applied sciences

MDPI

Article

Personalized Privacy Protection Based on Space Grid in Mobile Crowdsensing

Hengfei Gao, Ziqing Zhang and Hongwei Zhao *

College of Computer Science and Technology, Jilin University, Changchun 130012, China
* Correspondence: zhaohw@jlu.edu.cn

Abstract: The rapid proliferation of handheld intelligent devices and the advent of 5G technology have brought about convenient and fast services for people. In perception-oriented application services, participating users will upload sensitive mobile data in order to obtain benefits. While devising privacy protection strategies to ensure data security, it is crucial to accomplish task perception related to data collection to the fullest extent possible. To address this challenge, this paper proposes a personalized data privacy protection algorithm based on an adaptive dynamic adjustment grid and the minimum wage task allocation strategy. According to the different levels of users' needs for privacy protection, combined with the privacy budget allocation strategy, we design a different-level differential privacy protection mechanism and consider the reward mechanism in task allocation to balance the effectiveness and security of the location data uploaded by users. Experiments show that the strategy proposed in this paper can not only protect the data but also enable users to freely choose the level of privacy protection.

Keywords: differential privacy; personalized privacy protection; task assignment; mobile crowdsensing

Citation: Gao, H.; Zhang, Z.; Zhao, H. Personalized Privacy Protection Based on Space Grid in Mobile Crowdsensing. *Appl. Sci.* **2023**, *13*, 12696. https://doi.org/10.3390/app132312696

Academic Editor: Rosario Pecora

Received: 25 September 2023
Revised: 10 November 2023
Accepted: 20 November 2023
Published: 27 November 2023

1. Introduction

The emergence of mobile crowdsensing technology [1–3] comes from the integration of sensors and embedded devices. Handheld devices can acquire useful information in the real world by collecting, uploading and sharing data [4,5]. In various scenarios like social recommendations (e.g., food or hotel recommendations) and real-time monitoring of the surrounding environment (e.g., noise level monitoring) [6,7], an application uses a mobile device's perception capabilities to enhance data credibility, thereby minimizing the expenses of installing sensor equipment. However, its implementation may inadvertently expose sensitive information about mobile users, including social connections and location data [8,9]. The platform's task allocation relies on the user's location, causing the data collected by the user's device to potentially reveal their trajectory or other details. For instance, Google Maps utilizes "anonymous" location data from drivers to create real-time maps, yet it inadvertently discloses the driver's route and location. An attacker with prior knowledge will use the known information to mine the user's privacy to threaten the user's personal safety. Therefore, when designing perceptual service applications, the first consideration is to protect the privacy and security of collectors. Only by designing a reasonable and effective data protection mechanism can users accept more tasks.

Users may disclose their own sensitive information when participating in crowdsensing: a malicious cloud platform may use user information to make profits, which will cause immeasurable losses to users and thus dampen the enthusiasm of mobile users for participating in crowdsensing. Therefore, how to solve the privacy protection problem in crowdsensing is crucial; privacy protection has gradually been paid great attention by scholars, and a variety of privacy protection methods have been proposed [10–14]. In 2003, Beresford first proposed the concept of location privacy protection [15]. Privacy

protection is primarily focused on concealing users' identity and location information for safeguarding their confidentiality [16,17]. Currently, common privacy protection methods include anonymity-based methods, reputation-based methods and differential privacy-based methods.

There are several mobile crowd awareness schemes using anonymity technology to protect privacy data at home and abroad [18–21], such as k-anonymity technology [22]. The main idea of the k-anonymity technique proposed by Samarati and Sweeney is as follows: Assuming that the attacker is faced with massive data and there are a certain number of quasi-identifiers (such as age, gender, salary, etc.) in the massive data of which records cannot be identified, the attacker can use prior knowledge to narrow the data to a certain range of equivalence classes that satisfy their prior knowledge but cannot lock the attack target from the equivalence class [22,23]. This technology uses equivalence classes to protect personal privacy, and parameter k can measure the maximum information disclosure risk that users can bear [24]. Nevertheless, anonymization is insufficient for privacy preservation, since mobile users may be traced via travel routes and social relations. Meanwhile, despite the efficacy of k-anonymity technology in data protection, its reliance on the attacker's background knowledge poses a significant vulnerability. If faced with a new attacker possessing unknown prior knowledge, the attacker can potentially differentiate various records within the published equivalent dataset beyond the predicted scope, leading to deanonymization [25]. So, the security of the k-anonymous model depends on the knowledge possessed by the attacker. And since it is impossible to prove that the algorithm can evaluate the mathematical process of privacy levels, it is not a very perfect privacy protection strategy.

Reputation-based approaches are popular in mobile crowdsensing for allocating tasks to mobile users, but they have their inherent weaknesses. Reputation-based mechanisms [19,26–28] can assign tasks to mobile users. However, the mechanism reliant on reputation is vulnerable due to its dependency on a trusted third party (TTP) for managing reputation, which makes it vulnerable to reputation-linking attacks, in which anonymous mobile users can be reidentified based on their reputations. The SPOON framework, proposed by Ni et al., addresses this weakness by utilizing proxy re-encryption and BBS signature technology to safeguard sensitive information of mobile users and customers. Named SPOON [6], this framework enables registered customers and mobile users to anonymously demonstrate their ability and trustworthiness for participating in service perception tasks, ensuring the security of the tasks and reports. However, the effectiveness of this technology still relies on the guarantee provided by a trusted third party. If this third party breaches trust, the data remain vulnerable and unprotected.

The emergence of differential privacy technology effectively addresses the aforementioned issues. Differential privacy was originally used in the field of statistical databases (database). Dwork [24,25,29] first applied differential privacy technology to the statistical database field with the aim of safeguarding individuals' privacy information when publishing statistical data. Differential privacy is advantageous as it does not rely on an attacker's prior knowledge and can be mathematically proven using a quantitative evaluation method. This technology has gained significant attention from privacy protection researchers and has been implemented in various areas, including privacy-protected data publishing and mining.

While designing privacy protection strategies to ensure the security of data, we should complete the task perception related to data collection as much as possible. To address this problem, this paper proposes a personalized data privacy protection algorithm based on an adaptive dynamic adjustment grid and the lowest paid task allocation strategy. Considering the privacy protection requirements of users at different levels, we have designed various levels of differentiated privacy protection mechanisms by incorporating the privacy budget allocation strategy. In addition, we consider a reward mechanism in task assignment to balance the effectiveness and security of user-uploaded location data. Experiments

demonstrate that the strategy proposed in this paper not only safeguards the data but also empowers users to freely select the level of privacy protection.

In summary, this paper makes the following contributions:

- Privacy protection in perception: We investigate the privacy protection problem in perception-oriented application services. Our aim is to design privacy protection strategies ensuring data security, with the capability to comprehensively fulfill the task of data collection perception.
- Personalized data privacy protection: We propose a personalized data privacy protection algorithm based on an adaptive dynamic adjustment grid and the lowest paid task allocation strategy. Considering the varying levels of user privacy protection needs, we design different-level privacy protection mechanisms by incorporating the privacy budget allocation strategy. We also consider a reward mechanism in task allocation to balance the effectiveness and security of user-uploaded location data.
- Extensive evaluation: We conduct thorough evaluations on various real-world datasets. The results demonstrate that our proposed strategy not only safeguards the data but also allows users to freely choose their preferred level of privacy protection.

The remainder of this paper is organized as follows. After reviewing the related works in Section 2, we introduce the model framework in Section 3. Then, the personalized privacy protection is proposed in Section 4, followed by the evaluations in Section 5 and discussion in Section 6. Finally, we conclude this paper in Section 7.

2. Related Work

2.1. Differential Privacy

A differential attack occurs when the attacker leverages existing knowledge to compare it with the query information, attempting to deduce the most probable guess. For instance, if a school releases the results of a course with 100 students, revealing that only six students failed, an attacker possessing the score information of 99 students can attempt to guess whether the remaining student passed or failed. Differential privacy protection technology introduces noise to the original data to ensure a similar data distribution. Consequently, when an attacker adds or removes data, the disparity in data distribution becomes indistinguishable, preventing the identification of specific users within the published data. The basic concepts and definitions of differential privacy are as follows:

(ε-differential privacy) In the algorithm, a series of queries Q are performed on any two adjacent datasets D and $D^{'}$, and P_Q is the probability of the set of all possible outputs of Q. S_Q is any subset of P_Q. If the algorithm satisfies:

$$P[Q(D) \in S_Q] \leq e^{\varepsilon} \cdot P[Q(D^{'}) \in S_Q] \tag{1}$$

then this algorithm satisfies ε-differential privacy, where datasets D and $D^{'}$ are adjacent datasets. That is, even if any tuple is changed, the probability difference of the output results is very small. The attacker cannot guess the dataset and can play a role in protecting user data [30].

(Global sensitivity) When conducting a series of random queries Q on any two adjacent datasets D and $D^{'}$, the global sensitivity of the query function is the maximum Manhattan distance of the output. The global sensitivity can obtain the variation range of a query function on a pair of adjacent datasets. And the formula is as follows:

$$QS_{q(D)} = \max_{D,D'} \left\| Q(D) - Q(D') \right\|_1 \tag{2}$$

(Local sensitivity) When conducting a series of random queries Q on any two adjacent datasets D and $D^{'}$, the formula of local sensitivity is as follows:

$$LS_{q(D)} = \max_{D'} \left\| Q(D) - Q(D') \right\|_1 \tag{3}$$

The local sensitivity is determined via the query function and the distribution of the query dataset, whereas the global sensitivity is solely related to the query function and is independent of the query dataset. Utilizing local sensitivity poses a certain risk of data leakage. Therefore, in the algorithm proposed in this paper, we will adopt global sensitivity. As mentioned earlier, differential privacy can be achieved by introducing noise to the query results, yet excessive noise can impede data availability. Since sensitivity signifies a change in query results due to the deleting of any record in the dataset, it is commonly employed as a parameter to measure the amount of noise in differential privacy.

2.2. Noise-Adding Mechanism in Differential Privacy

Because differential privacy only allows access to the database through counting and summation, random noise needs to be added to each query result to protect data privacy. Only in this way can the attacker be prevented from guessing the existence of the target in the database and obtaining specific information from the set of query results. Next, we will introduce the common noise mechanism in differential privacy.

(Laplace mechanism) If we send a series of query requests to database D: Query = q_1, q_2, q_3, …, q_n, the database will produce a real answer $q(D)$. Differential privacy protection adds noise to the numerical results to obtain a series of results with the same probability distribution [31,32]. The Laplace mechanism solves this problem well and needs to provide a parameter λ. It is calculated as follows:

$$\lambda = \frac{QS}{\varepsilon} \tag{4}$$

The Laplace distribution has a mean of 0 and a variance of λ^2, and it satisfies ε-differential privacy. The noise added to the query set result $q(D)$ during the query is $\eta \sim Laplace(0, \lambda)$. Therefore, the returned result is:

$$A(D) = q(D) + \eta \tag{5}$$

Clearly, the smaller the privacy budget, the greater the noise when adding to the original data, thereby enhancing the level of protection, but inevitably compromising the data's availability. To cater to the personalized privacy protection requirements of users, this paper categorizes protection levels by assigning different privacy budgets.

2.3. Private Space Decomposition

In database management, there exists a specialized storage structure known as an index, which enables the rapid retrieval of required data. For instance, in a student information database, the index allows for the swift retrieval of specific student information. Common index storage is generally implemented using B-tree or B+tree, facilitating the quick location of specified data through dichotomy. However, these conventional indexes are designed solely for one-dimensional data. When dealing with data represented on two-dimensional coordinates, spatial indexes are employed in database management. Data storage commonly utilizes quadtree [33], R-tree, k-d tree and the geohash algorithm to convert two-dimensional data into one-dimensional data, enabling the use of B-tree indexes for efficient spatial data point searches.

Building upon the definition of query function in differential privacy, Graham Cormode et al. proposed the concept of "private space decomposition (PSD)" in Reference [34]. PSD is a concept applied in the field of data management. It refers to dividing the data space into smaller subspaces with the introduction of differential privacy protection measures in spatial data structures to achieve the goal of data privacy protection. Under the premise of privacy protection, introducing noise at different levels of the data space can safeguard data privacy while maintaining data structure effectiveness and retrievability. Private space decomposition technology is used in applications involving personal and private data to balance the requirements of data availability and privacy protection.

When using PSD for privacy protection in spatial indexing, it is common to utilize the spatial index tree structure and divide the geographical area into subintervals, which serve as the next-level nodes of the tree structure. Leaf nodes contain two-dimensional coordinates, while the parent node stores the summarized data information of child nodes adhering to differential privacy, essentially representing the index information. To ensure the objective of data privacy protection, noise is introduced to the spatial nodes during index establishment, ultimately leading to query results aligned with the distribution of differential privacy.

In addition to the tree hierarchy, the index for spatial data also has a plane structure, such as a grid structure. Wahbeh Qardaji et al. [30] proposed the concept of a multilevel grid index and compared it with the traditional tree index. Their findings indicate that the multilevel grid offers better performance in terms of differential privacy protection than the traditional hierarchy. They further introduced the unified grid granularity method [30]. Building upon the idea of unifying the grid granularity partition, Reference [35] presented an adaptive grid (AG) method characterized by robustness and simplicity. This method calculates the size of the second grid based on the results of the first grid query. In comparison to the PSD, the AG method also demonstrates robustness and simplicity. While the dynamic grid utilizes some data in the second calculation of grid granularity, its impact on privacy protection is not significant. This paper inherits the adaptive grid (AG) method and makes improvements based on personalized user needs. When adding noise to the grid by using the Laplace mechanism, the personalized user needs are considered.

3. System Model and Framework Overview

In this section, we first introduce the system model used in this work. Next, we propose a personalized privacy protection framework for mobile crowdsensing, with a detailed workflow presented in Figure 1.

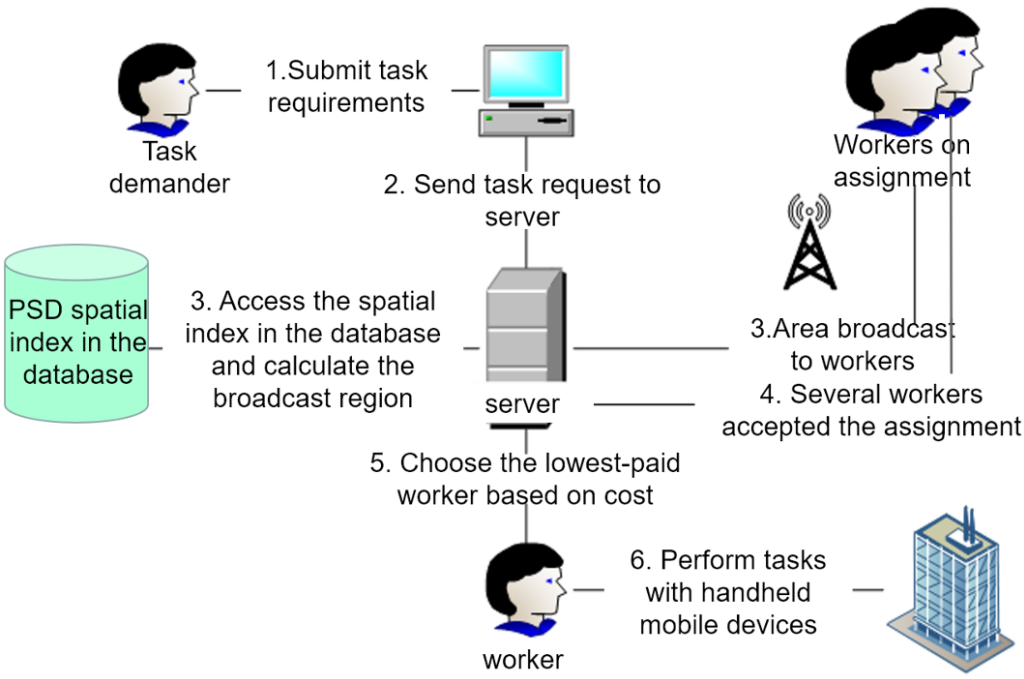

Figure 1. Mobile acquisition task allocation process based on differential privacy protection for mobile crowdsensing.

3.1. Dynamic Spatial Meshing

"Dynamic spatial meshing" refers to a technique utilized across various fields that improves simulation accuracy and efficiency by dynamically adjusting grid density to accommodate areas with complex or rapidly changing phenomena. The dynamic adaptive

grid index planning used in this paper was established in Reference [35]. As illustrated in Figure 2, it involves fixed first- and second-level privacy budget allocation strategies. The strategy calculates the granularity of the first-level grid based on the number of workers. It combines the privacy budget of the first-level grid with the Laplace mechanism to introduce noise into the nodes of the first-level grid. Subsequently, the query data of each grid in the first level are utilized to calculate the granularity of the second-level grid. This process involves establishing the second-level grid and adding noise twice in the second-level grid.

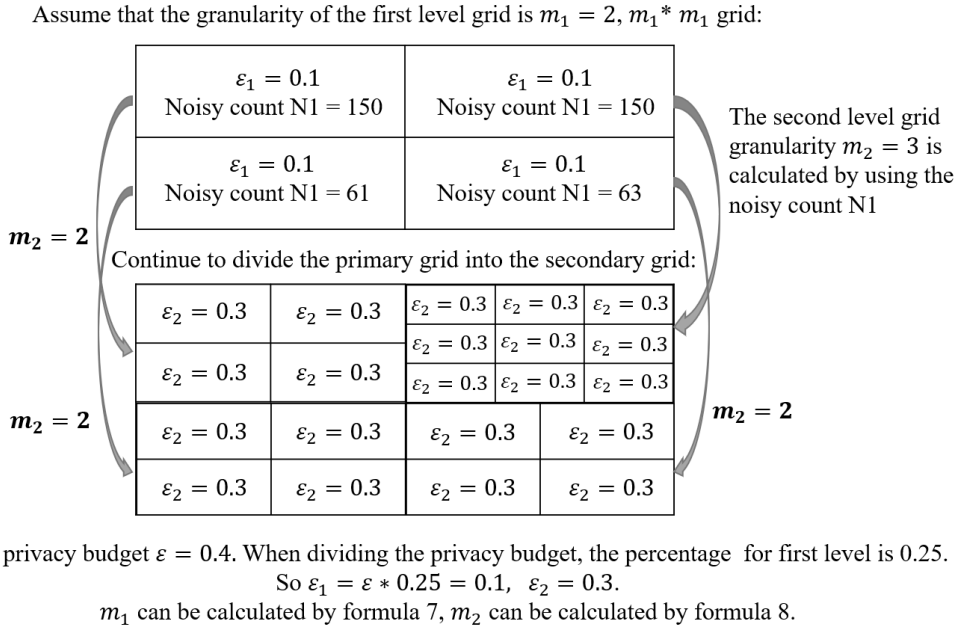

Figure 2. Dynamic spatial meshing strategy and meshing process.

3.2. System Model

This paper proposes a personalized strategy for user data privacy protection during the process of mobile data collection, which builds upon the dynamic division of plane space and the construction of a private space decomposition index proposed by previous researchers. Considering the diverse privacy protection needs of different workers, we introduce varying Laplace noises into leaf grid nodes while ensuring that tasks are allocated in the most cost-effective manner. In the simulation experiment, this paper adds noise to the node, queries the summarized data result within a specific range, compares it with the actual result and then assesses the level of privacy protection capability. The task allocation framework for users with different protection levels is evaluated using parameters such as task propagation time, completion percentage and forwarding distance. Experimental data demonstrate that the personalized privacy protection strategy proposed in this algorithm is more effective in the data collection process and performs better in task allocation results.

3.3. Framework Overview

We design a personalized privacy protection framework for mobile crowdsensing, as shown in Figure 1. The framework is mainly divided into three parts:

1. The platform provides differential privacy protection according to the location of workers, completes task allocation and ensures that the data are not excessively distorted.
2. According to the private space decomposition (PSD) broadcast area selection algorithm, the appropriate task broadcast range is calculated for task release and the task is accepted under the purpose probability.

3. Design cost reward mechanism and select appropriate workers to complete the task. Figure 1 describes the workflow of the whole platform when a task needs to be published.

In the process of crowdsensing, the task requester first submits the task requirements, and then the platform broadcasts the task to workers. Workers choose to accept the task, and the platform selects among them based on cost or optimal performance. The selected worker with a handheld mobile device performs sensing and returns the sensed data to the platform. During this process, workers need to upload their location and other privacy information to the platform, which may pose a risk of privacy leakage. Our model includes three parts: personalized private space decomposition framework, task broadcast strategy and task allocation algorithm based on worker compensation. These parts correspond to step 4 (workers accept the assignment), step 3 (area broadcast to workers) and step 5 (choose the lowest-paid worker based on cost) in Figure 1, respectively. Our model protects user privacy during the crowdsensing process to avoid privacy leakage. Details of these parts are introduced in Section 4.

4. Personalized Privacy Protection

In this section, we introduce the three parts of our personalized privacy protection framework in detail. We first design the personalized private space decomposition framework. Based on that, we further design the task broadcast strategy. Finally, the task allocation algorithm based on worker compensation is presented.

4.1. Personalized Private Space Decomposition Framework Design

4.1.1. PSD

When constructing a worker index in the database, we operate under the following assumptions: Firstly, the workers' residential places represent their current locations. Secondly, the platform possesses latitude and longitude coordinate data of workers. The third assumption is that the workers' residential places are distributed within a known area. In the simulation experiment, the residential location can serve as a substitute for real-time location data, facilitating comparisons with the spatial index privacy protection algorithm of other data structures. This section introduces the spatial index frame design for the spatial location of workers. The dynamic division of the spatial grid employs the dynamic spatial meshing strategy outlined in Section 3.1.

It is worth noting that the data privacy budget of each grid is the same when the above algorithm dynamically divides the spatial grid, so it cannot meet personalized needs. This paper makes some improvements on this basis.

4.1.2. Personalized Privacy Budget Allocation

When establishing the dynamic meshing results, using the conclusions in Reference [30], we will reduce the total budget ε. It is divided into two parts. The first part ε_1 is the privacy budget required to add noise to the node when establishing the first-level grid node. The second part $\varepsilon_2 = \varepsilon - \varepsilon_1$ will be used to calculate the noisy count of the second-level grid nodes. And the calculation formula of ε_1 is as follows:

$$\varepsilon_1 = \varepsilon \times \alpha \tag{6}$$

α is the percentage of the total budget divided when dividing the first-level grid ($0 < \alpha < 1$). According to the literature [30], the parameter α is not critical for privacy protection results. When the range of α is from 0.2 to 0.6, the noisy data are roughly similar to the original data. Therefore, in order to achieve a personalized level of privacy protection, we will set α to a random number from 0.3 to 0.6.

4.1.3. Spatial Grid Division Strategy

When constructing a spatial index, it is necessary to organize each index into a grid containing a limited number of user nodes or a grid covering a small area, with each leaf

node forming part of the subsequent tree structure. Drawing from index categories in the database, various forms of tree structure decomposition exist during spatial division:

No data decomposition required. The tree structure after division is precisely defined, and the division's outcome relies solely on the definition of the tree structure. For instance, the quadtree is established by recursively dividing each node's region into four subregions of equal area [36].

Data decomposition required. This decomposition relies on the internal data of the node. For example, Hilbert R-tree: R-tree is a spatial decomposition formed by nested rectangles that may overlap.

Hybrid tree. The definition of a hybrid tree is that the establishment of some level nodes needs data support and the division of the remaining level nodes does not need data support. The construction process of a hybrid tree needs to be designed in combination with a specific framework.

When providing differential privacy protection for nodes in space, some researchers have adopted the above tree structure [34,37]. It is common to use recursive partitioning to establish quadtree and k-d tree. Although effective in maintaining differential privacy protection, applying one-dimensional hierarchical methods to two-dimensional datasets often results in deeper trees. When dealing with massive data, the construction of spatial indexes can extend to more than ten levels, with grid divisions often operating at a scale level.

Dynamic spatial meshing strategy AG: Reference [35] proposed an adaptive grid (AG) method. This method aims to divide areas with concentrated worker data into more subareas and areas with sparse worker data into fewer subareas. In other words, AG addresses the drawback of the unified grid (UG) by calculating the sizes of second-level grid cells at different levels based on the data of the upper-level node.

The PSD algorithm in AG is introduced as follows: Firstly, we establish the root node of the mesh. Subsequently, we partition the root node into a grid of fixed size $m_1 \times m_1$. To introduce variability into the number of users in each grid cell, we add noise to the number of users in each grid cell by using Formula (5). The author of Reference [30] proposed the calculation formula for determining the granularity of the first-level grid node in AG, which is presented in Equation (7):

$$m_1 = \max\{10, \left\lceil \frac{1}{4}\sqrt{\frac{N \times \varepsilon}{c}} \right\rceil\} \tag{7}$$

where N represents the number of data points on the workers' location grid. The authors of Reference [38] presented a substantial number of data. According to the experimental results, it was demonstrated that setting c = 10 yields improved division results.

Subsequently, AG retrieves the count of workers in the first-level grids. With the number of these grids being $m_1 \times m_1$, we calculate the size of the second-level grid according to the number of query results. Each coarse-grained region is then adaptively subdivided into $m_2 \times m_2$ finer-grained areas. The calculation formula of m_2 is as follows:

$$m_2 = \left\lceil \sqrt{\frac{N' \times \varepsilon \times (1-\alpha)}{c_2}} \right\rceil \tag{8}$$

To increase the granularity m_2 and reduce the overhead, the author of Reference [35] incorporated an additional parameter c_2 into the heuristic algorithm, setting it to $c_2 = \sqrt{2}$. During the initialization phase, the particle size is set to $m_2 \times m_2$, and noise is added to the number of fine-grained grid cells using Formula (5). Subsequently, the perturbed count of the corresponding fine-grained mesh is published alongside the structure of AG. Based on Formula (1), the counting query algorithm of the AG model satisfies the requirements of ε-differential privacy.

Personalized dynamic spatial meshing strategy PAG: For personalization, we set the privacy budget percentage α to a random value. Refer to Algorithm 1 for the specific

algorithm. Table 1 presents a description of the special symbols utilized in Algorithm 1, while Figure 3 serves as an aid for comprehending the flowchart of Algorithm 1.

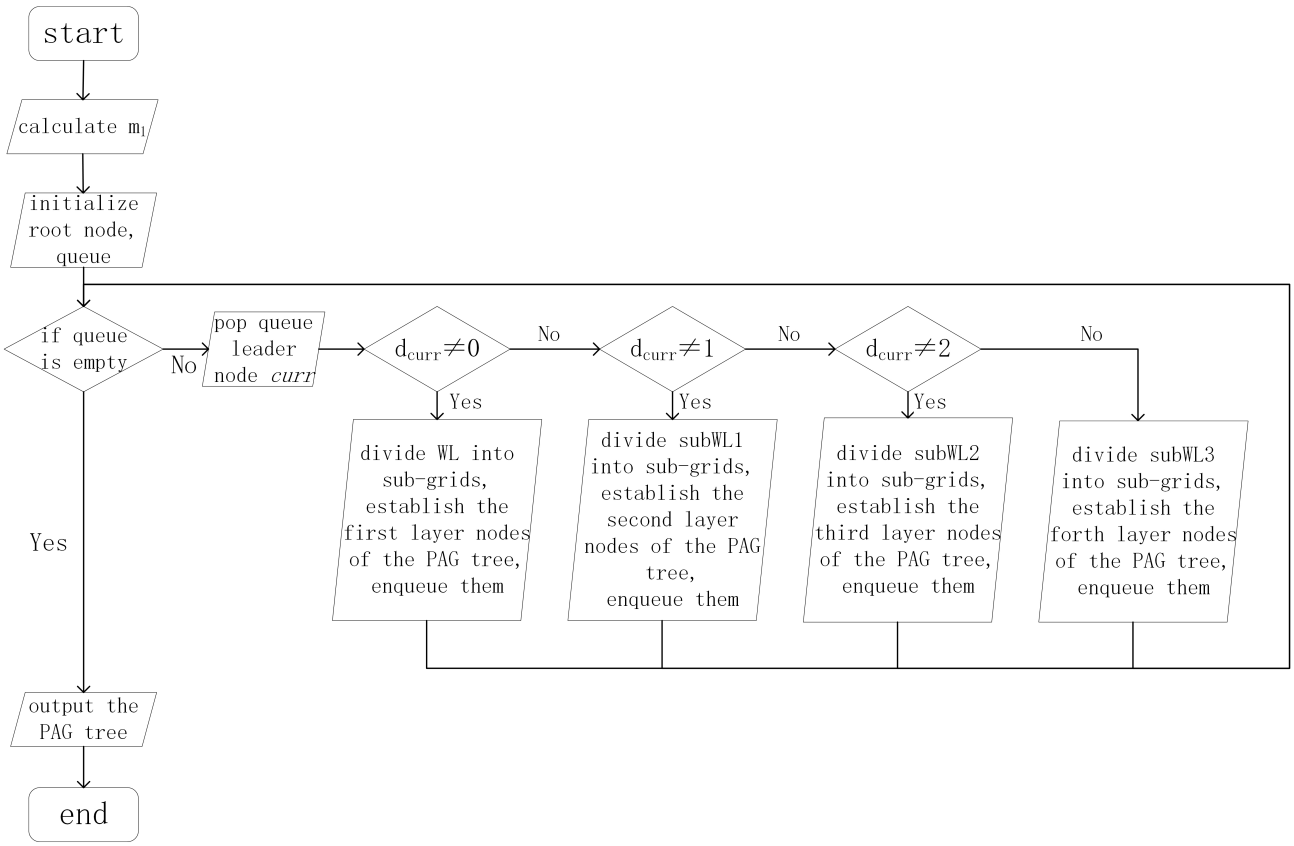

Figure 3. The flowchart of Algorithm 1.

Table 1. Symbols required for Algorithm 1.

Symbol	Definition
queue	auxiliary queue
curr	current team head node
d_{curr}	depth of node *curr* in PAG tree
$subWL1_{currId}$	subgrid corresponding to node curr
$\varepsilon_{1,node_i}$	personalized privacy budget of the first-level grid node i
$\varepsilon_{2,node_i}$	remaining privacy budget

According to References [34,35], we understand that the Laplace mechanism can introduce noise to the spatial index of the dataset, ensuring that the output of the counting query function satisfies the requirements of ε-differential privacy.

4.2. Task Broadcast Strategy Design

Compared to tasks assigned directly by the platform, workers are more willing to accept the broadcast task, so the platform needs to find a reasonable broadcast area. In essence, the task broadcasting strategy must access the spatial indexes of workers and calculate a designated broadcast grid area centered around the task location. The decisive condition of this area is that the task can be accepted by the staff in the platform when the probability of completion can be achieved. As users generally prefer tasks closer to

their current location for convenience [38,39], their acceptance rates vary depending on task proximity. The acceptance rate is a function that decreases with forward distance. In this paper, the linear decline function is selected. This chapter adopts the broadcast area algorithm described in Reference [35]. This paper finds the secondary grid node where the task is located and continuously adds the neighbor nodes with workers. When there is a certain probability that the task can be completed, it ends the expansion of the broadcast area.

Algorithm 1 PAG

Input: Grid of workers' location (WL)
Output: Grid of workers' location (WL)

calculate m_1
Initialize the root node, initialize the queue and queue the root node
if queue is empty:
  output the PAG tree
else:
  pop queue leader node curr
  If $d_{curr} = 0$:
    divide WL into $subWL1_1, subWL1_2, \ldots, subWL1_{m1}$
    establish first-layer nodes of PAG tree
    put first-layer nodes into queue
    randomly generated budget percentage α
    calculate $\varepsilon_{1,node_i}$ and remaining privacy budget $\varepsilon_{2,node_i}$
    execute step 3
  else:
    If $d_{curr} = 1$:
      divide $subWL1_{currId}$ into $subWL2_1, subWL2_2, \ldots, subWL2_{m1}$
      establish second-layer nodes of PAG tree
      add second-layer nodes to Laplace random noise count
      put second-layer nodes into queue
      execute step 3
    else:
      If $d_{curr} = 2$:
        calculate m_1
        divide $subWL2_{currId}$ into $subWL3_1, subWL3_2, \ldots, subWL3_{m2}$
        establish third-layer nodes of PAG tree
        add third-layer nodes to Laplace random noise count
        put third-layer nodes into queue
        execute step 3
      else:
        divide $subWL3_{currId}$ into $subWL4_1, subW4_2, \ldots, subWL4_{m2}$
        establish fourth-layer nodes of PAG tree
        add fourth-layer nodes to Laplace random noise count
        put fourth-layer nodes into queue
        execute step 3

According to the literature [39], approximately 10% of the platform's workforce is responsible for more than 80% of the tasks; these individuals are referred to as super workers. Moreover, within the group of super workers, around 90% of them travel less than 40 miles a day. This attribute has been labeled as task locality in the literature [37]. Additionally, prior research [40] addressed the issue of content locality among users of platforms such as Flickr and Wikipedia, proposing the spatial content generation model

(SCPM). The model calculates the average contribution distance (*MCD*) of each worker as follows:

$$MCD(w_i) = \sum_{i=1}^{n} \frac{d(w_i, t_j)}{n} \tag{9}$$

where $d(w_i, t_j)$ represents the distance between worker w_i and task t_j. As described in the literature [35], the formula for calculating the maximum travel distance (*MTD*) that workers are willing to accept for a task is as follows:

$$MTD = 90\% \times MCD \tag{10}$$

When the distance between the worker and the task exceeds the *MTD*, the worker will not choose the task. Here, MAR denotes the maximum probability of a worker accepting the task. Additionally, *AR* represents the probability that the staff will accept and complete the task. *AR* is calculated as follows:

$$AR = \max\{0, (1 - \frac{dist}{MTD}) \times MAR\} \tag{11}$$

Here, *dist* represents the distance between the worker and the task. We establish a threshold, *EU*. If the probability of completing the broadcast area surpasses this threshold, it implies that at least one worker can accept the task. Assuming uniform *AR* among workers within a single secondary grid, the distance between workers and tasks is the average distance between tasks and the four corners of the secondary grid.

n_c represents the noise count of workers within grid c. p_c represents the task acceptance rate of workers in grid c. The utility of grid c is calculated as follows:

$$U_c = 1 - (1 - p_c)^{n_c} \tag{12}$$

Next, two calculation schemes of broadcast area will be introduced.

Broadcast area calculation without considering the secondary grid division: Begin with the grid containing task t as the initial broadcast area. If the current broadcast area's utility is lower than *EU*, the nearest neighbor to the task within the maximum travel range will be added to the broadcast area. The specific flow of the algorithm is outlined in Algorithm 2, while Table 2 provides the interpretation of the symbols used in Algorithm 2.

Table 2. Symbols required for Algorithm 2.

Symbols	Definition
t	task
GR	broadcast area
Q	maximum heap of the second-level cells covering t
c_i	the second-level grid node
neighbours	space inner grid $c_i^{'}s$ neighbor

Broadcast area calculation considering two-level grid continuous division: In order to reduce the cost as much as possible, Algorithm 2 can divide the final secondary grid and find the adjacent subregions that can reach the threshold. The following are the improvement steps of Algorithm 2 and special symbol notes.

4.3. Task Allocation Algorithm Based on Worker Compensation

When facing the task allocation mechanism based on a dynamic grid, there are different task allocation methods according to different considerations of the decision-making mechanism. For example, the task is assigned to the workers who accept the task first (time first), the

task is assigned to the workers closest to the task (location first), and the task proposed in this paper is assigned to the workers with the lowest salary (salary first). In order to improve the speed of task acceptance, the first accept first assign model is used to assign the task to the staff with the earliest response time among all response staff. In order to reduce the travel distance of the staff, the nearest first assignment model is used to assign the task to the staff member closest to the task among all the responding staff. This paper chooses to assign tasks to the workers with the lowest salary in the unit cost, that is, the lowest reward-first allocation algorithm.

Algorithm 2 GDY

Input: Task t, utility threshold EU, maximum travel area MTD
Output: Broadcast area GR

1: Initialization GR is null, u = 0
2: Initialize the maximum heap Q, where q is the second-level grid covering t
3: Pop $\{ c_i, U_{c_i}\}$ from stack Q
4: If c_i = null then output broadcast area GR, where GR is greater than MTD. Otherwise $GR = GR \cup c_i$
5: If $U_{c_i} \geq 0$ then $U = 1 - (1 - U)(1 - U_{c_i})$
6: If $U \geq EU$ then output broadcast area GR. Otherwise look for neighbors of C not in $GR \cap MTD$
7: Add neighbors to Q and execute step 3

When designing the remuneration, the worker $w_i^{'}s$ privacy budget and the distance between w_i and task t $d(w_i, t)$ are taken into account. Reward $cost[i,0]$ is a linear combination of privacy budget and distance. The calculation formula is shown in Equation (13):

$$cost[i, 0] = 50 \times \frac{\varepsilon_2}{\varepsilon} + 50 \times d(w_i, t) \tag{13}$$

For personal reasons (such as completing a task), even if some workers have a high probability, they may not complete the task. Therefore, it is necessary to simulate whether workers complete the task by using a random function. The greedy algorithm is used to find the current lowest paid worker, and Formula (11) is used to calculate the probability AR that the worker accepts the task, and the random function is called to simulate the probability that the worker actually accepts the task. If the random number is greater than AR, the task is accepted and the assignment ends. Otherwise, continue to find the next worker until there is a worker to accept or no workers.

5. Experiments and Analyses

The personalized differential privacy protection strategy proposed in this paper adds complexity to task assignment. Introducing random Laplace mechanism noise can affect the efficiency and accuracy of task assignment to workers within the task assignment model. Hence, we aim to assess the effectiveness of the personalized privacy policy (PAG) proposed in this paper in safeguarding data against discrepancies between noisy and real data.

5.1. Experimental Settings

In our experiments, we utilized two widely used real-world datasets, Gowalla and Yelp, as the core foundations. Gowalla, a social network dataset, provides genuine location check-in data from users of the Gowalla mobile application, recording their activities at various global locations, including timestamps and user profiles. Within our model, we consider the users in Gowalla as workers, each check-in point being a previously accepted and executed task.

On the other hand, Yelp offers an extensive compilation of business- and user-related information sourced from the Yelp website. It encompasses intricate business data such as geographical coordinates, operating hours and user ratings, alongside user-generated

reviews. With data points comprising 15,583 restaurants, 70,817 users and 335,022 reviews, we regarded restaurant locations as tasks and user reviews as accepted tasks.

Our experiments were conducted on a computer equipped with an Intel i7 CPU and an NVIDIA 3090 GPU. Each experiment was repeated multiple times to derive the average value as the final result.

5.2. Performance Evaluation Criteria

In the context of the task allocation mechanism, we emphasize the following indicators:

- *averageErr*: Represents the average of the average difference between the actual query count and the noisy query count results. A smaller difference indicates better algorithm performance.
- n_{hops}: Denoting the number of broadcast hops in the platform's task dissemination region. A smaller number of hops suggests faster task completion.
- *ANW*: Indicating the average number of workers required for each task broadcast. A smaller workforce implies higher quality worker selection.
- *TASR*: Representing the acceptance rate by staff once the task is released on the platform. A higher ratio signifies more efficient worker selection.
- *WTD*: Signifying the average distance that staff members need to travel in order to complete their assigned tasks. A smaller distance suggests better suitability of the selected workers for the tasks.

5.3. Experimental Results

In this paper, the simulation experiment will utilize datasets from the literature [35]. The specific results of the simulation experiment evaluation are as follows:

5.3.1. Evaluation of Privacy Protection Performance of Personalized PSD Spatial Index

Based on the findings presented in Reference [35], the grid structure demonstrates superior performance compared to other spatial index data structures in terms of differential privacy protection. Consequently, we solely consider one spatial index data structure for comparison. Specifically, our experiments are conducted on four spatial indexes: k-d tree, UG unified grid, AG adaptive grid and PAG personalized adaptive grid. Each of these algorithms is utilized to identify different spatial indexes. Furthermore, the spatial indexes with added noise are repeatedly counted within a specific range, allowing us to calculate the disparity between the actual data count and the noise-infused count for the same query. A smaller variance between the actual count and the noise count signifies a stronger data protection effect.

Given that our partition percentage is a randomly generated number during the actual internal division of the PAG grid, it becomes apparent that most workers allocate a significant portion of their privacy budget to the division of the first-level grid, with a comparatively smaller share allocated to the second-level grid nodes. Consequently, we set the privacy budget for k-d, UG and AG at 0.4, with a percentage of 0.5, and the privacy budget in PAG at 0.6. To ensure the accuracy and effectiveness of the experimental data, we conducted separate queries across various spatial indexes under different total privacy budgets. The average results from multiple seeds were calculated, followed by the computation of the average discrepancy between the query results of different spatial indexes under different total privacy budgets and the real data. Figure 4 illustrates the experimental test data for different total privacy budgets in a line graph, while Table 3 presents the average experimental values for different privacy budgets, with averageErr representing the average of the average difference between the actual query count and the noisy query count results.

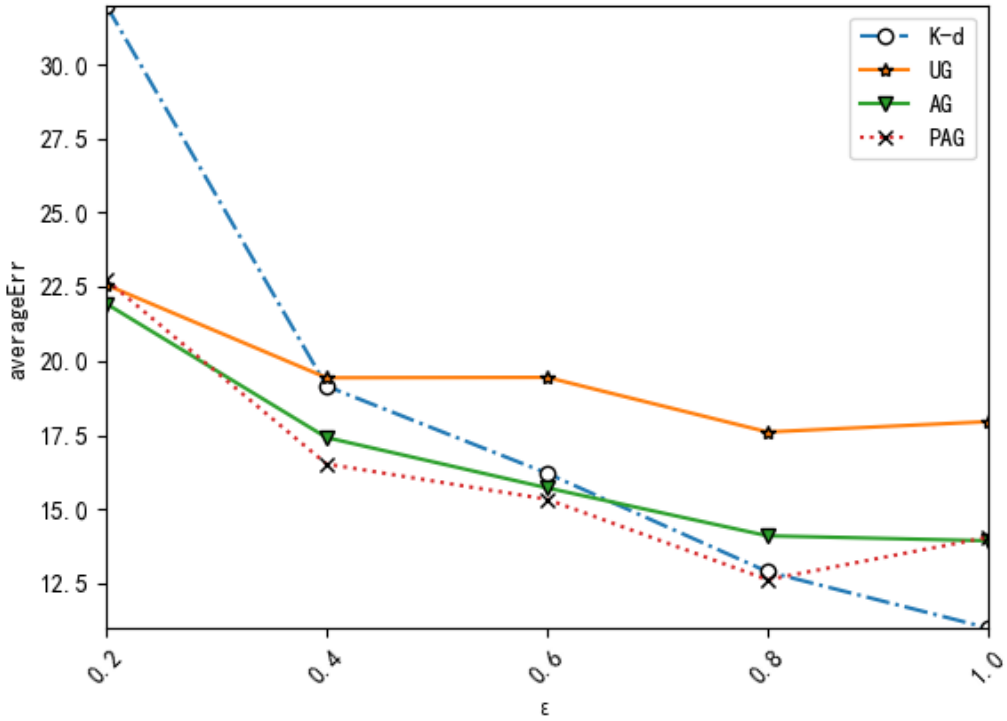

Figure 4. Values of averageErr under different spatial partition strategies.

Table 3. Average value of spatial index experimental data gap under different privacy budgets.

Data/Space Structure	k-d	UG	AG	PAG
averageErr	24.24	20.34	17.65	18.02

5.3.2. Impact of Personalized Grid Structure on System Performance

We conducted a comparison of the communication cost between the PAG structure and the AG structure in task broadcasting. Using a set of 100 task coordinates, along with the AG spatial index and the PAG spatial index, we generated the broadcast areas GR_{AG} and GR_{PAG} according to Algorithm 3, while Table 4 provides the interpretation of the symbols used in Algorithm 3. Subsequently, we initiated broadcasting within the designated broadcast areas. Following this, we calculated the average number of notified workers (ANW) and the number of propagation hops (n_{hops}) for the task broadcast. The specific results are presented in Tables 5 and 6.

Table 4. Symbols required for Algorithm 3.

Symbol	Definition
c_i	the last unit with excessive utility in the broadcast area
$c_{i'}$	subunit of c_i obtained according to Algorithm 3
$w_{required}$	the noise count of the subunit is required
w_{c_i}	noise count of c_i
percentile	percentage of subarea in the secondary grid

Table 5. ANW in broadcast areas under different privacy budgets.

ε	GR_{AG}	GR_{PAG}
0.1	46	270
0.4	30	134
0.7	30	111
1.0	27	92

Table 6. n_{hops} in broadcast areas under different privacy budgets.

ε	GR_{AG}	GR_{PAG}
0.1	6	14
0.4	4	11
0.7	4	9
1.0	4	8

Algorithm 3 Subunit division algorithm of the secondary grid in broadcast area *PCSH*

Input: Task location t, the last cell c_i, the probability of task acceptance in the current grid U_{curr}

Output: $c_{i'}$ (subunit of c_i)

1: Calculate dist (the distance between c_i and t) and ${p_{sub}}^a$ (the worker's probability)

2: $U_{required} = \dfrac{U - U_{curr}}{1 - U_{curr}}$ (probability required)

3: $w_{required} = \log_{1-U_{required}} 1 - {p_{sub}}^a$ (number of workers required)

4: Calculation of area percentile: $percentile = \dfrac{w_{required}}{w_{c_i}}$

5: If c_i covers t, then calculate the subunit C' of C-based percentile and output; otherwise, perform step 6

6: Find the subunit $c_{i'}$ of c_i adjacent to the current region and output it.

5.3.3. Impact of Minimum Reward Task Allocation Model on System Performance

Based on the simulation results of task broadcasting in Section 4 we can further analyze the impact of the minimum reward task allocation model on the platform system's performance. The primary indicators in this section are the success rate *TASR*, representing the acceptance rate by staff once the task is released on the platform, and the distance *WTD* that staff members need to travel in order to complete their assigned task. The specific experimental results are presented in Tables 7 and 8.

Table 7. *TASR* (%) of AG and PAG under different task allocation mechanisms.

ε	AG		PAG		
	Time First	**Location First**	**Time First**	**Location First**	**Salary First**
0.1	87	94	92	100	99
0.4	86	92	90	96	97
0.7	87	94	92	96	93
1.0	85	90	90	95	98

Table 8. *WTD* (km) of AG and PAG under different task allocation mechanisms.

ε	AG		PAG		
	Time First	**Location First**	**Time First**	**Location First**	**Salary First**
0.1	0.3561	0.2112	0.7456	0.2674	0.1790
0.4	0.3249	0.2064	0.5458	0.1891	0.1945
0.7	0.3130	0.2229	0.4871	0.2434	0.1683
1.0	0.3224	0.2322	0.4206	0.1649	0.2095

6. Discussion

Our method not only ensures that user privacy is not leaked but also considers the impact on system performance of completing the task as efficiently as possible. In this section, we will analyze our method from the three aspects of privacy protection performance, impact of personalized grid structure on system performance and impact of minimum reward task allocation model on system performance.

6.1. Evaluation of Privacy Protection Performance of Personalized PSD Spatial Index

Through the analysis of Figure 4, it can be concluded that when the total privacy budget remains constant, the adaptive grid demonstrates superior data protection capabilities compared to other tree structures. In scenarios with a small privacy budget, the AG slightly outperforms the PAG grid. However, as the privacy budget increases, the AG no longer exhibits its initial advantages, and at times, the PAG even outperforms the AG. This result can be attributed to the fixed allocation of the second-level grid's budget in the AG, which consistently accounts for half of the total privacy budget, while the PAG's allocation is not fixed. Consequently, the privacy budget percentage for the PAG's second-level grid may lean towards being less than half, leading to significant noise addition in the second-level grid of PAG and potentially compromising the data's authenticity. Nevertheless, Table 3 indicates that, without considering the size of the privacy budget, the overall difference in data protection capabilities between the PAG algorithm proposed in this paper and the AG algorithm is not pronounced. Furthermore, the PAG, which facilitates personalized privacy protection levels for workers, marks an improvement over the AG.

6.2. Impact of Personalized Grid Structure on System Performance

Tables 5 and 6 indicate that the disadvantage of this algorithm is the high communication cost of the PAG when using Algorithm 3 to calculate the task broadcast area GR_{PAG}. However, Figure 4 and Table 7 demonstrate that the PAG algorithm achieves enhanced privacy protection results and a higher task completion rate, even at the expense of increased communication cost. Therefore, the reduction in salary cost can make up for the increase in broadcast communication cost.

6.3. Impact of Minimum Reward Task Allocation Model on System Performance

Table 7 indicates that the task allocation success rate of the PAG significantly outperforms that of the AG algorithm. Furthermore, after selecting the PAG to establish a spatial index, Table 8 demonstrates that the time priority task allocation mechanism surpasses both the location priority mechanism and the reward mechanism in terms of the average travel distance required to complete tasks. This not only affects the time to complete the task but also increases the cost of the platform. Consequently, for data collection tasks, the PAG spatial index algorithm proposed in this paper, coupled with the reward-first task allocation mechanism, can efficiently realize cost-effective data collection on the platform.

The results indicate that the personalized adaptive grid (PAG) exhibits superior performance in terms of privacy protection and task allocation success rates compared to other spatial index structures, particularly when the privacy budget is substantial. However, the communication costs associated with the PAG are relatively higher, albeit justifiable

considering the enhanced privacy protection and task completion rates. Moreover, the use of a reward-first task allocation mechanism further optimizes data collection processes on the platform, leading to cost-effective and efficient data collection. The findings underscore the importance of considering privacy protection and task allocation strategies in designing efficient spatial index structures for data collection platforms.

7. Conclusions

During the process of mobile data collection, the protection of data collectors' location data has become an increasingly critical concern. To prevent exposing real staff location information to potential attackers, certain documents, such as [41–44], have employed differential privacy technology to obfuscate location information on the local client before uploading the perturbed data to the platform. Building upon this prior work, this paper presents implementation algorithms for different privacy protection levels within specific grid cells in the second-level grid. Finally, the task allocation strategy of minimum reward allocation is proposed. We compare the algorithm proposed in this paper with the algorithm proposed by the author of Reference [35] through experimental simulation. The purpose of the comparison is to analyze the effectiveness and practicability of the personalized privacy protection algorithm proposed in this paper in data protection. By analyzing the results, we can conclude that the task allocation model proposed in this paper has a positive impact on improving performance indicators such as the task completion rate in the task allocation process.

While the algorithm proposed in this paper yields favorable results in terms of data privacy protection and task allocation, it is not without its limitations. One prominent issue is that the algorithm sacrifices a portion of the communication cost for lower remuneration and a flexible privacy protection level. Our forthcoming area of improvement is to find a more suitable algorithm that minimizes the number of propagation hops during communication, consequently reducing the cost associated with it.

Author Contributions: Conceptualization, H.G. and H.Z.; methodology, H.G. and Z.Z.; formal analysis, H.G. and Z.Z.; investigation, H.G. and Z.Z.; project administration, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ganti, R.K.; Ye, F.; Lei, H. Mobile crowdsensing: Current state and future challenges. *IEEE Commun. Mag.* **2011**, *49*, 32–39. [CrossRef]
2. Capponi, A.; Fiandrino, C.; Kantarci, B.; Foschini, L.; Kliazovich, D.; Bouvry, P. A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 2419–2465. [CrossRef]
3. Zhang, D.; Wang, L.; Xiong, H.; Guo, B. 4w1h in mobile crowd sensing. *IEEE Commun. Mag.* **2014**, *52*, 42–48. [CrossRef]
4. Zhang, Y.; Li, M.; Yang, D.; Tang, J.; Xue, G.; Xu, J. Tradeoff between location quality and privacy in crowdsensing: An optimization perspective. *IEEE Internet Things J.* **2020**, *7*, 3535–3544. [CrossRef]
5. Kulshrestha, T.; Saxena, D.; Niyogi, R.; Cao, J. Real-time crowd monitoring using seamless indoor-outdoor localization. *IEEE Trans. Mob. Comput.* **2020**, *19*, 664–679. [CrossRef]
6. Ni, J.; Zhang, K.; Xia, Q.; Lin, X.; Shen, X. Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2019**, *19*, 1317–1331. [CrossRef]
7. Lane, N.D.; Miluzzo, E.; Hong, L.; Peebles, D.; Choudhury, T.; Campbell, A.T. A survey of mobile phone sensing. *IEEE Commun. Mag.* **2010**, *48*, 140–150. [CrossRef]
8. Liu, W.; Yang, Y.; Wang, E.; Wu, J. Dynamic User Recruitment with Truthful Pricing for Mobile CrowdSensing. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 1113–1122.

9. Liu, W.; Yang, Y.; Wang, E.; Wang, H.; Wang, Z.; Wu, J. Dynamic online user recruitment with (non-) submodular utility in mobile crowdsensing. *IEEE/ACM Trans. Netw.* **2021**, *29*, 2156–2169. [CrossRef]
10. Wang, H.; Wang, E.; Yang, Y.; Wu, J.; Dressler, F. Privacy-preserving online task assignment in spatial crowdsourcing: A graph-based approach. In Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer Communications, London, UK, 2–5 May 2022; pp. 570–579.
11. Wu, H.; Wang, L.; Xue, G.; Tang, J.; Yang, D. Enabling data trustworthiness and user privacy in mobile crowdsensing. *IEEE/ACM Trans. Netw.* **2019**, *27*, 2294–2307. [CrossRef]
12. Liu, Y.; Feng, T.; Peng, M.; Jiang, Z.; Xu, Z.; Guan, J.; Yao, S. COMP: Online control mechanism for profit maximization in privacy-preserving crowdsensing. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1614–1628. [CrossRef]
13. Li, L.; Shi, D.; Zhang, X.; Hou, R.; Yue, H.; Li, H.; Pan, M. Privacy preserving participant recruitment for coverage maximization in location aware mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2021**, *21*, 3250–3262. [CrossRef]
14. An, J.; Wang, Z.; He, X.; Gui, X.; Cheng, J.; Gui, R. PPQC: A blockchain-based privacy-preserving quality control mechanism in crowdsensing applications. *IEEE/ACM Trans. Netw.* **2022**, *30*, 1352–1367. [CrossRef]
15. Beresford, A.R.; Stajano, F. Location privacy in pervasive computing. *IEEE Pervasive Comput.* **2003**, *2*, 46–55. [CrossRef]
16. Xiao, X.; Tao, Y. Personalized privacy preservation. In Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, Chicago, IL, USA, 27–29 June 2006; pp. 229–240.
17. Ardagna, C.A.; Cremonini, M.; Damiani, E.; De Capitani di Vimercati, S.; Samarati, P. *Location Privacy Protection through Obfuscation-Based Techniques*; Springer: Berlin/Heidelberg, Germany, 2007.
18. Cornelius, C.; Kapadia, A.; Kotz, D.; Peebles, D.; Shin, M.; Triandopoulos, N. Anonysense: Privacy-aware people-centric sensing. In Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, Breckenridge, CO, USA, 17–20 June 2008; pp. 211–224.
19. Huang, K.L.; Kanhere, S.S.; Hu, W. A privacy-preserving reputation system for participatory sensing. In Proceedings of the 37th Annual IEEE Conference on Local Computer Networks, Clearwater Beach, FL, USA, 22–25 October 2012; pp. 10–18.
20. Dimitriou, T.; Krontiris, I.; Sabouri, A. PEPPeR: A querier's privacy enhancing protocol for participatory sensing. In Proceedings of the Security and Privacy in Mobile Information and Communication Systems: 4th International Conference, Frankfurt am Main, Germany, 25–26 June 2012; pp. 93–106.
21. Qiu, F.; Wu, F.; Chen, G. Privacy and quality preserving multimedia data aggregation for participatory sensing systems. *IEEE Trans. Mob. Comput.* **2014**, *14*, 1287–1300. [CrossRef]
22. Samarati, P.; Sweeney, L. Protecting Privacy When Disclosing Information: K-Anonymity and Its Enforcement through Generalization and Suppression. 1998. Available online: https://www.scirp.org/(S(lz5mqp453edsnp55rrgjct55.))/reference/referencespapers.aspx?referenceid=776999 (accessed on 19 November 2023).
23. Sweeney, L. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **2002**, *10*, 557–570. [CrossRef]
24. Abul, O.; Bonchi, F.; Nanni, M. Never walk alone: Uncertainty for anonymity in moving objects databases. In Proceedings of the IEEE International Conference on Data Engineering, Cancun, Mexico, 7–12 April 2008.
25. Gedik, B.; Ling, L. Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. *IEEE Trans. Mob. Comput.* **2007**, *7*, 1–18. [CrossRef]
26. Ren, J.; Zhang, Y.; Zhang, K.; Shen, X.S. SACRM: Social aware crowdsourcing with reputation management in mobile sensing. *Comput. Commun.* **2015**, *65*, 55–65. [CrossRef]
27. Mousa, H.; Mokhtar, S.B.; Hasan, O.; Younes, O.; Hadhoud, M.; Brunie, L. Trust management and reputation systems in mobile participatory sensing applications: A survey. *Comput. Netw.* **2015**, *90*, 49–73. [CrossRef]
28. Christin, D.; Roßkopf, C.; Hollick, M.; Martucci, L.A.; Kanhere, S.S. IncogniSense: An anonymity-preserving reputation framework for participatory sensing applications. *Pervasive Mob. Comput.* **2013**, *9*, 353–371. [CrossRef]
29. Dwork, C. Differential Privacy. In Proceedings of the 33rd international conference on Automata, Languages and Programming—Volume Part II, Venice, Italy, 10–14 July 2006.
30. Qardaji, W.; Yang, W.; Li, N. Differentially private grids for geospatial data. In Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE), Brisbane, Australia, 8–12 April 2013; IEEE: Piscataway, NJ, USA, 2013.
31. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput.* **2013**, *9*, 211–407. [CrossRef]
32. Dwork, C.; Naor, M.; Pitassi, T.; Rothblum, G.N. Differential privacy under continual observation. In Proceedings of the ACM Symposium on Theory of Computing, Cambridge, MA, USA, 5–8 June 2010.
33. Samet, H. *The Design and Analysis of Spatial Data Structures*; Addison-Wesley: Reading, MA, USA, 1990; Volume 85.
34. Cormode, G.; Procopiuc, M.; Shen, E.; Srivastava, D.; Yu, T. Differentially private spatial decompositions. In Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, Arlington, VA, USA, 1–5 April 2012.
35. To, H.; Ghinita, G.; Shahabi, C. A framework for protecting worker location privacy in spatial crowdsourcing. *Proc. Vldb Endow.* **2014**, *7*, 919–930. [CrossRef]
36. Berg, M.; Cheong, O.; Kreveld, M.; Overmars, M. Computational geometry: Algorithms and applications. *Math. Gaz.* **2000**, *19*, 333–334.

37. Xiao, Y.; Xiong, L.; Yuan, C. Differentially private data release through multidimensional partitioning. In Proceedings of the Secure Data Management, 7th VLDB Workshop, SDM 2010, Singapore, 17 September 2010.
38. Alt, F.; Shirazi, A.S.; Schmidt, A.; Kramer, U.; Nawaz, Z. Location-based crowdsourcing: Extending crowdsourcing to the real world. In Proceedings of the Nordic Conference on Humancomputer Interaction, Reykjavik, Iceland, 16–20 October 2010.
39. Musthag, M.; Ganesan, D. Labor dynamics in a mobile micro-task market. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, 27 April–2 May 2013.
40. Hecht, B.J.; Gergle, D. On the "localness" of usergenerated content. In Proceedings of the ACM Conference on Computer Supported Cooperative Work, Savannah, GA, USA, 6–10 February 2010.
41. Wang, L.; Yang, D.; Xiao, H.; Wang, T.; Ma, X. Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017.
42. Miao, C.; Jiang, W.; Su, L.; Li, Y.; Guo, S.; Qin, Z.; Xiao, H.; Gao, J.; Ren, K. Privacy-Preserving Truth Discovery in Crowd Sensing Systems. *ACM Trans. Sens. Netw.* **2019**, *15*, 1–32. [CrossRef]
43. Wang, C.J.; Ku, W.S. Anonymous Sensory Data Collection Approach for Mobile Participatory Sensing. In Proceedings of the IEEE International Conference on Data Engineering Workshops, Arlington, VA, USA, 1–5 April 2012.
44. Zhang, N.; Li, M.; Lou, W. Distributed Data Mining with Differential Privacy. In Proceedings of the IEEE International Conference on Communications, Kyoto, Japan, 5–9 June 2011; pp. 1–5.