Mathematical Biosciences and Engineering

MBE, 19(4): 3285–3312.
DOI: 10.3934/mbe.2022152
Received: 21 November 2021
Revised: 13 January 2022
Accepted: 16 January 2022
Published: 21 January 2022

http://www.aimspress.com/journal/MBE

Research article

A blockchain-based creditable and distributed incentive mechanism for participant mobile crowdsensing in edge computing

Shiyou Chen[1], Baohui Li[2], Lanlan Rui[1,*], Jiaxing Wang[1] and Xingyu Chen[1]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] Cyberspace Security Academy, Beijing University of Posts and Telecommunications, Beijing 100876, China

* **Correspondence:** Email: llrui@bupt.edu.cn; Tel: +8601062283412; Fax: +8601062283412.

Abstract: With the popularization of portable smart devices, the advance in ubiquitous connectivity and the Internet of Things (IoT), mobile crowdsensing is becoming one of the promising applications to acquire information in the physical world of edge computing and is widely used in Smart Cities. However, most of the existing mobile crowdsensing models are based on centralized platforms, which have some problems in reality. Data storage is overly dependent on third-party platforms leading to single-point failures. Besides, trust issues seriously affect users' willingness to participate and the credibility of data. To solve these two problems, a creditable and distributed incentive mechanism based on Hyperledger Fabric (HF-CDIM) is proposed in this paper. Specifically, the HF-CDIM combines auction, reputation and data detection methods. First, we develop a multi-attribute auction algorithm with a reputation on blockchain by designing a smart contract, which achieves a distributed incentive mechanism for participants. Second, we propose a K-nearest neighbor outlier detection algorithm based on geographic location and similarity to quantify the credibility of the data. It is also used to update the user's reputation index. This guarantees the credibility of sensing data. Finally, the simulation results using real-world data set verify the effectiveness and feasibility of above mechanism.

Keywords: hyperledger fabric; blockchain; mobile crowdsensing; incentive mechanism; smart contract

1. Introduction

It is predicted that the emerging Internet of Things (IoT) will connect more than 50 billion smart devices by the year 2025 [1], which will inevitably change the way we live and work, such as smart homes, work spaces, transportation and even future cities. With the development of the Internet and the proliferation of mobile devices, Mobile Crowdsensing (MCS) [2] has become a new paradigm for collecting real-time data from the physical world. MCS has been widely used in Smart Cities [3], such as real-time situation awareness in the aftermath of a disaster, detecting real-time traffic congestion [4], and so on. However, the MCS platform will face huge computing and storage pressure when processing the user's return results with large-scale data analysis. The latest technology of edge computing brings new opportunities for this issue, which pushes the frontiers of computing, services, and data collection to the edge of the network [5]. With the help of edge computing, the MCS platform can offload the processing and transmission of the sensing data to edge servers and devices.

The main idea of MCS-based edge computing is to crowdsource space-time sensing and processing tasks to a large number of smart devices (e.g., smartphones and users' portable electronic devices). MCS is also called "social sensing" or "human-enabled edge computing" [6]. Hence, the quantity and quality of MCS participants are the key issues in MCS-based edge computing: 1) The owner of the sensing device is a rational participant. The sensing task consumes the resources of sensing devices, and brings corresponding costs to the participants in terms of battery, mobile data communication, time and mobile space, etc. Some sensing tasks cannot bring direct benefits to the participants, while they require long-term participation. If there is no benefit compensation (e.g., economic or reputation), self-driven users are unwilling to spend their own resources on sensing tasks. 2) Sensing tasks are allocated to different mobile users with heterogeneous capabilities, the quality and credibility of user sensing data cannot be guaranteed. The quality of user sensing data greatly affects the quality of service (QoS) of MSC applications. Therefore, to ensure the high-quality completion of MCS tasks, we need to design an appropriate incentive mechanism to encourage a large number of users to participate in MCS actively and credibly.

The analysis and design of incentive mechanisms often usually involve game theory [7,8], because it is suitable for analyzing the interaction between individuals in the network group and determining the optimal strategy. The price-based mechanism is often realized through payment of remuneration and virtual points, and its essence is to obtain services or resources for a fee. This incentive mechanism combines game theory to construct various auction models [9]. However, most of the research is implemented on centralized platforms. The method based on the feasibility of virtual currency payment may have problems such as unfair pricing and false quotations. Besides, centralized systems require complex and secure audit and supervision mechanisms, which increase the corresponding communication and computing load. In addition, it is also prone to single-point failures. When the central node encounters an attack or fails, the entire system will face a crisis.

To tackle the above challenges while meeting the above incentive requirements, we introduced the idea of blockchain into MCS-based edge computing to develop a decentralized crowdsensing system. The development of Blockchain technology has well overcome the challenges of single-point failures and mistrust between users in centralized systems [10]. This technology is to construct a decentralized or peer-to-peer application ledger. The Distributed Ledger (DL) is used to store data related to the interaction between users, such as virtual currency transaction behavior, traceability behavior of commodity information, and storage records of data. The introduction of blockchain

technology can provide MCS with a transparent and reliable decentralized mechanism. We choose Hyperledger Fabric for implementation in this paper, because Hyperledger Fabric is a permissioned blockchain platform without Proof of Work (PoW) and encrypted mining, providing high scalability and fast transactions.

Based on the above analysis, our goal is to reduce the over-reliance of third-party platforms in a centralized system and to achieve credible tamper-proof sensing data with the help of blockchain technology. In this process, we need to solve the following two issues: First, how to implement the incentive mechanism in a distributed way? Second, how to evaluate the credibility of the sensing data provided by participants?

In this paper, we introduce a creditable and distributed incentive mechanism based on Hyperledger Fabric (HF-CDIM) for mobile crowdsensing. Specifically, the HF-CDIM combines auction and reputation methods. We implement the multi-attribute auction algorithm by designing a smart contract, which realizes an automated and transparent incentive mechanism for participants. Then, we propose a K-nearest neighbor outlier detection algorithm based on geographic location and similarity to calculate the credibility of the data, which is used to update the user's reputation. The proposed framework and algorithm overcome the outstanding problems faced by the centralized incentive mechanism. At the same time, the credibility of sensor data and non-tamperable characteristics are guaranteed. Our contributions can be summarized as follows:

1) In the scenario of MCS-based edge computing, we propose a distributed incentive architecture based on the Hyperledger Fabric and design multi-attribute auction based on a smart contract, which realizes the selection of the bid-winners to assign sensing tasks. This architecture combines smart contract and auction method to achieve automatically execution, realizes distributed incentive, and alleviates the over-dependence of the centralized platform and users.

2) We proposed a K-nearest neighbor outlier detection algorithm based on geographic location and similarity. After the user wins the bid, it is necessary to check the credibility of the sensing data submitted by the user. This algorithm can effectively detect the credibility of the sensing data submitted by users. It takes advantage of the information of the sensing data itself, including sampling geographical location, sampling time point and neighboring information. We establish users' reputation indicators based on the data credibility to realize a equitable effective incentive mechanism.

3) Considering the implementation performance of this mechanism and the limited resources on the blockchain, we store the large-scale sensing data off the chain and simultaneously store the hash value of the sensing data on the blockchain. This ensures that the data cannot be tampered with and avoids the performance degradation caused by the large-scale data on the blockchain.

4) The feasibility and effectiveness of the proposed mechanism are verified by simulation experiments using real world data set.

This paper is organized as follows: Section 2 introduces related work, Sections 3 and 4 present the proposed framework, our proposed detection algorithm based on geographic location and similarity is introduced in Section 5, and the experimental evaluation and performance results are shown in Section 6. Finally, we conclude this paper in Section 7.

2. Related work

According to our research topic, we first survey relevant research on Mobile Crowdsensing model and incentive mechanism. Then we propose the research process analysis.

2.1. Mobile Crowdsensing model

Mobile Crowdsensing model is generally as shown in Figure 1. It is composed of task publishers, task executors (participating users and devices), network (transmission of sensing data), and third-party platforms. The basic process is:

a. Task publishers publish tasks through third-party platforms.
b. Users view tasks, collect and upload data.
c. The third-party platform stores the data to the data center and returns it to the task publisher.
d. Users receive rewards.

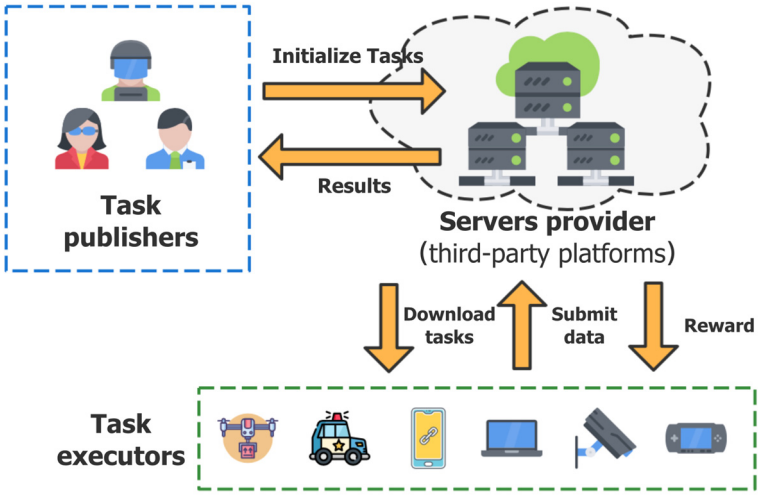

Figure 1. Mobile Crowdsensing model.

From the above model and process, we can see the current problems of Mobile Crowdsensing mainly include: Over-reliance on third-party platforms, data integrity cannot be guaranteed, and users face unfair incentive mechanisms.

2.2. Mobile Crowdsensing incentive mechanism

For Mobile Crowdsensing incentive mechanism, many scholars have researched and proposed a variety of methods to realize the incentive process, including incentive mechanisms based on user credibility, data quality, focusing on user privacy protection, and distributed technology.

The incentive mechanism based on user reputation is to evaluate user reputation and issue rewards after completing tasks. Guo et al. [11] proposed a dynamic and high-quality MCS incentive mechanism called TaskMe, which can achieve appropriate worker selection and high-quality data acquisition. Lin D et al. [12] proposed a multi-dimensional reputation evaluation model to objectively evaluate user reputation. Simulation experiments verified the effectiveness of simulation experiments. Kang et al. [13] proposed a joint optimization method based on the combination of reputation and contract theory to realize the incentive mechanism. From the perspective of significantly improving learning accuracy, the method proposed in the paper is an effective incentive mechanism. Li et al. [14] proposed a crowd-aware task selection algorithm and reward distribution incentive mechanism based on a reputation evaluation model (CTSRE). This method uses a reputation-weighted reward distribution method,

which can effectively encourage users to participate, but this method may face attacks such as whitewashing attacks. Some researchers combine auction mechanism with reputation. Luo et al. [15] incentive mechanism based on the reverse auction and fine-grained ability reputation system. Xu et al. [16] compatible user grouping and reverse auction and design truthful incentive mechanisms to minimize the social cost. However, the above methods evaluate the user reputation based on the user's historical interaction information. In the MCS based edge computing scenario, the user dependency is not determined at the time of deployment, the connection is established based on the requirements of sensing tasks. The above methods can not accurately evaluate the reputation of new nodes or unknown nodes. We found data quality evaluation method can solve this problem.

The incentive mechanism based on data quality mainly considers how to evaluate the reliability of the data, so as to obtain high-quality data. H. Gao et al. [17] proposed an online quality sensing incentive mechanism, which can additionally reward users based on their task completion level and historical performance. Xiang C et al. [18] proposed a measure based on the confidence interval to quantify the quality of uncertain data. The paper also uses the Fisher information of the Mobile Crowdsensing data to calculate the confidence interval, avoiding multiple experimental evaluations. Krontiris I et al. [19] proposed the use of a traditional reverse auction mechanism to achieve user incentives and obtain high-quality data. The simulation experiment demonstrated the advantages of this method compared with a single price attribute. Liu Y, et al. [20] proposed an incentive mechanism based on a new mobile Mobile Crowdsensing auction model. In this paper, a location-aware incentive method based on reverse auction (IMRAL) is proposed to improve user utility. Luo T et al. [21] proposed a cross-validation method. The verification result of this method indicates that the data can be reshaped into more credible data, and the data quality is enhanced.

However, when analyzing the above-mentioned literature, it is found that the realization of the above-mentioned incentive mechanism basically depends on the third-party platform, which faces the threat of excessive centralization for both users and platforms.

In response to the above problems, some scholars have begun to study distributed incentive mechanisms. For example, Zhou Z et al. [22] in order to solve the problem of large-scale spectrum sharing in 5G heterogeneous networks, the article proposes a blockchain-based framework for privacy preservation, incentive compatibility and spectrum efficiency, and uses cases to illustrate the security and effectiveness of the framework. The distributed incentives of our Mobile Crowdsensing scenarios are very referential. Gervais A et al. [23] and Bentov I et al. [24] theoretically proved the credibility and safety of distributed incentives, and laid a theoretical foundation for the study of blockchain-based incentive mechanisms. M. Li et al. [25] proposed a blockchain-based crowdsourcing framework, which implemented a distributed incentive framework, using smart contracts to implement management logic, and using miner nodes to implement data verification. Jia, B et al. [26] proposed a hybrid incentive mechanism that combines privacy protection and virtual credit, and realized user privacy protection based on blockchain. J. Wang et al. [27] proposed an incentive mechanism that applies Mobile Crowdsensing to the blockchain. As nodes in the blockchain, the platform and the sensing user carry out the sensing task, and their trading relationship is recorded in the blockchain, which is verified by the miners in the blockchain, effectively preventing the collusion attack launched by the sensing platform, and overcoming the security risks faced by the trusted third party. Lin X et al. [28] proposes a peer-to-peer (P2P) computing resource trading system to balance the space-time dynamic demand of computing resources in the vehicle IoT-assisted smart city, and combined with the blockchain to avoid centralized trusted third parties. This method can effectively encourage the cooperation between buyers

and sellers. However, the distributed incentive mechanisms proposed in some references put data completely on the blockchain. That will seriously affect the efficiency and scalability of the blockchain.

Table1 shows the comparison between our research and the above work survey.

Table 1. Comparison between our research and the above work survey.

Study	Incentive Mechanism	Features
[11–14]	Centralized platform based on user reputation	Use reputation to achieve incentives, users' reputation over-reliance on centralized platforms
[17,19–21]	Centralized platform based on data quality evaluation	Achieved high-quality data acquisition but single-point failures of centralized platforms may make data unavailable.
[23–25]	Distributed platform incentive framework	Propose a distributed incentive framework based on blockchain, but don't propose a specific realization method.
[26–28]	Distributed platform incentive realization	Realize distributed incentives, but storage sensing data on the blockchain, not suitable for large-scale tasks.
Our work	Distributed incentive platform based on Hyperledger Fabric	a. Realize distributed incentive mechanism based on smart contract. b. Obtaining reputation indication and credible sensing data by data detection. c. Non-tamperable large-sacale sensing data. d. Avoid large-scale data on the blockchain.

It can be seen from Table 1 that our work has made the following improvements to the current research status: First, we proposed a distributed incentive mechanism based on Hyperledger Fabric, which realizes high-quality sensing task assignment with the help of a multi-attribute auction algorithm on the blockchain. Then, we proposed a sensing data credibility detection algorithm that realizes credible evaluation of data and reputation value acquisition. In addition, we proposed to store hash value on the blockchain instead of large-scale sensing data, which avoids the inefficiency of large-scale on-chain data storage with the help of the non-tamperable feature of the blockchain and the verifiability of the hash value.

2.3. Research process analysis

Based on the previous reading of a large number of documents and the analysis of the current research status in this field, we have two questions: How to achieve distributed incentives? How to evaluate the credibility of sensing data? The research process of this article is as shown in Figure 2.

As is shown in Figure 2 we have mainly conducted specific research on the two issues of distributed incentives and sensing data credibility. Among them, the distributed incentive problem is analyzed in

terms of Blockchain-based incentive architecture, and multi-attribute auction contract design issues. For sensing data creditable problem of sensing data, we analyze and design the credibility detection of sensing data, the establishment of reputation indicators, and large-scale data storage issues.

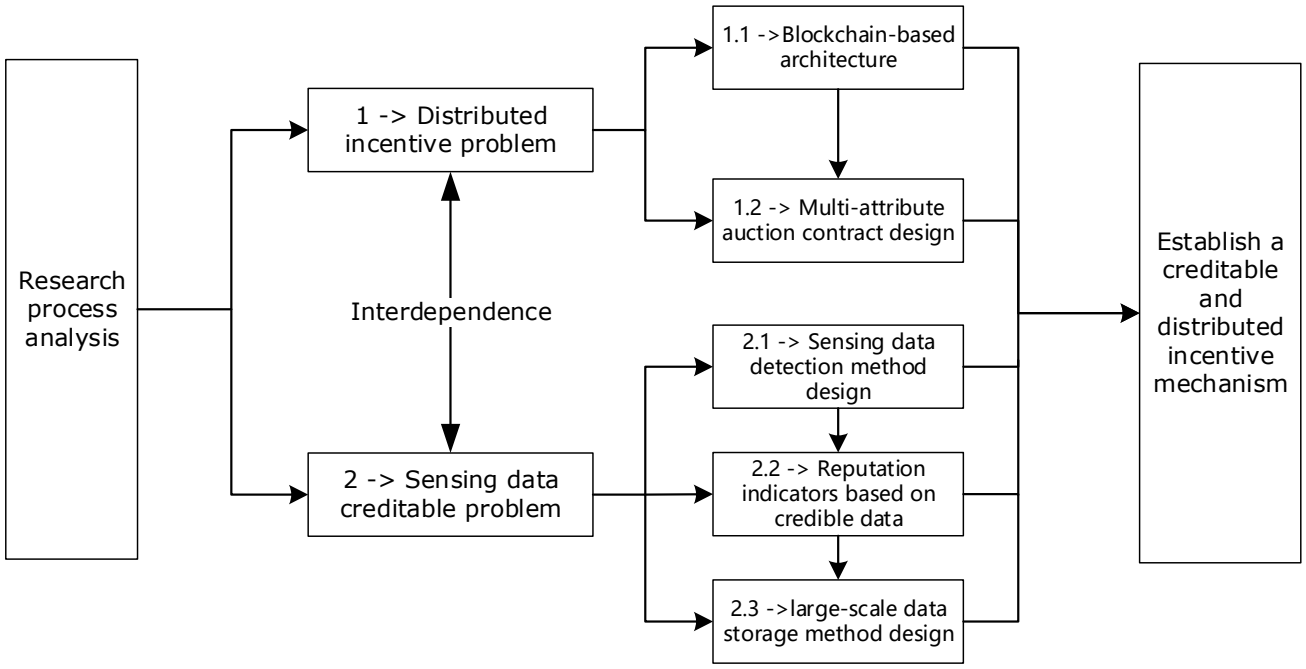

Figure 2. Research process.

Figure 2 also describes the relationship between the researched problems. Distributed incentives rely on users providing credible sensing data, and credible data can help to improve the fair problem of distributed incentives. The specific research on these two will be introduced in the following subsections.

3. System architecture

3.1. Blockchain-based creditable and distributed incentive architecture

Blockchain-based creditable and distributed incentive architecture is shown in Figure 3. We have modified the traditional centralized Mobile Crowdsensing incentive architecture in a distributed way.

Our architecture is mainly composed of three layers: users and devices, task platform and blockchain network. The users and devices layer includes users and devices in different sensing scenarios (i.e., participants in sensing tasks). The task platform layer contains edge servers, application servers, which are responsible for the task publishment, task assignment, reputation calculation after the task is completed. The blockchain network layer is mainly the underlying distributed storage architecture based on Hyperledger Fabric, which ensures that the uploaded data and reputation indicators cannot be tampered with. The smart contract deployed on the blockchain achieves a distributed and automated incentive mechanism.

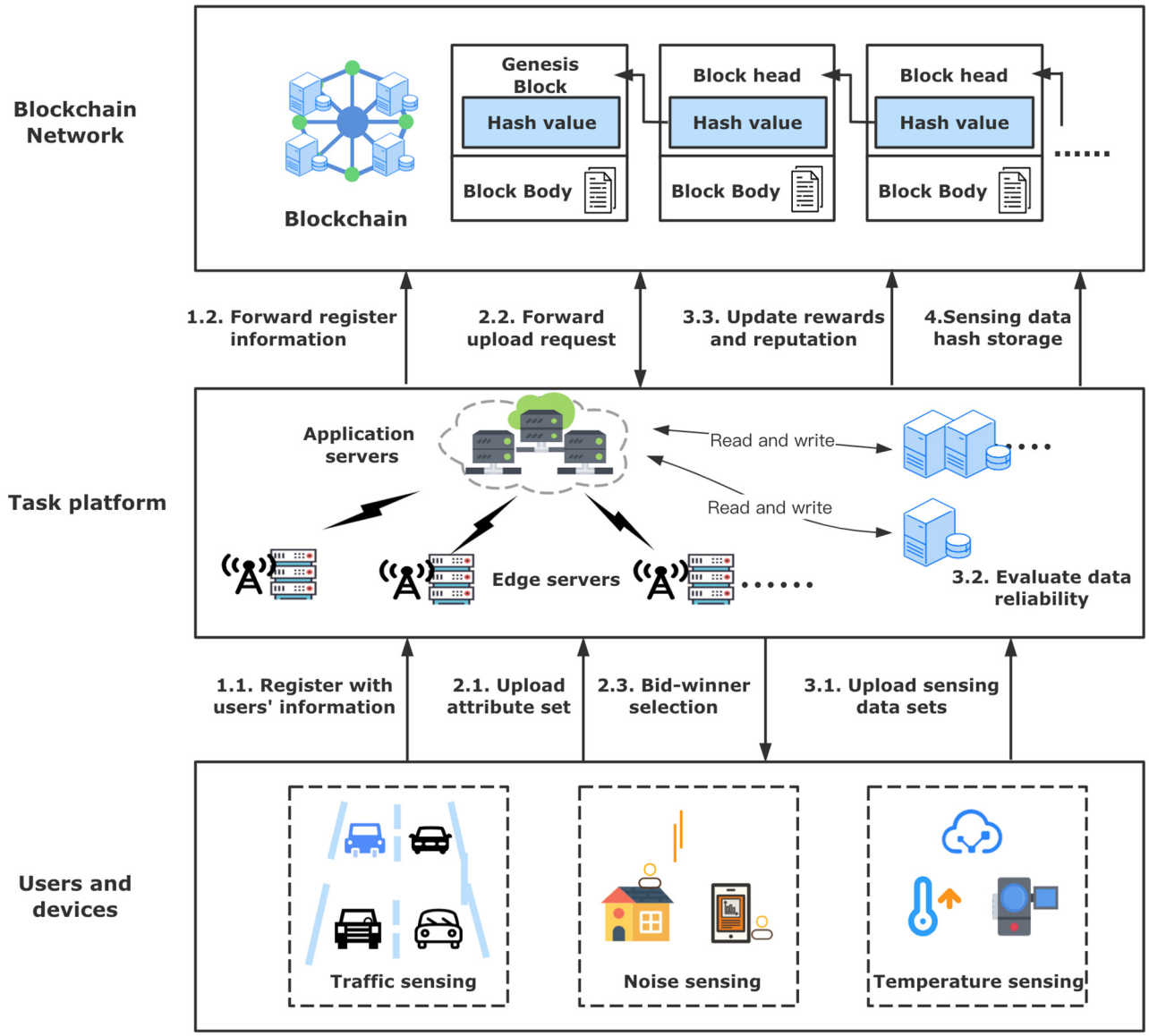

Figure 3. Blockchain-based creditable and distributed incentive architecture.

In this paper, we develop and deploy the following types of contracts with blockchain business logic on the Hyperledger Fabric to achieve credible and distributed incentives:

- *Contract A (Attribute Set Storage Contract):* Attribute set storage contract is implemented to upload and store the user's attribute set on the blockchain.
- *Contract B (Multi-attribute Auction Contract):* The multi-attribute auction contract is implemented based on the scoring function, which is calculated by user's attribute set. This contract consists of three steps: First, call contract A to obtain the user's attribute set. Then, calculate the scoring function and rank based on the score. Last, feedback the ranking results to the application or edge servers.
- *Contract C (Reputation Update Contract):* The reputation update contract is implemented to store basic information when users register, and update users' reputation and reward values after a transaction.

- *Contract D (Sensing Data Storage Contract):* The sensing data storage contract is implemented to store the hash value of the large-scale sensing data and ensure data integrity verification.

The overview of our blockchain-based creditable and distributed incentive architecture is described as follows:

Stage 1. User registration

First, users send registration information to the platform to finish registration. *Then,* the blockchain stores basic information about users' registration.

Stage 2. Sensing tasks assignment

First, Users upload the attribute set to the blockchain network according to the multi-attribute set by the buyer. *Second,* the multi-attribute auction contract of the blockchain network sorts the user attribute set according to the scoring function and blockchain returns the results to the platform. *Third,* the task platform selects bid-winners to assign sensing tasks based on the ranking results on the blockchain and notifies users to upload sensing data.

Stage 3. Sensing data detection

First, Users upload sensing data according to the assigned sensing tasks. *Then*, the platform evaluates data reliability with a sensing data detection algorithm. *Then,* the reputation update contract conducts reputation of the blockchain network with the evaluation of user sensing data, and update user reputation and reward values.

Stage 4. Sensing data storage

Sensing data storage contract of the blockchain network synchronizes the hash value of the sensing data corresponding to this task on the blockchain, and uses the stored hash to verify the integrity of the data.

4. Sensing tasks assignment based on multi-attribute auction algorithm

The purpose of sensing tasks assignment is to select suitable users (bid-winners) to be sensing participants who process and send sensing data to the platform. The selection basis of participants mainly includes scoring function and reputation value based on users' attributes. However, this process is usually performed by a third-party platform, which lacks transparency to users and cannot guarantee the credibility and fairness of the selection process. Therefore, users are still reluctant to participate in the sensing task and the effectiveness of the incentive mechanism cannot be guaranteed, and it is difficult to mobilize the enthusiasm of users to participate in mobile crowdsensing. In this paper, we proposed a multi-attribute auction based on smart contract. In the selection process of bid-winners, multiple attributes of users were comprehensively considered to ensure the benefits of the platform. The auction process is automatically executed by smart contracts on the blockchain to ensure the credibility of the results.

4.1. Multi-attribute auction description model

The multi-attribute auction model is simplified to M with 7-tuple.

$$M = \{P, I, A_{attr}, U_b, U_s, B_{attr}, S_{attr}\} \tag{1}$$

P: Task publishers or servers who are task auctioneers (buyers).

I: Task bidders (sellers). The total number of task bidders is N.

$$I = \{1,2, \dots , N\} \tag{2}$$

A_{attr}: Attribute collection. Including m attributes (ep: user bid price, ad: frequency of adoption, at: adoption time, pl: sensing location, pa: positioning accuracy, bn: number of unsuccessful bids).

$$A_{attr} = \{ep, ad, at, pl, pa, bn\} \tag{3}$$

B_{attr}: Vector collection of attributes submitted by task bidders.

$$B_{attr} = \{B_1, B_2, \dots, B_n \dots, B_N\}, B_n = \{ep_n, ad_n, at_n, pl_n, pa_n, bn_n\} \tag{4}$$

U_b: The utility function of buyers, which reflects the value of sensing data.

U_s: The utility function of sellers, which reflects the difference between user bid price and cost of sensing data.

S_{attr}: The scoring function set by task auctioneers. Buyers calculate the bid submitted by the seller through the scoring function to determine the bid-winners.

Based on the above auction description model, we deployed an auction contract on the blockchain to achieve automation and transparency of distributed incentives. The smart contract of multi-attribute auction (Contract A and Contract B) is built on the logic of user participation and auction process. The multi-attribute auction process based on smart contract is defined as follows:

1) Buyer's task publication to blockchain: Before the auction, the buyers announce the multiple attributes ($\mathrm{A_{attr}}$) (price attributes and non-price attributes), data quantity and scoring rules contained in the auction item.

2) Sellers' bidding pricing decision: The sellers calculate the bid price (the first attribute) by solving the decision-making problem of maximizing the expected utility (related to the bid price and sensing cost) according to its utility function ($\mathrm{U_s}$). The sellers make price decision by themselves and the seller's bidding pricing decision process is executed off-chain. This part does not need to be written into the smart contract, so we will not discuss it in detail in this paper. For the specific bid pricing decision-making algorithm, we implement it with reference to [29].

3) Seller's bid sending to blockchain: Then the sellers seal the bid with multiple attributes to the buyers ($\mathrm{B_{attr}}$) based on the information provided by buyers.

4) Bid-winner selection by smart contract: According to the scoring rules, the bidding items are scored ($\mathrm{S_{attr}}$). According to the buyer's demand for the number of data, the top x sellers with the highest scores won the bid. The pricing decision is consistent with the bid of winners.

5) Sensing data transaction through blockchain: After obtaining the optimal auction solution, the sellers who win the bid will provide the buyer with the sensing data according to the attribute combination of the bid. The buyer will pay according to the price attribute in the bid and sellers will receive corresponding rewards. The sellers who did not win the bid received zero revenue.

4.2. Utility function of auction parties

Utility of Buyers: This utility function can evaluate the value of sensing data to buyers, including the influence of all attributes of the task bidders performing the sensing task on the buyer. In this paper, we design the utility function as follows.

$$U_b = \sum_{1}^{m} w_i * {A_i}^{\alpha}, \sum w_i = 1, 0 < \alpha_i < 1 \tag{5}$$

where w is the corresponding attribute weight, A_i corresponds to the i-th attribute in attribute space A_{attr}, and the parameter α ensures that the marginal utility of the attribute is not increasing.

While in actual situations, different types of attributes can't be calculated directly, and the elements in A_{attr} need to be normalized first. In this paper, we divide the seller's attributes into three categories, namely cost type, benefit type and interval type. The following describes how to normalize different types of attribute values.

1) Cost type: user bid price and positioning accuracy (assuming n bidders, v_{min} represents the minimum of attribute values, v_{max} represents the maximum of attribute values). A_i includes ep and pa.

$$r_{attr} = \frac{v_{min}}{A_i} \tag{6}$$

2) Benefit type: frequency of adoption, adoption time and number of unsuccessful bids. A_i includes ad, at, bn.

$$r_{attr} = \frac{A_i}{v_{max}} \tag{7}$$

3) Interval type: Sensing location (pl). A_i includes pl.

$$r_{attr} = \begin{cases} 1 - \dfrac{\max(q_1 - A_i, A_i - q_2)}{\max(q_1 - v_{min}, v_{max} - q_2)}, A_i \notin [q_1, q_2] \\ 1 \qquad\qquad\qquad\qquad\qquad\quad , A_i \in [q_1, q_2] \end{cases} \tag{8}$$

where $[q_1, q_2]$ is a fixed interval. Sensing location(pl) is an interval attribute, the closer the attribute value is to the fixed interval $[q_1, q_2]$, the better utility it will be. If the attribute is inside the interval, it should be 1.

The utility of sellers will not deploy on the smart contract. We implement it with reference to [29]. Due to space constraints, this article will not be introduced in detail.

4.3. Optimal auction solution based on smart contract

After the blockchain receives the information of buyer's task publication and seller's bid, the smart contract will automatically execute the bid-winner selection process according to scoring rules. The scoring rules are formulated by the buyers according to their utility function

Scoring function: The auctioneers describe the characteristics of the item through the scoring rule, and calculate the attributes submitted by the bidders through the scoring rule to determine the bid-winners, its description is as follows.

$$S_{attr} = \sum_{1}^{m} w_i * {A_i}^{\alpha} + \beta * ur, \sum w_i = 1, 0 < \alpha_i < 1 \tag{9}$$

We define the scoring function to be consistent with the task publishers' utility function, adding an extra parameter ur. ur is the user's reputation value, which measures the credibility of bidders. The calculation and acquisition method of ur will be described in detail in Section 5.3. β is the weight parameter that can adjust the influence of the user's reputation value on the scoring function. The scoring function reflects the score of the data set by calculating the weighted attributes of the attribute set.

For N submitted bids, the smart contract determines the x bid winners according to the data demand of buyers, as given by

$$max \sum_{n \in X} S_{attr}(B_n), where\ 1 \le n \le N \tag{10}$$

The set of bid winners is represented by X, $X \subseteq I$.

Then, according to the normalized data of users, the utility function is calculated according to Formula (5), and the scoring function is calculated according to Formula (6). The scoring function results are sorted and stored by smart contract A and smart contract B on the Hyperledger Fabric. Then the top k bidders with higher scores will be the bid-winners, the platform will assign the corresponding sensing tasks to them. The above multi-attribute auction algorithm based on smart contract is summarized as Algorithm 1, which is described as follows.

Algorithm 1: Multi-attribute auction algorithm based on smart contract

Input The users' attribute set : $AttrSet(A_i) = \{A_1, A_2, \dots, A_n\}, A_i = \{ep, ad, at, pl, pa, bn\}$, users' reputation $ur(i) = \{ur_1, ur_2, \dots, ur_n\}$, task publisher attribute weight setting: $W = \{w_1, w_2, w_3, w_4, w_5, w_6\}$

Output RankDlict

Initialize the dlict: $Score = \{s_1, s_2, \dots, s_n\}, RankDlict\{Score, user_i\}$
Read the $AttrSet(A_i)$ stored on the contract according to the $task_i$
for j = 0; j < length(AttrSet); j++
    Normalize the A_i according to Formulas (6)–(8)
    Calculate the s_j based on the W and Formula (9)
    Appending s_j the corresponding $Score$
end for
According to $Score$ and $user_i$ modify $RankDlict$
Write to the corresponding ledger of the contract
Return $RankDlict$

4.4. Theoretical analysis

Effective incentive mechanism design usually needs to meet some properties. We will provide proof of the properties of our proposed mechanism in this section, including individual rationality, incentive compatibility and efficient.

- Individual rationality (IR): Each participant in the auction will build its own strategy and provide its own price. When gets rational participants a non-negative utility, they are willing to participate in the auction.
- Individual rationality (IC): Participants can get the maximum benefit only when they honestly disclose their intentions.

Theorem 1. The proposed multi-attribute auction algorithm based on smart contract has the

properties of individual rationality and individual rationality.

Proof. First of all, the price attribute of the bidder is determined by itself according to its positive utility function. If the bidder fails to win the bid, it does not need to perform the task with zero cost. The bidder has no loss of interest. Therefore, the individual rationality (IR) is satisfied. Then, the scoring function of multi-attribute auction is not only related to the price attribute. It can be seen from Formulas (6) and (7) that the bidder's score is a decreasing function of the price attribute and an increasing function of the income attribute. If the bid price is too high, it will reduce its bid winning rate. Therefore, the bidders can get the benefit only by reporting his real attribute. Therefore, the mechanism is incentive compatible (IC).

Then we analyze the computational complexity of the proposed algorithm.

Computational complexity: The algorithm is divided into three steps. The first step calls Contract A to store the data set. Then calls Contract B to execute auction solution. The second step normalizes the data, and finally step calculates the ranking according to the scoring function. The computational complexity of our proposed algorithm is $o(n)$. The model implements business logic based on blockchain contracts, thus providing a solution for users to participate in distributed incentives.

5. Sensing data detection and integrity verification

After sensing tasks assignment with auction algorithm, bid-winners will send the data of sensing task to the platform. But this process cannot guarantee the reliability of sensor data. In this paper, we propose a sensing data detection method with a K-nearest neighbor outlier detection algorithm based on geographic location and similarity. The process of sensing data detection is mainly to check the reliability of the sensing data uploaded by users, and update the users' reputation and store it in the blockchain. Then, in order to solve the efficiency problem of storing large-scale sensing data on the blockchain, we propose to store the data hash on the blockchain. By comparing hash values of large-scale off-chain data, the integrity of sensing data is verified.

5.1. K-nearest neighbor outlier detection algorithm based on geographic location and similarity

In the actual situation of Mobile Crowdsensing, the sensing data of users with close geographical positions are likely to be similar. Therefore, we can consider combining geographical position and data similarity to evaluate data credibility. Based on this, we propose a geographical K-nearest neighbor outlier detection algorithm based on geographic location and similarity. The algorithm implementation process is as follows.

The sensing data set uploaded by the user is:

$$DataSet(T_i) = \{T_1, T_2, \dots, T_n\} \tag{11}$$

$$T_i = \{id_i, time_i, longitude_i, latitude_i, dataitem_i\} \tag{12}$$

First, the user (X, Y) data similarity $F_{\sin(X,Y)}$ is expressed by the following formula.

$$F_{\sin(X,Y)} = \frac{\sum_1^d \frac{m_i}{|x_i - y_i| + m_i}}{d} \tag{13}$$

where X, Y are the sensing data vectors provided by the users, m_i is the absolute value of the average

value of the *i*-th column data, and d is the dimension.

Then, we could obtain the user's similarity matrix based on Formula (13).

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} \tag{14}$$

$$w_{ij} = F_{\sin(i,j)}$$

For each $user_i$, according to $longitude_i, latitude_i$, find out the distance between other users and him.

$$dist_i = geohash(longitude_i, latitude_i) \tag{15}$$

$$dist_{ij} = \left|dist_i - dist_j\right| \tag{16}$$

$$dist_{sorted-i} = sort(dist_{i1}, dist_{i2}, \ldots, dist_{in}) \tag{17}$$

where $geohash$ converts the two-dimensional latitude and longitude into a string.

And then we pick the k nearest neighbors.

$$Neighbors_{k-i} = dist_{sorted-i}[:k] \tag{18}$$

Finally, according to $Neighbors_k$, the k neighbor degree matrix (the sum of the similarities of the k nearest users in space) is obtained according to the similarity matrix W, which reflects the degree of similarity between each data set and the neighbor data sets.

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix} \tag{19}$$

$$d_{ii} = sum(Neighbors_{k-i})$$

Determine the outlier by the similarity in the degree matrix and the set threshold:

$$ranking_{ratio} = \frac{ranking}{number} \tag{20}$$

where $number$ is the total number of participants who submitted data. $ranking_{ratio}$ is used to judge data as outliers of users who sent unreliable data. Then we set outlier threshold θ_r. Data with $ranking_{ratio}$ outside the threshold range θ_r is an outlier. The above algorithm is summarized as Algorithm 2.

5.2. Build reputation indicators

Users' reputation marks the honesty of users, which is an important factor that affects the assignment of sensing tasks. Therefore, we add user reputation indicator management to get corresponding punishment after users submit false data. After the user submits the sensing data, the user's reputation is updated. In this paper, we use a joint weight method based on the user's current

reputation and data credibility to update user's reputation.

The update formula is as follows.

$$ur = \begin{cases} ur^0 & if\ data \notin outliers\ and\ |D| \le ur^0, \\ w_1 * |D| + w_2 * ur^0 & if\ data \notin outliers\ and\ |D| > ur^0 \\ w_2 * ur^0 & if\ data \in outliers \end{cases} \quad (21)$$

where $|D|$ is the value of the normalized degree matrix, ur^0 is the user's current reputation, w_1, w_2 are the corresponding weight values, and $outliers$ is the set of outliers judged to be outliers.

After updating the reputation of each participating, the honesty of users can be further judged by calling contract C according to the reputation values. After the contract determines the user whose reputation meets the threshold, the attribute set uploaded by the user in the first step will be called to obtain the user's bid price. The bid price is the user reward value after completing the sensing task.

Algorithm 2: K-nearest neighbor outlier detection algorithm based on geographic location and similarity

Input Mobile Crowdsensing Dataset (Read in the data set csv. file): $DataSet(T_i) = \{T_1, T_2, \dots, T_n\}$, $T_i = \{id_i, time_i, longitude_i, latitude_i, dataitem_i\}$, and data time range of k neighbors: $tlimit$, threshold: θ_r

Output $Outlier$s

1 Map $longitude_i, latitude_i$ to strings (as:$geographic_i$) with the Geohash function

2 Sort the Dataset by multiple attributes: first by $geographic_i$ location, and then in ascending order by $time_i$

3 Initialize the similarity array:

$$Score = \{s_1, s_2, \dots, s_n\}, Dlict = \{Score, id_i\}$$

4 for each j = 1 to $j \le length(DataSet)$; j++ **do**

5 $NearK = \{id_1, id_2, \dots, id_k\}$ according to $id_i, geographic_j, tlimit$

6 **for** $u = 1; u \le length(NearK); u$++ **do**

7 Calculate the $dataitem_j$ similarity $simi_{ju}$ according to Formula 18 (id_j, id_u)

8 $s_j = s_j + simi_{ju}$

9 **end for**

10 Appending s_j to the corresponding $Score$

11 end for

12 According to $Score\ and\ id_i$ modify $Dlict$

13 Sort the $Dlict$ based on s_i

14 Obtain $Outliers$ that are not within the threshold range according to the threshold θ_r

15 return $Outliers$

5.3. Large-scale data storage and integrity verification

According to the distributed incentive architecture designed above, we need to consider how to store large-scale sensing data. The existing methods include off-chain and on-chain storage. From the perspective of security, storage on the blockchain is safer than storage outside the chain because each node will save a copy of all the data. However, from the point of view of performance analysis, data storage and data reliability analysis of nodes on the blockchain, a large amount of storage space of blockchain is required. In connection with large-scale participation user scenarios, there will be

obvious problems such as poor performance when tasks are executed at the same time.

In order to balance the security and performance, we adopted an on-chain verification and off-chain storage (OVOS) method to ensure that sensing data cannot be tampered with while ensuring the execution performance of distributed incentive mechanism. This method is explained below.

We take a task as a unit, assuming the task data set is (the user-submitted data set corresponding to the task, in order of ranking) $data_{taskdown}$. The SHA256 algorithm is a cryptography hash function, which produces as output a 256-bit message digest of the input. [30] The fingerprint corresponding to the task data set (hash value corresponding to the task data set) is defined as:

$$hash_{taskdown} = sha256(data_{taskdown}) \tag{22}$$

$hash_{taskup}$ is the initial hash value of the task data set:

$$hash_{taskup} = sha256(data_{taskdown}) \tag{23}$$

Storage content description under the chain:

$$data_{down} = \{data_{taskdown}, hash_{taskdown}\} \tag{24}$$

$$data_{up} = \{task_{id}, hash_{taskup}\} \tag{25}$$

When the task publisher requires the use of data, the hash value corresponding to the task on the blockchain can be read first, and the hash value of the task data set on and off the chain (the hash needs to be recalculated each time) can be compared to determine whether the data has been tampered with. A large number of sensing data storage problems can be realized while ensuring that data is not tampered with. Storage validation is shown in Figure.4.

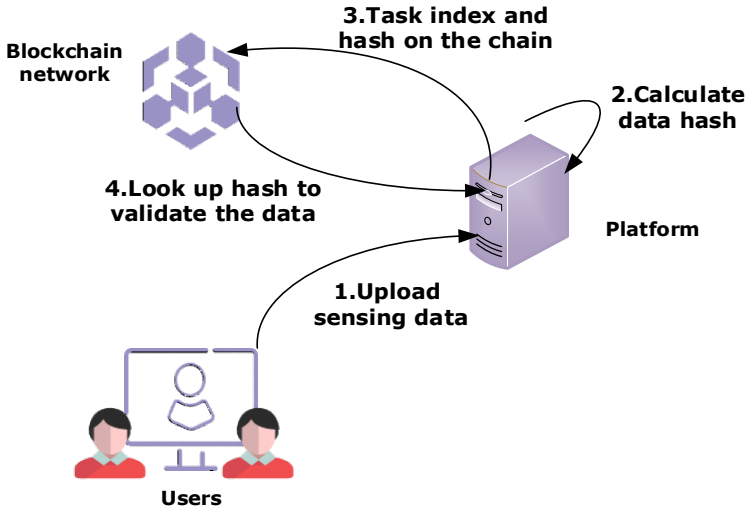

Figure 4. Data storage and validation.

Non-tamper-proof (data integrity) verification process is as follows (Assuming that no hash conflicts have occurred):

Without being tampered with:

$$hash_{taskup} = hash_{taskdown} \tag{26}$$

$$hash_{taskup} = \left(sha256(data_{taskdown})\right) \tag{27}$$

The data is tampered with, the hash is not tampered with:

$$hash_{taskup} = hash_{taskdown} \tag{28}$$

$$hash_{taskup} \neq \left(sha256(data_{taskdown})\right) \tag{29}$$

The data is not tampered with, the hash is tampered with:

$$hash_{taskup} \neq hash_{taskdown} \tag{30}$$

$$hash_{taskup} = \left(sha256(data_{taskdown})\right) \tag{31}$$

Data and hash were tampered with:

$$hash_{taskup} \neq hash_{taskdown} \tag{32}$$

$$hash_{taskup} \neq \left(sha256(data_{taskdown})\right) \tag{33}$$

The above validation procedure can determine the state of the data, and two comparisons are required each time.

6. Results and discussion

In this section, we first explain the simulation experiment settings. Then we carry out the experiments on the performance of different algorithms and on-chain execution contracts respectively, at the same time discuss the simulation results.

6.1. Simulation settings

To analyze the performance of the creditable and distributed incentive mechanism based on Hyperledger Fabric (HF-CDIM) for MCS in this paper. We set up an experimental environment based on Hyperledger fabric. The hardware environment is windows10 (AMD4800H + 16G + working frequency 2.9GHZ) + Linux (Ubuntu 18.04 64-bit) to deploy hyperledger2.0. The blockchain network is implement with docker (two organizations and four nodes) and the smart contracts are written in go language and deployed on the blockchain test network. Then, we will introduce the basic experiment settings and the used datasets in this section.

Our simulation is mainly divided into the following three experiments:

First, we analyzed the performance of multi-attribute auction algorithm. Specifically, we compared our multi-attribute auctions method with traditional single-attribute auction method (price as a single attribute). We respectively compared the impact of the number of attributes, bid-winners, and participants on performance.

Then, we analyzed the performance of proposed data credibility detection algorithm. We selected outdoor temperature data set collected by taxis in Rome, Italy [31] as the test set and inserted tampered data for users in different areas, which is different from normal data. We respectively compared the impact of the number of participants and different k value on performance of sensing data detection algorithm.

Finally, we evaluated the performance of entire distributed incentive mechanism with smart contracts based on Hyperledger Fabric. We deployed the chain-code of attribute set storage contract, multi-attribute auction contract, reputation update contract and sensing data storage contract on Hyperledger Fabric, and analyzed the running time and transactions per second (TPS) as evaluation indicators.

The simulation test data sets in this article are described in Table 2.

Table 2. Simulation data sets description.

Experiment name	Data set description	Source
Sensing task assignment based on multi-attribute auction algorithm	Users' attribute data set	Simulate a 1000*1000 area, limit different attribute size ranges, and generate corresponding attribute values uniformly and randomly.
K-nearest neighbor detection algorithm based on geographic location and similarity	Sensing data set	Outdoor temperature data collected by taxis in Rome, Italy, a single csv file, size 406KB. [31]
Performance of the entire incentive mechanism based on Hyperledger Fabric	Users' attribute data set and sensing data	Same as the data set of multi-attribute auction algorithm.

6.2. Performance of sensing task assignment based on multi-attribute auction algorithm

The input to the sensing task assignment based on multi-attribute auction algorithm is the users' attribute data set submitted by the bidders. The data set was first normalized to avoid the influence of different types of attributes. Figure 5 compares the effect of the number of attributes on the utility value of the task publisher in the case of a single attribute and multiple attributes. As shown in Figure 5, the performance of sensing task assignment based on multi-attribute auction algorithm is better than a single-attribute one. The utility of task publisher increases as the number of attributes increases in multi-attribute auction, and the utility value of a single price attribute in single-attribute auction does not change. This is because that not only the bid price attribute affects the decision of sensing tasks assignment, other user attributes (e.g., frequency of adoption, adoption time and sensing location, etc.) have an impact on the utility of task publisher. Multi-attribute auction algorithm that we proposed in this paper can reflect the task publisher's preference of the data through the score function compared with a single attribute of bid price.

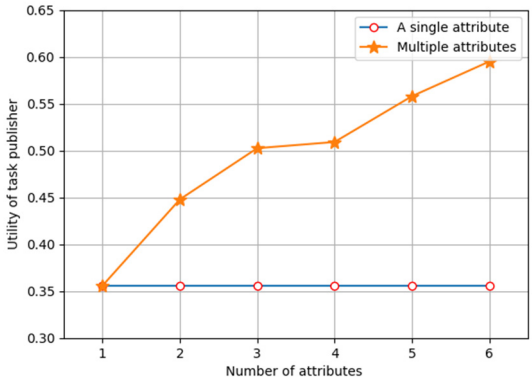

Figure 5. Comparison of single-attribute auction and multi-attribute auction.

Figures 6 and 7 show the impact of the number of bidders and bid-winners. Figure 6 shows the relevance between the number of bid-winners and utility of task publisher in multi-attribute auction algorithm and single-attribute auction algorithm. The number after the symbol '-' in legend (200, 400, 600) means the total number of bidders who participate in the auction. As shown in Figure 6, when the number of bidders is the same, as the number of bid-winner settings increases, the utility of the task publisher gradually decreases (here, the utility of the task publisher means the average utility that each bidder-winner can provide). This is because the auction algorithm in this paper assigns sensing tasks in descending order of the user's score function. The user's score function is related to the utility of the task publisher. Therefore, the more bid-winners the task publisher sets, the average utility that each bidder-winner can provide to task publishers will inevitably gradually decrease. In addition, when the bid-winner number set by the task publisher is given, it can be found that the more the total number of people participating in the auction, the higher the utility of the task publisher. It also shows that it is necessary to deploy appropriate incentive mechanisms in mobile crowdsourcing to motivate more people to participate in sensing tasks.

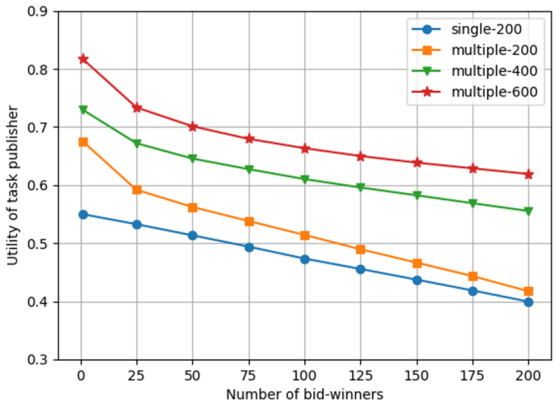

Figure 6. Utility comparison with different number of bid-winners and total bidders.

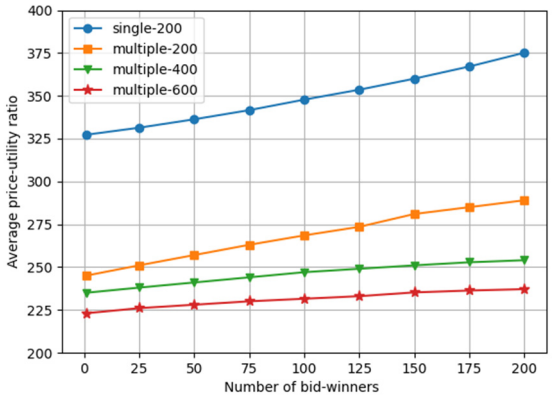

Figure 7. Price-utility ratio comparison with different number of bid-winners and total bidders.

Figure 7 simulates the relevance between the number of bid-winners and the price-utility ratio of task publisher in multi-attribute auction algorithm and single-attribute auction algorithm. Price is the first attribute of bidders, which needs to be paid by task publishers. The price-utility ratio refers to the cost-effectiveness ratio of the task publisher to complete the task. As shown in Figure 7, the price utility ratio increases as the number of bid-winners increases, that is, the task publisher gets the data with the same utility, but the cost increases. We could also find that tasks using a single attribute algorithm cost more than a multi-attribute one. In addition, as the number of participating bidders increases, the cost of obtaining the data with the same utility can be effectively reduced. Because when the number of bid-winners is fixed and the number of participating bidders increases, high quality-data can be obtained more widely and the cost of publishers will be reduced due to the randomness of the bidding price of users. Similarly in the case of a fixed number of participating bidders. The above results are consistent with actual results. The result also proved the effectiveness of our proposed method.

6.3. Analysis of K-nearest neighbor outlier detection algorithm based on location and similarity

We will design experiments to analyze the performance of K-nearest neighbor outlier detection algorithm based on geographic location and similarity. The data set is the same as in literature [31]. Figure 8 is a scatter plot of a real sensing dataset collected by taxis in Rome, Italy. The horizontal and vertical axes represent latitude and longitude. The blue dots in the figure are the normal points with true data, and the red dots are the abnormal points we have tampered with.

In this paper, accuracy, precision, recall and F1-measure [32] are used to evaluate the performance of the outlier detection algorithm. Accuracy represents the ratio of correct judgments for all data. Precision represents the ratio of correctly detected abnormal points to all abnormal points detected. Recall represents the ratio of correctly detected abnormal points to all actual abnormal points. F1-measure achieves a trade-off between recall ratio and precision. The calculation method is as follows:

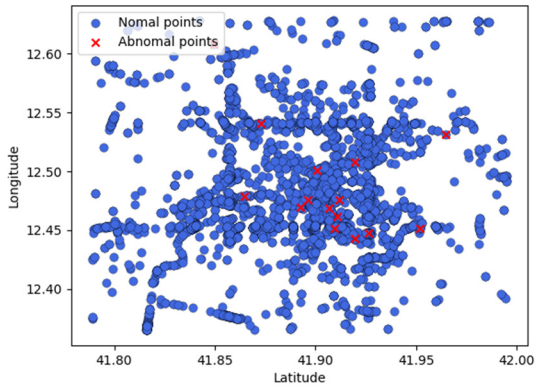

Figure 8. Normal points and abnormal points of the dataset.

$$precision = \frac{TP}{TP + FP} \tag{34}$$

$$recall = \frac{TP}{TP + FN} \tag{35}$$

$$F1 - measure = \frac{2 * precision * recall}{precision + recall} \tag{36}$$

True positive (TP) represents the number of real anomalies that are correctly detected as anomalies. True negative (TN) represents the normal points that are correctly identified. False positive (FP) presents the normal points that are incorrectly identified as abnormal points. False negative (FN) represents the abnormal points that are identified as normal.

Next, we will conduct an experimental analysis on the impact of the three parameters (including outlier threshold θ_r, the number of participants N and the size of the k value) on the performance of the proposed method.

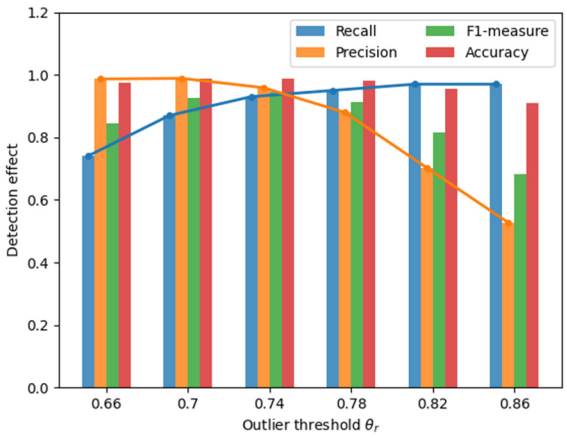

Figure 9. Detection effect of different outlier thresholds ($N = 1000$).

Figures 9 and 10 show the impact of outlier threshold θ_r on detection effect. As shown in Figure 9, our proposed outlier detection algorithm could reach more than 95% accuracy when threshold $\theta_r = [0.66,0.82]$. And detection accuracy reaches 98.9% when $\theta_r = 0.74$. The experimental results proved the high effectiveness of our proposed algorithm.

We can also find that as the threshold increases, the recall gradually becomes larger and the precision gradually becomes smaller in Figure 9. We analyze the reasons for the above phenomenon as follows: When θ_r is set too small, some abnormal points will escape detection, which causes a larger precision and a smaller recall. When θ_r is set too large, some normal points with similarity not that large will be detected as abnormal points. In this case, there will be a larger recall and a smaller precision. Therefore, in the following experiments, we use F1-measure (harmonic mean of precision and recall) to evaluate the detection effect.

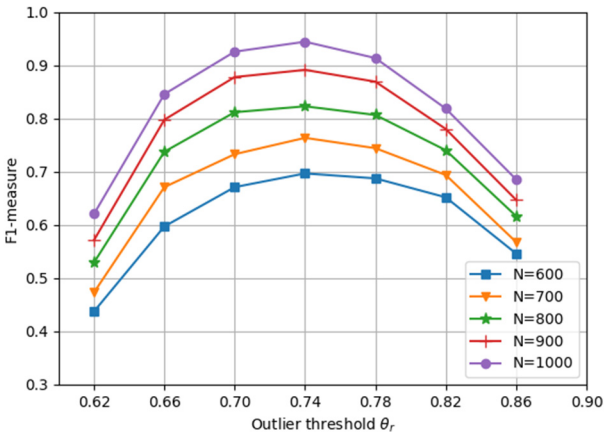

Figure 10. Detection effect of different outlier thresholds and N $(k = 6)$.

Figure 10 shows detection effect of different outlier thresholds ($\theta_r = \{0.62,0.66,0.70,0.74,0.78, 0.82,0.86\}$) under different number of participants ($N = \{600,700,800,900,1000\}$). The outlier threshold is set to judge unreliable data based on the data similarity matrix, which has a great influence on the detection effect. As the outlier threshold is set larger, F1-Measure first becomes larger and then smaller, and reaches the peak value at $\theta_r = 0.74$. And the above conclusions are all established under different N values. We have obtained θ_r that can make the best detection effect, and the following experiment sets $\theta_r = 0.74$.

From the longitudinal observation of Figure 10, we can find that the F1-measure increases with the increase of the number of participants, indicating that the more people participating in the sensing task, the better the detection effect of abnormal data. Because within the same geographic area, the more people participate, the closer locations of the k neighbors can be obtained by the data detection algorithm. Then we can more accurately evaluate the similarity between data. If users tamper with or upload malicious data, the similarity will be fluctuating thereby detecting abnormal data. It shows the necessity of distributed incentive mechanism and the effectiveness of our proposed outlier detection algorithm in mobile crowdsensing with large-scale edge devices.

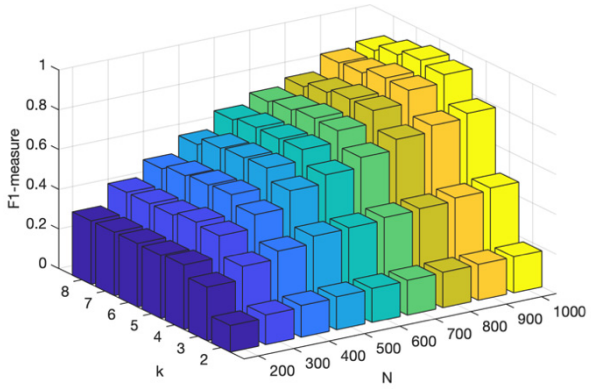

Figure 11. Detection effect of different k values and N ($\theta_r = 0.74$).

Figure 11 shows the detection effect of different k value ($k = \{2,3,4,5,6,7,8\}$) and $N(N = [200,1000])$. Under the same number of participants, F1-measure increases first and then decreases with the increase of k, and reaches its peak value at $k = 6$. This is because if the k value is too large, neighbors that are farther away will be found, and the similarity results will be inaccurate. If the k value is too small, the similarity calculated by the neighbor's data is accidental, and the detection result may also be inaccurate.

Table 3. Comparison of proposed data detection method and other method.

Data detection method	Proposed data detection method	Random forest-based method	One-class SVM based method
F1-measure	0.9441	0.9059	0.9148
Accuracy	0.9891	0.8436	0.8572

Finally, we compare the performance of our proposed data detection method with the two baseline methods. The baseline methods including Random forest-based method and One-class SVM based method. The method proposed in this paper can better capture the characteristics of sensing data. We add the analysis of data similarity on the basis of considering the location feature information of data, then obtain better performance. It can be seen from the experimental results in Table 3 that our proposed method performs better than the baseline algorithm in terms of accuracy and F1-measure.

6.4. Running time and TPS of entire distributed incentive mechanism with smart contracts based on Hyperledger Fabric

We evaluated the performance of the entire distributed incentive mechanism with smart contracts based on Hyperledger Fabric in this section. We built a blockchain network based on Hyperledger Fabric and deployed the chain-code of attribute set storage contract, multi-attribute auction contract, reputation update contract and sensing data storage contract on Hyperledger Fabric.

In the actual crowdsensing process, we first considered the running time of executing the multi-attribute auction algorithm on the blockchain, whether the model can rapidly execut task will be an important factor that the efficiency of the sensing task completion. In view of the actual situation, the

users submit the attribute set at different time periods during the sensing task, so the following running time does not consider the writing process of the attribute set (i.e., the processing of attribute set storage contract). We mainly consider the running time of processing the multi-attribute auction contract, including reading the attribute set corresponding to the users, executing the multi-attribute auction algorithm with the contract and writing the sorting results of the corresponding task into the ledger. The experimental results are shown in Figure 12 and Table 3. We tested the average running time of each participant and the average running time of sensing task with a different number of participants. We selected a sensing task and performed the multi-attribute auction contract 10 times and tested the running time for sensing task (setting the number of participants as 50, 100, 150 and 200). Then we calculated the average running time.

Table 4. Average running time of executing the multi-attribute auction.

Number of participants	50	100	150	200
Average running time of sensing task (ms)	613	806	1159	1380
Average running time of each participant (ms)	12.275	8.06	7.73	6.9

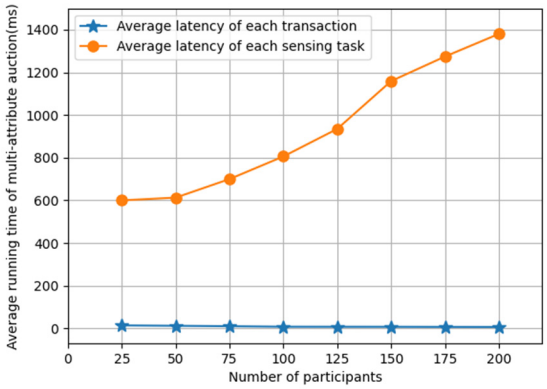

Figure 12. Average running time of different participants.

As shown in Figure 12 and Table 3, as the number of participants increases, the running time will increase monotonically. However, the average running time of each participant does not fluctuate greatly with the number of participants. The main reason is that the blockchain itself will have a certain delay in writing data and the consensus process, and as the number of participants increases, this process running time must increase, but for the average running time, the running time of writing and consensus is certain, and there will be no big fluctuations.

TPS means transactions per second, which is obtained by dividing the number of transactions in each block by the block generation time. The TPS of blockchain network is an important indicator of our distributed incentive mechanism. With the help of the Tape tool, we tested the read-write throughput of the built network and compared our blockchain network with Ethereum and Bitcoin.

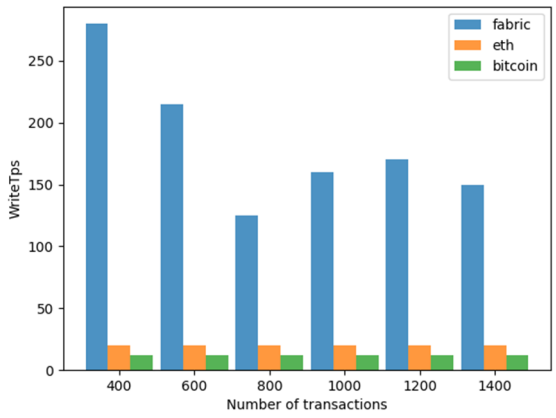

Figure 13. Bitcoin, Ethereum, Hyperledger fabric TPS comparison.

As shown in Figure 13, the TPS of the Hyperledger is obviously faster than that of Bitcoin and Ethereum. This is because the latter two are public chains, which involve the process of coin issuance, and the consensus protocols used between them are also different, so the TPS of the Hyperledger is relatively high overall.

It can be known from the real effect of algorithm simulation in this section that it is feasible and effective to implement distributed incentive mechanisms on Hyperledger Fabric.

7. Conclusions

In our paper, we propose a creditable and distributed incentive mechanism based on Hyperledger Fabric for mobile crowdsensing-based edge computing, which can effectively achieve distributed mechanism and data credibility. The distributed incentive mechanism proposed in this paper realizes the multi-attribute auction algorithm on the blockchain through the smart contract. And the K-nearest neighbor outlier detection based on geographical location and similarity can effectively detect the data credibility. The reputation indicator is established based on the data credibility, which enhances the fairness of the incentive and promotes the enthusiasm of participants. Simulation experiments using real data set proved the effectiveness of the proposed mechanism. In future research, we plan to conduct an in-depth exploration of data privacy-preserving method to improve participants' privacy, making the system model to meet real-world users' needs.

Acknowledgments

This work was supported by National Key R&D Program of China (2020YFB1807802, 2020YFB1807800).

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. H. Vestberg, CEO to Shareholders: 50 Billion Connections 2020, Ericsson Inc., Stockholm, Sweden, 2010. Available from: https://www.ericsson.com/en/press-releases/2010/4/ceo-to-shareholders-50-billion-connections-2020
2. R. K. Ganti, F. Ye, H. Lei, Mobile crowdsensing: current state and future challenges, *IEEE Commun. Mag.*, **49** (2011), 32–39. https://doi.org/10.1109/MCOM.2011.6069707
3. R. Sánchez-Corcuera, A. Nuñez-Marcos, J. Sesma-Solance, A. Bilbao-Jayo, R. Mulero, Smart cities survey: Technologies, application domains and challenges for the cities of the future, *Int. J. Distib. Sens. N.*, **15** (2019). https://doi.org/10.1177/1550147719853984
4. L. Kong, Z. Wu, G. Chen, M. Qiu, J. Rodrigues, Crowdsensing based cross-operator switch in rail transit systems, *IEEE T. Commun.*, **68** (2020), 1–1. https://doi.org/10.1109/TCOMM.2020.3019527
5. T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, H. Flinck, Mobile edge computing potential in making cities smarter, *IEEE Commun. Mag.*, **55** (2017), 38–43. https://doi.org/10.1109/MCOM.2017.1600249CM
6. P. Bellavista, S. Chessa, L. Foschini, L. Gioia, M. Girolami, Human-enabled edge computing: Exploiting the crowd as a dynamic extension of mobile edge computing, *IEEE Commun. Mag.*, **56** (2018), 145–155. https://doi.org/10.1109/MCOM.2017.1700385
7. V. S. Dasari, B. Kantarci, M. Pouryazdan, L. Foschini, M. Girolami, Game theory in mobile crowdsensing: A comprehensive survey, *Sensors*, **20** (2020), 2055. https://doi.org/10.3390/s20072055
8. N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, Z. Han, Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey, *IEEE Commun. Surv. Tut.*, **18** (2016), 2546–2590. https://doi.org/10.1109/COMST.2016.2582841
9. L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, M. Huang, TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial internet of things, *IEEE T. Mob. Comput.*, **99** (2021). https://doi.org/10.1109/TMC.2021.3064314
10. R. Iqbal, T. A. Butt, M. Afzaal, K. Salah, Trust management in social internet of vehicles: factors, challenges, blockchain, and fog solutions, *Int. J. Distrib. Sens. Netw.*, **15** (2018). https://doi.org/10.1177/1550147719825820
11. B. Guo, H. Chen, Z. Yu, W. Nan, X. Xie, D. Zhang, et al., TaskMe: Toward a dynamic and quality-enhanced incentive mechanism for mobile crowd sensing, *Int. J. Hum.-Comput. Stud.*, **102** (2017), 14–26. https://doi.org/10.1016/j.ijhcs.2016.09.002
12. D. Lin, Q. Wang, P. Yang, Z. Zhang, A multidimensional reputation evaluation model for mobile crowd sensing, in *2019 15th International Wireless Communications and Mobile Computing Conference, IEEE*, (2019), 2070–2073. https://doi.org/10.1109/IWCMC.2019.8766533
13. J. Kang, Z. Xiong, S. Xie, J. Zhang, Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory, *IEEE Int. Things J.*, **6** (2019), 10700–10714. https://doi.org/10.1109/JIOT.2019.2940820
14. Q. Li, H. Cao, S. Wang, X. Zhao, A reputation-based multi-user task selection incentive mechanism for crowdsensing, *IEEE Access*, **8** (2020), 74887–74900. https://doi.org/10.1109/ACCESS.2020.2989406
15. J. Xu, Z. Rao, L. Xu, D. Yang, T. Li, Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities, *IEEE T. Mob. Comput.*, **19** (2019), 1618–1633. https://doi.org/10.1109/TMC.2019.2911512

16. Z. Luo, J. Xu, P. Zhao, D. Yang, L. Xu, J. Luo, Towards high quality mobile crowdsensing: Incentive mechanism design based on fine-grained ability reputation, *Comput. Commun.*, **180** (2021), 197–209. https://doi.org/10.1016/j.comcom.2021.09.026
17. H. Gao, C. H. Liu, J. Tang, D. Yang, P. Hui, W. Wang, Online quality-aware incentive mechanism for mobile crowd sensing with extra bonus, *IEEE T. Mob. Comput.*, **18** (2018), 2589–2603. https://doi.org/10.1109/TMC.2018.2877459
18. C. Xiang, P. Yang, X. Fan, L. Gong. Quantifying sensing quality of crowd sensing networks with confidence interval, in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, IEEE*, (2017), 1–6. https://doi.org/10.1109/UIC-ATC.2017.8397538
19. I. Krontiris, A. Albers, Monetary incentives in participatory sensing using multi-attributive auctions, *Int. J. Parallel, Emerg. Distrib. Syst.*, **27** (2012), 317–336. https://doi.org/10.1080/17445760.2012.686170
20. Y. Liu, X. Xu, J. Pan, J. Zhang, G. Zhao, A truthful auction mechanism for mobile crowd sensing with budget constraint, *IEEE Access*, **7** (2019), 43933–43947. https://doi.org/10.1109/ACCESS.2019.2902882
21. T. Luo, J. Huang, S. S. Kanhere, J. Zhang, S. K. Das, Improving IoT data quality in mobile crowd sensing: A cross validation approach, *IEEE Int. Things J.*, **6** (2019), 5651–5664. https://doi.org/10.1109/JIOT.2019.2904704
22. Z. Zhou, X. Chen, Y. Zhang, S. Mumtaz, Blockchain-empowered secure spectrum sharing for 5G heterogeneous networks, *IEEE Network*, **34** (2020), 24–31. https://doi.org/10.1109/MNET.001.1900188
23. A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, S. Capkun, On the security and performance of proof of work blockchains, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, (2016), 3–16. https://doi.org/10.1145/2976749.2978341
24. I. Bentov, R. Kumaresan, How to use bitcoin to design fair protocols, in *International Cryptology Conference Springer Berlin Heidelberg*, (2014), 421–439. https://doi.org/10.1007/978-3-662-44381-1_24
25. N. More, D. Motwani, A blockchain-based decentralized framework for crowdsourcing, in *International Conference on Image Processing and Capsule Networks*, (2020), 448–460. https://doi.org/10.1007/978-3-030-51859-2_41
26. B. Jia, T. Zhou, W. Li, Z. Liu, J. Zhang, A blockchain-based location privacy protection incentive mechanism in crowd sensing networks, *Sensors*, **18** (2018), 3894. https://doi.org/10.3390/s18113894
27. J. Wang, M. Li, Y. He, H. Li, K. Xiao, C. Wang, A blockchain based privacy-preserving incentive mechanism in crowdsensing applications, *IEEE Access*, **6** (2018), 17545–17556. https://doi.org/10.1109/ACCESS.2018.2805837
28. X. Lin, J. Wu, S. Mumtaz, S. Garg, J. Li, M. Guizani, Blockchain-based on-demand computing resource trading in IoV-assisted smart city, *IEEE Trans.Emerg. Topics Comput.*, **9**(2020), 1373–1385. https://doi.org/10.1109/TETC.2020.2971831
29. E. David, R. Azoulay-Schwartz, S. Kraus, Bidding in sealed-bid and English multi-attribute auctions, *Decis. Support Syst.*, **42** (2007), 527–556. https://doi.org/10.1016/j.dss.2005.02.007
30. P. Wei, D. Wang, Y. Zhao, S. K. S. Tyagi, N. Kumar, Blockchain data-based cloud data integrity protection mechanism, *Future Generat. Comput. Syst.*, **102** (2020), 902–911. https://doi.org/0.1016/j.future.2019.09.028

31. Crawdad Citation CRAWDAD Dataset, 2015. Available from: http://crawdad.org/
32. S. Lu, X. Wei, Y. Li, L. Wang, Detecting anomaly in big data system logs using convolutional neural network, in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, (2018), 151–158. https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00037

AIMS Press