Tingxuan Lianga, Lingyi Chena, Mingfeng Huanga, Xiaoheng Denga,\*, Shaobo Zhangb, Neal N. Xiongc,\*, Anfeng Liua

*a School of Computer Science and Engineering, Central South University, Changsha, 410083, China*

*b School of Computer Science and Engineering of the Hunan University of Science and Technology, Xiangtan, 411201, China.*

*c Department of Computer Science and Mathematics, Sul Ross State University, Alpine, TX 79830, USA*

RLTD: A Reinforcement Learning-based Truth Data Discovery Scheme for Decision Support Systems under Sustainable Environments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A R T I C L E I N F O

*Article history:*

Received

Received in revised form

Accepted

Available online

*Keywords:*

Truth data discovery

Digital technology

Decision support systems

Reinforcement learning

Mobile Crowd Source

A B S T R A C T

The online world and associated information and communication technologies have generated digital networks by processing massive volumes of data and have a significant impact on the environmental sustainability. Mobile Crowd Source (MCS) is one of the digital technologies that can help humanity to better sense, understand and protect the environment by using vast amounts of data obtained to construct intelligent decision support systems (DSS). However, as the false data submitted by dishonest and malicious workers will cause the data-based DSSs to make wrong decisions and thus cause great harm, it is an urgent issue to propose an effective Truth Data Discovery (TDD) scheme for MCS. To tackle this issue, a Reinforcement Learning-based Truth Data Discovery (RLTD) scheme is proposed to obtain truth data in MCS at low cost in this paper. The main innovations of the RLTD scheme are as follows: (1) A novel trustworthiness-based TDD scheme is proposed to obtain truth data accurately at low cost, which can facilitate data-based DSSs in MCS. (2) Combined with Matrix Factorization (MF), we propose a worker recruitment method that only needs to recruit () workers for TDD in tasks, which reduces the data collection cost significantly than previous TDD schemes. (3) We propose a Reinforcement Learning-based Site Selection (RLSS) method that intelligently selects as few sites as possible for worker recruitment with guaranteed high data quality. Experimental results demonstrate that the RLTD scheme can improve the accuracy of data collection by 1.31%-21.02%, reduce the data collection cost by 81.39%-85.50% compared to the traditional TDD schemes, and identify workers with accuracy of 86.67%-94.58%.

———————

∗ Corresponding author.

*E-mail address: dxh@csu.edu.cn (X. Deng), xiongnaixue@gmail.com (N. Xiong).*

**1 Introduction**

According to the experts, humanity is currently in the "Digital Age" [1-3]. The online world and associated information and communication technologies have generated digital networks by processing massive volumes of data [4-6]. Moreover, the development of digital technologies and the integration of digital activities with physical world are gaining traction [7-13]. Mobile Crowd Source (MCS) is one of the digital technologies that can help humanity to better sense, understand and protect the environment by using vast amounts of data obtained to construct intelligent decision support systems (DSSs) [14-16], which are claimed to have a significant impact on the environmental sustainability [17-20]. Before the emergence of MCS, it was difficult to perform environmental monitoring tasks that distribute over millions of kilometers and span decades of time due to the huge cost and difficulty of implementation [21-23]. For example, in studies to understand the migratory habits of migratory birds and their dependence on the environment. Some migratory birds, such as the Arctic terns, migrate up to 40,000 kilometers during their annual migration between the Arctic and the Antarctic Circle. And it needs to take decades of observation to fully understand their migratory habits. Thus, it was difficult to obtain high-quality monitoring data at low cost over such a wide range and for such a long period of time. MCS, however, offers a practical solution to these large-scale environmental protection and monitoring problems. In MCS, after the platform publishes the environmental monitoring tasks, people or devices with sensing equipment can collect various environmental monitoring data such as images, sounds, and videos in a participatory way [24-25]. Then, the platform uses the collected data to construct DSSs and makes wise decisions for environmental sustainability. Due to the outstanding advantages such as wide spatial and temporal coverage, low cost and good scalability, MCS has been applied to various fields of environmental sustainability, such as VTrack [26], NoiseTube [27], CrowdPark [28] and Parknet [29].

However, the data submitted by workers with unknown reliability may greatly affect the decisions made by data-based DSSs [30]. In MCS, workers need to pay certain costs to move to designated sites for data collection, such as movement costs, sensing resource costs, time and communication costs, etc. Although the platform will pay them some rewards, there are still some dishonest or malicious workers submitting false data to profiteer illegal rewards [31-32]. And the false data submitted to the platform will cause the constructed data-based DSSs to make wrong decisions and cause significant loss of life and property [30, 33]. Worse yet, due to the high cost of acquiring Ground Truth Data (GTD), the platform cannot verify the authenticity of the submitted data, which is known as the Information Elicitation Without Verification (IEWV) [34] problem. Therefore, it is urgent for the MCS platform to identify and recruit trustworthy workers to obtain truth data and avoid false data for reliable services, so-called the Truth Data Discovery (TDD) problem, which is one of the key issues in MCS [30, 33].

In recent studies, many schemes have been proposed to address the TDD problem, which can be divided into two categories [33-47]. One category is the mathematical and statistical methods [33-35], such as Mean, Median and Majority Voting (MV). This type of method believes that most workers are trustworthy. So, it recruits workers to collect data for each task, and calculates the average, median or majority value of the collected data as the Estimated Truth Data (ETD) of each task. Although these methods are reasonable, they are vulnerable to attacks and failure. For example, if multiple malicious workers jointly launch an attack, the platform will get the false ETD easily. In addition, a fatal disadvantage of these methods is their high cost. In theory, only one worker needs to be recruited for a single data collection task, while these methods recruit workers, where the cost is times higher.

The other category is the trustworthiness-based methods [36-47]. This type of method uses trustworthiness as an indicator of the intrinsic characteristic of workers. And it believes that the data submitted by trustworthy workers has a higher probability of being truth data, while the data submitted by malicious workers has a higher probability of being false data. So, different from the previous category, the platform identifies the trustworthiness of workers first, and then recruits the identified trustworthy workers for data collection in order to obtain reliable ETD. However, these methods have the same challenge as TDD to ensure that the worker identification process is accurate and cost effective. Most of the trustworthiness-based methods assume that the trustworthiness of workers can be easily acquired, or obtained by traditional trustworthiness reasoning and evolution methods, while they are difficult to apply in MCS.

To address the TDD problem in MCS, a Reinforcement Learning-based Truth Data Discovery (RLTD) scheme is proposed to identify trustworthy workers and obtain truth data at low cost. The main innovations of the RLTD scheme that differ from the previous schemes are as follows:

(1) First, we propose a novel worker identification method that evaluates workers’ trustworthiness in two ways: (a) Using data submitted by known trustworthy workers as GTD to evaluate the trustworthiness of other workers. In trustworthy sites (defined in Section 3.1), the platform regards the data submitted by the known trustworthy worker with the highest trust score as GTD. If the accuracy of the data submitted by another worker is larger than the threshold, increases its trust score; otherwise, decreases its trust score. After few rounds of evaluation, workers with trust score larger than threshold will be grouped into the set of trustworthy workers, and workers with trust score smaller than threshold will be grouped into the set of malicious workers. (b) Using data recovered by Matrix Factorization (MF) algorithm as sub-GTD to evaluate the trustworthiness of undetermined workers. Numerous studies [48-51] have demonstrated that due to the spatiotemporal correlation between data, even though the data missing rate reaches 50%, the accuracy of the data recovered by data inference methods can still be over 90% with an error rate of less than 10%. So, we can use the accurately recovered data by MF as sub-GTD to evaluate the trustworthiness of workers like (a), which can speed up the worker identification process at low cost.

(2) Second, combined with MF algorithm, we propose a worker recruitment method which reduces the number of workers to be recruited and improves the data quality than previous TDD schemes. This paper proposes and demonstrates for the first time that it only needs to recruit () workers for TDD in tasks. Firstly, in theory, it only needs to recruit one trustworthy worker for a single data collection task to guarantee its authenticity, so workers are sufficient for tasks instead of as in previous schemes. Secondly, due to the spatiotemporal correlation between data, we only need to select a portion of trustworthy sites to recruit trustworthy workers for data collection while using MF algorithm to recover the data of unselected sites accurately. So, the number of recruited workers can be further reduced to (). Moreover, the MF algorithm can also improve the collected data quality. In practice, due to cost consideration, the identified trustworthy workers are often not sufficient to cover all sites [30]. So, there are always a part of sites that have no trustworthy worker to collect high-quality data. In this case, MF that using data collected from trustworthy sites to recover the data of other sites tends to yield better data quality than recruiting multiple workers per site.

(3) Finally, we propose a Reinforcement Learning-based Site Selection (RLSS) method to reduce the number of sites selected for the MF algorithm. In MF algorithm, using data collected from different site combinations will significantly influence the recovered data quality. So, we need to select the optimal site combination to achieve the highest data quality, which we call the site selection problem [30]. To address the site selection problem, we can first use the backtracking algorithm to get all the site combinations, and then iterate through them to find the optimal one. But the time complexity of backtracking is , which can’t be solved in polynomial time. Moreover, to calculate the recovered data accuracy of each site combination, we need to know the GTD of each site in advance, which is absolutely impossible in reality. Actually, the site selection process can be formulated as a finite Markov Decision Process (MDP), which selects the best possible sites sequentially and stops when it meets the sparsity requirement of MF. Among the solutions for MDP, reinforcement learning (RL) is a promising solution for sequential decision-making, especially the off-policy and value-based Deep Q Network (DQN) [61] and its variants [62]. Therefore, to solve the site selection problem efficiently, we propose a Reinforcement Learning-based Site Selection (RLSS) method, which defines the objective function as the quality of data recovered from different site combinations by MF. Specifically, RLSS first uses the historical data in previous cycles to train a site selection agent by learning the spatiotemporal correlation between them, and then uses the trained agent to guide the site selection process in current cycle. In this way, we can get the optimal site combination effectively in the absence of GTD, and then get the highest quality data through MF at low cost.

To summarize, the innovations of this paper are as follows:

(1) We propose a novel trustworthiness-based TDD scheme to obtain truth data accurately and quickly at low cost, which can facilitate data-based DSSs in MCS.

(2) Combined with MF algorithm, we propose a worker recruitment method that only needs to recruit () workers for TDD in tasks, which significantly reduces the data collection cost and improves the collected data quality compared to the previous TDD schemes.

(3) We propose a Reinforcement Learning-based Site Selection (RLSS) method to select the optimal site combination for MF-based data inference. RLSS uses the historical data to train a site selection agent and uses the trained agent to guide the site selection process in current cycle, which reduces the data collection cost with guaranteed data quality.

(4) Experimental results demonstrate that under the same network condition, compared with traditional TDD schemes, the RLTD scheme can improve the accuracy of collected data by 1.31%-21.02%, reduce the data collection cost by 81.39%-85.50%, and identify workers with accuracy of 86.67%-94.58%.

The rest of this paper is organized as follows: The related works are given in Section 2 and the network model and problem formulation are presented in Section 3. Then, the proposed RLTD scheme is introduced in Section 4. The performance analysis of RLTD is presented in Section 5. Finally, Section 6 provides the policy implication, conclusions and future works.

**2 Related works**

*2.1 Truth data discovery*

As one of the most important digital technologies, MCS uses the vast amounts of collected data to construct data-based DSSs, and make wise decisions for environmental sustainability. Therefore, the quality of the collected data will significantly affect the quality of the decisions made by DSSs. And the false data even will cause the DSSs to make wrong decisions, potentially resulting in severe loss of life and property [30]. So, it is essential to ensure that the MCS platform obtains high-quality data while avoiding false data, which is known as the TDD problem [33]. The TDD problem is a very common problem in MCS, while it has not been properly addressed yet. The main challenge of the TDD problem is the high cost of obtaining the GTD of tasks, which incurs the difficulty in verifying the authenticity of the collected data, so-called the IEWV problem [34]. To address this challenge, researchers have done extensive studies in this area.

A straightforward kind of TDD scheme is the redundancy-based scheme, which recruits multiple workers for each task and then aggregates the data submitted by different workers to infer the ETD of each task [33-35]. The most classic redundancy-based schemes are the Mean [33], Median [33] and Majority Voting (MV) [33] methods, which calculate the mean, median or majority value of the collected data as the ETD of each task. These methods assume that most workers in the network are trustworthy. Thus, most of the collected data is reliable and the inferred ETD is close to the GTD. However, due to the heterogeneous nature of the network, it is impractical to assume that most workers are trustworthy and regard them equally. So, researchers proposed another kind of redundancy-based scheme, called the weighted scheme [33], which believes that the quality of workers is different and assigns workers with different weights according to their quality. For example, Sun et al. [35] assume that the data submitted by workers follows a normal distribution. And workers who submit data closer to the normal distribution center are assigned with higher weights, while workers who submit data further away from the normal distribution center are assigned with lower weights. In this way, the ETD of each task is the weighted mean value of data submitted by different workers.

However, in the redundancy-based scheme, the GTD is actually unknown *a priori* and it is impossible to verify the reliability of the inferred ETD. Moreover, these methods are vulnerable to attacks and failure. For example, if multiple malicious workers jointly submit false data for the same task, the inferred ETD will easily deviate from the GTD. To address these problems, some researchers proposed a trustworthiness-based TDD scheme [36-47]. The trustworthiness-based scheme uses trustworthiness as an indicator of the intrinsic characteristic of workers. It believes that the data submitted by trustworthy workers has a higher probability of being truth data, while the data submitted by malicious workers has a higher probability of being false data. So, the platform first identifies the trustworthiness of workers, and then recruits the identified trustworthy workers for data collection in order to obtain reliable ETD. And many trustworthiness reasoning and evolution methods have been proposed to evaluate the workers’ trustworthiness. For example, in research [41], they first use the data collected by UAV as GTD to identify a portion of trustworthy workers, and then evaluate the trustworthiness of other workers through their interaction with identified trustworthy workers. However, in MCS, the cost of dispatching UAV to collect data from such a wide range is very high and workers do not interact with each other in general. But, compared to the redundancy-based scheme, the ETD obtained by the trustworthiness-based scheme turns to be more reliable.

*2.2 Data inference*

Although the quality of data submitted by trustworthy workers is high, the identified trustworthy workers are often not sufficient to cover all sites in MCS. Fortunately, numerous studies have demonstrated that data in MCS often exhibits spatiotemporal correlations. So, even if data is missing from some sites, the missing data can be accurately recovered by data inference methods [30]. In this way, we only need to collect data from the trustworthy sites, while using the data inference methods to recover the data of other sites accurately. In order to improve the accuracy of the recovered data, many researchers have proposed many different data inference methods.

Rana et al. [48] ignored the cell selection problem and used the Compressive Sensing (CS) algorithm for the first time to recover the missing data in the large-scale sensing area, so as to construct a global noise map. He et al. [49] further proposed a Bayesian Compressive Sensing algorithm, which further outputs the confidence intervals for each recovered data based on the inference results. Marchang et al. [50] proposed an improved KNN algorithm named KNN-ST (KNN-Spatial-Temporal), which can extract the spatiotemporal correlation between data and recover the missing data accurately. In recent studies, Xie et al. [51] found that combined with cell selection algorithm, Matrix Completion (MC) algorithm can further reduce the number of selected cells compared to CS algorithm, and proposed a bipartite graph-based MC algorithm for data inference. The experimental results demonstrated that the accuracy of the data recovered by this algorithm is better than that by CS at the same data sparsity. Moreover, with the development of recommendation system technologies in recent years [52-53], Matrix Completion algorithm based on Matrix Factorization has also been widely applied to data inference [54-59]. Therefore, in this paper, we are intended to use the Matrix Factorization (MF) algorithm as the data inference method to recover the data of unselected sites from the collected high-quality data, which reduces the data collection cost with guaranteed data quality. To the best of our knowledge, there is no research on applying data inference methods to TDD so far, and this paper is the first attempt to do so.

*2.3 Site selection*

In addition to the data inference method, another key issue in MCS is the cell selection (i.e., site selection) problem. In data inference, using data collected from different numbers of cells has a great impact on the accuracy of the recovered data. Besides, with a fixed number of selected cells, using data collected from different cell combinations has an impact, too. Therefore, the objective of cell selection is to select the optimal cells one by one, so as to achieve the highest data inference accuracy. To achieve this goal, He et al. [49] designed an incentive mechanism to lead workers to some valuable cells for data collection, where the data collected in current cycle has a great difference with that in previous cycles. Combined with CS algorithm, Liu et al. [60] proposed an active learning-based cell selection algorithm that iteratively selects the most valuable cells for data collection and used the Expectation Maximization (EM) algorithm to guide the cell selection process.

In recent studies, with the continuous development of reinforcement learning (RL) algorithm [61] and its variants [62] in sequential decision making, Liu et al. [63-64] proposed two reinforcement learning-based cell selection algorithms, which use the historical data to train a cell selection agent and use the

Table 1. A brief overview of the related works

|  |  |  |
| --- | --- | --- |
| Literature | Field | Main contributions |
| Zheng et al. [33] | Truth data discovery | Proposed the Mean, Median and Majority Voting (MV) TDD methods |
| Ye et al. [34] | Truth data discovery | Proposed a mean and median check (MMC) framework to iteratively check the distance between mean and median value and remove the error value |
| Sun et al. [35] | Truth data discovery | Proposed a weighted TDD scheme, which assigns workers with different weights and calculates the weighted mean value as the ETD |
| Guo et al. [41] | Truth data discovery | Proposed an active and verifiable trust evaluation method to evaluate workers’ trust by data collected by UAV and the interaction between workers |
| Rana et al. [48] | Data inference | Proposed to use Compressive Sensing (CS) algorithm for data inference in MCS for the first time |
| He et al. [49] | Data inference | Proposed a Bayesian Compressive Sensing algorithm, which returns not only the recovered data, but also the confidence intervals for incentive control |
| Marchang et al. [50] | Data inference | Proposed an improved KNN algorithm named KNN-ST for data inference |
| Xie et al. [51] | Data inference | Proposed a bipartite graph-based Matrix Completion algorithm for data inference |
| Wang et al. [58] | Data inference | Proposed a deep matrix factorization method, which retains the nonlinear spatiotemporal correlation, and performs high-precision data inference |
| He et al. [49] | Site selection | Proposed an incentive mechanism to lead workers to some valuable cells for data collection |
| Liu et al. [60] | Site selection | Proposed an active learning-based cell selection algorithm |
| Liu et al. [63] | Site selection | Proposed a reinforcement learning-based cell selection algorithm, which selects the least cells under the constraint of inference error |
| Liu et al. [64] | Site selection | Proposed a reinforcement learning-based cell selection algorithm, which selects the optimal cell combination under the constraint of cell number |
| Han et al. [65] | Site selection | Proposed an update method to update the reinforcement learning trained cell selection agent to adapt to the time-dependent environment |

trained agent to guide the future cell selection process. Based on the different optimization objectives, they can be classified into two types: (1) minimizing the number of selected cells under the constraint of the upper inference error limit [63]; and (2) minimizing the data inference error under the constraint of a fixed number of selected cells [64]. However, Han et al. [65] believed that the cell selection model trained by previously proposed reinforcement learning-based methods is fixed, which can’t be applied in the time-dependent environment. So, they proposed an update method to continuously update the cell selection model, so as to adapt to the time-dependent environment. Inspired by these cell selection algorithms, this paper designs a Reinforcement Learning-based Site Selection (RLSS) algorithm to further reduce the number of trustworthy sites selected for data collection, which further reduces the data collection cost.

In summary, the brief overview of the related workers in truth data discovery, data inference, and site selection is shown in Table 1.

**3 System model and problem formulation**

*3.1 System model*

In this paper, we work on a MCS network that continuously monitors the environment, as shown in Fig. 1, which consists of three main components: (1) Sites (Tasks); (2) Workers; and (3) MCS platform.

(1) Sites (Tasks). To monitor the environment effectively, we split the large-scale sensing area into sites, and the task to monitor the whole environment is split into pieces correspondingly. Each site represents a subarea and workers in the subarea can submit data for the corresponding task. We use to denote the set of sites, where represents the -th site. Moreover, for the convenience of later discussion, we refer to sites with known trustworthy workers as trustworthy sites and use to represent the set of them. Correspondingly, we refer to sites without known trustworthy workers as undetermined sites and use to represent the set of them. Therefore, the relationships between , and are and . In addition, we split each task into cycles of equal length, and use to denote them, where represents the -th cycle.

(2) Workers. Workers refer to people or devices that can collect data from specific sites. After the platform publishes the data collection tasks, workers can apply for them, and then the platform will recruit workers who meet the requirements to perform the tasks. Suppose that in cycle , the number of workers recruited from site is , then the set of recruited workers is denoted as , where represents the -th worker recruited from site in cycle . Moreover, according to the worker identification process, we classify workers into three categories (see Section 4.2). And we denote the set of known trustworthy workers as , the set of undetermined workers as and the set of malicious workers as .

(3) MCS platform. The MCS platform receives the request from workers, and recruits workers who meet the requirements to collect data. Then it uses the collected data to infer the ETD of each task, and uses the inferred ETD to construct data-based DSSs and various applications for users.

*3.2 Problem formulation*

**Definition 1**. (**Ground Truth Matrix, GTM**). The GTD is the fractal truth of tasks in each site. So, we denote the ground truth matrix as , as shown in Formula (1), where the element represents the GTD of site in cycle .

**Definition 2**. (**Site Selection Matrix, SSM**). In this paper, combined with MF algorithm, we only need to collect data from a portion of trustworthy sites, while recovering the data of the other sites accurately. The site selection matrix, denoted as a binary matrix , is used to record which sites are selected in each cycle. means that the site is selected in cycle . Otherwise, .

**Definition 3**. (**Partial Estimated Truth Matrix, PETM**) The ETD is the estimated trustworthy data of each task obtained by TDD schemes. The partial estimated truth matrix, denoted as , is used to



Fig. 1. System model

record the ETD of the selected trustworthy sites in each cycle, whose site selection value . If we use the function to indicate the TDD scheme, and use to indicate the set of data submitted by workers in set , then the value of element is shown in Formula (2).

**Definition 4**. (**Estimated Truth Matrix, ETM**) In PETM, some sites are not selected to recruit workers for data collection, so the ETD of them is . However, we can use data inference method to recover the missing ETD of them, and call the recovered matrix as the estimated truth matrix, denoted as , which is shown in Formula (3). The element represents the ETD of site in cycle .

The relationship between PETM and ETM is shown in Formula (4), where represents the data inference method.

(1) Maximize the collected data quality

In this paper, the platform uses the inferred ETD of each site to construct the data-based DSSs. According to the above definitions, the ETD of site in cycle is , and the corresponding GTD is . Then the quality of the data collected from site in cycle is calculated as Formula (5), .



Fig. 2. Data quality is related to the deviation of the collected data

The image of is shown in Fig. 2. Its physical meaning is that the closer the ETD is to the GTD , the higher the data quality , which will reach its maximum value (i.e., ) when is equal to . On the contrary, if deviates far away from , will decrease rapidly. Then, the average data quality of all the sites in all cycles is calculated as Formula (7).

Obviously, the higher the average data quality, the closer the ETD is to the GTD, and then the better the performance of the TDD scheme. Therefore, our objective is to maximize the average quality of the collected data: .

(2) Minimize the data collection cost

The data collection cost includes the cost of worker recruitment, data transmission and processing, and so on. In this paper, we mainly consider the cost of worker recruitment. And we use to denote the cost function, which is monotonically increasing with respect to the number of recruited workers. Simply put, we assume that the cost of recruiting different workers is equal. And in Section 3.1, we denote the set of workers recruited from sites in cycle as . Then the data collection cost of all the sites in all cycles is calculated as Formula (8).

In Formula (8), indicates the operation to calculate the number of elements in a set. So, denotes the number of workers recruited from site in cycle , i.e., . As we all know, the lower the data collection cost, the better the system performance. Therefore, our objective is to minimize the comprehensive data collection cost: .

(3) Maximize the F1 score of worker identification

In this paper, we propose a novel trustworthiness-based TDD scheme to identify workers by their trust score and recruit the identified trustworthy workers to collect high-quality data. The worker identification process can essentially be viewed as a binary classification process. And we use a confusion matrix shown in Table 2 to record the identification result. In Table 2, indicates the number of correctly identified trustworthy workers; indicates the number of misidentified trustworthy workers; indicates the number of misidentified malicious workers; indicates the number of correctly identified malicious workers.

In practical, the worker identification results cannot be completely accurate, and some workers may be misidentified due to various uncontrollable reasons. To evaluate the identification result, we use the precision rate to represent the proportion of the identified trustworthy workers who are truly trustworthy, and use the recall rate to represent the proportion of the truly trustworthy workers who are correctly identified, which are calculated as Formula (9) and (10).

Table 2. Confusion matrix of worker identification result

|  |  |  |
| --- | --- | --- |
| **Identified**  **Reality** | **Trustworthy** | **Malicious** |
| **Trustworthy** |  |  |
| **Malicious** |  |  |

Of course, we hope both the precision rate and the recall rate to be as large as possible. However, they are contradictory in some cases. Thus, to balance the precision rate and recall rate, we use the F1 score to evaluate the worker identification result, which is the harmonic mean value of and , calculated as Formula (11). Then, our objective is to maximize the F1 score of the worker identification: .

To summarize, the optimization objectives of this paper are as follows:

The notations of main variables used in this paper are shown in Table 3.

Table 3. Notation table

|  |  |
| --- | --- |
| **Notation** | **Meaning** |
|  | The -th data collection cycle |
|  | The set of cycles |
|  | The -th site |
|  | The set of sites |
|  | The set of trustworthy sites |
|  | The set of undetermined sites |
|  | The number of workers recruited from site in cycle |
|  | The -th worker recruited from site in cycle |
|  | The set of workers recruited from site in cycle |
|  | The set of known trustworthy workers |
|  | The set of undetermined workers |
|  | The set of malicious workers |
|  | The data submitted by worker |
|  | The set of data submitted by workers in set |
|  | The ground truth matrix |
|  | The site selection matrix |
|  | The partial estimated truth matrix |
|  | The estimated truth matrix |
|  | The TDD scheme |
|  | The data inference method |
|  | The collected data quality of site in cycle |
|  | The average quality of data collected from all sites in all cycles |
|  | The cost function of worker recruitment |
|  | The comprehensive data collection cost |
|  | The number of correctly identified trustworthy workers |
|  | The number of misidentified trustworthy workers |
|  | The number of misidentified malicious workers |
|  | The number of correctly identified malicious workers |
|  | The precision rate |
|  | The recall rate |
| F1 | The harmonic mean value of and |

**4 The Design of RLTD scheme**

*4.1 Overall structure of RLTD scheme*

The main idea of RLTD scheme is illustrated in Fig. 3, which includes three main processes: (1) worker identification process; (2) worker recruitment process; and (3) site selection process. These three processes interact with each other and sometimes cannot be clearly separated. The details are as follows.

(1) First, the worker identification process. The worker identification process in this paper can be divided into two stages: (a) The first stage is the expansion stage, which aims to identify as many trustworthy workers in the network as possible (see from Fig. 3 (a)-(c)). In the initial stage, there are few known trustworthy workers in the network, and the platform cannot recruit known trustworthy workers from sufficient trustworthy sites to collect high-quality data for data inference. Then, the platform will recruit many undetermined workers from trustworthy sites and use the data submitted by known trustworthy workers as GTD to update the trust score of them, and then identify their trustworthiness (see from Fig. 3(a)-(b)). When the number of trustworthy sites meet the sparsity requirement of data inference, the platform will further use the data recovered by data inference as sub-GTD to identify the trustworthiness of undetermined workers in corresponding sites (see from Fig. 3 (c)). Thus, this stage costs more to the platform, while it lays the foundation for the high-quality data collection in future cycles. (b) The second stage is the stabilization stage, which can be seen from Fig. 3(d). In the stabilization stage, there are abundant known trustworthy workers and sufficient trustworthy sites in the network for data inference to recover the data of undetermined sites accurately. So, the platform only needs to recruit known trustworthy workers to collect high-quality data in the trustworthy sites, while obtaining the data of other sites by data inference method. However, due to the dynamic nature of the network, workers are moving through different sites. Therefore, it is necessary to recruit a certain number of undetermined workers from trustworthy sites for worker identification, so as to maintain the number of trustworthy sites in the network at a certain high level.

(2) Second, the worker recruitment process. The worker recruitment process is a challenging process combined with worker identification and data inference. And we divide the process into the following four cases according to the number of known trustworthy workers and trustworthy sites in the network.

(a) The case of insufficient known trustworthy workers (see from Fig. 3(a)-(b)). As we illustrated above, the objective in this case is to identify as many trustworthy workers as possible. Therefore, the worker recruitment strategy in this case is to recruit one trustworthy worker with the highest trust score and undetermined workers in the decreasing order of trust score in the trustworthy sites; and recruit undetermined workers in the decreasing order of trust score in the undetermined sites. Note that if the number of undetermined workers in a site is less than or , recruit all of them. In this way, the ETD of the trustworthy sites is the data submitted by the recruited trustworthy worker and the ETD of the undetermined sites is the weighted mean value of data submitted by the recruited undetermined workers. Note that the obtained ETD in undetermined sites is less trustworthy, so we only perform worker identification in trustworthy sites in this case.

(b) The case of basically sufficient known trustworthy workers (see from Fig. 3(c)). Although this case is still in the expansion stage of worker identification, while there are sufficient trustworthy sites for data inference to recover the data of other sites accurately. So, the worker recruitment strategy in this case is the same as in case (a), while the ETD is more reliable. In this case, the ETD of the trustworthy sites is the data submitted by the recruited trustworthy workers, and the ETD of the undetermined sites is the data recovered by data inference. And due to the high quality of data recovered by data inference, the platform can use it and the data collected from trustworthy workers for worker identification jointly.



Fig. 3. Main processes of RLTD scheme. (a) The case of insufficient known trustworthy workers; (b) The number of known trustworthy workers and trustworthy sites increases; (c) The case of basically sufficient known trustworthy workers; (d) The case of sufficient known trustworthy workers; (e) The case of abundant known trustworthy workers and recruit trustworthy workers from all trustworthy sites; (f) Recruit trustworthy workers from part of the trustworthy sites.

(c) The case of sufficient known trustworthy workers (see from Fig. 3(d)). In this case, there are more known trustworthy workers than case (b). Although we can obtain the ETD of undetermined sites through data inference, it is difficult to maintain the number of trustworthy sites at the same level due to the worker mobility. So, the worker recruitment strategy in this case is to recruit one known trustworthy worker with the highest trust score and a small number of undetermined workers in the decreasing order of trust score in trustworthy sites; and recruit no worker in undetermined sites. In this way, the ETD in this case is the same as in case (b), while the platform only needs to identify the trustworthiness of undetermined workers recruited from part of the trustworthy sites.

(d) The case of abundant known trustworthy workers (see from Fig. 3(e)-(f)). In this case, there are abundant trustworthy sites, and only a few undetermined sites in the network. Actually, due to the spatiotemporal correlation between data, we only need to select part of the trustworthy sites to collect high-quality data, while recovering the data of the unselected trustworthy and undetermined sites by data inference accurately. Therefore, there are two worker recruitment strategies in this case. One is to recruit one trustworthy worker with the highest trust score from every trustworthy site (see from Fig. 3(e)). The other is to use the RLSS algorithm to select part of the trustworthy sites first, and then recruit one trustworthy worker with the highest trust score from each selected site (see from Fig. 3(f)). Then, in this case, the ETD of the selected sites is the data submitted by the recruited trustworthy workers, and the ETD of the unselected sites is the data recovered by data inference.

(3) Finally, the site selection process. Due to the spatiotemporal correlation between data, although there are abundant trustworthy sites, the platform only needs to select part of them to collect high-quality data, while recovering the data of unselected sites by data inference accurately, so as to save cost. However, collecting data from different trustworthy site combinations will result in different data inference accuracy. Therefore, we propose a Reinforcement Learning-based Site Selection (RLSS) method to get the optimal trustworthy site combination for data inference under the constraint of the number of selected sites, which reduces the data collection cost with guaranteed data quality. As shown in Fig. 3(f), there are 23 trustworthy sites in the network, but we only need to select 14 of them to collect high-quality data under the guidance of RLSS algorithm. As for the unselected trustworthy and undetermined sites, we can use data inference method to recover the data of them accurately. In this way, the number of workers to be recruited is further reduced, which further reduces the data collection cost.

In summary, based on the four cases of worker recruitment, we illustrate the process of the RLTD scheme with the pseudocode shown in Algorithm 1.

|  |
| --- |
| **Algorithm 1:** RLTD Algorithm |
| **Initialization**: , , , , , , , ,, , , ,, , , , ,  **Case 1**: The case of insufficient known trustworthy workers  1: **For** each site in  2:  **If**  3: add worker with the highest trust score into  4: calculate the number of undetermined workers  5: **If**  6: add undetermined workers in the decreasing order of trust score into  7: **Else**  8: add undetermined workers into  9: **End If**  10: receive the data submitted by workers in  11:  12: , , ***WorkerIdentification***(, , , , )  13: **Else**  14: add undetermined workers in the decreasing order of trust score into  15: receive the data submitted by workers in  16:  17:  18: **End If**  19: **End For**  **Case 2**: The case of basically sufficient known trustworthy workers  1: **For** each site in  2:  **If**  3: add worker with the highest trust score into  4: calculate the number of undetermined workers  5: **If**  6: add undetermined workers in the decreasing order of trust score into  7: **Else**  8: add undetermined workers into  9: **End If**  10: receive the data submitted by workers in  11:  12: **Else**  13: add undetermined workers in the decreasing order of trust score into  14: receive the data submitted by workers in  15:  16: **End If**  17: **End For**  18: ***MatrixFactorization***(, )  19: **For** each site in  20: , , ***WorkerIdentification***(, , , , )  21: **End For**  **Case 3**: The case of sufficient known trustworthy workers  1: **For** each site in  2:  **If**  3: add worker with the highest trust score into  4: add undetermined workers in the decreasing order of trust into  5: receive the data submitted by workers in  6:  7: , , ***WorkerIdentification***(, , , , )  8: **Else**  9:  10: **End If**  11: **End For**  12: ***MatrixFactorization***(, )  **Case 4**: The case of abundant known trustworthy workers  1: initialize as Formula (22)  2: = ***RLSS\_Train***(, , , )  3: = ***RLSS\_Test***(,, , )  4: **For** each site in  5: **If**  6: add worker with the highest trust into  7:  8: **Else**  9:  10: **End If**  11: **End For**  12: ***MatrixFactorization***(, )  13:  14: append to  15: calculate and as Formula (30) and (31)  16: update as Formula (32) |

Next, we use an example to show the RLTD process. Initially, as shown in Fig. 3(a), there are only a small number of known trustworthy workers (colored in green) exist in tree trustworthy sites (colored in gray) , and (i.e., sites , and ) in the network. Then, according to the workers recruitment strategy, we denote the sets of workers recruited from sites , and in the first cycle as , and . And we assume that the first worker in above sets is the trustworthy worker. Note that the number of undetermined workers in site is less than , so the platform recruits all of them (i.e., ). In addition, we denote the set of workers recruited from each undetermined site as without loss of generality. Then, the ETD of the trustworthy sites is , and , while the ETD of other sites is the weighted mean value of the collected data. And then, the platform uses , and as GTD to identify the trustworthiness of undetermined workers in sites , and respectively. In this way, after several cycles, the number of known trustworthy workers will increase, as shown in Fig. 3(b). At this time, the platform can select five trustworthy sites , , , and (i.e., sites , , and ) to collect high-quality data, and the number of undetermined workers can be identified increases to .

Performing above worker recruitment and identification process repeatedly, the number of known trustworthy workers and trustworthy sites will further increase. When the number of trustworthy sites meets the sparsity requirement of data inference, which is the case of basically sufficient known trustworthy workers, the platform then can use the data inference method to recover the data of undetermined sites accurately. As shown in Fig. 3(c), the platform uses the data collected by the trustworthy workers recruited from 13 trustworthy sites (i.e., ) to recover the data of other 12 undetermined sites. Furthermore, the recovered high-quality data can act as sub-GTD to identify the trustworthiness of undetermined workers in corresponding sites. Then, theoretically, undetermined workers can be identified, which accelerates the worker identification process.

After few more cycles, more trustworthy workers are identified. As shown in Fig. 3(d), the platform first recruits one trustworthy worker with the highest trust score in each trustworthy site, and then further recruits a small number of undetermined workers from part of these sites (i.e., , , , , , and ) for worker identification, so as to maintain the number of trustworthy sites. And for the undetermined sites, the platform recruits no worker. In this way, the number of recruited workers is reduced.

Finally, when the number of known trustworthy workers and trustworthy sites increases to a quite high level, the RLSS is applied to the trustworthy site selection process. As shown in Fig. 3(f), there are 23 trustworthy sites, while the platform only selects 14 of them to collect high-quality data. And the data of unselected trustworthy and undetermined sites is recovered by the data inference method.

*4.2 Worker identification*

In this section, we will illustrate the proposed worker identification method, which updates the trust scores of workers based on the data submitted by known trustworthy workers or the data recovered by data inference and identifies workers’ trustworthiness according to their updated trust scores.

For each worker, we use a trust score to evaluate its trustworthiness. And there are two ways to update the trust score in this paper: (1) using data submitted by known trustworthy workers as GTD to update it; (2) using data recovered by data inference as sub-GTD to update it. Regardless of which way is used, the trust score is essentially updated based on the ETD of corresponding sites.

Here, as shown in Formula (13), we use to represent the accuracy of the data submitted by worker , which indicates the proximity between and the corresponding ETD . Thus, the greater , the more proximate between and , which means that is more reliable. In this way, we define a threshold to determine the update strategy of the trust score. If is larger than or equal to the threshold , which means that worker submits reliable data in cycle , and its trust score should be increased. On the contrary, if is smaller than the threshold , which means that worker submits false data in cycle , and its trust score should be decreased.

|  |
| --- |
| **Algorithm 2:** Worker Identification Algorithm |
| **Input**: , , , ,  **Output**: , ,  **Initial**: , , , , ,  1: **For** in  2: calculate of as Formula (13)  3: **If** :  4:  5: **Else:**  6:  7: **End If**  8: **If** :  9: add into  10: **Else if** :  11: add into  12: **Else:**  13: add into  14: **End If**  15: **End For**  16: **Return** , , |

Then, the update strategy is shown in Formula (14). In Formula (14), represents the updated trust score of , and represents its historical trust score. is the maximum value of trust score, which is generally set to 1, and it is used to avoid the trust score getting too large. If the trust score of trustworthy workers is too large, it will take a long time to decrease their trust score to when some trustworthy workers become malicious. On the contrary, is the minimum value of trust score, which is generally set to 0, and it is used to avoid the trust score getting too small. What’s more, is a decaying factor that we define to undermine the influence of the historical trust score, and its value is 0~1.

After the trust scores of the recruited workers are updated, the workers are identified according to their updated trust scores. Specifically, if the trust score of a worker is greater than or equal to the threshold , it is considered to be trustworthy and will be grouped into the set of trustworthy workers. If the trust score of a worker is smaller than the threshold , it is considered to be malicious and will be grouped into the set of malicious workers. In addition, workers with trust score between and are labeled as undetermined workers and are grouped into the set of undetermined workers. The trustworthy workers are considered to submit data honestly, while the data submitted by malicious workers is considered to be of low quality. Thus, if the platform already knows that a worker is malicious, it will not recruit it to collect data. In addition, the undetermined workers are essentially either trustworthy or malicious, however, they cannot be identified due to insufficient evaluation.

The worker identification process is shown in Algorithm 2. In the algorithm, , and are the sets of workers of different trustworthiness that we defined in Section 3.1.

*4.3 Data inference*

As we illustrated in Section 4.1, when the number of trustworthy sites meets the sparsity requirement of data inference, we can use the data collected from all the trustworthy sites to recover the data of undetermined sites by data inference. Moreover, when the number of trustworthy sites is quite large, we only need to select part of them for data collection, while recovering the data of unselected sites. In this section, we propose a Matrix Factorization (MF) algorithm for data inference, which uses the historical data and the data collected from a part of sites in current cycle to recover the data of other sites accurately.

The basis of the MF algorithm is the latent factor model, which tries to explain the spatiotemporal correlation between data by characterizing both sites and cycles on several latent factors inferred from the historical data distribution pattern. Here, we adopt a matrix representation to represent the site-cycle data pair. As shown in Fig. 4, the row of the matrix represents the data of the same site in different cycles, while the column of the matrix represents the data of different sites in the same cycle. For the missing data in current cycle, we can use the historical data and the collected data in current cycle to recover them. However, as increasingly more data is collected as the number of cycles increases, the size of the matrix that needs to be recovered is also increasingly getting larger, and it will take longer for the MF to recover the missing data. To save the computation time, we consider the matrix consists of the data of sites in recent cycles instead of all of the historical cycles as the input of MF algorithm.

To recover the missing data in the incomplete matrix , the MF algorithm is to take full rank decomposition of the recovered matrix by using the advantage that the matrix has the low rank feature. According to the latent factor model, we use two latent factor matrices and to represent the spatial correlation between different sites and the temporal correlation between different cycles respectively, where and . Then we can use the product of and shown in Formula (15) to approximate the recovered matrix .

Accordingly, data in each site is associated with a vector , and data in each cycle is associated with a vector . Then, the resulting dot product, , approximates the missing data of site in cycle , which is shown in Formula (16).

Formula (16) tries to capture the comprehensive data distribution pattern between sites and cycles. However, much of the observed variation in data distribution is due to the effects associated with sites or cycles only, known as biases or intercepts. Taking the Beijing air pollution data [67-68] for example, the PM2.5 concentration in the early morning is always lower than that during rush hour, and the PM2.5 concentration in suburban is always lower than that in urban city. Thus, it would be unwise to approximate



Fig. 4. The process of Matrix Factorization algorithm

the missing data by only. Instead, we try to identify the portion of these values that individual site or cycle biases can explain. And we denote the bias involved in by , which is calculated as Formula (17).

Here, represents the average value of , and indicate the observed deviations of site and cycle from the average values respectively. Taking these biases into account, the recovered data is calculated as Formula (18).

Next, we apply stochastic gradient descent algorithm to learn the parameters , , and . The objective of gradient descent is to minimize the regularized squared error on the known data between origin matrix and recovered matrix , which is shown in Formula (20).

Here, we introduce four regularization penalties , , and to avoid overfitting during the learning process. is the operation to calculate the sum of the squares of elements in a vector. And the constant controls the extent of regularization. Note that we only compute the prediction error on the known data, i.e., .

The process of MF algorithm is shown in Algorithm 3. Assume that the current is . It uses the PETM of current cycle and the ETM of recent cycles to construct the incomplete matrix to be recovered. Then, it iterates for times. In each iteration, it uses *,* , , , to calculate the approximate data as Formula (18) and the associated prediction error as Formula (19). Then it updates the parameters , , and by a magnitude proportional to in the opposite direction of the gradient as Formula (21). Here, is called the learning rate, which is applied to adjust the magnitude of the gradient drop at each iteration.

|  |
| --- |
| **Algorithm 3:** Matrix Factorization Algorithm |
| **Input**: ,  **Output**:  **Initialize**: , , , , , , , , , , ,  1:  2:  3: **While**  4: **While**  5: **While**  6:  7:  8: Update parameters as Formula (21)  9:  10: **End While**  11:  12: **End While**  13:  14: **End While**  15:  16:  17: **Return** |

*4.4 Reinforcement Learning-based Site Selection*

When the number of the trustworthy sites is quite large, due to the strong spatiotemporal correlation between the data, we only need to select a part of trustworthy sites from set instead of all of them to collect data, and use the data inference method to recover the data of unselected trustworthy sites and undetermined sites together. In this way, as the number of selected sites decreases, the data to be collected (i.e., data collection cost) will be reduced, which improves the system performance.

However, choosing which sites from is a critical issue that will impact the performance of data inference, which we call the site selection problem. The objective of site selection is to select the optimal site combination to collect data for data inference, so as to achieve the highest data inference accuracy. To tackle this issue, we propose a Reinforcement Learning-based Site Selection (RLSS) algorithm. RLSS uses the historical data in previous cycles to train a site selection agent first, and then uses the trained agent to guide the site selection process in current cycle in order to select the optimal site combination. In this way, the platform can get the data of the highest quality by using the data collected from the selected sites to recover the data of unselected sites by data inference method. Specifically, RLSS models the state, action and reward of the site selection problem, and then uses a Deep Q-Network (DQN) to approximate the Q-function, which outputs the optimal action that achieves the highest reward under certain states. Then it uses the Double DQN method to train the DQN agent. Finally, it uses the trained agent to guide the site selection in current cycle. The detailed illustration is as follows.

### 4.4.1 Modeling state, action and reward

First, we model the state, action and reward in reinforcement learning.

(1) *State*. The state represents the site selection condition, that is which sites are selected in a data collection cycle. We use a binary vector to describe the state , where means that site has been selected and means that site has not been selected.

(2) *Action*. The action represents the site to select. In our system model, we select trustworthy sites from one by one. This is because that selecting multiple sites at the same time will result in a large action space, which will take longer training time to cover the whole action space. So, we get choices when we select the next site. In practice, however, each site is selected at most once in a cycle and the set is different in different cycles. To make the action space consistent under different states, we denote the action space as , while we will use the invalid action masking [71] technique to mask the action which selects undetermined sites and the selected trustworthy sites during the training process.

(3) *Reward*. The reward represents the benefit of an action under a certain state . In each cycle,



Fig. 5. The relationship between state, action, reward and DQN

the platform selects sites one by one until the number of selected sites reaches the site selection limit . denotes the operation of calculating the sum of elements in the set , which is equivalent to the number of selected sites for its binary nature. Then, the objective of site selection is to find an optimal subset of size from to achieve the highest data inference accuracy, which should be reflected in the reward model.

Here, we introduce a training data matrix for site selection agent training. As the selection process proceeds, when the number of selected sites reaches , we use the historical ETM matrix and the site selection vector to construct the training data matrix as Formula (22). In Formula (22), represents the current cycle . Thus, consists of the historical data in recent cycles and the collected data in current cycle. Then, we use the MF algorithm to recover the missing data in .

Denoting the recovered matrix as , the inference accuracy can be calculated as Formula (23).

Then, the reward model is shown in Formula (24). When the number of the selected sites reaches , give the action a positive reward . Otherwise, give the action an opposite constant reward , because select sites to collect data will take cost.

### 4.4.2 Training and Test process of RLSS

In addition to the above modeling of state, action and reward, we also need to learn a Q-function. In reinforcement learning, Q-function outputs the reward of every possible action under a certain state, and can be represented as Q-table or Q-network depending on the scale of state space. For the site selection problem in the practical MCS, there are usually a large number of sites, and the state space is extremely large. For example, if there are 20 sites in the network and the site selection limit , then the size of state space is , which is intractable in practice. Therefore, to tackle the dimension curse problem, we apply a Deep Q-Network (DQN) consisting of two fully connected layers to represent the Q-function in this paper. The relationship between state, action, reward and DQN is shown in Fig. 5. At each step, the site selection condition represents a certain state, then DQN calculates the rewards of every possible action under the state, and selects the action with the highest reward. According to the selected action, a site is selected and then the system transfers to a new state, and the corresponding reward is incurred. If the number of the selected sites reaches , the MF algorithm will be applied to recover the data of unselected sites, and the reward is 100 times the inference accuracy. Otherwise, the reward is the opposite constant reward .



Fig. 6. The training process of RLSS

The training process of the RLSS algorithm is shown in Algorithm 4 and Fig. 6. In reinforcement learning, the Q-function is used to represent the reward of action under certain state . And its optimal value is calculated based on the Bellman Equation, which is shown in Formula (25). In Formula (25), represents the expectation and denotes the reward decay coefficient, which is used to determine the relative importance of future reward compared with immediate reward.

According to the Bellman Equation, the optimal policy is learned based on the agent-environment interactions by using Q-learning algorithms. In this paper, we use DQN to approximate the Q-function and use deep Q-learning algorithm to update the parameters of DQN iteratively. Moreover, in order to eliminate the strong correlation between continues observations, we use the experience replay technique for model training [61]. Specifically, in each step, the experience is generated by the -greedy policy, and is stored in the experience replay pool . Then, DQN randomly samples several experiences from to learn the parameters and update the network. In this way, old experiences and new experiences are mixed and used together, which eliminates the strong correlation between continues observations and makes the state transform more independent.

We apply stochastic gradient algorithm to learn the parameters of DQN and the loss function is shown in Formula (26).

Then, the gradient of the loss function is shown Formula (27).

In addition, the target in Formula (25) is nonstationary and Q-function may changes rapidly during the training process, which is known as the oscillation problem

|  |
| --- |
| **Algorithm 4:** Training process of RLSS (RLSS\_Train) |
| **Input**: , , ,  **Output**:  **Initialize**: , , , , , , randomly initialize the parameters of DQN, initialize target Q-network by  1: **While**  2: Calculate by DQN as Formula (25)  3: Select an action with -greedy policy  4:  5: **If** :  6:  7:  ***MatrixFactorization***(, )  8: calculate as Formula (23)  9:  10:  11: **Else**  12:  13:  14: **End If**  15:  16: randomly select some experiences for training  17: Update of DQN as Formula (32)  18:  19: **If**  20:  21: **End If**  22: **End While**  23: **Return** |

[61]. To overcome the oscillation problem, we use Double DQN technique to train DQN, that is, we use another Q-network called target Q-network, whose parameters are cloned from the original Q-network periodically, to slow down the change of , and thus improves the stability. Specifically, the target Q-network and original Q-network (i.e., DQN) are initialized to be the same (i.e., ) in the beginning. In each step, RLSS uses the DQN to select the action with the highest reward under state . Then, it performs the selected action and uses the target Q-network to calculate the reward of every possible action under the new state . Finally, RLSS uses output by target Q-network to train DQN. Thus, the loss function of Double DQN training is as follows.

The gradient of new loss function is shown in Formula (29).

Note that RLSS uses DQN to select the action with the highest reward under certain state , while uses the reward calculated by target Q-network to train the DQN in each step. And the target Q-network is updated periodically by cloning the parameters of DQN every steps. In this way, the update of target Q-network are lagged behind DQN, which slows down the change of Q-function , and thus improves the stability.

After the training process, we can use the trained DQN to guide the site selection process in the next cycle, which is the reinforcement learning test process. The test process of RLSS is shown in Algorithm 5. The test process is relatively simple, which inputs the state into the trained DQN, and selects the sites under the guidance of the trained DQN, and finally outputs the optimal trustworthy site combination.

|  |
| --- |
| **Algorithm 5:** Test process of RLSS (RLSS\_Test) |
| **Input**: , ,,  **Output**: site selection vector  **Initialize**: ,  1: **While**  2: select action  3:  4: **End While**  5: **Return** |

### 4.4.3 Update training data

Using the Double DQN technique, RLSS can use the historical data to train a site selection model effectively and use the trained model to guide the site selection process in current cycle. However, due to the complexity and variability of the MCS environment, the site selection model trained with fixed training data does not always work well. Therefore, we should update the training data continuously in order to keep the site selection model up-to-date to adapt the time-dependent environment [65].

We comprehensively consider the gradual changes of data in time and space and use Zooming technique to evolve the data distribution patterns in different sites and cycles [65]. Zooming technique is a classic technique used in image processing and information spaces, which can extract the data distribution patterns but not the values. After the site selection and data collection in a cycle, we drop the oldest column in and add the newly collected data to . Then, we use the newly collected data and the site selection vector in current cycle to update . Specifically, we first zoom the training data based on the average value of the training data in last cycle, and the average value  of the newly collected data in current cycle, which are shown in Formula (30) and (31).

Then, we update the training data as Formula (32) [65]. In this way, the training data is updated every cycle, so that the site selection model can be updated to adapt to the time-dependent environment.

**5 Performance Analysis**

In this section, we will introduce the datasets used in this paper. Then, simulation experiments are conducted, and the performances of the proposed RLTD scheme are presented and analyzed.

*5.1 Experimental setup*

First, we adopt two real-life datasets, Sensor-Scope [66] and U-Air [67-68], to evaluate the performance of the MF and RLSS algorithm in data accuracy. These two datasets contain various types of data collection tasks, including humidity, temperature, air quality measurements, and so on. The detailed information is as follows.

*Sensor-Scope* [66]: The Sensor-Scope dataset is comprised of the humidity and temperature data collected from the EPFL campus within a sensing area about 500m×300m every 30 minutes. This target area is equally divided into 100 sites with the size of 50m×30m. Since there are only 57 out of these 100 sites are deployed with valid sensors, and the distribution of most sensors is too sparse, we just use the collected humidity data from 12 centrally distributed sites from 2007-01-01 to 2007-01-07 (336 cycles in total) for experimental evaluation.

*U-Air* [67-68]: The U-Air dataset is comprised of the air quality measurements (CO, NO2, O3, SO2 and PM2.5) data collected from 36 stations in Beijing every 1 hour. Similar to the Sensor-Scope dataset, we only use the collected data at 13 centrally distributed stations from 2015-2-19 to 2019-2-25 (168 cycles in total) for experimental evaluation. However, note that there are missing data in these 13 stations, and for the reliability and validity of the experiment, we filter out the sites with missing values, instead of



|  |  |  |
| --- | --- | --- |
| (a) | (b) | (c) |
| Fig. 7. The performance of Matrix Completion Algorithm. (a) The impact of learning rate , regularization factor on the performance of Matrix Completion Algorithm; (b) The impact of the rank of latent matrix on the performance of Matrix Completion Algorithm; (c) The performance of Matrix Completion Algorithm on three real-life data collection tasks. | | |

filling them. After missing data process, we use the NO2 and SO2 data from 12 sites for experimental evaluation.

However, the data in above three data collection tasks are all ground truth and there is no data submitted by workers, which can’t be applied to evaluate the performance of RLTD directly. Thus, we generate three synthetic datasets based on the humidity data in Sensor-Scope dataset to evaluate the performance of our proposed RLTD scheme on the F1 score of worker identification, data collection cost and data quality.

In these three synthetic datasets, we assume that there are 20 workers in each site (i.e., 240 workers in total). The difference among them is the proportion of malicious workers to all workers, which is 30%, 50%, and 70% in high quality, medium quality, and low quality dataset respectively. For example, there are 12 sites in high quality dataset. And in each site, there are 14 trustworthy workers and 6 malicious workers. That’s to say, there are 168 trustworthy workers and 72 malicious workers in the whole sensing area. According to Ref. [69-70], the data submitted by trustworthy workers follows normal distribution, while the data submitted by malicious workers follows uniform distribution. So, we generate the data submitted by trustworthy worker following , where is the ground truth of site in cycle and is the standard deviation of data submitted by worker , which is randomly generated within . And the data submitted by malicious worker is generated following .

*5.2 The performance of Matrix Factorization*

First, we will analyze the performance of MF algorithm on three original data collection tasks. In MF algorithm, there are three parameters impacting its performance, which are the learning rate , regularization factor and the rank of latent matrix, and we should assign the optimal parameters to it, so as to achieve the best performance. First, we set the number of historical cycles used in MF to 24, because the data collection tasks used in this paper are performed every 1 or 0.5 hour. And we sample 30% of the data in the first 24 cycles in humidity data to test the impact of three parameters.

By fixing the rank of latent matrix , Fig. 7 (a) shows the training error changing with the iteration times under different learning rate and regularization factor . It can be seen that training error decreases as the iteration increases. On the one hand, the algorithm converges faster with the larger , which can be seen comparing the curves with the same . This is because that determines the magnitude of each gradient descent. However, if is too large, the gradient descent algorithm may fall into a local optimum and fail. On the other hand, the training error is smaller with the smaller , which can be seen comparing the curves with the same . Comprehensively, the training error of the green curve is the smallest and decreases the fastest. Thus, we set the value of and to 0.05 and 0.01, and analyze the impact of the rank of latent matrix on MF algorithm. Fig. 7 (b) shows the training error changing with the iteration times under different . It can be seen that the algorithm converges fastest with . This is because that the larger the rank , the larger the latent matrix, which means that more latent information can be represented. However, larger matrix brings a longer training time in the meanwhile. Therefore, we should comprehensively consider the efficiency of the algorithm and the optimizing effect to choose the rank .



|  |  |  |
| --- | --- | --- |
| (a) | (b) | (c) |
| Fig. 8. The performance of RLSS Algorithm. (a) The performance of RLSS algorithm on humidity task; (b) The performance of RLSS Algorithm on NO2 task; (c) The performance of RLSS Algorithm on SO2 task. | | |

According to the above analysis, we set the parameters as , and , and perform the MF algorithm on three data collection tasks to test the data inference accuracy under different sample rate. The sample rate refers to the ratio of the selected sites to all sites, that is, the proportion of sites with collected data. We randomly extract different proportion (10%~100%) of data from the ground truth matrix in each cycle as the collected data. And then, we apply the MF algorithm to recover the missing data in the constructed matrix and calculate the mean inference accuracy under different sample rate. Fig. 7 (c) shows that the inference accuracies of MF algorithm show an upward trend as the sample rate increases in three tasks. This is because that the higher the sample rate, the more data are collected, and thus the more information is contained in the matrix. And the average inference accuracies on humidity, NO2 and SO2 data are 99.10%, 91.77% and 84.59%, respectively, which demonstrates the efficiency of MF algorithm in data inference.

*5.3 The performance of Site Selection*

However, in Fig. 7 (c), the inference accuracies do not strictly increase with the sample rate. For example, in SO2 dataset, the accuracy of sample rate 0.5 is smaller than that of sample rate 0.4. As explained in Section 4.4, this is because that different site combinations under the same sample rate will result in different inference accuracy. In the experiment of Fig. 7 (c), we randomly extract different proportion (10%~100%) of data from the ground truth matrix for test. Therefore, the selected site combination may not be optimal and the inference accuracy may not be the highest.

In Section 4.4, we proposed the RLSS algorithm to guide the site selection process, so as to select the optimal site combination with the highest data inference accuracy. In this section, we design a simulation experiment to analyze the performance of it. We use the historical data in previous 24 cycles to train site selection agent under different sampling rates (10%~100%), and use the trained agent to guide the site selection process in the 25th cycle. Then, we use the ground truth of the selected sites as the collected data, and use MF algorithm to recover the data of unselected sites. Finally, the inference accuracy is calculated between the recovered data and original data. Moreover, we use the inference accuracy under the guidance of the random selection method as the baseline to demonstrate the efficiency of RLSS algorithm.

Fig. 8 shows that under the guidance of RLSS algorithm, the inference accuracies of MF algorithm strictly increase with the sample rate in three tasks. Moreover, the inference accuracy of RLSS is higher than that of random selection method under all sample rates, especially on NO2 and SO2 tasks, where the average inference accuracy improves by 5.68% and 7.23% respectively, which demonstrates the efficiency of RLSS algorithm in site selection. Moreover, when the sample rate is 40%, under the guidance of RLSS, the data inference accuracy by MF algorithm reaches 99.88%, 96.37% and 97.84% in three tasks respectively. That’s to say, when the sampling rate is 40%, the quality of data collected by RLSS and MF algorithm is reduced by 3.63% at most, while the data collection cost is reduced by 60%, which shows that the RLSS and MF algorithms can greatly reduce the data collection cost with guaranteed high data quality.

*5.4 The performance of Worker Identification*

In Section 4.2, we proposed a worker identification algorithm to identify the trustworthiness of workers, in order that the platform can recruit trustworthy workers for data collection and improve the collected



Fig. 9. The impact of parameters and on the performance of worker identification algorithm on high quality dataset

**

Fig. 10. The impact of parameters and on the performance of worker identification algorithm on medium quality dataset



Fig. 11. The impact of parameters and on the performance of worker identification algorithm on low quality dataset

data quality. Based on the three synthetic datasets generated in Section 5.1, we design a simulation experiment to analyze the performance of it. In the experiment, we assume that there are only two trustworthy sites that contain one trustworthy worker in the network initially. The trust score of trustworthy workers is initialized as 0.95 and the trust score of other workers is initialized as 0.5. From Fig. 8 (a), we can know that when the sample rate is 40%, the data inference accuracy reaches 99.88% under the guidance of RLSS. So, we assume that when 40% of the sites are trustworthy sites in the network, we can use the MF algorithm to recover the data of unselected sites. And we use the F1 score introduced in Formula (11) to evaluate the worker identification result.

There are four parameters may impact the performance of worker identification, which are the accuracy threshold , the decaying factor , and the trust score thresholds and . First, the accuracy threshold and the decaying factor may impact the update of trust score and further impact the result of worker identification. Setting the value of and to 0.9 and 0.4, the impact of parameters and on the performance of worker identification on three synthetic datasets is shown in Fig. 9 to Fig. 11. It can be seen that, with the same threshold , when the value of increases, the F1 score decreases and the identification process gets slower. This is because determines the impact of the historical trust score on the updated trust score. If is too large, the trust score will update slowly and the abnormal situation can’t be detected in time (e.g., a trustworthy worker turns malicious and submits false data). However, if is too small, the trust score will be easily affected by the data submitted by workers in current cycle, which ignores the historical trust score. Besides, the parameter impacts the trust score



Fig. 12. The impact of parameters and on the performance of worker identification algorithm on three synthetic datasets

updating method. In theory, represents the accuracy of data submitted by workers, so the larger value of , the better the performance. Because larger can filter out the workers who submit more accurate data. However, due to some irresistible factors, the data submitted by trustworthy workers is sometimes not very accurate, but they are still trustworthy. So, if is too large, some trustworthy workers who submit less accurate data may be misidentified malicious. Thus, we should compressively consider the impact of parameters and .

Second, the other two parameters that may affect the performance of worker identification are the trust score thresholds and . According to the experiment result shown in Fig. 9 to Fig. 11, we set the value of parameters and to 0.8 and 0.6 in the experiment on three synthetic datasets. Then, the impact of parameters and on the performance of worker identification is shown in Fig. 12. It can be seen that, the F1 score is the highest when with the same . As explained before, some trustworthy workers may submit less accurate data, which means that they will be filter out the trustworthy group if is too large. And if is too small, some undetermined workers may be identified trustworthy in advance while they are actually malicious, because of the insufficient evaluation. Same for , which should also be considered comprehensively. If is too large, the trust of some trustworthy workers who submit less accurate data will update slowly, and those workers may be identified malicious. And, if is too small, it will take a long time to identify the malicious workers, which reduces the system performance.

Comprehensively considering the impact of above parameters, we set the value of them as Table 4.

Table 4. The parameters used in worker identification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset  Quality | Parameters | | | |
|  |  |  |  |
| High | 0.8 | 0.6 | 0.9 | 0.3 |
| Medium | 0.8 | 0.6 | 0.9 | 0.4 |
| Low | 0.8 | 0.6 | 0.9 | 0.3 |

*5.5 The performance of RLTD*

In this section, we are going to evaluate the comprehensive performance of RLTD scheme on collected data quality and data collection cost. In addition, we compare our proposed scheme with two traditional TDD schemes: Mean [33] and Median [33].

The comparison on collected data quality is shown in Fig. 13. It can be seen that, at the beginning, the quality of the ETD obtained by RLTD is relatively poor. This is because that at the beginning, the known trustworthy workers in the network are insufficient and the platform needs to recruit undetermined workers in undetermined sites to collect data and use the weighted average value of data submitted by undetermined workers as ETD, which leads to the poor data quality. However, as the worker identification process proceeds, the trustworthy workers in the network are continuously identified, and the number of trustworthy sites will continuously increase. So, the platform can recruit trustworthy workers to collect data in more trustworthy sites, and the data quality is improved. Moreover, when the number of trustworthy sites exceeds 40%, the platform can select only part of the trustworthy sites by RLSS algorithm for high-quality data collection, and then use MF algorithm to recover the data of unselected sites accurately.



Fig. 13. The accuracy of estimated truth data of three TDD schemes.



Fig. 14. The data collection cost of three TDD schemes.

In detail, the performances of Mean, Median and RLTD schemes are not significantly different in the high quality dataset. This is because that most of the workers in high quality dataset are trustworthy and the data submitted by malicious workers do not have a significant impact. However, in the medium quality and low quality dataset, the performances of Mean and Median schemes get worse as the number of malicious workers in the network increases. But the quality of data obtained by RLTD scheme is high in all the datasets, which exceeds the Mean and Median schemes by 6.1% and 1.31% on average in the high quality dataset, exceeds the Mean and Median schemes by 11.06% and 4.06% on average in the medium quality dataset, and exceeds the Mean and Median schemes by 21.02% and 15.93% on average in the low quality dataset. This is because that in the stabilization stage, the RLTD only recruits trustworthy workers to collect data from trustworthy sites and recovers the data of unselected sites by MF algorithm accurately. Therefore, the RLTD scheme has good robustness against malicious workers in practical MCS applications.

Next, we set the cost of recruiting each worker to be equal and equal to 1. Then, the data collection cost in each cycle can be regarded as the number of recruited workers. The comparison on data collection cost is shown in Fig. 14. Since Mean and Median schemes need to recruit all the workers to collect data in all sites, the data collection cost of them in every cycle is same and extremely high. However, although the number of workers recruited in the first few cycles in RLTD is relatively large, when the number of trustworthy sites exceeds 40%, RLTD only need to select 40% of all sites for data collection, while recovering the data of unselected sites, which significantly reduces the data collection cost. Therefore, the high cost of worker identification in the first few cycles is diluted relative to the overall long-term data collection. Comprehensively, we can see from the experiment result that compared with the traditional TDD schemes, RLTD reduces the average data collection cost in 20 cycles by 81.39%, 83.77% and 85.50% on three synthetic datasets respectively, which significantly improves the system performance.

*5.6 Running Time*

Finally, we list the running times of the main methods on different datasets in the Table 5. Our experiment platform is equipped with Intel(R) Xeon(R) Platinum 8222CL CPU @ 3.00GH and 128.00 GB RAM, and we use the PyTorch framework to run the experiments. Note that we don’t use the GPU to accelerate the training process of the reinforcement learning agent. This is because that the neural network used for the DQN only contains two fully connected layer, which is very simple. So, the GPU acceleration is very slight. What’s more, the tensor transferring between GPU and CPU will consume a lot of unnecessary running time.

Table 5 shows the running time of the MF algorithm to recover a matrix for each cycle and the training and testing time of the RLSS algorithm to select the optimal site selection in a cycle. The MF algorithm costs 0.31-0.34 seconds to recover the missing data. And although the RLSS algorithm costs 113.47-127.32 minutes to train a site selection agent, it only costs 0.01 seconds to guide the platform to select the optimal site combination through the test process.

In practice, although the RLSS algorithm takes a little long time and expensive computation resources to train the agent, it reduces the data collection cost. In MCS, as the data collection tasks are widely distributed in a large scale, the cost to collect data from all the sites is extremely high. So, the computational cost of RLSS algorithm is very small compared to the cost of collecting data from all the sites, which will improve the system performance.

Table 5. the running time

|  |  |  |  |
| --- | --- | --- | --- |
|  | MF  (s) | RLSS\_Train  (min) | RLSS\_Test  (s) |
| Humidity | 0.34 | 127.32 | 0.01 |
| SO2 | 0.31 | 113.47 | 0.01 |
| NO2 | 0.31 | 116.42 | 0.01 |

**6 Policy implication, conclusion and future work**

*6.1 Policy implication*

As one of the most important digital technologies, MCS uses the vast amounts of collected data to construct data-based DSSs, and then make wise decisions for environmental sustainability. To guarantee the fairness of the decisions made by DSSs, the TDD problem is a critical issue that need to be properly addressed, whose objective is to obtain the truth data while avoiding the false data. In addition, the cost of the TDD scheme should be as low as possible. The RLTD scheme proposed in this paper uses worker identification method to identify trustworthy workers for data collection, so as to collect high-quality data. Moreover, combined with the MF algorithm, the cost of the TDD scheme is significantly reduced.

The RLTD can help the data-based DSSs to make fair decisions for environment sustainability in many fields, such industrial monitoring, intelligent transportation, medical health, environmental monitoring, etc. For example, in the studies to understand the migratory habits of migratory birds and their dependence on the environment. The MCS platform can recruit the trustworthy workers along the migration route to collect the required data, such as images and videos, which significantly reduces the data collection cost. Then, it provides the collected data to experts to analyze the habits of the migratory birds. Another example is the urban rail transit planning. The platform can collect the traffic and passenger flow data through the drivers and passengers, and use the collected data to develop an urban transportation network that facilitate the travels. What’s more, the well-designed transportation network can encourage people to travel by public transport rather than by car, which helps to improve the air quality.

*6.2 Conclusion*

In this paper, we investigate the TDD problem in MCS. As one of the most important digital technologies, MCS uses the vast amounts of collected data to construct data-based DSSs, and make wise decisions for environmental sustainability. However, the false data submitted by dishonest or malicious workers may cause the DSSs to make wrong decisions, potentially resulting in severe loss of life and property. Therefore, we propose a Reinforcement Learning-based Truth Data Discovery (RLTD) scheme to identify trustworthy workers in MCS, in order that the MCS platform can obtain truth data while avoiding false data for reliable services. First, we propose a novel trustworthiness-based worker identification method to accurately identify the trustworthy and malicious workers in the network. Second, we find a new application of data inference to TDD, and propose a worker recruitment method that only needs to recruit () workers for TDD in tasks, which significantly reduces the data collection cost and improves the collected data quality. Finally, we propose a Reinforcement Learning-based Site Selection (RLSS) method to select the optimal site combination for data inference. By using RLSS, we can reduce the sites to be selected for data inference and thus reduce the data collection cost. After simulation experiments on two real-life datasets and three synthetic datasets, our proposed RLTD scheme can improve the collected data quality by 1.31%-21.02%, reduce the data collection cost by 81.39%-85.50% compared to the traditional TDD schemes, and identify workers with accuracy of 86.67%-94.58%.

*6.3 Future work*

Although the proposed RLTD can improve the collected data quality and reduce the data collection cost in MCS, and thus help the constructed data-based DSSs to make fair decisions, there are still some issues that need to be considered. First, in the case of insufficient trustworthy workers, we use the weighted mean value of the data collected by undetermined workers as the ETD, which is less trustworthy. Second, we use a fixed value as an upper limit on the number of selected sites to train the site selection agent. However, the value of is highly dependent on the characteristics of tasks. Therefore, in future work, we will further study a better way to obtain the ETD of the undetermined sites and improve the data quality in the case of insufficient trustworthy workers. Moreover, we will exploit a better site selection algorithm to select sites without the manually set upper limit on the number of selected sites intelligently and further improve the site selection performance.

**CRediT authorship contribution statement**

**Tingxuan Liang**: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing-original draft. **Lingyi Chen**: Review & editing. **Mingfeng Huang**: Review & editing. **Xiaoheng Deng**: Supervision, Review & editing. **Shaobo Zhang**: Review & editing. **Neal N. Xiong**: Supervision, review & editing. **Anfeng Liu**: Conceptualization, Funding acquisition, Resources, Supervision, Writing-original draft, Writing-review & editing.

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Data availability**

Data will be made available on request.

**Acknowledgment**

This work was supported in part by the National Natural Science Foundation of China (62072475, 61772554).

**Reference**

1. M. Castells, The information age: Economy, society and culture (3 volumes), Blackwell, Oxford. 1997 (1996) 1998.
2. K. Arts, R. van der Wal, W.M. Adams, Digital technology and the conservation of nature, Ambio. 44 (2015) 661–673. https://doi.org/10.1007/s13280-015-0705-1.
3. D. Reinsel, J. Gantz, and J. Rydning, Data age 2025: the digitization of the world – from edge to core, IDC White Paper Doc#US44413318 (2018) 1-29.
4. J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, B. Fang, A Survey on Access Control in the Age of Internet of Things, IEEE Internet of Things Journal. 7 (2020) 4682–4696. https://doi.org/10.1109/JIOT.2020.2969326.
5. M. Karatas, L. Eriskin, M. Deveci, D. Pamucar, H. Garg, Big Data for Healthcare Industry 4.0: Applications, challenges and future perspectives, Expert Systems with Applications. 200 (2022) 116912. https://doi.org/10.1016/j.eswa.2022.116912.
6. Y. Chai, L. Du, J. Qiu, L. Yin, Z. Tian, Dynamic Prototype Network based on Sample Adaptation for Few-Shot Mal-ware Detection, IEEE Transactions on Knowledge and Data Engineering. (2022) 1–1. https://doi.org/10.1109/TKDE.2022.3142820.
7. H. Hellani, L. Sliman, A.E. Samhat, E. Exposito, Overview on the Blockchain-Based Supply Chain Systematics and Their Scalability Tools, Emerging Science Journal. 4 (2021) 45–69. https://doi.org/10.28991/esj-2021-SP1-04.
8. M. ALMAKHOUR, A. WEHBY, L. SLIMAN, A.E. SAMHAT, A. MELLOUK, Smart Contract Based Solution for Secure Distributed SDN, in: 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2021: pp. 1–6. https://doi.org/10.1109/NTMS49979.2021.9432647.
9. H. Hellani, L. Sliman, A.E. Samhat, E. Exposito, On Blockchain Integration with Supply Chain: Overview on Data Transparency, Logistics. 5 (2021) 46. https://doi.org/10.3390/logistics5030046.
10. Z. Tian, X. Gao, S. Su, J. Qiu, X. Du, M. Guizani, Evaluating Reputation Management Schemes of Internet of Vehicles Based on Evolutionary Game Theory, IEEE Transactions on Vehicular Technology. 68 (2019) 5971–5980. https://doi.org/10.1109/TVT.2019.2910217.
11. Z. Tian, M. Li, M. Qiu, Y. Sun, S. Su, Block-DEF: A secure digital evidence framework using blockchain, Information Sciences. 491 (2019) 151–165. https://doi.org/10.1016/j.ins.2019.04.011.
12. H. Lu, C. Jin, X. Helu, C. Zhu, N. Guizani, Z. Tian, AutoD: Intelligent Blockchain Application Unpacking Based on JNI Layer Deception Call, IEEE Network. 35 (2021) 215–221. https://doi.org/10.1109/MNET.011.2000467.
13. P. Rani, A.R. Mishra, M. Deveci, J. Antucheviciene, New complex proportional assessment approach using Einstein aggregation operators and improved score function for interval-valued Fermatean fuzzy sets, Computers & Industrial Engineering. 169 (2022) 108165. https://doi.org/10.1016/j.cie.2022.108165.
14. H.A. Dolatsara, E. Kibis, M. Caglar, S. Simsek, A. Dag, G.A. Dolatsara, D. Delen, An interpretable decision-support systems for daily cryptocurrency trading, Expert Systems with Applications. 203 (2022) 117409. https://doi.org/10.1016/j.eswa.2022.117409.
15. I. Gokasar, M. Deveci, O. Kalan, CO2 Emission based prioritization of bridge maintenance projects using neutrosophic fuzzy sets based decision making approach, Research in Transportation Economics. 91 (2022) 101029. https://doi.org/10.1016/j.retrec.2021.101029.
16. R. Krishankumar, S. Supraja Nimmagadda, A.R. Mishra, D. Pamucar, K.S. Ravichandran, A.H. Gandomi, An integrated decision model for cloud vendor selection using probabilistic linguistic information and unknown weights, Engineering Applications of Artificial Intelligence. 114 (2022) 105114. https://doi.org/10.1016/j.engappai.2022.105114.
17. Z. Mi, D. Coffman, The sharing economy promotes sustainable societies, Nat Commun. 10 (2019) 1214. https://doi.org/10.1038/s41467-019-09260-4.
18. X. Chen, M. Despeisse, B. Johansson, Environmental Sustainability of Digitalization in Manufacturing: A Review, Sustainability. 12 (2020) 10298. https://doi.org/10.3390/su122410298.
19. M. Deveci, D. Pamucar, I. Gokasar, M. Isik, D. Coffman, Fuzzy Einstein WASPAS approach for the economic and societal dynamics of the climate change mitigation strategies in urban mobility planning, Structural Change and Economic Dynamics. 61 (2022) 1–17. https://doi.org/10.1016/j.strueco.2022.01.009.
20. I.M. Hezam, A.R. Mishra, P. Rani, A. Alshamrani, Assessing the barriers of digitally sustainable transportation system for persons with disabilities using Fermatean fuzzy double normalization-based multiple aggregation method, Applied Soft Computing. 133 (2023) 109910. https://doi.org/10.1016/j.asoc.2022.109910.
21. S.L. Pan, L. Carter, Y. Tim, M.S. Sandeep, Digital sustainability, climate change, and information systems solutions: Opportunities for future research, International Journal of Information Management. 63 (2022) 102444. https://doi.org/10.1016/j.ijinfomgt.2021.102444.
22. M.A. Reuter, Digitalizing the Circular Economy, Metall Mater Trans B. 47 (2016) 3194–3220. https://doi.org/10.1007/s11663-016-0735-5.
23. G.T. Pecl, M.B. Araújo, J.D. Bell, J. Blanchard, T.C. Bonebrake, I.-C. Chen, T.D. Clark, R.K. Colwell, F. Danielsen, B. Evengård, L. Falconi, S. Ferrier, S. Frusher, R.A. Garcia, R.B. Griffis, A.J. Hobday, C. Janion-Scheepers, M.A. Jarzyna, S. Jennings, J. Lenoir, H.I. Linnetved, V.Y. Martin, P.C. McCormack, J. McDonald, N.J. Mitchell, T. Mustonen, J.M. Pandolfi, N. Pettorelli, E. Popova, S.A. Robinson, B.R. Scheffers, J.D. Shaw, C.J.B. Sorte, J.M. Strugnell, J.M. Sunday, M.-N. Tuanmu, A. Vergés, C. Villanueva, T. Wernberg, E. Wapstra, S.E. Williams, Biodiversity redistribution under climate change: Impacts on ecosystems and human well-being, Science. 355 (2017) eaai9214. https://doi.org/10.1126/science.aai9214.
24. M. Karaliopoulos, E. Bakali, Optimizing Mobile Crowdsensing Platforms for Boundedly Rational Users, IEEE Trans. Mob. Comput. 21 (2022) 1305–1318. https://doi.org/10.1109/TMC.2020.3023757.
25. Z. Zhang, Z. Shi, Y. Gu. Ziv-Zakai Bound for DOAs Estimation. IEEE Transactions on Signal Processing, 71 (2022) 136-149. https://doi.org/10.1109/TSP.2022.3229946.
26. A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, J. Eriksson, VTrack: accurate, energy-aware road traffic delay estimation using mobile phones, in: Proc. 7th ACM Conf. Embed. Networked Sens. Syst., Association for Computing Machinery, New York, NY, USA, 2009: pp. 85–98. https://doi.org/10.1145/1644038.1644048.
27. N. Maisonneuve, M. Stevens, M.E. Niessen, L. Steels, NoiseTube: Measuring and mapping noise pollution with mobile phones, in: I.N. Athanasiadis, A.E. Rizzoli, P.A. Mitkas, J.M. Gómez (Eds.), Inf. Technol. Environ. Eng., Springer, Berlin, Heidelberg, 2009: pp. 215–228. https://doi.org/10.1007/978-3-540-88351-7\_16
28. T. Yan, B. Hoh, D. Ganesan, K. Tracton, T. Iwuchukwu, and J.- S. Lee, CrowdPark: A crowdsourcing-based parking reservation system for mobile phones, University of Massachusetts at Amherst Tech. Report, 2011.
29. S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, W. Trappe, ParkNet: drive-by sensing of road-side parking statistics, in: Proc. 8th Int. Conf. Mob. Syst. Appl. Serv., Association for Computing Machinery, New York, NY, USA, 2010: pp. 123–136. https://doi.org/10.1145/1814433.1814448.
30. L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, A. M’hamed, Sparse mobile crowdsensing: challenges and opportunities, IEEE Commun. Mag. 54 (2016) 161–167. https://doi.org/10.1109/MCOM.2016.7509395.
31. J. Xu, Z. Luo, C. Guan, D. Yang, L. Liu, Y. Zhang, Hiring a Team from Social Network: Incentive Mechanism Design for Two-tiered Social Mobile Crowdsourcing, IEEE Trans. Mob. Comput. (2022) 1–1. https://doi.org/10.1109/TMC.2022.3162108.
32. C. Huang, M. Li, F. Cao, H. Fujita, Z. Li, X. Wu. Are Graph Convolutional Networks with Random Weights Feasible? IEEE Transaction on Pattern Analysis and Machine Intelligence, 45 (3) (2023) 2751-2768. https://doi.org/10.1109/TPAMI.2022.3183143.
33. Y. Zheng, G. Li, Y. Li, C. Shan, R. Cheng, Truth inference in crowdsourcing: is the problem solved?, Proc. VLDB Endow. 10 (2017) 541–552. https://doi.org/10.14778/3055540.3055547.
34. S. Ye, J. Wang, H. Fan, Z. Zhang, Probabilistic model for truth discovery with mean and median check framework, Knowl.-Based Syst. 233 (2021) 107482. https://doi.org/10.1016/j.knosys.2021.107482.
35. P. Sun, Z. Wang, L. Wu, Y. Feng, X. Pang, H. Qi, Z. Wang, Towards Personalized Privacy-Preserving Incentive for Truth Discovery in Mobile Crowdsensing Systems, IEEE Trans. Mob. Comput. 21 (2022) 352–365. https://doi.org/10.1109/TMC.2020.3003673.
36. K. Li, S. Wang, X. Cheng, Q. Hu, A Misreport- and Collusion-Proof Crowdsourcing Mechanism Without Quality Verification, IEEE Trans. Mob. Comput. 21 (2022) 3084–3095. https://doi.org/10.1109/TMC.2021.3052873.
37. J. Tang, S. Fu, X. Liu, Y. Luo, M. Xu, Achieving Privacy-preserving and Lightweight Truth Discovery in Mobile Crowdsensing, IEEE Trans. Knowl. Data Eng. (2021) https://doi.org/10.1109/TKDE.2021.3054409.
38. Y. Liu, M. Dong, K. Ota, A. Liu, ActiveTrust: Secure and Trustable Routing in Wireless Sensor Networks, IEEE Trans. Inf. Forensics Secur. 11 (2016) 2013–2027. https://doi.org/10.1109/TIFS.2016.2570740.
39. J. Guo, H. Wang, W. Liu, G. Huang, J. Gui, S. Zhang, A lightweight verifiable trust based data collection approach for sensor–cloud systems, J. Syst. Archit. 119 (2021) 102219. https://doi.org/10.1016/j.sysarc.2021.102219.
40. J. Guo, G. Huang, Q. Li, N.N. Xiong, S. Zhang, T. Wang, STMTO: A smart and trust multi-UAV task offloading system, Inf. Sci. 573 (2021) 519–540. https://doi.org/10.1016/j.ins.2021.05.020.
41. J. Guo, A. Liu, K. Ota, M. Dong, X. Deng, N.N. Xiong, ITCN: An Intelligent Trust Collaboration Network System in IoT, IEEE Trans. Netw. Sci. Eng. 9 (2022) 203–218. https://doi.org/10.1109/TNSE.2021.3057881.
42. J. Bai, Z. Zeng, T. Wang, S. Zhang, N.N. Xiong, A. Liu, TANTO: An Effective Trust based Unmanned Aerial Vehicle Computing System for the Internet-of-Things, IEEE Internet Things J. (2022) https://doi.org/10.1109/JIOT.2022.3150765.
43. M. Huang, A. Liu, N.N. Xiong, J. Wu, A UAV-Assisted Ubiquitous Trust Communication System in 5G and Beyond Networks, IEEE J. Sel. Areas Commun. 39 (2021) 3444–3458. https://doi.org/10.1109/JSAC.2021.3088675.
44. W. Mo, W. Liu, G. Huang, N.N. Xiong, A. Liu, S. Zhang, A Cloud-Assisted Reliable Trust Computing Scheme for Data Collection in Internet of Things, IEEE Trans. Ind. Inform. 18 (2022) 4969–4980. https://doi.org/10.1109/TII.2021.3108149.
45. C. Zhang, L. Zhu, C. Xu, X. Liu, K. Sharif, Reliable and Privacy-Preserving Truth Discovery for Mobile Crowdsensing Systems, IEEE Trans. Dependable Secure Comput. 18 (2021) 1245–1260. https://doi.org/10.1109/TDSC.2019.2919517.
46. Y. Ren, W. Liu, A. Liu, T. Wang, A. Li, A privacy-protected intelligent crowdsourcing application of IoT based on the reinforcement learning, Future Gener. Comput. Syst. 127 (2022) 56–69. https://doi.org/10.1016/j.future.2021.09.003.
47. N. Jiang, W. Jie, J. Li, X. Liu, D. Jin, GATrust: A Multi-Aspect Graph Attention Network Model for Trust Assessment in OSNs, IEEE Trans. Knowl. Data Eng. (2022) https://doi.org/10.1109/TKDE.2022.3174044.
48. R.K. Rana, C.T. Chou, S.S. Kanhere, N. Bulusu, W. Hu, Ear-phone: an end-to-end participatory urban noise mapping system, in: Proc. 9th ACMIEEE Int. Conf. Inf. Process. Sens. Netw., Association for Computing Machinery, New York, NY, USA, 2010: pp. 105–116. https://doi.org/10.1145/1791212.1791226.
49. S. He, K.G. Shin, Steering Crowdsourced Signal Map Construction via Bayesian Compressive Sensing, in: IEEE INFOCOM 2018 - IEEE Conf. Comput. Commun., IEEE, Honolulu, HI, 2018: pp. 1016–1024. https://doi.org/10.1109/INFOCOM.2018.8485972.
50. N. Marchang, R. Tripathi, KNN-ST: Exploiting Spatio-Temporal Correlation for Missing Data Inference in Environmental Crowd Sensing, IEEE Sens. J. 21 (2021) 3429–3436. https://doi.org/10.1109/JSEN.2020.3024976.
51. K. Xie, X. Li, X. Wang, G. Xie, J. Wen, D. Zhang, Active Sparse Mobile Crowd Sensing Based on Matrix Completion, in: Proc. 2019 Int. Conf. Manag. Data, Association for Computing Machinery, New York, NY, USA, 2019: pp. 195–210. https://doi.org/10.1145/3299869.3319856.
52. H.-J. Xue, X. Dai, J. Zhang, S. Huang, J. Chen, Deep Matrix Factorization Models for Recommender Systems, in: Proc. Twenty-Sixth Int. Jt. Conf. Artif. Intell., International Joint Conferences on Artificial Intelligence Organization, Melbourne, Australia, 2017: pp. 3203–3209. https://doi.org/10.24963/ijcai.2017/447.
53. Y. Koren, R. Bell, C. Volinsky, Matrix Factorization Techniques for Recommender Systems, Computer. 42 (2009) 30–37. https://doi.org/10.1109/MC.2009.263.
54. W. Hu, X. Zhu, T. Wang, Y. Yi, G. Yu, Discrete subspace structure constrained human motion capture data recovery, Applied Soft Computing. 129 (2022) 109617. https://doi.org/10.1016/j.asoc.2022.109617.
55. Y. Meng, M. Jin, X. Tang, J. Xu, Drug repositioning based on similarity constrained probabilistic matrix factorization: COVID-19 as a case study, Applied Soft Computing. 103 (2021) 107135. https://doi.org/10.1016/j.asoc.2021.107135.
56. S. Yan, A. Yang, S. Kong, B. Bai, X. Li, Predictive intelligence powered attentional stacking matrix factorization algorithm for the computational drug repositioning, Applied Soft Computing. 110 (2021) 107633. https://doi.org/10.1016/j.asoc.2021.107633.
57. X. Zhu, Y. Luo, A. Liu, W. Tang, Md.Z.A. Bhuiyan, A Deep Learning-Based Mobile Crowdsensing Scheme by Predicting Vehicle Mobility, IEEE Trans. Intell. Transp. Syst. 22 (2021) 4648–4659. https://doi.org/10.1109/TITS.2020.3023446.
58. E. Wang, M. Zhang, X. Cheng, Y. Yang, W. Liu, H. Yu, L. Wang, J. Zhang, Deep Learning-Enabled Sparse Industrial Crowdsensing and Prediction, IEEE Trans. Ind. Inf. 17 (2021) 6170–6181. https://doi.org/10.1109/TII.2020.3028616.
59. X. Liu, Y. Liu, N. Zhang, W. Wu, A. Liu, Optimizing Trajectory of Unmanned Aerial Vehicles for Efficient Data Acquisition: A Matrix Completion Approach, IEEE Internet Things J. 6 (2019) 1829–1840. https://doi.org/10.1109/JIOT.2019.2894257.
60. T. Liu, Y. Zhu, Y. Yang, F. Ye, $ALC2$ : When Active Learning Meets Compressive Crowdsensing for Urban Air Pollution Monitoring, IEEE Internet Things J. 6 (2019) 9427–9438. https://doi.org/10.1109/JIOT.2019.2939552.
61. V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature. 518 (2015) 529–533. https://doi.org/10.1038/nature14236.
62. G. Lample, D.S. Chaplot, Playing FPS games with deep reinforcement learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
63. W. Liu, L. Wang, E. Wang, Y. Yang, D. Zeghlache, D. Zhang, Reinforcement learning-based cell selection in sparse mobile crowdsensing, Comput. Netw. 161 (2019) 102–114. https://doi.org/10.1016/j.comnet.2019.06.010.
64. W. Liu, Y. Yang, E. Wang, J. Wu, User Recruitment for Enhancing Data Inference Accuracy in Sparse Mobile Crowdsensing, IEEE Internet Things J. 7 (2020) 1802–1814. https://doi.org/10.1109/JIOT.2019.2957399.
65. L. Han, Z. Yu, L. Wang, Z. Yu, B. Guo, Keeping Cell Selection Model Up-to-Date to Adapt to Time-Dependent Environment in Sparse Mobile Crowdsensing, IEEE Internet Things J. 8 (2021) 13914–13925. https://doi.org/10.1109/JIOT.2021.3068415.
66. F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, M. Parlange, SensorScope: Application-specific sensor network for environmental monitoring, ACM Trans. Sen. Netw. 6 (2010) 1–32. https://doi.org/10.1145/1689239.1689247.
67. Y. Zheng, X. Chen, Q. Jin, Y. Chen, X. Qu, X. Liu, E. Chang, Y. Ma, Y. Rui, W. Sun, A Cloud-Based Knowledge Discovery System for Monitoring Fine-Grained Air Quality, (2014) 4.
68. Y. Zheng, F. Liu, H.-P. Hsieh, U-Air: when urban air quality inference meets big data, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Chicago Illinois USA, 2013: pp. 1436–1444. https://doi.org/10.1145/2487575.2488188.
69. X. Zhang, Y. Wu, L. Huang, H. Ji, G. Cao, Expertise-Aware Truth Analysis and Task Allocation in Mobile Crowdsourcing, IEEE Trans. on Mobile Comput. 20 (2021) 1001–1016. https://doi.org/10.1109/TMC.2019.2955688.
70. V.C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning From Crowds, Journal of Machine Learning Research. 11 (2010) 1297–1322.
71. S. Huang, S. Ontañón, A Closer Look at Invalid Action Masking in Policy Gradient Algorithms, in: The International FLAIRS Conference Proceedings, 2022. https://doi.org/10.32473/flairs.v35i.130584.