A Differential Privacy Budget Allocation Method Combining Privacy Security Level

***Abstract*—Trajectory privacy protection schemes based on suppression strategies rarely take geospatial constraints into account, which is made more likely for an attacker to determine the user's true sensitive location and trajectory. To solve this problem, this paper presents a privacy budget allocation method based on privacy security level(PSL). Firstly, in a custom map, the idea of P-series is contributed to allocate a given total privacy budget reasonably to the initially sensitive locations. Then, the size of privacy security level for sensitive locations is dynamically adjusted by comparing it with the customized initial level threshold parameter μ. Finally, the privacy budget of the initial sensitive location is allocated to its neighbors based on the relationship between distance and degree between nodes. By comparing the PSL algorithm with the traditional allocation methods, the results show that it is more flexible to allocate a privacy budget without compromising location privacy under the same preset conditions.**

***Index Terms*—location privacy, differential privacy, privacy budget, budget allocation, PSL**

I. Introduction

In the Internet era, with the continuous development of digital services such as medical treatment [1], shopping [2], payment [3], navigation [4], etc., data collection is becoming simpler and simpler. Publishing and sharing of collected data is of great value, but it also facilitates the collection and mining of data by attackers, which seriously affects the privacy and information security of users. For example, on May 30, 2018, Kromtech Security Center revealed Honda India, a subsidiary of Honda in India, leaked personal details from more than 50,000 customers because of unsafe Amazon Web Services (AWS) Simple Storage Service (S3) buckets. Researchers found it possible for hackers to steal these data because the company accidentally stored personal details in a publicly accessible Amazon S3 bucket. Information included names, phone numbers, passwords, gender, email addresses, vehicle identification numbers (VIN), connect identities (CID), etc. Therefore, it is an urgent problem to ensure that users' sensitive information is not leaked while using social network platforms normally [5], [6].

For privacy protection of specific data, many scholars have carried out fruitful research [7], [8]. Traditional privacy protection methods [9], [10] rely heavily on known background knowledge. Current privacy information has characteristics of many categories, complex hierarchical relationships, and large numbers. To meet the need for privacy data protection, researchers focus on differential privacy, a concept proposed by Dwork in 2006 for privacy data disclosure in statistical databases [11]. Differential privacy has become a hotspot in location-based services (LBS) privacy protection [12], [13] due to its strong mathematical foundation.

One feature of differential privacy is that the privacy budget ε measures resilience. Smaller ε implies greater privacy protection. However, privacy budget allocation is hard to control. Chen et al. [14] uses uniform distribution, wasting part of the budget. Chen et al. [15] uses an adaptive method based on Markov prediction but doesn't meet ε-differential privacy requirements.

Li et al. [16] propose a budget allocation algorithm based on out-of-bag estimation in random forests. Zhang et al. [17] propose an iterative method for blockchain systems to minimize deviation. Wang et al. [18] proposes arithmetic and geometric sequences for personalized choices but introduces excessive noise, reducing accuracy. These strategies are only suitable for specific spatial data applications. Based on this analysis, this paper makes the following contributions:

• To avoid excessive injection noise, this paper effectively allocates the total privacy budget to each initially sensitive location using the P-series allocation method.

• To solve the problem that attackers can infer sensitive locations based on geographic relationships between locations, this paper presents an algorithm that assigns budgets for initial sensitive locations to their neighboring locations. The algorithm effectively reduces the noise level injection, while playing a strong privacy protection role.

II. Related Concepts and Definitions

A. Data Protection Framework

There are two main protection frameworks for differential privacy data publishing, which are interactive and non-interactive.

1) Interactive Framework: An interactive framework, also known as an online query framework, is shown in Fig. 1. When the user submits the function q through the query interface, the database server designs a query function f which satisfies the differential privacy according to the query requirements. After filtering by the algorithm f , the database server returns the noise result z' to the user. For example, there is a patient data set D, and the request query q here is the number of diabetes patients in the query data set D. Surely, modifying the data of any patient in the data set will result in the query result increasing by 1 at most, then the query function f will add noise that obeys the Laplace distribution in the query result, so that the output of the query result will not be changed because of modifying one person's data, Therefore, f satisfies the differential privacy requirements. In this process, the datasets

and the user are separated, that is, the user cannot directly access the data to achieve the purpose of privacy protection.

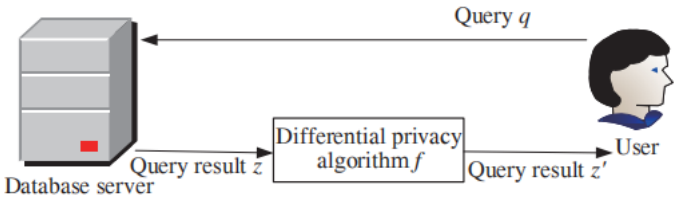


Fig. 1: Interactive framework

2) Non-Interactive Framework: A non-interactive framework, also known as an offline publishing framework, is shown in Fig. 2.

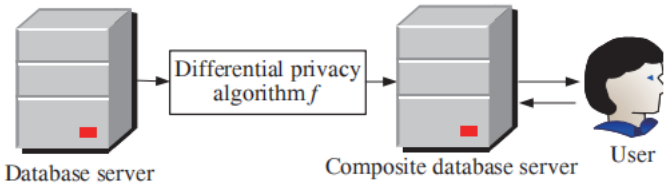


Fig. 2: Non-interactive framework

After the original data is processed by some differential privacy algorithm f , the database server transforms it into a composite dataset server and publishes it for user searching.

B. Related Definitions

The basic idea of differential privacy is to add random noise to the original data, query results, or functions on the original data so that there is no effect on the output when inserting or deleting a record in the dataset. Even if an attacker has any background knowledge, he or she cannot determine whether any real data in the dataset from the published data to protect his or her privacy.

Definition 1 (Adjacent dataset). Gives two datasets D and D', where $D = \{d_1, d_2, \cdots, d_{n-1}\}$, $D' = \{d'_1, d'_2, \cdots, d'_{n-1}\}$, if two datasets differ by only one record, D and D' are adjacent datasets when they satisfy (1).

$$D \cap D' = \{d_1, d_2, \cdots, d_{n-1}\}. \tag{1}$$

Definition 2 (ϵ-Differential privacy) [19]. Given adjacent datasets D and D', f is a random query algorithm on D and D'. The result of any output of the algorithm f on datasets D and D' is Z. If (2) is satisfied, then algorithm f is said to satisfy differential privacy.

$$\frac{\Pr[f(D) = z]}{\Pr[f(D') = z]} \leqslant e^{\varepsilon \cdot l_0(D,D')}. \tag{2}$$

where ϵ indicates the degree of privacy protection, and the smaller the ϵ, the higher the degree of privacy protection.When $\epsilon \to 0$, privacy protection is increasing, indicating that the output of the two datasets is closer. lo(c,d) is a measurement matrix, and different lo(c,d) models represent different scenarios in differential privacy problems. If lo(c,d) = l2(c,d), then it represents the Euclidean distance between two locations. Differential privacy can be used to protect privacy disclosure when LBS location queries occur.

Definition 3 (Location Sensitivity). Assuming any function f whose inputs are datasets D and D' and outputs are d-dimensional real vectors, for any datasets D and D', (3) is satisfied.

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\|. \tag{3}$$

Δf denotes the position sensitivity of function f and the effect of adding noise on data queries.$\|f(D) - f(D')\|$represents the Manhattan distance between $f(D)$ and $f(D')$. It is worth noting that sensitivity is independent of the dataset and only related to the query results.

In the differential privacy introduced by Dwork et al., the mechanism becomes the standard tool for ϵ-differential privacy, which can be achieved by adding noise that obeys the Laplace distribution to the private data information.

Definition 4 (Laplace mechanism) [20]. For any function f on dataset D, if the output of algorithm M satisfies (4), then algorithm M satisfies ϵ-differential privacy.

$$M(D) = f(D) + \text{Lap}\left(\frac{\Delta f}{\varepsilon}\right)^d. \tag{4}$$

where $\text{Lap}\left(\frac{\Delta q}{\varepsilon}\right)^d$ is the added Laplace noise, and the noise variable is directly proportional to the location sensitivity and inversely proportional to the privacy budget ϵ.

Lemma 1 (Differential privacy serial combination principle). Assumes that $f_1, f_2, \ldots, f_n$ represents a random algorithm and that any two algorithms are independent of each other, where f_i satisfies ε_i-differential privacy, and $0 \leqslant i \leqslant n$, then these algorithms $<f_1, f_2, \ldots, f_n>$ are combined to satisfy ε-differential privacy, that is

$$\varepsilon = \sum_{i=1}^{n} \varepsilon_i. \tag{5}$$

Lemma 2 (Differential privacy sequence integration principle). Given dataset D, and differential privacy algorithm $f_1, f_2, \ldots, f_n$, any two algorithms are independent of each other, and each algorithm can satisfy ϵ-differential privacy. Then if a given algorithm is combined together, the combination result can satisfy $\sum_{i=1}^{n} \varepsilon_i$ differential privacy.

In LBS, because of the spatial characteristics of geographic locations, users may provide the same or similar query results to service providers in different locations, thus defining service similarity.

Definition 5 (Service similarity). For two different location points a and b, their service similarity is defined as

$$S = \text{sim}((a.x, a.y), (b.x, b.y)) = \frac{|R_k(a.x, a.y) \cap R_k(b.x, b.y)|}{k} \tag{6}$$

where $(a.x, a.y)$ represents the coordinates of the location a, and $R_k(a.x, a.y)$ represents the sorted result set of k points of interest queried at coordinate $(a.x, a.y)$.

Given a map of an area, service similarity queries are performed on the map, the set of locations with the same similarity is merged, and the merged map is divided and numbered, as shown in Fig. 3.

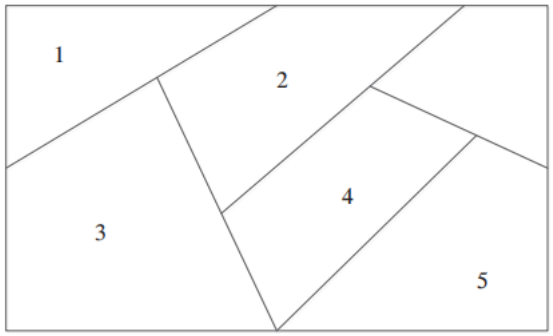


Fig. 3: Regional division

After the zoning map is obtained, it is converted to a weighted undirected graph $U = <V, E>$, V represents each region after division, and E represents the edges between the regions, as shown in Fig. 4.

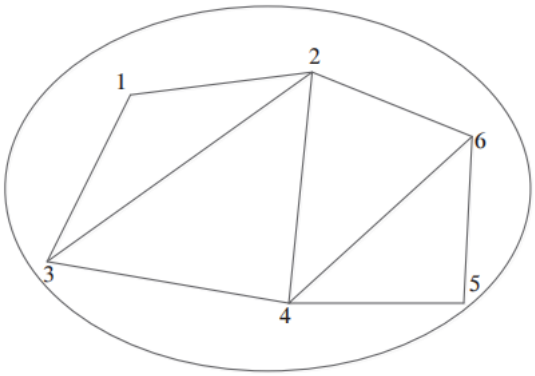


Fig. 4: Undirected graph representation

When two regions can be reached directly, their corresponding nodes are connected, and the direct distance between the two regions is represented by the weight of the edge.

III. PRIVACY BUDGET ALLOCATION

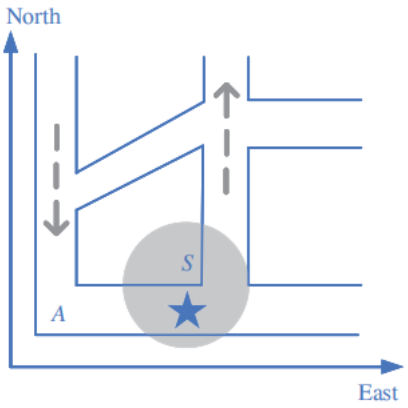
A. Problem Definition

The research objective is to allocate a reasonable privacy budget for sensitive points and the nodes adjacent to sensitive points under the given total privacy budget ϵ. While protecting the privacy information, the query function f can maximize the data availability. In other words, the error between the query data results and the original query data results after disturbing according to the privacy budget allocation is the minimum. Assuming that the real location of the location z_{ij} published at a certain time is o_{ij}, the distance between z_{ij} and o_{ij} is used as the error evaluation. The problem formula is defined as

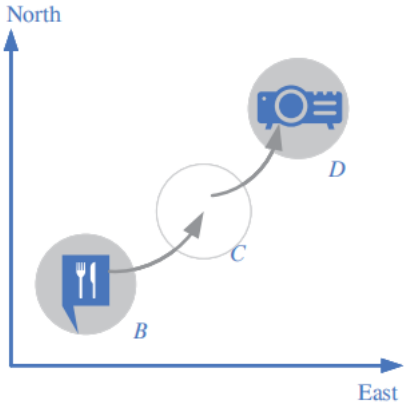
$$\begin{aligned} E &= \min\left[\mathrm{dis}(z_{ij}, o_{ij})\right] = \min\left|z_{ij} - o_{ij}\right| \\ &= \min\left|f\left(\mathrm{location}\left(\sum_{i=1}^{n}\sum_{j=1}^{k}\varepsilon_{ij}\right)\right) - f(\mathrm{location}(0))\right| \end{aligned} \tag{7}$$

where i represents the number of sensitive points, j represents the number of adjacent nodes of sensitive points, ϵ_{ij} represents the privacy budget allocated by adjacent nodes j at sensitive point i, and $i, j \in N^+$.

Some personalized trajectory privacy protection mechanisms (i.e., user-defined privacy levels for different locations, such as suppression policies) only consider the privacy level of a single location, while ignoring the impact of geographical topology on the location privacy level. As shown in Fig. 5(a), set the sensitive position as S (marked with a star), and generalize S to the gray area (that is, the gray area is the sensitive area), which is currently moved by the gray arrow on the left.



(a)



(b)

Fig. 5: Sensitive location and privacy protection: (a) Geographic topology;(b) Temporal relationship

In the suppression strategy, when the user moves to location A, the location will be truthfully reported (because A is insensitive). However, if the user moves to the arrow on the right after a period of time, even if the location of the user is not reported within the gray area, the attacker can easily guess that the user has passed through or stayed at S according to the geographic location relationship between A and S, thus revealing the user's location privacy or even the trajectory privacy.

There may be some correlation between the geographical topology and the location of continuous moments. Even if the sensitive location of a single moment is protected, the continuous trajectory will still have a great possibility of exposing the user's real location. As shown in Fig. 5(b), even if the user's sensitive location areas B and D and location C near the sensitive point are regularly generalized, an attacker is likely to speculate that the user is at the cinema at this time, resulting in a disclosure of privacy based on the user's behavior pattern (e.g., 80% of users who pass this trajectory after 9 p.m. are people who go to the movies after dinner), geographic constraints, and other information.

In this paper, the total privacy budget is effectively allocated to each initial sensitive location using the P-series allocation method, which effectively reduces the noise injection and improves the utilization efficiency of the privacy budget. To solve the problem of attackers inferring sensitive locations based on geographic relationships between locations, an algorithm for allocating budgets for initial sensitive locations to adjacent locations is presented. The algorithm sets a security level for locations near sensitive points and takes into account the correlation between nodes and time factors, dynamically adjusting the security level of locations to achieve an effective privacy budget allocation, while protecting the user's location privacy and improving the availability of published location data.

B. Sensitive Location Adjacent Node Privacy Budget

In the differential privacy location scrambling algorithm, it is unreasonable to scramble each location point to the same degree. On the one hand, users do not have a high demand for privacy protection in some places. Large disturbances to these locations will reduce the quality of service users get. On the other hand, excessive disruption can result in low trajectory accuracy and is of little value to service providers for mining private data. Therefore, this paper uses the idea of the series to allocate different privacy budgets to different sensitive locations.

Set the initial set of sensitive locations $\mathrm{ML}^{\mathrm{initial}} = \{ml_1, ml_2, \ldots, ml_n\}$, where ml_n is the zone number after zoning. The entire map can be divided into $H = \{\mathrm{ML}^{\mathrm{initial}}, \mathrm{NML}^{\mathrm{initial}}, \mathrm{NR}\}$, where $NML_initial$ represents a collection of insensitive areas, and NR represents geographically inaccessible places, such as the ocean.

P-series, also known as hypertonic and series, is a special positive term series. When $p > 1$, the P-series $\sum_{m=1}^{\infty} \frac{1}{m^p}$ converges, and remember that it converges to $\zeta(p)$, that is, $\sum_{m=1}^{\infty} \frac{1}{m^p} = \zeta(p)$. Therefore, $\varepsilon = \frac{\varepsilon}{\zeta(p)} \sum_{m=1}^{\infty} \frac{1}{m^p}$.

In the differential privacy protection method, the privacy budget is allocated using the P-series as

$$\varepsilon_m = \frac{\varepsilon}{\zeta(p)} \times \frac{1}{m^p}, \quad m \in \mathbb{N}^+ \tag{8}$$

where $p > 1$, and $\zeta(p)$ represents the convergence value of the P-series.

Assign the privacy budget $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ to the initial sensitive location set $\mathrm{ML}^{\mathrm{initial}} = \{ml_1, ml_2, \ldots, ml_n\}$. According to Lemma (1), when (9) is satisfied, ε-differential privacy is still satisfied after infinite budget allocation using P-series.

$$\varepsilon = \sum_{m=1}^{n} \varepsilon_m = \frac{\varepsilon}{\zeta(p)} \sum_{m=1}^{n} \frac{1}{m^p}, \quad 1 \leqslant n < \infty \tag{9}$$

Definition 6 (Location privacy) [21]. Assumes that the location of a publication satisfies the location privacy. If its privacy security level (PSL) sl and differential privacy protection budget ε satisfies (10) assigned to that point.

$$\epsilon \times sl = \gamma \tag{10}$$

Definition 6 shows that, given γ, PSL sl is inversely proportional to the privacy budget ϵ. After calculating the privacy budget according to (8), the value of the corresponding PSL can be obtained by substituting it into (10). It is worth noting that the privacy budget allocated according to the P-Series in this paper is fixed.

The PSL corresponding to the node k can be calculated from (8) and (10), then the PSL of the node is as

$$k_m.sl = \lambda \times \frac{\zeta(p)}{\varepsilon} \times m^p, \quad 1 \leqslant m < \infty \tag{11}$$

The idea of a suppression strategy is to choose a sensitive location without publishing the current location. If a suppression strategy is used only for sensitive locations, the effect of geographic relationships on sensitive locations will be ignored. Therefore, a certain level of privacy security should be set for locations near sensitive locations. Although randomly assigned PSLs are more secure, location availability decreases. To improve the availability of locations and reduce the probability of leaking sensitive locations, the relationship between nodes is taken into account in this paper. Set appropriate PSLs for sensitive locations based on the distance between nodes and the degree of nodes, taking into account that location PSLs may change over time. Take time into account to provide more detailed PSL calculations.

(11) gives the privacy level $k_m.sl$ of the sensitive node k_m. The critical point set of k_m is NPS, and its size is the degree of node k_m in graph U. For any node k_{mh} in NPS, the privacy level assigned to it is shown below

$$k_{mh}.sl = \frac{[1/l(k_{mh}, k_m)] \times (k_m.sl)}{\sum_{k'_{mh} \in \mathrm{NPS}} 1/l(k'_{mh}, k_m)} \tag{12}$$

where h represents the size of the critical point set, $k_{mh}.sl$ represents the PSL assigned to the node k_{mh}, and $l(k_{mh}, k_m)$ represents the distance between node k_h and node k_m.

With the change of time, the PSL of sensitive locations will also change dynamically. This change may increase or decrease from the initial value, but some areas are unsuitable for change due to their very high level of privacy security. Therefore, the initial hierarchical threshold parameter μ is introduced. When the PSL in your area is greater than the initial level threshold parameter μ, the level size does not change. Directly substitute into (12) to calculate the PSL of its neighboring nodes. When the PSL in the region is less than the initial level threshold parameter μ, substitute into (13) to calculate the new PSL of the node. It changes dynamically over time, but if the PSL of the node is too small to protect privacy. Adjusting the threshold parameter ϕ in (13) plays a role in adjusting the size of PSL. Finally, the final PSL of the node is substituted into (12) to get the PSL of the neighboring nodes.

$$k_m.sl = \begin{cases} k_m.sl, & k_m.sl \geqslant \mu, \\ k_m.sl \times |\sin t| + \phi, & k_m.sl < \mu. \end{cases} \tag{13}$$

The privacy budget corresponding to adjacent nodes can be calculated from (10), and the results are as

$$\varepsilon_{mh} = \frac{\lambda}{k_{mh}.sl}. \tag{14}$$

In traditional computing methods, the security level of adjacent nodes only concerns the degree of sensitivity and the distance between critical points. It assumes that adjacent nodes closer to sensitive nodes have higher levels, but ignores the timeliness of privacy security levels. In this paper, time factors are taken into account, and the initial level threshold parameters μ and adjustment threshold parameters ϕ are set to adjust the PSL, and the dynamic privacy security level is used to allocate the corresponding privacy budget. It can not only improve the quality of services for users, but also enhance the value of private data mining for service providers. According to this idea, an PSL algorithm is proposed, and the pseudocode is shown in Algorithm 1.

In the PSL algorithm, the PSL threshold parameter ζ plays a regulatory role. When $\zeta = 0$, all locations on the map will execute a privacy algorithm, while when $\zeta = 1$, only the most sensitive locations will execute a privacy algorithm. The difference between this paper and the traditional suppression strategy is that it is a location-disrupted differential privacy protection method, and does not completely prohibit publishing.

Algorithm 1 PSL Algorithm

Require: Total privacy budget ε, map $U = \langle V, E \rangle$, area division H, time t, parameters μ, ϕ, ξ.
Ensure: Sensitive area set and budget ε^*.

Initialize ε
for $m = 1$ to n **do**
  $\varepsilon_m = \frac{\varepsilon}{\zeta(p)} \times \frac{1}{m^p}$
  GetNoise(ε_m)
end for
$ML = \text{ML}^{\text{initial}}$
$k_m = ML$.head()
while $k_m \neq$ NULL **do**
  if $k_m.sl \leq \mu$ **then**
    new $k_m.sl = k_m.sl$
  else
    new $k_m.sl = k_m.sl \times |\sin t| + \phi$
  end if
  NPS = findNPS(new k_m) {Get adjacent nodes}
  for all $k_{mh} \in$ NPS **do**
    if $k_m.sl \leq \mu$ **then**
    end if
    newsl = allocPrivLevel(k_{mh})
    if newsl $< \xi$ **then**
    end if
    if $k_{mh} \in ML$ **then**
      $k_{mh}.sl = \max(k_{mh}.sl,$ newsl)
    else
      $k_{mh}.sl =$ newsl
    end if
    $\varepsilon_{mh} = \frac{\lambda}{k_{mh}.sl}$
    ML.append(ε_{mh})
  end for
  $k_m = k_m$.next()
end while
return ML

C. Algorithm Analysis

1) Convergence Analysis: The PSL algorithm achieves personalized allocation for sensitive points combined with Pseries. The convergence of the allocation scheme combined with P-series is proved below. For P-series, there are

$$\lim_{n\to\infty} \frac{u_{n+1}}{u_n} = \lim_{n\to\infty} \left(\frac{n}{n+1}\right)^p < 1. \tag{15}$$

As can be seen from the ratio convergence method, $\sum_{n=1}^{\infty} \frac{1}{n^p} = 1 + \frac{1}{2^p} + \frac{1}{3^p} + ... + \frac{1}{n^p} + ...(p > 1)$ converges, so the privacy budget $\epsilon = \sum_{m=1}^{n} \epsilon_m = \sum_{m=1}^{n} \frac{\varepsilon}{\zeta(p)} \times \frac{1}{m^p} = \frac{\varepsilon}{\zeta(p)} \sum_{m=1}^{n} \frac{1}{m^p}$ converges, where $\sum_{m=1}^{\infty} \frac{1}{m^p} = \zeta(p), p > 1, 1 < n < \infty$.

2) Complexity Analysis: In the PSL algorithm, the initial sensitive points are allocated the privacy budget with P-series. The expansion term of P-series depends on the number of sensitive points. Therefore, the time complexity of the algorithm combined with the P-series to allocate the location privacy of sensitive points is $O(n)$. When allocating the privacy budget of adjacent locations of sensitive nodes, the security level needs to be determined by traversing the distance between sensitive points and other nodes in the graph, so the time complexity is $O(nm)$. Therefore, the overall time complexity of the algorithm is $O(nm)$. In summary, the complexity of this algorithm is positively linear with the number of data and will not increase significantly when applied to large-scale data operations.

IV. EXPERIMENTAL ANALYSIS

This paper analyses the impact of the initial level threshold parameter, the PSL threshold parameter and the initial sensitive set size on the running time of the algorithm in the two data sets. At the same time, the PSL algorithm is compared with the commonly used arithmetic sequence allocation method, uniform distribution allocation method, and geometry allocation method.

A. Experimental Environment Deployment

The main part of the PSL algorithm is implemented in Python programming language, which is installed and tested on Windows 10 platform with 3.60 GHz CPU and 16.00 GB RAM. The two travel datasets used in this analysis are the Geolife and Gowalla datasets. The Geolife dataset collected real data on 182 users'activities in Beijing from April 2007 to August 2012. The dataset contains 17 621 trajectories, and the attributes in the dataset include user numbers, timestamps, longitude, latitude, elevation, and so on. In this paper, the first five attributes of the trajectories within the six rings of Beijing are selected as new datasets. The Gowalla dataset collects data from 15 116 users who checked in on mobile social networking sites (California-wide) from February 2009 to October 2010. This paper extracts California-wide user numbers, timestamps, longitude, latitude, and elevation as new datasets.

For the PSL algorithm, the map is converted to an undirected graph representation, and the χ ($\chi \in [10, 20, 30, 40, 50, 60]$) areas with the longest user stay or most visits are used as the initial sensitive location set. The PSL corresponding to the initial sensitive location set is calculated from (11). The effects of PSL threshold parameterζ ($\zeta \in [0.2, 0.4, 0.6, 0.8, 1]$) and initial level threshold parameter μ ($\mu \in [0.1, 0.2, 0.3, 0.4, 0.5]$) on the running time of the algorithm are also analyzed.

The specific parameter settings are shown in Tab. I.

TABLE I: Experimental Parameters

Parameter	Description	Value Range
χ	Size of initial sensitive set	$10 \leq \chi \leq 60$
ξ	PSL threshold parameter	$0.2 \leq \xi \leq 1$
μ	Initial hierarchy threshold parameter	$0.1 \leq \mu \leq 0.5$

B. Experimental Analysis

Firstly, the impact of the initial sensitive set size on the PSL algorithm run time is analyzed, as shown in Fig. 6. In this experiment, the default, $\zeta = 0.2$, $\mu = 0.5$ is used.

As can be seen from Fig. 6, with the gradual growth of χ, the running time of the PSL algorithm on the two data sets increases accordingly. This is because when χ is large, the geographic space traversed by the algorithm also increases, resulting in an increasing amount of time required. At the same time, it can be seen from the figure that the running time on the Geolife dataset is slightly longer than that on the Gowalla dataset. The reason is that the geographic semantic set of Los Angeles is smaller than that in Beijing. When traversing the same situation, the Gowalla dataset takes less time.

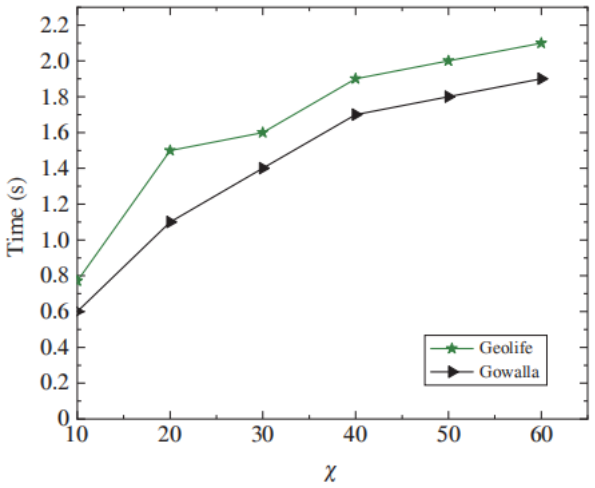


Fig. 6: Influence of χ on the running time of the PSL algorithm

Secondly, the impact of PSL threshold parameter ζ on the PSL algorithm running time is analyzed, and the result is shown in Fig. 7.

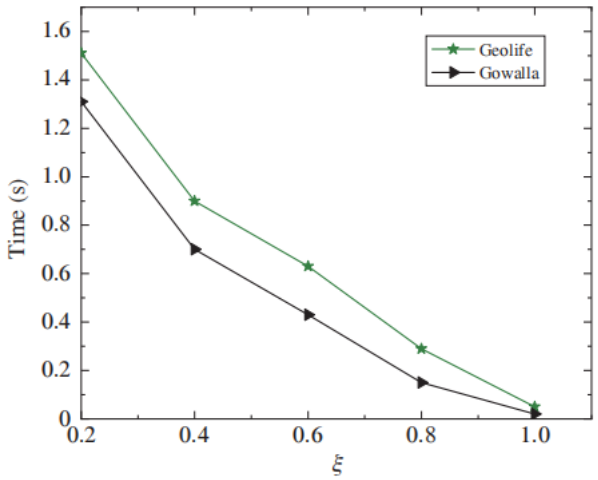


Fig. 7: Influence of ζ on the running time of the PSL algorithm

In this experiment, $\chi = 10$, $\mu = 0.5$ is set by default. As can be seen from Fig. 7, when ζ increases, the running time of the PSL algorithm on two data sets decreases. This is because when ζ gradually increases, the pruning effect of the PSL algorithm becomes increasingly obvious, so the efficiency of the PSL algorithm will gradually improve.

Thirdly, the effect of the initial hierarchical threshold parameter μ on the running time of the PSL algorithm is analyzed, as shown in Fig. 8.

In this experiment, $\chi = 10$, $\mu = 0.5$ is set by default. As can be seen from Fig. 8, the PSL algorithm runtime gradually decreases as μ increases. The main reason is that when μ increases, the probability of choosing the second scenario

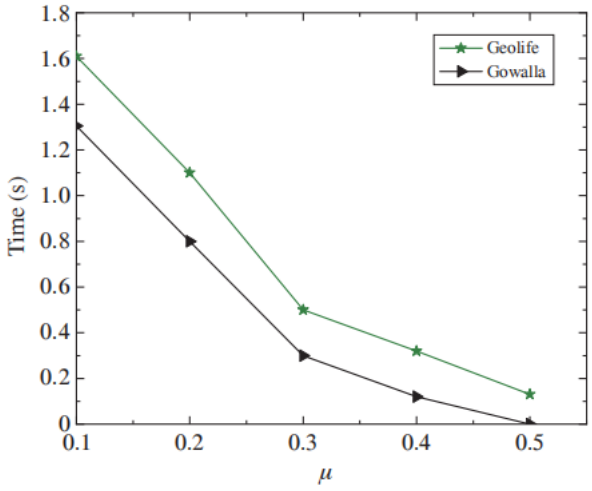


Fig. 8: Influence of μ on the running time of the PSL algorithm

increases in the calculation of (13). Then the PSL of the initial sensitive location changes dynamically, and the maximum value is the assigned PSL. As a result, the pruning operation becomes more and more effective in the PSL algorithm, so the efficiency of the algorithm is greatly improved.

Finally, this paper compares the PSL algorithm with the uniform distribution allocation method [14], the arithmetic sequence allocation method [18] and the geometric distribution allocation method [19] in the privacy budget, and analyses the total query error as shown in Fig. 9.

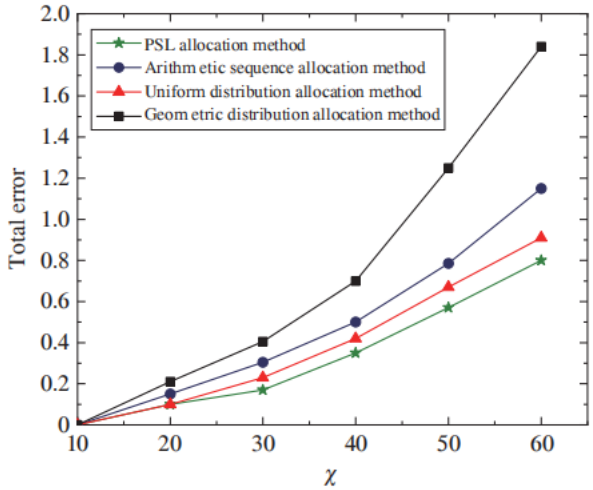


Fig. 9: Total error comparison under different allocation methods

In this experiment, $\chi = 10$, $\mu = 0.5$ is set by default. Take the average of the d values of the arithmetic sequence allocation method as 0, 0.01, 0.02, and 0.025. As can be seen from Fig. 9, the total errors of the four privacy budget allocation methods increase with the increase of the number of locations. Among them, the geometric distribution allocation method produces the largest total errors, the PSL algorithm produces the smallest total errors. The total errors produced by the arithmetic sequence allocation method and the uniform distribution method are similar, and they are slightly larger than the total errors produced by the PSL algorithm. Both the PSL algorithm and the arithmetic sequence allocation method can flexibly allocate the privacy budget according to the different demands of users for sensitive locations, but the PSL algorithm is more flexible. In the arithmetic sequence allocation method, the setting of d value is limited by the number of χ, which satisfies $0 \leqslant d \leqslant \frac{2}{[\chi(\chi+1)]}$. In the PSL algorithm, $p > 1$, which is not limited by the number of χ.

V. CONCLUSIONS

This paper presents a differential privacy budget allocation method that combines PSLs. Firstly, the total privacy budget is allocated to the initial sensitive location according to the idea of P-Series, and then the corresponding PSL is obtained by using the γ-model. Finally, the undirected graph is used to represent the topological relationship of geographical location, and the PSL of the initial sensitive location is allocated to the neighboring nodes according to the degree of nodes and the distance between nodes, to obtain its corresponding privacy budget. It effectively avoids the hidden danger of attackers by using the geographical relationship between locations to infer sensitive locations.

REFERENCES

[1] SINGH S, RATHORE S, ALFARRAJ O, et al. A framework for privacy-preservation of IoT healthcare data using federated learning and blockchain technology[J]. Future Generation Computer Systems, 2022, 129: 380-388.

[2] LIN J, NIU J, LIU X, et al. Protecting your shopping preference with differential privacy[J]. IEEE Transactions on Mobile Computing, 2021, 20(5): 1965-1978.

[3] JIN R, HE X, DAI H. Minimizing the age of information in the presence of location privacy-aware mobile agents[J]. IEEE Transactions on Communications, 2021, 69(2): 1053-1067.

[4] LI M, CHEN Y, ZHENG S, et al. Privacy-preserving navigation supporting similar queries in vehicular networks[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 1133-1148.

[5] ZHANG K, TIAN Z, CAI Z, et al. Link-privacy preserving graph embedding data publication with adversarial learning[J]. Tsinghua Science and Technology, 2022, 27(2): 244-256.

[6] TIAN Y, ZHANG Z, XIONG J, et al. Achieving graph clustering privacy preservation based on structure entropy in social IoT[J]. IEEE Internet of Things Journal, 2022, 9(4): 2761-2777.

[7] WEI J H, LIN Y P, YAO X, et al. Differential privacy-based location protection in spatial crowdsourcing[J]. IEEE Transactions on Services Computing. 2022, 15(1): 45-58.

[8] YANG Z, WANG R, WU D, et al. Local trajectory privacy protection in 5G enabled industrial intelligent logistics[J]. IEEE Transactions on Industrial Informatics, 2022, 18(4): 2868-2876.

[9] LIU B, DING M, SHAHAM S, et al. When machine learning meets privacy: a survey and outlook[J]. ACM Computing Surveys, 2021, 54(2): 1-36.

[10] PARMAR D, RAO U P. Towards privacy-preserving dummy generation in location-based services[J]. Procedia Computer Science, 2020, 171: 1323-1326.

[11] DWORK C. Calibrating noise to sensitivity in private data analysis[J]. Lecture Notes in Computer Science, 2012, 3876(8): 265-284.

[12] MIN M, WANG W, XIAO L, et al. Reinforcement learning-based sensitive semantic location privacy protection for VANETs[J]. China Communications, 2021, 18(6): 244-260.

[13] KIM J W, EDEMACU K, KIM J S, et al. A survey of differential privacy-based techniques and their applicability to location-based services[J]. Computers and Security, 2021, 111: 1-29.
[14] CHEN R, FUNG B C M, DESAI B C, et al. Differentially private transit data publication: a case study on the montreal transportation system[C]//Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM, 2012: 213-221.
[15] CHEN R, ACS G, CASTELLUCCIA C. Differentially private sequential data publication via variable-length n-grams[C]//Proceedings of the 2012 ACM conference on Computer and communications security. New York: ACM, 2012: 638-649.
[16] LI X, QIN B, LUO Y, ZHENG D. A differential privacy budget allocation algorithm based on out-of-bag estimation in random forest[J]. Mathematics, 2022, 10(22): 4338.
[17] ZHANG K, TIAN J, XIAO H, et al. A numerical splitting and adaptive privacy budget allocation based LDP mechanism for privacy preservation in blockchain-powered IoT[J]. IEEE Internet of Things Journal, 2022, early access.
[18] WANG C, ZHANG L, ZHANG C. Privacy protection method for random transformation of adjacent sensitive areas[J]. Application Research of Computers, 2020, 37(10): 3083-3085+3090.
[19] CORMODE G, PROCOPIUC C, SRIVASTAVA D, et al. Differentially private spatial decompositions[C]//Proceedings of IEEE 28th International Conference on Data Engineering. Piscataway: IEEE Press, 2012: 20-31.
[20] LIU H, WU Z, PENG C, et al. Bounded privacy-utility monotonicity indicating bounded tradeoff of differential privacy mechanisms[J]. Theoretical Computer Science, 2020, 816: 195-220.
[21] WU Y, CHEN H, ZHAO S, et al. Differentially private trajectory protection based on spatial and temporal correlation[J]. Jisuanji Xuebao/Chinese Journal of Computers, 2018, 41(2): 309-322. ABOUT THE AUTHORS