problem with redundancy constraint and divide this problem into two sub-problems solved by the Hungarian method and dynamic programming. However, they have not considered the incentive to motivate users to contribute high-quality data. [2] detected data quality from both signal and semantic dimensions. However, it is assumed that the incorrect data considered are caused by device damage rather than malicious tampering by users.

Because the data quality from consumers' devices in MEC heavily relies on the hardware of devices such as CPUs, GPUs, and memory capacity, as well as the network performance [22], it is imperative to analyze the data quality from multiple aspects to screen users into various categories and design feasible incentive mechanisms to meet the demand for high data-quality and low latency.

3. System Model

In this section, we describe the system model for edge computing with collaborative tasks and the data quality evaluation model based on signal and semantic aspects.

3.1 Edge Computing for Collaborative Tasks

The edge computing system for data collection comprises an MEC server s and a group of users U, where $U = \{1, \ldots, N\}$. The server publishes a set of tasks T, denoted as $T = \{1, \ldots, M\}$. In case of the collaborative tasks, each task j requires the participation of at least m_j users, and m_j is referred to as the *task threshold* for task j. These task thresholds are collectively represented as W, that is, $W = \{m_1, \ldots, m_M\}$. The server communicates the task set and the corresponding task thresholds to users, and it earns a value v_j when it successfully recruits no fewer than m_j users to provide high-quality data for task j. Otherwise, the server can not benefit from any reward from task j.

The properties of task j, m_j and v_j depend on the requirement of task j. Moreover, each user i has the capacity to perform a subset of tasks, i.e., T_i, $T_i \subset T$. The set of all users' task capacities is denoted as $\mathcal{T}$, i.e., $\mathcal{T} = \{T_1, \ldots, T_N\}$. User i incurs a cost c_i^j to perform task j, which depends on

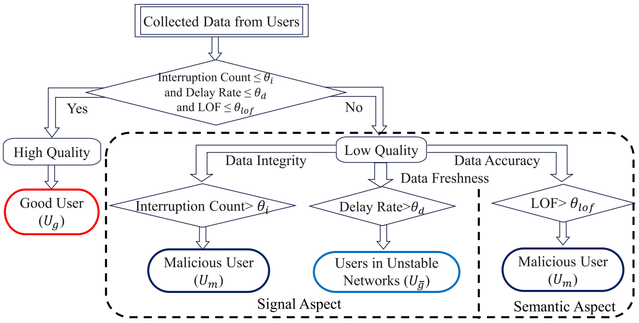


Fig. 2 Data quality evaluation method.

the specific application. The set of all users' costs is denoted as C, i.e., $C = \{C_1, \ldots, C_M\}$, each of which is the users' cost set to perform each task.

The server-user interaction in our system involves a four-step process, outlines as follows.

- The users report their cost and task capacity to the server, denoted as $(C, \mathcal{T})$.
- The edge server announces (R, W), where R is the total reward to all users, and $W = \{m_1, \ldots, m_M\}$, each of which is the minimum number of users required with high-quality data to achieve the corresponding task completion.
- Taking the information $(C, \mathcal{T}, R, W)$ into account, each user makes its strategy, determining the set of tasks user i performs, and submits their results to the server.
- The server employs the data quality evaluation method, as depicted in Fig. 2 to classify users into three categories, and allocates R to users by the dynamic reward allocation scheme, designed in Sect. 4.

Each user is assumed to be rational and not willing to do tasks if the obtained reward is less than the cost. Hence, we define the utility functions of users and the server. The user i's utility is defined as the reward minus the cost, i.e.,

$$u_i = r_i - \sum_{j \in T_c^i} c_i^j \tag{1}$$

where r_i is the reward that user i receives and T_c^i indicates the completed tasks that user i participated in.

Similarly, the server's utility function is defined as the total value of completed tasks minus the total reward, i.e.,

$$u_s = \sum_{j \in T_c} v_j - R \tag{2}$$

where T_c includes all completed tasks. The server is aimed to maximize its utility by selecting appropriate users and allocating its tasks accordingly, while individual users make their task strategies to maximize their own utilities.

3.2 Data Quality Evaluation Method

Since rational users has no inherent intention to provide data with high quality, it becomes essential to ensure effective data collection through initial data validation. To achieve this, we propose a data quality evaluation method that considers both signal and semantic aspects of the received data.

Regarding the signal aspect, the collected data can be influenced by various network factors, such as network types and noise environment. Evaluating the Signal-to-Noise Ratio (SNR) becomes crucial to assess the signal strength of received data. To achieve reliable transmission, the system typically uses TCP (Transmission Control Protocol) and employs retransmission operations when the SNR falls below a given threshold. Consequently, data reception at the server may experience significant delay. If the latency surpasses the predefined waiting time threshold, an interruption occurs. Frequent and excessive interruptions can signify malicious intent on the user's part or unavoidable scenarios, such as device malfunction, resulting in the user being eliminated from eligible candidates. If the interruption happens frequently and the *Interruption Count* is above the threshold (θ_i), these users who have uploaded data that fails to guarantee the property of data integrity are placed in the set U_m.

In data collection scenarios, data are usually sent to the server at a predetermined frequency, resulting in fixed time intervals for receiving the data within a specific range. However, these time intervals may vary due to the influence of the communication environment. Time intervals that exceed a certain proportion of the normal value (determined by the specific application) are considered delay intervals. The *Delay Rate* refers to the proportion of these delay intervals. If the delay rate exceeds the threshold (θ_d), it indicates that the user sending the data is experiencing an unstable network connection, classified as $U_{\overline{g}}$.

Consequently, after successful data decoding, we evaluate the data quality based on its semantic value. Data may be subject to tampering for personal gains, we exploit the metric of Local Outlier Factor (LOF) [23] to detect the abnormal data. LOF quantifies the isolation level of an object concerning its neighboring data points. The definition of LOF is given as follows.

$$LOF_k(A) = \frac{\sum_{o \in N_k(A)} \frac{lrd(o)}{lrd_k(A)}}{|N_k(A)|} \tag{3}$$

where $lrd_k(A)$ is local reachability density of an object A in its k-distance neighborhood $N_k(A)$, and $|N_k(A)|$ denotes the size of $N_k(A)$, i.e., the number of objects in A's k-distance neighborhood. LOF is the average ratio of the local reachability density of data point A to that of its k-nearest neighbors. If the LOF value of a data point exceeds 1 and surpasses the predefined threshold (θ_{lof}), it is considered as abnormal data, leading to the classification of the data provider as a malicious user. Conversely, if data point A and A's k-nearest neighbors exhibit similar behaviors, the LOF value of A approaches 1, indicating high data accuracy.

If the data satisfy the properties of data integrity, freshness and accuracy, their providers are grouped into the set of good users (U_g). Upon completing the data quality assess-

Table 2 Consolidate abbreviations and generic terms.

Consolidate abbreviations and generic terms	Denotation
QoS	Quality of Service
MEC	Mobile Edge Computing
DRIVE	Data-quality awaRe IncentiVe mEchanism
NE	Nash Equilibrium
GPS	Global Positioning System
SNR	Signal to Noise Ratio
TCP	Transmission Control Protocol
LOF	Local Outlier Factor
WQA	Without Quality Aware

ment, users are categorized into three classes: good users, users in unstable networks and malicious users. The flow chart of our data-quality evaluation method is illustrated in Fig. 2. Table 2 lists all the consolidate abbreviations and generic terms in our paper.

4. Incentive Mechanism Design

In this section, we design the dynamic reward allocation scheme based on users' different classes. Subsequently, the Data-quality awaRe IncentiVe mEchanism (DRIVE) is proposed to encourage users with high-quality data.

4.1 Dynamic Reward Allocation Scheme

In this section, we introduce the reward function for three users' classes.

We first define the data quality level of user i, l_i computed as

$$l_i = k/LOF(i) \tag{4}$$

where k is a predetermined constant from the application served by the MEC server. In order to detect the data quality among users, another metric is presented as $\rho_i = \frac{l_i}{\sum_{i \in U_g} l_i}$.

For each user i in U_g, the reward is

$$r_i = |T_c^i| \cdot \frac{\rho_i R_j}{\sum_{i \in S_j} \rho_i}. \tag{5}$$

where $|T_c^i|$ is the number of tasks in T_c^i. Here S_j is the set of users involved in completing task j. The total reward for all good users is denoted as R', i.e. $R' = \sum_{i \in U_g} r_i$. R_j is the reward allocated to the users who perform task j, and the reward should be proportional to the number of users who perform it. Here R_j is computed as

$$R_j = \frac{R' \sum_{i \in S_j} \rho_i}{\sum_{j \in T_c} \sum_{i \in S_j} \rho_i}. \tag{6}$$

On the other hand, for users in $U_{\overline{g}}$, despite contributing nothing due to their poor network conditions, they will receive a basic reward denoted as δ, calculated as follows:

$$\delta = \frac{R - R'}{|U_{\overline{g}}|} \tag{7}$$

where R represents the total reward provided by the server, and $|U_{\overline{g}}|$ is the number of users in $U_{\overline{g}}$. Since the malicious users give no benefit to the task completion, the reward for the users in U_m is set as zero.

In summary, the reward provided for users in different classes is defined as follows:

$$r_i = \begin{cases} 0 & i \in U_m \\ \delta & i \in U_{\overline{g}} \\ \frac{|T_c^i| \rho_i}{\sum_{j \in T_c} \sum_{i \in S_j} \rho_i} & i \in U_g \end{cases} \tag{8}$$

4.2 Data Quality Aware Incentive Mechanism

The quality-aware data collection system involves two types of game relationships: a game between the edge server and users and a game between users themselves. Therefore, it can be modeled using a two-stage Stackelberg game. In stage I, the server first announces the reward for good users R' and task threshold W to users. (Note: since δ is determined by the specific application, the server only need to compute R') This stage primarily focuses on the game between the server and users. The server's decision is how to determine R' to maximize its utility. In stage II, based on (R', W), each user devises a task strategy $(\mathcal{S}_i)$ to maximize their own utility. Here, $\mathcal{S}_i$ represents the set of tasks performed by user i. This stage mainly concerns the game among users.

The Stackelberg game is typically solved using backward induction, and the main process is illustrated in Alg. 1.

Algorithm 1 Data quality Aware Incentive Mechanism

```
Input: users' normalized costs $\overline{C}$, task thresholds $W$ and task capacity $\mathcal{T}$
Output: NE of the reward $R'$ for good users and NE of $u_s$
{User selection}
Initialization: $M = \{1, \ldots, m_M\}, U_s = \emptyset, T_c = \emptyset$
for all task $j \in M$ do
    $\overline{c}_j^*$= the $m_j - th$ smallest quality among $U_g$
    if $v_j > \sum (\overline{c}_i^j | \overline{c}_i^j < \overline{c}_j^*)$ then
        $S_j = \{i | \overline{c}_i^j < \overline{c}_j^*\}, T_c = T_c \cup \{j\}$
$U_g = \cup_{j \in T_c} S_j$
{Compute the NE of the reward $R'$ for good users}
$R'^* = \max\{\frac{c_i \sum_{j \in T_c} \sum_{i \in S_j} \rho_i}{|T_c^i| \rho_i}, i \in U_g\}$
{Compute the NE of server's utility $u_s$}
$u_s^* = \sum_{j \in T_c} v_j - R'^* - \delta |U_{\overline{g}}|$
```

We first analyze stage II, where, given a reward R' offered to high-quality users, each user determines their task strategy to maximize their own utility. As long as their utility is non-negative, rational users will choose to participate in tasks. Since the user's cost is related to the provided data quality, we introduce a normalized cost $\overline{C}$ to represent the actual cost C under the same data quality, given by $\overline{c}_i^j = \frac{c_i^j}{\rho_i}$. Therefore, the Nash Equilibrium (NE) of the user strategy profile $\mathbb{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_N\}$ depends on the NE of the normalized cost threshold $(\overline{c}_j^*)$. If user i's normalized cost is no more than the threshold, i.e., $\overline{c}_i^j \leq \overline{c}_j^*$, then user i participates in task j. Otherwise, user i will not. It can be deduced that

$\overline{c}_j^*$ is the m_j-th minimum normalized cost among U_g (Line 3). If $v_j > \sum(\overline{c}_i^j | \overline{c}_i^j < \overline{c}_j^*)$, then task j can recruit a sufficient number of good users. The union of all S_j ($j \in T_c$) is the set of all selected users with high data quality, symbolized as U_g (Line 6). The user selection process is now complete.

Next, in Stage I, the key process is to determine the NE of R' to maximize the server's utility. The key insight for computing R'^* is to ensure $u_i \geq 0$ ($i \in U_g$), i.e., $r_i - \sum_{j \in T_c^i} \overline{c}_i^j \geq 0$ ($i \in U_g$). Hence, R'^* is computed as the equation in Line 7. Finally, we calculate the NE of the server's utility, which is the total obtained task values minus the reward for good users and users in unstable networks (Line 8).

5. A Practice Case Study

5.1 Dataset Introduction

We use the T-Drive dataset containing one-week taxi trajectories (2008) from Beijing, encompassing 10357 taxis [11], [24]. Each trajectory consists of taxi ID, timestamp, and GPS data. Our focus is on treating each taxi driver as a user, suspecting potential GPS tampering for fare inflation. The taxi communication network is dynamic due to a collective distance of 9 million kilometers covered. This dynamic nature, along with potential human manipulation, affects data quality, subject to evaluation via our model.

5.2 Thresholds Determination

We establish critical thresholds in our data-quality evaluation method — delay rate, velocity eccentricity, and interruption thresholds.

Beginning with time intervals between adjacent GPS readings, let's consider taxi No.19 as an example. On February 5th, 2008, the proportion of its intervals is shown in Fig. 3. Standard interval is 600s, and tolerance includes 599s and 601s. Intervals of 1200s and 1199s result from data loss and subsequent retransmission. Similarly, 1799s indicates two instances of data loss. For intervals of 826s and 974s, it was caused by the delayed delivery. If the time interval exceeds 50% of the normal one, it is regarded as a delayed interval.

(1) Delay Rate Threshold Determination:

We computed the delay rate for 100 taxis on February 5, 2008. To account for various factors contributing to delays like weather and network conditions, we employed log-normal distribution for fitting GPS transmission delays. The density function of this distribution is:

$$p(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp^{-\frac{(\ln x - \mu)^2}{2\sigma^2}} \tag{9}$$

Illustrated in Fig. 4(a), the fitting outcome for delay rate yields parameters $\mu = -2.7242$ and $\sigma = 0.4991$. Subsequently, we derive the delay rate threshold using the following formula:

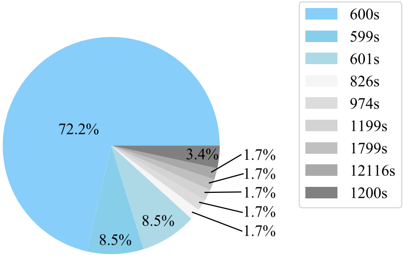
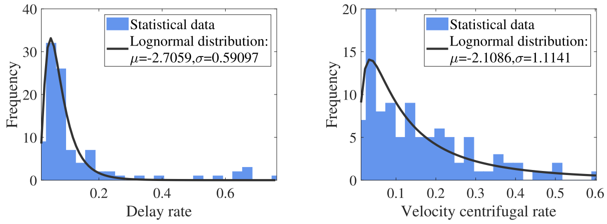


Fig. 3 The time-interval proportion of No. 19 taxi sending GPS.



(a) The fitting results of delay rate (b) The fitting results of velocity eccentricity

Fig. 4 Real data based simulation results.

$$\frac{\int_{0^+}^{a} p(x)\,\mathrm{d}x}{\int_{0^+}^{\infty} p(x)\,\mathrm{d}x} = k \tag{10}$$

The parameter k can be tailored based on specific demand scenarios. Subsequently, the value of a can be computed, serving as the delay rate threshold. For instance, with k set at 0.6, the resulting delay rate threshold is 7.45%. This implies that taxi drivers exhibiting a delay rate exceeding 7.45% are categorized within an unstable network context and fall under $U_{\overline{g}}$.

(2) Velocity Eccentricity Threshold Determination: We determined the abnormal point rate, which represents velocity eccentricity. By computing distances from latitude and longitude data and then deriving velocities from time differences, we established abnormal velocity values through LOF analysis. The velocity eccentricity rate, denoting the ratio of LOF-detected abnormal points to all points, was then calculated. The velocity's abnormal values can stem from various factors. For fitting velocity eccentricity, we employed the lognormal distribution, yielding parameters $\mu = -2.1086$ and $\sigma = 1.1141$, as shown in Fig. 4(b). Employing Eq. (10), we once again set k to 0.6, resulting in a velocity eccentricity threshold of 16.1%. This implies that users exhibiting velocity eccentricity surpassing 16.1% are flagged as malicious users and grouped under U_m.

(3) Interruption Threshold Determination:

If the taxi does not send GPS after the 1800s, we treat it as an interruption. Considering the time for the driver's three meals a day, we allow 3 interruptions.

5.3 Performance Evaluation

In this section, we assess the proposed incentive mechanism across varied scenarios to discern the effects of LOF, DRIVE,

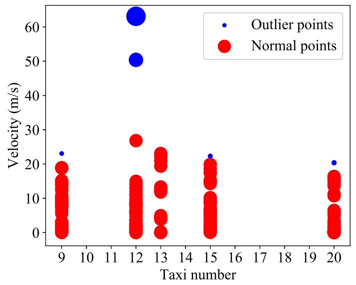


Fig. 5 Data quality evaluation.

and key parameters. All simulations were conducted on a PC equipped with an Intel I7-7700 CPU and 16 GB memory.

5.3.1 Results on Data-Quality Evaluation

We simulated input data by selecting 60 car tracks for one day. In Fig. 5, the velocity of taxis is depicted. Blue points signify velocity outliers, with larger points indicating greater outlier count. Through dataset processing, we acquired time intervals for each vehicle's geographic location uploads, averaging its speed during this interval. We further processed and analyzed data to derive velocity eccentricity and upload delay rate, aiding in taxi status determination. Figure 5 demonstrates that, according to the previously mentioned evaluation process, car No. 12's velocity exhibits multiple irregularities. Although its speed eccentricity surpasses the threshold, the upload delay rate remains low. Consequently, car No. 12 is identified as malicious due to fraudulent behavior. Subsequent assessment of other cars follows. Car No. 13 exhibits favorable behavior, belonging to the set U_g. Cars No. 9 and No. 20 both possess minimal outliers. Evaluating their latency and upload delay rate, car No. 9's values surpass the threshold while No. 20's remain below it. This categorizes car No. 9 under $U_{\overline{g}}$ and car No. 20 under U_g.

5.3.2 Result on DRIVE Performance

To validate DRIVE's effectiveness, a comparison was drawn between our incentive mechanism, DRIVE, and an alternative mechanism, Without Quality Aware (WQA) [25] by evaluating u_s. Three cases were examined using the taxi data mentioned earlier, with their parameter settings outlined in Table 3. Figure 6 illustrates that WQA's u_s values are all negative. This is because WQA might select users from sets $U_{\overline{g}}$ and U_m, leading to incomplete task execution. Conversely, DRIVE consistently yields positive u_s values across all three cases, underscoring its effectiveness. Moreover, we investigate the impact of N and M on the cost threshold. Figure 7(a) and 7(b) indicate that during the first period cost threshold grows along with reward R, but it peeks at a certain point soon after, that is because all tasks are done and no more users are needed. We learn that the cost threshold decreases in M and N.

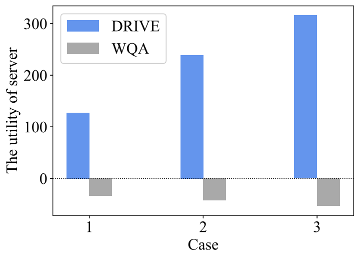
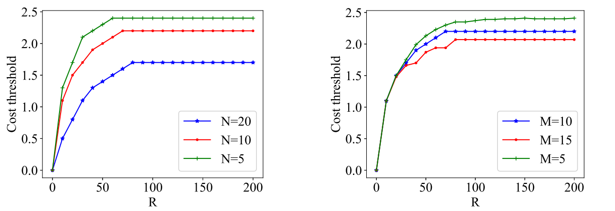


Fig. 6 The evaluation of server's utility.

Table 3 Parameters setting.

Parameter	Case 1	Case 2	Case 3
M	2	3	4
θ_i	3		
θ_d	7.45%		
θ_{lof}	16.1%		
N	60		
k	3		
v_j	Integer in [50,100]		
c_i	$N(2,1)$		
m_j	Integer in [6,12]		



(a) Impact of N to cost threshold (b) Impact of M to cost threshold

Fig. 7 Impact of M and N to cost threshold.

6. Conclusion

In this paper, we proposed a data-quality aware incentive mechanism that aims to solve the problems that crowdsourcing users don't have the long-lasting passion to provide high-quality data and that some malicious users might cheat in data collection. First, we design a data-quality evaluation method from both signal and semantic perspectives and deploy it in the mobile edge computing framework for real-time data gathering. Then we propose a dynamic reward allocation scheme based on the LOF method for classified users. Moreover, we design the DRIVE based on the Stackelberg game for collaborative tasks. Finally, a realistic case study is given and real-data experiments are conducted to validate the superiority of the proposed incentive mechanisms.

Acknowledgment

This research was supported by the Fundamental Research Funds of Zhejiang Sci-Tech University (Grant No. 23232138-Y), Liaoning Provincial Natural Science Foundation of China (Grant No. 2022-KF-21-01), the National Natural Science Foundation of China (Grant No. U22A2004) and the Key

Research and Development Program of Zhejiang Province, China (Grant No. 2022C01079).

References

[1] 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, 2019.

[2] R. Casado-Vara, F. Prieto-Castrillo, and J.M. Corchado, "A game theory approach for cooperative control to improve data quality and false data detection in WSN," International Journal of Robust and Nonlinear Control, vol.28, no.16, pp.5087–5102, 2018.

[3] P.-Y. Hsueh, P. Melville, and V. Sindhwani, "Data quality from crowdsourcing: A study of annotation selection criteria," Proc. NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing, pp.27–35, 2009.

[4] B. Yin and J. Lu, "A cost-efficient framework for crowdsourced data collection in vehicular networks," IEEE Internet Things J., vol.8, no.17, pp.13567–13581, 2021, IEEE.

[5] D. Kuemper, T. Iggena, R. Toenjes, and E. Pulvermueller, "Valid.IoT: A framework for sensor data quality analysis and interpolation," Proc. 9th ACM Multimedia Systems Conference, pp.294–303, 2018.

[6] H. Chen, L.F. Pieptea, and J. Ding, "Construction and evaluation of a high-quality corpus for legal intelligence using semiautomated approaches," IEEE Trans. Rel., vol.71, no.2, pp.657–673, 2022, IEEE.

[7] F. Li, Y. Wang, Y. Gao, X. Tong, N. Jiang, and Z. Cai, "Three-party evolutionary game model of stakeholders in mobile crowdsourcing," IEEE Trans. Comput. Social Syst., vol.9, no.4, pp.974–985, 2021, IEEE.

[8] X. Hu, J. Duan, and D. Dang, "Crowdsourcing-based semantic relation recognition for natural language questions over RDF data," Enterprise Information Systems, vol.13, no.7-8, pp.935–958, 2019.

[9] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects," IEEE Commun. Surveys Tuts., vol.23, no.2, pp.1160–1192, 2021.

[10] J. Nakazato, M. Nakamura, T. Yu, Z. Li, K. Maruta, G. Tran, and K. Sakaguchi, "Market analysis of MEC-assisted beyond 5G ecosystem," IEEE Access, vol.9, pp.53996–54008, 2021.

[11] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," Proc. 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.316–324, 2011.

[12] S. Liu, Z. Zheng, F. Wu, S. Tang, and G. Chen, "Context-aware data quality estimation in mobile crowdsensing," Proc. IEEE Conference on Computer Communications, pp.1–9, 2017.

[13] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," Proc. 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp.177–186, 2015.

[14] J. Yang, L. Fu, B. Yang, and J. Xu, "Participant service quality aware data collecting mechanism with high coverage for mobile crowdsensing," IEEE Access, vol.8, pp.10628–10639, 2020.

[15] J. An, S. Wu, X. Gui, X. He, and X. Zhang, "A blockchain-based framework for data quality in edge-computing-enabled crowdsensing," Front. Comput. Sci., vol.17, no.4, p.174503, 2023.

[16] J. An, J. Cheng, X. Gui, W. Zhang, D. Liang, R. Gui, L. Jiang, and D. Liao, "A lightweight blockchain-based model for data quality assessment in crowdsensing," IEEE Trans. Comput. Social Syst., vol.7, no.1, pp.84–97, 2020.

[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. Intelligent Systems and Technology (TIST), vol.10, no.6, pp.1–19, 2019.

[18] Z. Wang, Y. Li, D. Li, M. Li, B. Zhang, S. Huang, and W. He, "Enabling fairness-aware and privacy-preserving for quality evaluation in vehicular crowdsensing: A decentralized approach," Security and Communication Networks, vol.2021, 2021.

[19] W. Zhang, Z. Li, and X. Chen, "Quality-aware user recruitment based on federated learning in mobile crowd sensing," Tsinghua Sci. Technol., vol.26, no.6, pp.869–877, 2021.

[20] H. Lamaazi, R. Mizouni, H. Otrok, S. Singh, and E. Damiani, "Smart-3DM: Data-driven decision making using smart edge computing in hetero-crowdsensing environment," Future Generation Computer Systems, vol.131, pp.151–165, 2022.

[21] X. Gao, H. Huang, C. Liu, F. Wu, and G. Chen, "Quality inference based task assignment in mobile crowdsensing," IEEE Trans. Knowl. Data Eng., vol.33, no.10, pp.3410–3423, 2020.

[22] J. Kim and Y.H. Song, "Dynamic transaction management for system level quality-of-service in mobile APs," IEEE Trans. Consum. Electron., vol.64, no.2, pp.204–212, 2018.

[23] M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," Proc. 2000 ACM SIGMOD international conference on Management of data, pp.93–104, 2000.

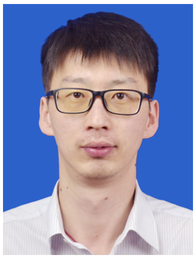
[24] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving directions based on taxi trajectories," Proc. 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp.99–108, 2010.

[25] S. Luo, Y. Sun, Y. Ji, and D. Zhao, "Stackelberg game based incentive mechanisms for multiple collaborative tasks in mobile crowdsourcing," Mobile Netw. Appl., vol.21, pp.506–522, 2016.

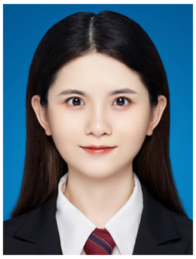
Shuyun Luo received her Ph.D. degree in communication and information system from Beijing University of Posts and Telecommunications, China, in 2016. She is currently an assistant professor in Zhejiang Sci-tech University, China. Her research interests include edge intelligence, Industrial Internet of things and network economics.

Wushuang Wang received the M.S. degree in information and communication engineering from Zhejiang Sci-Tech University, Hangzhou, China, in 2022. Now she is a research student majoring in policy and planning sciences at The University of Tsukuba. Her research interests are shop scheduling, intelligent edge computing and reinforcement learning.

Yifei Li received the B.E. degree in computer science and technology from Zhejiang Sci-Tech University and the M.S. degree in advanced computer science from University of Leeds in 2021 and 2022, respectively. Now he works as a data analyst. His research interests are intelligent edge computing and big data system.



Jian Hou received the Ph.D. degree in control theory and control engineering from Zhejiang University, Hangzhou, China, in 2013. He is currently with the School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou. His current research interests include multi-agent systems, consensus, and reinforcement learning.



Lu Zhang graduated from Beijing Information Science and Technology University, majored in Information Management and Information System. Now she is a graduate student majoring in software engineering at Zhejiang Sci-Tech University. Her research interest is spectral reconstruction.