Journal of Network Intelligence
Taiwan Ubiquitous Information
©2022 ISSN 2414-8105 (Online)
Volume **7**, Number **2**, May **2022**

QPDPLP: A Novel Location Privacy Protection Method

Zihao Shen

School of Computer Science and Technology
Henan Polytechnic University
Jiaozuo, 454000, China
hpuxxfzyjs@qq.com

Mengke Liu

School of Computer Science and Technology
Henan Polytechnic University
Jiaozuo, 454000, China
804432001@qq.com

Hui Wang*

School of Computer Science and Technology
Henan Polytechnic University
Jiaozuo, 454000, China
*Corresponding Author:wanghui_jsj@foxmail.com

Peiqian Liu

School of Computer Science and Technology
Henan Polytechnic University
Jiaozuo, 454000, China
2362695089@qq.com

Kun Liu

School of Computer Science and Technology
Henan Polytechnic University
Jiaozuo, 454000, China

Fangfang Lian

School of Computer Science and Technology
Henan Polytechnic University
Jiaozuo, 454000, China

Received December 2021; revised March 2022

Abstract. *For location information publishing, its sparsity can seriously affect the effectiveness of traditional location differential privacy protection algorithms. In this paper, a user interest-based differential privacy location publishing algorithm combined with quadtree partition (QPDPLP) is proposed, which uses a mapping function to process perturbed locations outside the user's area of interest to reduce the amount of noise injection. First, the disturbance location of the user interest area is combined with the quadtree structure to divide. Second, the ratio of each node weight to the total weight is the basis for area division, and then according to the sparsity of the divided area position points to add noise to the area position points. Finally, using the real location and the processed disturbance location of the randomly selected location as a published location. Experimental results show that, given the same privacy budget, the QPDPLP algorithm is more effective than traditional algorithms in protecting users' location privacy.*
Keywords: Quadtree, Privacy Protection, Differential Privacy, User Interest Area

1. **Introduction.** In recent years, Location-based Service (LBS) has brought great convenience to people, but with the continuous development of the Internet, mobile terminals, and location technology, the number of users communicating through social networking platforms has increased rapidly, resulting in the accumulation of user behavior databases in social networks to expand, including the user's sensitive information, such as home address, ID number, etc. Once these sensitive information is obtained by criminals, it will pose a great threat to the interests of users [1,2]. For example, the March 2018 Facebook user data breach that influenced the U.S. election originated from a partnership between Facebook and Cambridge Analytica, which claimed that the company had created an app on Facebook for predicting users' personality preferences for academic research purposes, when in fact Cambridge Analytica had not only collected test results from users. The personal information of 50 million users on Facebook was collected in passing without permission. Cambridge Analytica, after acquiring these user data, built a user portrait without authorization, and statistically analyzed each user's interests, personality, and behavioral characteristics through data analysis to predict their political leanings, then targeted to push news to users, with the help of Facebook's advertising delivery system to influence users' voting behavior. As a result, Facebook has had an inescapable impact on the U.S. election. Therefore, how to make users use the social networking platform normally and at the same time ensure that sensitive information is not leaked is an urgent issue that needs to be addressed.

Most traditional location privacy protection techniques are based on k-anonymity [3–5], t-closeness [6], and l-diversity [7], but they ignore the influence of an attacker's knowledge of known background and population density, thus failing to adequately protect LBS privacy security. Therefore, how to protect the location information published by users from illegal acquisition by LBS providers and against hackers' background knowledge attacks is one of the challenging research topics in the field of location data publishing privacy protection technology. To solve this problem, this paper presents a location publishing algorithm based on differential privacy [8, 9]. This algorithm does not care about the background knowledge of an attacker. Even if an attacker already has all information of the record except one, the privacy of the record will not be compromised. Considering the sparsity of location points in the area of interest, an appropriate perturbation algorithm is selected according to the sparsity. The main contributions of this paper are as follows:

1. For different areas with different densities of locations, quadtree and relative weights are introduced to judge the sparsity of the divided region, and the disturbance algorithm is flexibly selected according to the sparsity of the region. The influence

of the number of location points and the sparsity of location in the region with the published location is reduced.
2. The mapping function is used to deal with the disturbed locations outside the area of interest, and the relative weights are used to avoid adding too much noise when the locations in the quadtree recursively divided area are sparse, effectively reducing the amount of added noise.
3. The sparsity of location points and the number of location points are analyzed experimentally. By comparing with other schemes, it is proved that the algorithm can effectively improve the location privacy protection under the same privacy budget.

2. **Related Works.** Differential privacy [10–13] applies a random algorithm to add noise to the real location, and the generated published location is verified to be valid. Current research on differential privacy is divided into designing algorithms that satisfy differential privacy and reduce the amount of differential privacy noise. Li et al. [14] proposed a personalization-based algorithm to construct user interest areas based on individualized user preferences, but the privacy protection of user interest areas constructed using this algorithm is low; Yan et al. [15] proposed a user demand privacy protection framework based on differential privacy and association rules, which enhance the protection of user privacy, but ignore the impact of location differences within user areas on privacy protection. Liu and Zhu [16] partition the user's area of interest and add noise to the locations in the areas of interest by using different perturbation strategies and query functions, but this method requires high computational and memory space, and when the perturbation strategy is used inappropriately, it can seriously increase the injection of noise; Dong et al. [17] partition the data equally without considering the distribution of location data, and create buckets to represent these data. However, this method also adds noise to regions where no data exists, which invariably increases the amount of noise injected, and if the number of buckets is reduced, then the distribution of data cannot be adequately represented; Deldar and Abadi [18] proposed the PDLP-TD algorithm, which constructs a noisy tree using a database and assigns privacy levels and adds privacy parameters to the branches of the tree, respectively, but the regular construction of the tree results in low efficiency of the algorithm; Huo and Meng [19] designed algorithms for several spatial index structures to satisfy differential privacy, but did not consider the amount of injection of noise into the spatial structure.

To address the relevant shortcomings of the above algorithm, this paper proposes a location differential privacy publishing algorithm QPDPLP based on the user's area of interest, which considers the effect of both the number of position points and the sparsity of position points on the amount of noise added by combining the properties of quadtree when publishing locations in the area of interest.

3. **Problem Definition.** The basic idea of differential privacy is to addinterference noise to the raw data, the functions of the raw data and the dataset of the query results are a part to achieve privacy protection. Differential privacy algorithms are usually done on the basis of the concept of adjacent datasets, which guarantees that the ratio of the probability that an operation to insert or delete a record from an adjacent dataset will produce the same result is close to 1.

Definition 3.1. *Adjacent datasets. Given two datasets A_1 and A_2, A_1 and A_2 are said to be neighboring datasets when A_1 and A_2 satisfy the same structure and one and only one data is different, as shown in Equation (1).*

$$|A_1 \Delta A_2| = 1 \tag{1}$$

Definition 3.2. *Differential privacy. Given adjacent datasets A_1 and A_2, f is a random query algorithm on A_1 and A_2. The arbitrary output of algorithm f on datasets A_1 and A_2 is Z. If Equation (2) is satisfied, then algorithm f is said to satisfy ε-differential privacy.*

$$\frac{\Pr[f(a_1) = z]}{\Pr[f(a_2) = z]} \leq e^{\varepsilon}, a_1 \in A_1, a_2 \in A_2, z \in Z \tag{2}$$

where ε denotes the privacy budget factor in units of distance, and by varying the value of the difference factor, you can control the magnitude of noise addition, that is, the degree of privacy of the published data. When $\varepsilon \to 0$, the higher the degree of privacy, indicating that the output of the two datasets are not very different from each other.

Definition 3.3. *Position sensitivity. Suppose any function $q : A \to R^d$, where the inputs are the data sets A_1 and A_2, and the outputs are $d-$dimensional real vectors, For any data sets A_1 and A_2, there is:*

$$\Delta q = \max_{A_1, A_2} ||q(A_1) - q(A_2)|| \tag{3}$$

Δq denotes the position sensitivity of the function q, indicating the effect of noise addition on the data query. $\|q(A_1) - q(A_2)\|$ shows the Manhattan distance between $q(A_1)$ and $q(A_2)$. Note that the sensitivity is independent of the dataset and only correlates with the query results.

In this paper, a quadtree based method is used for the partitioning of region, the extent of which depends on the relative weight values of the following functions.

Definition 3.4. *Weighting function. In a quadtree, suppose that each node is denoted as $e(key, parent)$ and the parent of that node is denoted as $parent(Key, Sub)$, then the weight of that node is denoted as follows:*

$$w[e] = \frac{e.key}{parent.Key} \tag{4}$$

Where, key represents the weight of each node, namely the number of position points in the region; Key represents the weight of the parent node, that is, the number of locations in the region where the parent node is located; Sub represents the subscript of the parent node.

After partitioning the regions of interest, noise needs to be added to each region to achieve a perturbation effect on the true location, introducing the notion of a distance function in the selection of the perturbation algorithm, defined as follows:

Definition 3.5. *Distance function. Assume that n position objects on the two-dimensional plane K, then the distance between any two points can be expressed as:*

$$dist(o_1, o_2) = \sqrt{(o_1.x - o_2.x)^2 + (o_1.y - o_2.y)^2} \tag{5}$$

where, $K = \{o_i(x_i, y_i) : i = 1, 2, \ldots, n\}$, o_1 denotes the horizontal coordinates of point $o_1.x$ and the vertical coordinates of point $o_1.y$; $o_2.x$ denotes the horizontal coordinates of point o_2 and the vertical coordinates of point $o_2.y$.

4. Location Differential Privacy Publishing Algorithm.

4.1. Traditional Perturbation Publishing Algorithm.

4.1.1. *Independent Perturbation Publishing Algorithm.* The Laplace mechanism is one of the most basic noise mechanisms for differential privacy protection. This mechanism adds random noise that obeys the Laplace distribution to the returned result, so that the result meets the differential privacy and is suitable for the protection of numerical data.

Theorem 4.1. *Laplace perturbation [20]. For any function q on data set A, if the output of Algorithm D satisfies Equation(6), then Algorithm D satisfies ε-differential privacy.*

$$D(A) = q(A) + Lap(\frac{\Delta q}{\varepsilon})^d \tag{6}$$

Where $Lap(\frac{\Delta q}{\varepsilon}) \triangleq$ denotes the added Laplace noise, and the noise variable is proportional to the location sensitivity and inversely proportional to the privacy budget ε.

In two-dimensional space, noise cannot be added directly using the above Equation, this paper gives the probability function of the noise that needs to be added to satisfy ε-differential privacy in the case of a single location.

Suppose the set of true positions is A_1, the ε-differential privacy is satisfied when the elements in the set of disturbed positions A_2 after adding noise satisfy Equation (7).

$$Pr[D(a_1) = a_2] = \frac{\varepsilon^2}{2\pi} e^{-\varepsilon * dist(a_1,a_2)}, a_1 \in A_1, a_2 \in A_2 \tag{7}$$

Where $Pr[D(a_1) = a_2]$ denotes the probability that the true position a_1 after adding noise is a_2. When ε is certain, it can be seen from Equation(7) that $Pr[D(a_1) = a_2]$ is only related to the distance between a_1 and a_2, and that $Pr[D(a_1) = a_2]$ decreases as the distance between the two increases.

In order to simplify the representation, the polar function can be used to achieve the following:

$$Pr[D(a_1) = a_2] = \frac{\varepsilon^2}{2\pi} e^{-\varepsilon * dist(a_1,a_2)}, a_1 \in A_1, a_2 \in A_2 \tag{8}$$

Where $dist(a_1, a_2)$ denotes the distance between a_1 and a_2, and a denotes the angle between a_1 in polar coordinates and the polar axis.

To facilitate the solution, the Equation (9) is decomposed onto the distance $dist(a_1, a_2)$ and the angle a to obtain the following Equation:

$$\begin{aligned} \Pr[D_{\varepsilon,dist(a_1,a_2)}[dist(a_1,a_2)]] &= \int_0^{2\pi} \Pr[Q_\varepsilon[dist(a_1,a_2),\alpha]]d\alpha \\ &= \varepsilon^2 dist(a_1,a_2) e^{-\varepsilon dist(a_1,a_2)} \end{aligned} \tag{9}$$

$$\begin{aligned} \Pr[D_{\varepsilon,\alpha}(\alpha)] &= \int_0^{+\infty} \Pr[Q_\varepsilon[dist(a_1,a_2),\alpha]]dr \\ &= \frac{1}{2\pi} \end{aligned} \tag{10}$$

According to the above Equation, noise can be added to the real location set A_1 and a random disturbed location set A_2, can be generated to achieve the purpose of differential privacy location protection, where

$$B\{b_i = a_i + [dist(a_1,a_2)\cos\alpha, dist(a_1,a_2)\sin\alpha]\} \tag{11}$$

The independent perturbation algorithm is to add Laplace noise to each position in the true position A_1 separately, which satisfies ε-differential privacy and whose degree of privacy protection is the sum of the privacy protection budgets for each location. When

the number of real locations is small and the real locations are far apart, the effect of the noise addition algorithm proposed in this section is ideal, but the degree of privacy protection will decrease as the number of locations in the collection increases, and satisfies ε-geography indistinguishability [21], that is, the closer the disturbed location to the real location, the easier it is to publish. When the real location is too close and too much, the degree of privacy will decrease.

4.1.2. *Centroid Perturbation Publishing Algorithm.* This section addresses the above problem by adding noise to the centroid of a polygonal model, the idea of this model is: firstly, the protected points and the surrounding points that meet the requirements are formed into a polygon, secondly to calculate its centroid coordinates using the centroid Equation, and finally to add noise to it by using an independent perturbation algorithm. Suppose that the set of positions F in a user's trajectory map represents protected position points, Location set A represents two-dimensional information about the user's historical position points, and the points in the location set A with the number of users visits greater than m are added to the pending set H.

For these position points, According to the algorithm, the protected points $f_i(x_i, y_i)(i = 1, 2, \ldots, n)$ and the randomly selected points $h_i(x_i, y_i)(i = 1, 2, \ldots, n)$ from the undetermined set H form a polygon and calculate its centroid. The centroid $I(x, y)$ is computed as shown in Equation (11). Figure 1 shows an example of a triangle.

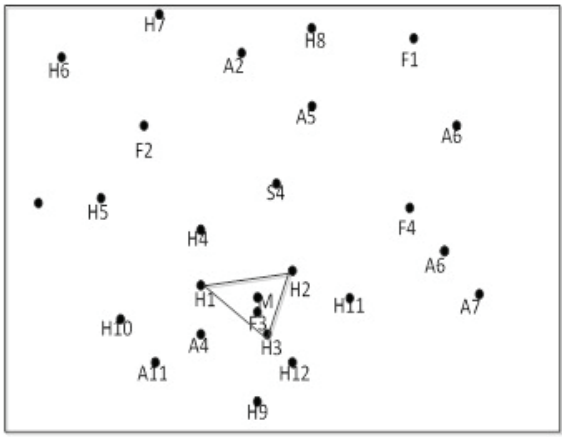


FIGURE 1. Constructing triangle to solve the centroid

When constructing the polygon, assume that the protected point $f_i(x_i, y_i)(i = 1, 2, \ldots, n)$ is the center of the circle and draw acircle with radius R, the point where the polygon is constructed should fall in the circle. It should be noted that the value of R should not be too small, otherwise the range of the generated polygon is too small to play the role of blurring the true position. The R-value should not be too large either, as it will reduce the level of privacy protection. Therefore, we need to set $R_{\max}$ and $R_{\min}$, so:

$$R_{\min} \leq R \leq R_{\max} \tag{12}$$

According to the method described above, the set of centroid M corresponding to the set H of protected position points is obtained, and then noise is added to each particle by the Laplace algorithm, and the resulting results are added to the database C.

4.1.3. *QPDPLP Algorithm.* Suppose that the user interest area is a square with r as its side length. When the perturbation location a_2 generated by perturbation algorithm D is within the user's area of interest, then a_2 is added to database C; When the perturbation

location generated by the perturbation algorithm D is not in the user's zone of interest, the mapping function is used to project onto the area of interest, and the points resulting from the projection are added to the database C.

At a certain moment, given the prior probability $p^{(t)-}(a_1)$ and the perturbation algorithm D, solve for the posterior probability $p^{(t)+}(a_1)$. According to the Bayes' theorem, there is:

$$p^{(t)+}(a_1) = p(a_1|a_2) = \frac{p(a_2|a_1)p^{(t)-}(a_1)}{\sum_{a' \in A} p(a_2|a_1{}')p^{(t)-}(a_1{}')} \tag{13}$$

The mapping function F is expressed as follows:

$$F(a_2) = \arg\min_{c \in C} \sum_{a_1 \in A_1} p^{(t)+}(a_1)\delta(a_1, c) \tag{14}$$

Among them, $\delta(a_1, c)$ is used to measure the degree of service quality *loss* when publishing c instead of a_1, and the service quality *loss* is the quality proportional to the accuracy of the location that the service provider should provide and the location it receives.

Based on the user's background knowledge it is known that in the user interest area, users are not only active in one region. When the position points frequently accessed by users are not close together or in small numbers, the privacy protection of using independent perturbation algorithms is high. To take an extreme example, Suppose the set of user locations contains three locations, a_{11}, a_{12}, a_{13}, where a_{11} is located at a location in Tian An Men Square in Beijing, a_{12} is located at a location in the Yuntai Mountain Scenic Spot in Jiaozuo, and a_{13} is located at a location in Fante Joy World in Zhengzhou. Suppose the user's current location is in position a_{12}, when the user wants to search for a hotel within 1 km of the current location, if the three positions of the centroid perturbation to protect the location, will lead to three positions of the centroid far away from the current position, the search result is certainly not Jiaozuo Yuntai Mountain near the hotel, so the error is very large, although it also achieves the role of privacy protection, but seriously affects the quality of service. In this case, the independent perturbation method is obviously better than the centroid perturbation method; however, when the visited position points are close to each other or there are too many of them, the use of a centroid perturbation algorithm is preferable to an independent perturbation algorithm. For an extreme example, when a user sends multiple service requests at the same location, the centroid perturbation method only needs to add noise once, but the independent disturbance method requires multiple noise additions. Therefore, when the location is close or the number of when there is too much, the centroid perturbation method is better than the independent perturbation method.

To address the aforementioned problems, this section proposes a differential privacy location publishing algorithm that combines quadtree partitioning, which effectively combines an independent perturbation algorithm and a centroid perturbation algorithm by introducing relative weights threshold ξ and sparsity threshold ψ to enhance the protection of location privacy.

The idea of a quadtree is to recursively divide geospatial space into different levels of tree structure, it divides a space of known extent into four equal subspaces, and so on recursively until the tree hierarchy reaches a certain depth or satisfies some requirement and then stops dividing, as shown in Figure 2(a). The traditional method uses a complete quadtree, and although it is easy to implement differential privacy analysis, the method results in a large number of empty nodes due to the large height of the quadtree when the location distribution is unbalanced, resulting in excessive noise injection overall. In this

paper, we design a quadtree splitting method based on relative weights, which effectively reduces the noise injection.

In the first step the weights of each node in the quadtree can be calculated according to definition 3.4; Since the subscripts of child nodes are associated with the subscripts of their parents for each partition of the region, the node subscripts are defined as follows: State i represents the subscript of the parent node corresponding to the desired node; The state ij represents the subscript of the child node of a node with state i, where $j = 1, 2, 3, 4$.

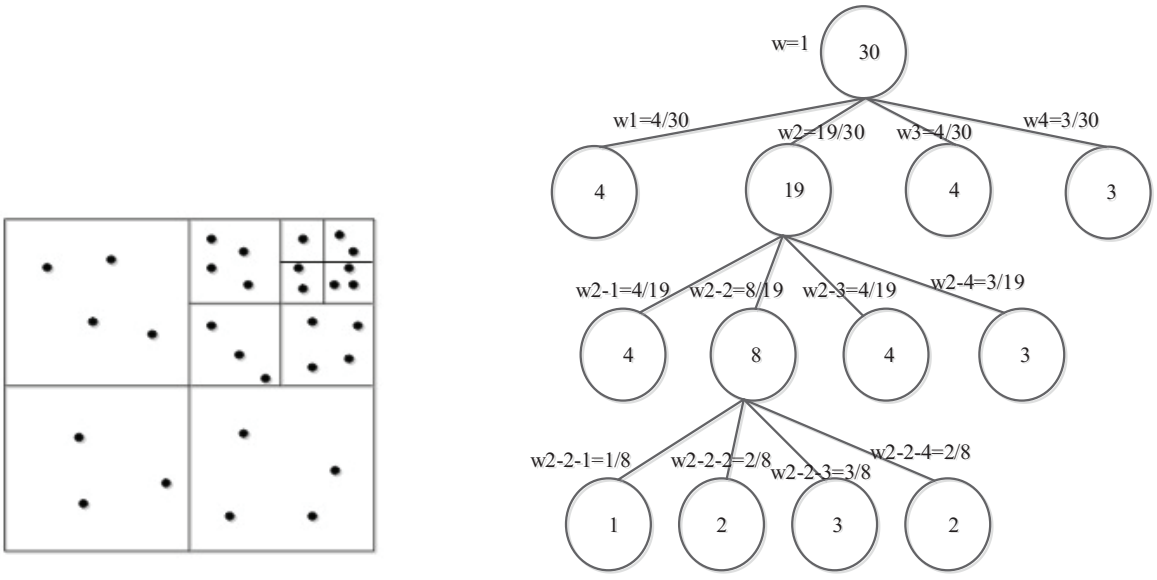
The second step calculates the weight of each area relative to the area of interest, as follows:

$$W[e] = w_{ij}[e] * W[parent], 1 \leq j \leq 4 \tag{15}$$

Where $w_{ij}[e]$ denotes the weight of the requested region. The third step compares the result calculated in the second step with ξ, as follows:

$$Divide = \begin{cases} True, W[e] \geq \xi \\ False, W[e] < \xi \end{cases} \tag{16}$$

Among them, $Divide$ represents division of areas, $True$ means continue to divide, $False$ indicates that the division is stopped, ξ denotes the threshold value of relative weights, when $\xi = 1/6$, the area division is shown in Figure 2(b).



(a) Area division (b) $\xi = 1/6$

FIGURE 2. Area division method based on quadtree weight

Step four and so on until the entire area of interest can no longer be divided.

The next key question is how to choose the appropriate perturbation algorithm for each area. Assuming that $V_1 = \{v_{1i} : i = 1, 2, \ldots, m\}$, then the sparsity of the area is calculated as follows:

$$S_1 = \sqrt{\frac{\sum_{j=1}^{m} [dist(v_{1i}, v_{1j}) - \overline{v_1.dist}]^2}{m-1}}, 1 \leq i \leq m \tag{17}$$

Where v_1 denotes region 1, S_1 indicates the sparsity of region 1, $dist(v_{1i}, v_{1j})$ means the distance between any two points in v_1, and $\overline{v_1.dist}$ denotes the average of the point-to-point distances in v_1.

The sparsity of any region $v_i(i = 1, 2, \ldots, n)$ can be obtained from Equation (18), where S_i denotes the sparsity of the $i-th$ region, $dist(v_{ij}, v_{ik})$ denotes the distance between any two points in v_i, and $\overline{v_i.dist}$ denotes the average of the point-to-point distances in v_i calculated as follows:

$$S_i = \sqrt{\frac{\sum_{j=1}^{m} [dist(v_{ij}, v_{ik}) - \overline{v_i.dist}]^2}{m-1}}, 1 \le i \le n, 1 \le k \le m \tag{18}$$

Based on the above derived Equation, the entire regional division can be calculated. For the distribution of true position points in each region, we can calculate its sparsity level $S_i(i = 1, 2, \ldots, n)$ using Equation(18) and compare it with the sparsity threshold ψ, when the former is small, an independent perturbation algorithm is used, otherwise a centroid perturbation algorithm is used, so that the error-disturbance caused by adding noise can be reduced as much as possible.

The main flow of the QPDPLP algorithm is given below.

Step 1: Establishing a square with r as its side length as a user interest area;

Step 2: According to the quadtree idea, the user's interest area is divided into several small areas, and the quadtree sp litting is controlled by the weight threshold ξ.

Step 3: From Equation (17) and Equation (18), calculate the sparsity $S_i(i = 1, 2, \ldots, n)$ corresponding to each region, compare it with the sparsity threshold ψ, select the perturbation algorithm based on the comparison result, and put the calculated perturbation location into the database C;

Step 4: For the disturbed locations that fall outside the interest area, use Equation (14) to map them to the interest area, and add the mapping result to the database C;

Step 5: Add the real location A to database C, and then randomly select a number of position points from database C to be published location Z.

The overall algorithm of the differential privacy publishing algorithm combined with the quadtree division is as follows:

4.1.4. *Algorithmic Analysis.* The total error of the independent perturbation algorithm is the sum of the noise added at each true position. Assuming that the privacy budget consumed at each location is ε the expectation of the added noise at each location is $E[dist(a_1, a_2)]_i(i = 1, 2, \ldots, n)$, so:

$$\begin{aligned} E[dist(a_1, a_2)]_i &= \int_0^{+\infty} dist(a_1, a_2) * P[dist(a_1, a_2)] d[dist(a_1, a_2)] \\ &= \int_0^{+\infty} dist(a_1, a_2) * \varepsilon^2 dist(a_1, a_2) e^{-\varepsilon * dist(a_1, a_2)} ddist(a_1, a_2) \\ &= \frac{2}{\varepsilon} \end{aligned} \tag{19}$$

From Equation (19), the total error of the independent perturbation algorithm is:

Algorithm 1 QPDPLP algorithm

Input: Real position set A, Length of area of interest r, the prior probability $p^{(t)-}$ of the non-interest area before moment t, Differential privacy budget ε, $R_{\min}$, $R_{\max}$, Relative weight threshold ξ, Sparsity threshold ψ.

Output: Publish location Z

```
nshoCount = Initialize();
for nshoCount > ξ
Recursion(e); // Iterative division of areas
end for
Calculate; // Equation (18)
if S_i > ψ Then
D=Choose(0); //0 indicates an independent perturbation algorithm
else
D=Choose(1); //1 represents the centroid perturbation algorithm
C_in=Disturbed(D);
P^(t)+= ChooseLoc(p^(t)-); // Equation (13)
C_out ← CalRelease(p^(t)+);
C ← (C_in ∪ C_out ∪ A);
Z=RandomChoose(C);
return Z;
```

$$\begin{aligned} Error_1 &= \sum_{i=1}^{n} error_i \\ &= n * E[dist(a_1, a_2)]_i \\ &= \frac{2n}{\varepsilon} \end{aligned} \tag{20}$$

The total error of the centroid perturbation algorithm takes into account not only the error caused by adding noise to the centroid, but also the error between the true position and the sought-after centroid. Assume that the privacy budget consumed per location is ε, but in fact, the privacy budget consumed by each location is $\varepsilon^{'}/n$, then the error added to the centroid is $E_s[dist(a_1, a_2)] = \frac{2}{n^2\varepsilon}, 1 \le s \le m$, then the total error of the centroid perturbation algorithm is:

$$\begin{aligned} Error_2 &= \sum_{i=1}^{n} (dist(a_{1i}, a_{1s}^{'}) + dist(a_{1s}, a_{1s}^{'})) \\ &= \sum_{i=1}^{n} dist(a_{1i}, a_{1s}^{'}) + \frac{2}{n^2\varepsilon}, 1 \le s \le m \end{aligned} \tag{21}$$

where a_{1i} denotes the $i-th$ true position, a_{1s} denotes the centroid position, and $a_{1s}^{'}$ denotes the disturbed position obtained after adding noise to the centroid position.

The QPDPLP algorithm is an algorithm that combines the independent perturbation algorithm and the centroid perturbation algorithm based on the relative weight threshold ξ and the sparsity threshold Ψ, So the total error is:

$$Error_3 \ = \frac{2j}{\varepsilon} + \sum_{i=\varsigma+1}^{n} dist(a_{1i}, a_{1s}^{'}) + \frac{2}{(n-\varsigma)^2\varepsilon}, 1 \le j \le \varsigma, 1 \le s \le m \tag{22}$$

Among them, a_{1i} represents the $i-th$ real position, and $a_{1s}^{'}$ represents the disturbed position obtained by adding noise to the centroid position.

5. **Experiments and Analysis.** Comparative experiments will be performed QPDPLP algorithm, the differential privacy location protection method (ISTDP) for irregular line trees [22] and the Haar wavelet zero-tree compression algorithm (EH-WT-DP), which is representative for improving the accuracy of differential privacy queries [23].

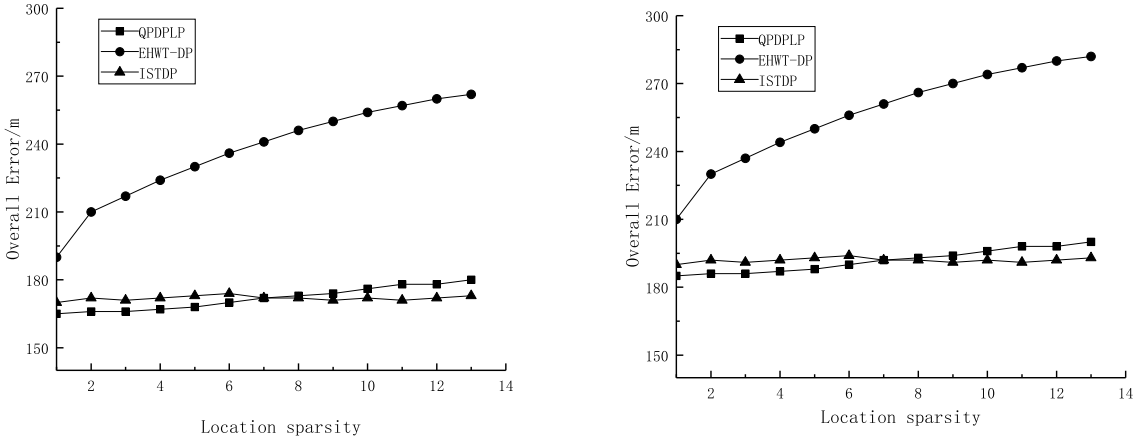
Since the value of the privacy budget directly affects the size of the noise addition and thus affects the error analysis, Therefore, this paper uses the same privacy budget to evaluate the accuracy of the algorithm query and the time complexity of the algorithm in terms of location sparsity and the number of position points.

5.1. **Experimental Setup.** The QPDPLP algorithms were implemented using MATLAB, and the experimental environment was a Windows 7 operating system with 8.00 GB RAM and a 3.60 GHz CPU. The datasets used in the experiments are the landmark and storage real datasets, the former being the landmarks consisting of geographic coordinates of the 48 large U.S. states provided by Infochimps big data site with about 880k data points, and the latter being the U.S. storage facility location data with sparse data points of about 10,000 data points. The specific parameter settings are shown in Table 1.

Table 1. Experimental parameters

Parameter	Description	Parameter value range
ε	Privacy budget	$1 <= \varepsilon <= 10$
S_i	Location sparsity	$1 <= S_i <= 13$
n	Number of location points	$0 <= n <= 200$

5.2. **Variation of Query Error at Different Location Sparsity.** To test the comparison of the query errors of three algorithms, QPDPLP, ISTDP, and EHWT-DP, in the experiment, the parameters was set up with ε=1, and the degree of sparsity was taken from $1 \sim 13$. For different location sparsity in storage dataset and landmark dataset, the results of the experiment are shown in Figure 3.

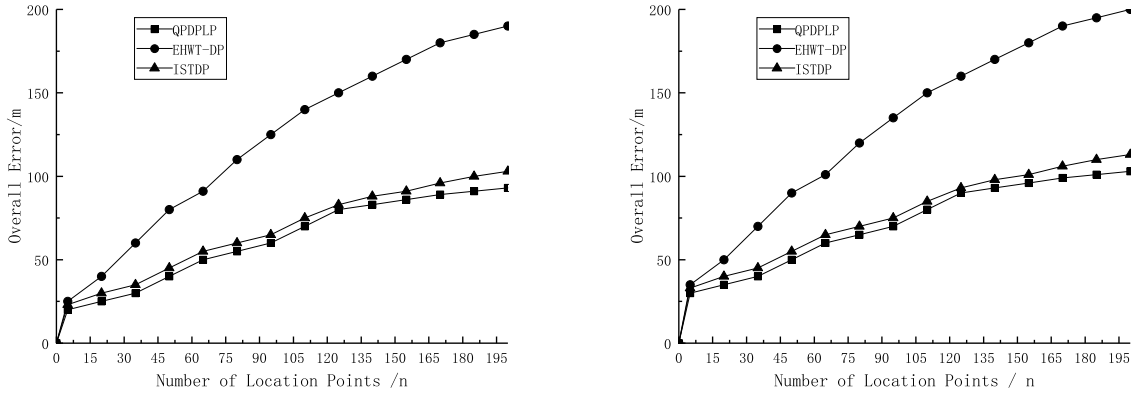


(a) Storage Dataset (b) Landmark Dataset

Figure 3. Error in different degree of sparsity

As can be seen from Figure 3(a), the EHWT-DP algorithm has the highest total error and the lowest location availability, i.e., the lowest location security, because EHWT-DP is strongly influenced by location sparsity; The ISTDP algorithm, whose total error does not fluctuate much and essentially shows a slow growth trend, has the best location availability, this is because the algorithm is essentially independent of location sparsity when privacy budgets are the same, resulting in better query accuracy, minimal errors and low fluctuations; The QPDPLP algorithm in this paper is intermediate between the two algorithms and close to the performance of the ISTDP algorithm because the algorithm combines an independent perturbation publishing algorithm and a centroid perturbation publishing algorithm, so it is affected by location sparsity and therefore location availability is slightly lower than the ISTDP algorithm. Figure 3(b) shows similar results on the landmark dataset. Also, the performance of the three algorithms on the storages dataset is better than the performance on the landmark dataset because the data distribution of storages is denser and the data distribution of landmark is sparse, so the total error of the former is smaller than the latter.

5.3. **Variation of Query Error for Different Number of Position Points.** In addition to location sparsity influencing the three algorithms, the number of position points is an important factor in the error. In this paper, we experimentally analyze the effect of the number of position points on the total error of three algorithms. In the experiment, the parameters was set up with ε=1, and sparsity is 2. The results of the experiment are shown in Figure 4.



(a) Storage Dataset (b) Landmark Dataset

FIGURE 4. Error under the number of different position points

It can be seen from Figure 4(a) that as the number of position points increases, the total error of the three algorithms increases. where the EHWT-DP algorithm shows the largest increase due to its large query error for individual position points, so that the total error increases linearly as the number of position points increases; The QPDPLP algorithm and the ISTDP algorithm have a smaller increase in the total error than the EHWT-DP algorithm due to the smaller query error at a single location, and the QPDPLP algorithm is also superior to the ISTDP algorithm due to the introduction of a centroid perturbation publishing algorithm in this paper, which reduces the amount of noise injection. Figure 4(b) shows similar results on the landmark dataset.

5.4. **Comparison of Algorithmic Time Complexity.** This experiment was conducted on the storage dataset and the landmark dataset. To compare the run times of QPDPLP,

EHWT-DP and ISTDP, in the experiment, the parameters was set up with $n = 200$, and sparsity is 2. The results are shown in Figure 5.

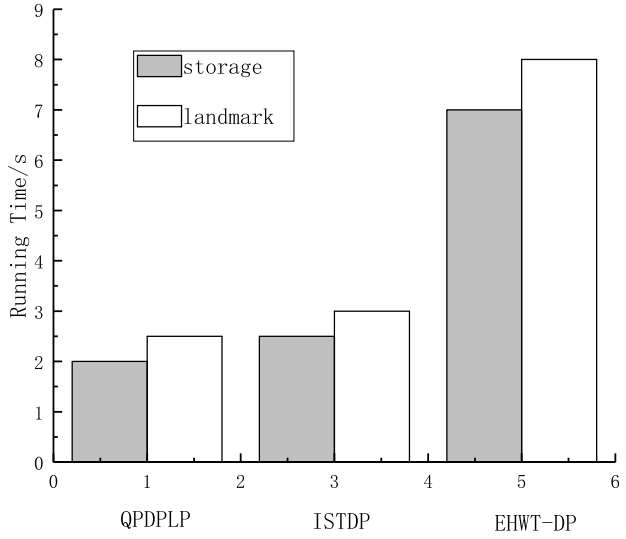


FIGURE 5. Algorithm running time

It can be seen from the figure that the running time of QPDPLP and ISTDP are very close, this is because these two algorithms are improved based on the tree structure and are only related to the location set n. The grid division of the region is performed in an offline environment, so the running time is less than EHWT-DP. The main time complexity of ISTDP is to recursively divide the sub-line segment tree, so the time complexity of the algorithm is $O(logn)$; QPDPLP is similar to ISTDP, and the time complexity is also $O(logn)$. Since QPDPLP introduces a relative weight threshold ξ , which can effectively control the number of divisions, the operational efficiency is improved compared to that of ISTDP, while it is superior relative to EHWT-DP, the time complexity of which is $O(nlogn)$.

6. **Conclusions.** The core of the differential privacy protection algorithm based on user's interest area proposed in this paper is to introduce relative weight threshold and sparsity threshold based on traditional position disturbance, and propose a QPDPLP algorithm based on user's interest area. The algorithm firstly uses the idea of quadtree to divide the user's area of interest; then the sparsity of the divided area is judged; and finally, the perturbation algorithm is selected according to the sparsity. Compared with the traditional algorithm, the algorithm considers both the influence of the number of position points on the published location and the influence of the sparsity of the position points on the published location, and the algorithm designed in this paper effectively reduces the total error between the scrambled location and the real location after adding noise.

After analyzing and comparing the experimental results, it is demonstrated that the amount of noise added is effectively reduced under the same privacy budget, thus ensuring that the user's private information is not leaked. Future research can continue to optimize the QPDPLP algorithm to reduce the running time of the algorithm and disturb the usability of location publishing for better application to location publishing services.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (61300216), Doctoral Scientific Fund of Henan Polytechnic University(B2022-16) and Youth Fund of Henan Polytechnic University(Q2014-05).

REFERENCES

[1] L. Zhang, D. Liu, M. Chen, H. Li, C. Wang, Y. Zhang, and Y. Du, A user collaboration privacy protection scheme with threshold scheme and smart contract, *Information Sciences*, vol. 560, no. 1, pp. 183-201, 2021.

[2] K. Wang, C. Chen, Z. Tie, M. Shojafar, S. Kumar, and S. Kumari, Forward Privacy Preservation in IoT-Enabled Healthcare Systems, *IEEE Transactions On Industrial Informatics*, vol. 18, no. 3, pp. 1991-1999, 2022.

[3] S. Zhang, X. Li, Z. Tan, T. Peng, and G. Wang, A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services, *Future Generation Computer Systems*, vol. 94, pp. 40-50, 2019.

[4] O. Lu, Q. Zheng, L. Shaolin, H. Yuan, and X. Jia, Releasing Correlated Trajectories: Towards High Utility and Optimal Differential Privacy, *IEEE Transactions On Dependable and Secure Computing*, vol. 17, no. 5, pp. 1109-1123, 2020.

[5] Y. Liu and Q. Zhao, E-voting scheme using secret sharing and K-anonymity, *World Wide Web-Internet and Web Information Systems*, vol. 22, no. 4, pp. 1657-1667, 2019.

[6] R. Bild, J. Eicher and F. Prasser, Efficient Protection of Health Data from Sensitive Attribute Disclosure,*Studies in Health Technology and Informatics*, vol. 270, pp. 193-197, 2020.

[7] M.A. Mohamed, S.M. Ghanem and M.H. Nagi, Privacy-preserving for distributed data streams: towards l-diversity., *International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 52-64, 2020.

[8] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li and Y. Li, Towards Privacy-preserving Incentive for Mobile Crowdsensing Under An Untrusted Platform,*IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 2053-2061, 2019.

[9] T. Wang, Y. Mei, W. Jia, X. Zheng, G. Wang, and M. Xie, Edge-based differential privacy computing for sensor–cloud systems, *Journal of Parallel and Distributed Computing*, vol. 136, pp. 75-85, 2020.

[10] P.C.M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, A Trustworthy Privacy Preserving Framework for Machine Learning in Industrial IoT Systems, *IEEE Transactions On Industrial Informatics*, vol. 16, no. 9, pp. 6092-6102, 2020.

[11] T. Zhang, D. Ye, T. Zhu, T. Liao, and W. Zhou, Evolution of cooperation in malicious social networks with differential privacy mechanisms, *Neural Computing and Applications*, vol. 2020, no. 4, pp. 156-168, 2020.

[12] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, Differentially private Naive Bayes learning over multiple data sources, *Information Sciences*, vol. 444, pp. 89-104, 2018.

[13] C. Yin, J. Xi, R. Sun, and J. Wang, Location Privacy Protection Based on Differential Privacy Strategy for Big Data in Industrial Internet of Things, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3628-3636, 2018.

[14] Y. Li, S. Liu, D. Li, and J. Wang, Release Connection Fingerprints in Social Networks Using Personalized Differential Privacy, *Chinese Journal of Electronics*, vol. 27, no. 5, pp. 1104-1110, 2018.

[15] C. Yan, Z. Ni, B. Cao, R. Lu, S. Wu, and Q. Zhang, UMBRELLA: user demand privacy preserving framework based on association rules and differential privacy in social networks, *Science China Information Sciences*, vol. 62, no. 3, pp. 205-207, 2018.

[16] S. Liu and Y. Zhu, Differential privacy protection for social network edge weights, *Computer Engineering and Design*, vol. 39, no. 1, pp. 44-48, 2018.

[17] S.U. Dong, J.N.E. Cao, L.I. Ninghui, N.O. ELISA BERTI, M. Lyu, and H. Jin, Differentially Private K-Means Clustering and a Hybrid Approach to Private Optimization, *Acm Transaction on Information & System Security*, vol. 20, no. 4, pp. 11-16, 2017.

[18] F. Deldar and M. Abadi, PLDP-TD: Personalized-location differentially private data analysis on trajectory databases, *Pervasive and Mobile Computing*, vol. 49, pp. 1-22, 2018.

[19] Z. Huo and X. Meng, A Trajectory Data Publication Method under Differential Privacy, *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 41, no. 2, pp. 400-412, 2018.

[20] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi, Privacy-Preserving Crowd-Sourced Statistical Data Publishing with An Untrusted Server, *IEEE Transactions On Mobile Computing*, vol. 18, no. 6, pp. 1356-1367, 2019.

[21] P. Zhang, C. Hu, D. Chen, H. Li, and Q. Li, ShiftRoute: Achieving Location Privacy for Map Services on Smartphones, *IEEE Transactions On Vehicular Technology*, vol. 67, no. 5, pp. 4527-4538, 2018.
[22] D.Hu and Z.Liao, Differential privacy location privacy preserving method for irregular line segment trees, *Microcomputer system*, vol. 41, no. 2, pp. 333-337, 2020
[23] X. Xiao, G. Wang and J. Gehrke, Differential Privacy via Wavelet Transforms, *IEEE Transactions on Knowledge & Data Engineering*, vol. 23, no. 8, pp. 1200-1214, 2011.