Research Article

Travel Trajectory Frequent Pattern Mining Based on Differential Privacy Protection

Weiya Wang^{1,2}, Geng Yang¹, Lin Bao², Ke Ma¹, Hao Zhou¹ and Yunlu Bai¹

¹School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China

²China Telecom Jiangsu Branch, Nanjing, China

Correspondence should be addressed to Geng Yang; yangg@njupt.edu.cn

Received 11 May 2021; Revised 24 June 2021; Accepted 7 July 2021; Published 10 August 2021

Academic Editor: Cong Pu

Copyright © 2021 Weiya Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Now, many application services based on location data have brought a lot of convenience to people's daily life. However, publishing location data may divulge individual sensitive information. Because the location records about location data may be discrete in the database, some existing privacy protection schemes are difficult to protect location data in data mining. In this paper, we propose a travel trajectory data record privacy protection scheme (TMDP) based on differential privacy mechanism, which employs the structure of a trajectory graph model on location database and frequent subgraph mining based on weighted graph. Time series is introduced into the location data; the weighted trajectory model is designed to obtain the travel trajectory graph database. We upgrade the mining of location data to the mining of frequent trajectory graphs, which can discover the relationship of location data from the database and protect location data mined. In particular, to improve the identification efficiency of frequent trajectory graphs, we design a weighted trajectory graph support calculation algorithm based on canonical code and subgraph structure. Moreover, to improve the data utility under the premise of protecting user privacy, we propose double processes of adding noises to the subgraph mining process by the Laplace mechanism and selecting final data by the exponential mechanism. Through formal privacy analysis, we prove that our TMDP framework satisfies ϵ -differential privacy. Compared with the other schemes, the experiments show that the data availability of the proposed scheme is higher and the privacy protection of the scheme is effective.

1. Introduction

With the rapid development in the field of artificial intelligence and big data technology, data mining and data analysis have become an important tool for researchers to extract useful knowledge from data. Location data is a kind of large-scale and fast-changing location information, which mainly comes from automobile network, mobile devices, and social networks. As an important part of mobile Internet services, location data mining and analysis have brought unprecedented changes and convenience to people's work and life. With the increasing popularity of personal location information, people are increasingly using the system of recording and processing location data, which is usually called "location-based system." These systems include (a) location-based services (LBSs), in which a user obtains, typically in real-time, a service related to his current location, and

(b) location data mining algorithms, used to determine points of interest and traffic patterns. A large number of applications of location data provide convenience for people's daily life, so location data service is called a new type of mobile computing service. For example, the personalized recommendation based on LBSs can recommend the interesting content to users according to their location POIs (points of interest) and movement trajectory [1]. Most mobile Internet services are based on the combination of location data with social data (personal information, social relationship information, etc.), and the crossduplication of these data will lead to privacy leakage problem [2]. Privacy-preserving data mining (PPDM) is a data mining method to reduce the possibility of sensitive data leakage. PPDM encrypts and sanitizes sensitive information; however, if the location data is "excessively" protected, it is difficult for users to obtain relatively accurate services [3, 4].

How to protect sensitive information while providing location data mining service is the key to the current location data privacy protection. The traditional approach of PPDM considers only how to effectively hide sensitive information through heuristics which try their best to minimize side effects of a known NP-hard problem. In PPDM, a complex process of data sanitization using deletion operations on transactions is the most common approach to hide confidential information. The existing location privacy protection schemes mainly include location anonymity, location ambiguity, location transformation, and differential privacy. Among them, the location anonymity scheme [5] is to add the false location to the location data, so that the user's real location is difficult to distinguish from other false location, to achieve the purpose of protecting the user's location privacy, such as the more commonly used K -anonymous scheme [6, 7]. In the location ambiguity scheme [8], the true location of the user is generalized to an area or replaced by other locations around it, which makes it difficult for the attacker to determine the true location of the user, to protect the privacy of the user's location. In the location transformation scheme, the user location is transferred to a different location, so the location service provider cannot obtain the user's true location. In the differential privacy scheme, controllable random noise is mainly added to protect the user's location data security [9, 10]. Most of the above schemes are aimed at protecting the discrete location data records in the location database. However, at the present stage, many Internet application services, such as Uber and Keep, provide information services based on the continuous movement trajectory of people; therefore, this paper is aimed at finding an efficient protection scheme for user trail frequent pattern mining. At present, many methods cannot balance the availability of location data with the protection of trajectory frequent pattern mining. Therefore, it is necessary to combine the characteristics of mobile location data and user trajectory to design an effective data mining privacy protection scheme suitable for location data. This method can handle the contradiction between privacy protection and quality of service well so that users can obtain relatively safe, more accurate, and convenient Internet data services [11].

2. Contributions

In this paper, we propose a differential privacy protection scheme for location data records based on the trajectory model frequent pattern mining to protect high-frequency access location data (which is related to the user's location preference). Our contributions are as follows:

- (1) In this scheme, we construct the user trajectory model from the database. Such a method has the following advantages: (a) it can intuitively describe the relationship of each path point and the importance of each path point in the user trajectory (as shown in Figure 1); (b) it can improve the protection effectiveness of location data: when we add the noises to the corresponding graph nodes, it may protect the

FIGURE 1: Trajectory graph model G_{example} based on the trajectory in Figure 4.

relationship of location data and the combination of location data and accessing frequency

- (2) In the differential privacy protection phase, we make double processes of selecting data. The first selection is based on the Laplace mechanism; noises are added into the accessing frequencies of the candidate sub-graph by the Laplace mechanism. The second selection is based on the exponential mechanism; the frequent trajectory graph with the final differential privacy protection is selected from the candidate subgraphs

3. Related Work

Currently, the existing location data privacy protection methods are mainly classified to three categories: the heuristic privacy-measure methods, the probability-based privacy inference methods, and the privacy information retrieval methods. The heuristic privacy-measure methods are mainly to provide the privacy protection measure for some no-high required users, such as K -anonymity [12], t -closing [13], m -invariability [14], and l -diversity [15]. Although the mechanisms such as anonymization [6] and spatial obfuscation [16] provide location privacy through hiding identity or reporting fake locations, they ignore adversary's knowledge about user's access pattern and LPPM algorithm and disregard the optimal attack where an attacker may design in an inference attack to reduce his calculation error. The information retrieval privacy protection methods may result in no data that can be released, and these methods have high overhead. Further, the three kinds of method are based on a unified attack model [17], which depends on certain background knowledge to protect location data. The works [18, 19] showed the shortages of the relationship-privacy protection methods. Gedik and Liu [20] proposed the first effective location-privacy preserving mechanism (LPPM) that enables a designer to find the optimal LPPM for a location-based service. Such an LPPM can maximize the expected distortion (error) when the adversary incurs in reconstructing the actual location of a user. Shokri et al. [21] propose an

optimization framework to determine the most optimal LPPM against the most effective/optimal inference attack.

However, with the deepening of the research, it is not enough to protect the user's location data only. Therefore, a trajectory privacy protection method is proposed based on location privacy protection. At present, the privacy protection methods of trajectory data mainly include the following: (1) the protection method based on K -anonymous generalization, (2) the privacy protection method based on dynamic kana, (3) the privacy protection method based on noise data, (4) the protection method based on differential privacy, (5) the privacy protection method based on similarity matrix, and (6) locality sensitive hashing. The track privacy protection method based on K -anonymous generalization is mainly to treat continuous n queries as n independent LBS service requests, that is, to construct anonymous space for these n queries, respectively. Spatial and temporal cloaking [6] is the main technology to realize this method. It uses a spatial region CR(Clocking Region) to replace the real location of the user. Or by delaying the response time, the number of users in the empty area is large enough that the attacker cannot deduce the real user. Cheng et al. proposed schemes delaying and patching [22], where delaying is realized by delaying the request and patching is realized by expanding the camouflage area. However, this method reduces the service quality of LBS. Xu and Cai proposed a K -anonymous hidden area generation algorithm, KAA, which supports continuous spatial query [7]. The vicinity information can then be used by the adversary to carry out inference attack, background knowledge attack, or center-of-ASR attack in K -anonymous principle. Gupta and Rao develop a vicinity protection technique, called VIC-PRO, that strengthens the location privacy of the user [23].

The main idea of the trajectory privacy protection method based on dynamic kana is that the user uses the pseudonym to replace the real identity when sending an LBS request. However, mobile users are real and visible in the geographic space, and using the same pseudonym for a long time cannot effectively protect users' privacy. Freudiger et al. proposed a dynamic kana technology based on the mixed region (mix-zone) in literature [24], which constructed the mixed region according to the background knowledge that the attacker may use, such as the starting point of the trajectory, moving speed, and other information. Deploying too many mixed zones, however, can seriously affect the quality of service. Xu et al. proposed a deployment scheme through multiple mix-zones [25]. However, the above method requires a third-party organization to modify the user's pseudonym centrally, and the reliability of the third party determines the effectiveness of this method.

The main idea of the trajectory privacy protection method based on noise data is that several false locations (user peripheral location) are included in LBS request to protect the real location so that the attacker cannot confirm the real location of the user. However, the quality of privacy protection and service is related to the distance between true and false locations. The degree of privacy protection and service quality of this method is related to the distance between true and false locations. Kido et al. randomly moved the real posi-

tion as a false position [26], but the false position generated by the noise mechanism was not consistent with the real movement characteristics of the user, and it was easy for attackers to distinguish the true and false positions. To solve this problem, Suzuki et al. added constraints such as moving speed and road network when generating false positions [27]. Kato et al. believed that mobile users would not move continuously all the time [28], so when generating noisy data points, the moving object would randomly pause according to the surrounding environment, so as to prevent attackers from distinguishing true and false positions. However, the above methods are not suitable for the attackers who have more background knowledge, and it is difficult to design a good noise mechanism.

Presently, it is the key of protecting location data to provide a privacy protection method that is not sensitive to background knowledge. Based on the requirement, differential privacy protection technology can exactly satisfy it. Literature [29] proposed the differential privacy method for the first time to solve the problem of privacy protection in the release of trajectory data of the release scale. The method uses a prefix tree to store the trajectory data and Laplacian noise mechanism to add the noise value to the nodes. The method supports the technique and frequent pattern query of the trajectory data. Hua et al. proposed a differential privacy algorithm based on spatial generalization for the first time [30]. This method solves the requirement that trajectories must have the same prefix in most current research methods. However, moving trajectories in the road network generally have time correlation. If these correlations are ignored, privacy will be compromised. Therefore, Xiao and Xiong proposed a differential privacy protection technology based on " δ -location set" [31], which used the function sensitivity measurement method and the location disturbance mechanism to hide the sensitive locations in the location set, so as to achieve the purpose of privacy protection. This method ignores the relevance of user location points and is vulnerable to a large number of inference attacks. Chatzikokolakis et al. [32] showed a formal notion of privacy that protects the user's exact location—"geoindistinguishability." In [32], they proposed two mechanisms to protect the privacy of user when dealing with location-based services. They extended their mechanisms to limit the degradation of the privacy guarantees due to the correlation between the points. However, when the cumulative privacy parameters of differential privacy exceed the given privacy budget, it will lead to privacy leakage. Therefore, when the number of users' queries accumulates to a certain extent, the effective duration of privacy protection cannot be guaranteed. Wang et al. [33] proposed a real-time spatiotemporal crowd-sourced data publishing scheme with differential privacy. Gupta and Rao propose a mechanism that provides a three-layer iterative RDV masking that exploits the basic geometry mathematics concept of Delaunay triangulation and Voronoi polygon formation to geo-mask the location attribute value before publishing the record [34].

The privacy protection method based on similarity matrix takes outline of information acquired for a client as a similarity matrix. Similarity matrix demonstrates the

similarity of the result set respecting the client's present location. Dewri and Thurimella [35] display a novel trusted intermediate server-based framework for LBS applications.

Locality sensitive hashing is a proficient mechanism algorithm to enhance the scalability of geofencing. It comprises two principal phases. In the first stage, in order to find out whether the point of concern is within the rectangle with the smallest boundary, an R tree is established for rapid recognition. In the second phase, an edge-based locality sensitive hashing arrangement is outlined and adjusted to the crossing point number calculation.

4. Preliminaries

4.1. Frequent Graph Pattern Mining. A graph $G = (V, E)$ is made of two sets, the set of vertices V and the set of edges E . Each vertex is associated with a label, which is drawn from a set of vertex labels. Weighted graph is expressed as $G = (V, E, W)$, where W is the weight set of edges. The weight represents the degree of relationship between two nodes in the graph, such as the degree of connection between two location points or the amount of transaction between businesses.

Suppose there are two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Let $L(u)$ denote the label of vertex u . We say that G_1 is contained in G_2 if there exists a function $f : v_1 \rightarrow v_2$, such that $\forall (u, v) \in E_1$; we have $(f(u), f(v)) \in E_2$, $L(u) = L(f(u))$, and $L(v) = L(f(v))$. If G_1 is contained in G_2 , we say that G_1 is a subgraph of G_2 , and G_2 is a supergraph of G_1 , denoted by $G_1 \subseteq G_2$.

A graph database $GD = \{G_0, G_1, \dots, G_n\}$ is a multiset of input graphs, where each input graph represents an individual's record. $\text{sup}(g)$ is the support of graph g which is the number of graphs in GD that contain g . Given a threshold, a graph is called frequent if its support is no less than this threshold.

Definition 1. Noisy support. Add the noise to the support of subgraph g . $\text{nsup}(g) = \text{sup}(g) + \text{noise}$, where $\text{nsup}(g)$ represents the noisy support of subgraph g . In this paper, we add Laplace noise to the subgraph.

Definition 2. Frequent subgraph mining (FGM). Given a graph database and a threshold, FGM is aimed at finding all frequent subgraphs for the threshold and computing the support of each frequent subgraph [36, 37].

Definition 3. Subgraph isomorphism. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if they are topologically identical to each other; that is, there is a mapping from V_1 to V_2 such that each edge in E_1 is mapped to a single edge in E_2 and vice versa. In the case of labeled graphs, this mapping must also preserve the labels on the vertices and edges. Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the problem of subgraph isomorphism is to find an isomorphism between G_2 and a subgraph of G_1 , i.e., to determine whether or not G_2 is included in G_1 . As shown in Figures 2 and 3, G_2 is a subgraph of G_1 , and there is a subgraph isomorphism relationship between the two graphs.

FIGURE 2: Graph G_1 .

FIGURE 3: Graph G_2 .

4.2. Differential Privacy. Differential privacy [38] is a recent privacy model which provides a strong privacy guarantee. The main idea of differential privacy is that after adding or deleting a record to the database, there is almost no difference in the output of the same algorithm applied to the database. Formally, differential privacy is defined as follows.

Definition 4. ϵ -Differential privacy. A private algorithm A gives ϵ -differential privacy if for any neighboring databases D_1 and D_2 , and for any possible output $S \in \text{Range}(A)$,

$$\Pr [A(D_1) = S] \leq e^\epsilon \times \Pr [A(D_2) = S], \quad (1)$$

where $\Pr$ represents the probability and $\text{Range}(A)$ represents the value range of the output result of algorithm A .

Definition 5. Sensitivity. For any neighboring databases D_1 and D_2 , the sensitivity of f is

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|. \quad (2)$$

Currently, differential privacy protection has two main methods [39]: the Laplace mechanism [40] and the exponential mechanism [41].

Theorem 6. Laplace's mechanism. The Laplace mechanism is to add noise following Laplace distribution to the output of the algorithm, so that the algorithm meets the differential privacy protection. For any function $f : D \rightarrow \mathbb{R}^n$ with sensitivity Δf , the algorithm A

$$A(D) = f(D) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right). \quad (3)$$

The Laplace distribution with magnitude λ , i.e., $\text{Lap}(\lambda)$, follows the probability density function as $\text{Lap}(\lambda) = (1/2\lambda)e^{-|x|/\lambda}$, where $\lambda = \Delta f/\epsilon$ is determined by the sensitivity Δf and the privacy budget ϵ .

FIGURE 4: Location data source. The trajectory of an anonymous user in the real world.

Theorem 7. Exponential mechanism. The exponential mechanism uses a utility score function u to measure all possible results in the output space of the algorithm, then select an element x from the output space with the probability of Equation (4) as the output result.

$$\Pr [A(D) = x] \propto \exp \left(\frac{\varepsilon \times u(D, x)}{2\Delta u} \right). \quad (4)$$

According to the scoring function, each x of the output space is assigned a weight $u(D, x)$. After the weight is amplified by the exponential function, the result is selected according to the weight.

Theorem 8. Sequential composition. For the same data set, if the whole privacy protection process is divided into the different privacy protection algorithms $(A_1, A_2, \dots, A_n)$ whose privacy protection levels are $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$, then the privacy protection level $\sum_{i=1}^n \varepsilon_i$ of the whole process needs to satisfy differential privacy protection.

Theorem 9. Parallel composition. For the disjoint data set, if the whole privacy protection process is divided to the different privacy protection algorithms $(A_1, A_2, \dots, A_n)$ whose privacy protection levels are $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$, then the privacy protection level $\max(\varepsilon_i)$ of the whole process needs to satisfy differential privacy protection.

5. The Location Data Privacy Protection Scheme

In this section, we proposed a location data record privacy protection scheme based on a weighted graph model. In the proposed scheme, we first build the user's location database based on the user's location data record (Figure 4), and then, we construct the user travel trajectory model according to the characteristics of mobile location data. After introducing the weighted graph model to construct the complete user travel trajectory graph database, we propose a three-stage differential privacy protection algorithm based on subgraph edge

growth. In the first stage, standardize and code the user travel trajectory graph database, and sort it according to the number of edges. In the second stage, the subgraph data sets generated by the edge-growing pattern are added with Laplace noise. After threshold judgment and selection, candidate frequent graph data sets are obtained. In the third stage, this algorithm generates frequent subgraph data sets by the exponential mechanism (Figure 5).

The main steps of the user trajectory graph privacy protection framework are described as follows:

Step 1. Design weighted trajectory graph model according to user's trajectory characteristics.

Step 2. Construct the trajectory graph database GD based on the trajectory database $\text{TrDatabase} = \{\text{Tr}1, \text{Tr}2, \dots, \text{Tr}n\}$.

Step 3. Add Laplace noise to the candidate subgraph for the first screening, introduce the exponential mechanism to carry out the second screening, and obtain the final result.

5.1. Establishment of Trajectory Model Graph. The user trajectory database is preliminarily established through the starting time of movement, the stay time, and the stop position of the user movement location database, as Table 1 shows.

The trajectory: β is set as the user residence time threshold based on the regularly reported user location data. When the user stops at a certain location for longer than β , the location is determined to be the user stop position: $n_i = \{p, a\}$ ($1 \leq i \leq k$), where i represents the sequential value of the sampling positions in the trajectory. p represents the stop position; there are three values of a : (1) if $i = 1$, a is the "starting position"; (2) if $1 < i < k$, a is the "passing position"; and (3) if $i = k$, a is the "end position." The movement trajectory of the user in the sampling period r is denoted as $\text{Tr} = \{n_1, n_2, n_3 \dots n_k\}, k = |\text{Tr}|$.

Trajectory model graph: the trajectory graph $G = \{V, E, L\}$ is constructed by the user trajectory. V is the set of user's

FIGURE 5: TMDP framework. The trajectory model is built, and weighted frequent graph mining based on differential privacy algorithm is introduced.

TABLE 1: Location data sample.

Encrypted id	Longitude	Latitude	City	Grid	Time (m)
W90H0925	118.803655	32.067106	Nanjing	0203	10/1/14:30
W90H0925	118.784718	32.073593	Nanjing	0209	10/1/16:38
W90H0925	118.788294	32.066525	Nanjing	0221	10/1/17:00
W90H0925	118.797348	32.072255	Nanjing	0347	10/1/18:26
W90H0925	118.796001	32.064865	Nanjing	0309	10/1/19:55
W90H0925	118.788294	32.066525	Nanjing	0221	10/1/22:24

stop position $V = \{v_1, v_2, \dots, v_m\} (m = |V|)$, which represents the area where the user stays, such as school, company, and bank, as shown in Figure 1.

$v_i = \{id, s_sum, m_num, e_num\}$, where id is the code of the user's stop position. Between the user's two stop position is a path. s_sum represents the number of paths with id as a "starting position." m_num is the number of paths with id as a "passing position." e_num is the number of paths with id as an "end position." $E = \{e_1, e_2, \dots, e_m\}$ is the set of edges; E represents the user's paths between id . The weight of e is the sum of the number of trajectories in this path; for example, for edge $e_1 = \langle v_1, v_2 \rangle$, if there is a trajectory Tr which $n_i = v_1$ and $n_{i+1} = v_2$, then the weight of edge e_1 add one.

This section proposes an algorithm (the TGDC algorithm) to construct a trajectory model graph from the trajectory data of users' travel. Firstly, the set of all "starting position" in the trajectory database will be obtained, and every starting position corresponds to a travel trajectory subgraph. After traversing all the trajectory data, store the trajectory subgraph corresponding to the "starting position" as Tr . For each trajectory, positions and edges will be added to the trajectory model to construct the trajectory graph database. After executing the algorithm, label function is applied to the graph structure to obtain the label value of the final point and edge. Finally, we will get the user's trajectory graph database.

5.2. Trajectory Frequent Graph Support Calculation Algorithm. In the frequent pattern mining of users' trajectories, the primary focus of our research is the relationship between the stop positions in users' moving trajectories and the frequent graph structure constructed by the trajectories. Therefore, we simplify the study of users' trajectory model into the weighted undirected graph study.

Canonical code of trajectory subgraph: in frequent subgraph mining, graph data is usually represented by adjacent matrix. There are three kinds of adjacent matrices in general use: (1) node-centered; (2) edge-centered; and (3) the combination of nodes and edges.

The purpose of trajectory frequent subgraph is not only to discover frequent subgraph but also to find implicit relationship between user's trajectories. In this paper, an edge-centered adjacent matrix representation method is adopted. The definition of adjacent matrix is given below.

$G = \langle V, E \rangle$ has m nodes, which is $V(G) = \{v_1, v_2, \dots, v_m\}$; the m -order matrix $A(G) = (x_{ij})$ is called the adjacent matrix of G , as Figure 6 shows.

$$x_{ij} = \begin{cases} e, & \text{there is an edge } e \text{ between } v_i \text{ and } v_j, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

In order to improve the efficiency of the algorithm, the upper triangular matrix of the adjacency matrix is selected according to the symmetry of the matrix. Different orders of nodes will generate different adjacent matrices; if two graphs are isomorphic to each other, they will be assigned the same code. To ensure the uniqueness of the adjacency matrix [42], we introduce a simple way of defining the canonical label of a graph that is as the string obtained by concatenating the upper triangular entries of the graph's adjacency matrix when this matrix has been symmetrically permuted so that this string becomes the lexicographically largest (or smallest) over the strings that can be obtained from all such permutations.

Let

$$X = \begin{Bmatrix} x_{11} & \dots & x_{1m} \\ & \ddots & \\ x_{m1} & \dots & x_{mm} \end{Bmatrix} \quad (6)$$

be an adjacent matrix of graph G , and the code of matrix defining X is $\text{Code}(X) = x_{12}x_{13} \dots x_{1m}x_{23} \dots x_{2m} \dots x_{m-1m}$.

The canonical matrix encoding of graph X is $\text{CanonicalCode}(X) = \max(\text{Code}(X))$. Therefore, if two graphs are isomorphic, they must have the same *CanonicalCode* [43].

In order to reduce the computational complexity of canonical matrix coding and improve computational efficiency, a fast coding strategy (*FastCanonicalCode(X)*) is

Input: Trajectory database $TrDatabase = \{Tr1, Tr2, \dots, Trn\}, Tr = \{n_1, n_2 \dots n_k\}$
Output: Trajectory Graph Database $GD = \{G_1, G_2, \dots, G_n\}$

```

1. for each  $v = Tr.start\_node$  in  $TrDatabase$ : /* Summarize all trajectory data by traversing the starting points  $v = Tr.start\_node$  in  $TrDatabase$  */
2.    $Vstart = null$ 
3.   if  $v$  not in  $Vstart$ :
4.      $Vstart += v$ 
5.   for each  $v$  in  $Vstart$ :
6.      $Trs = Null$ 
7.     for each  $Tr$  in  $TrDatabase$ :
8.       if  $Tr.start\_node = v$ :
9.          $Trs += Tr$ 
10.    end for
11. /*For each trajectory, points and edges are added to the trajectory graph to construct a complete graph structure */
12. for each  $Tri$  in  $Trs$ :
13.   for  $j$  in range ( $len(Tri) - 1$ ):
14.     if  $Tri.n[j]$  not in  $G$ :
15.        $G.add\_node(Tri.n[j])$ 
16.     if  $Tri.n[j + 1]$  not in  $G$ :
17.        $G.add\_node(Tri.n[j + 1])$ 
18.     Edge ( $Tri.n[j], Tri.n[j + 1]$ ).w += 1 /* Add edge weight to the graph */
19.   end for
20.    $GD += G$ 
21. end for
22. return  $GD$ 

```

ALGORITHM 1: Trajectory graph database construction.

	V_0	V_1	V_2	V_3	V_4	V_5
V_0	0	e_1	0	0	0	0
V_1	e_1	0	e_2	e_3	0	0
V_2	0	e_2	0	e_4	0	e_5
V_3	0	e_3	e_4	0	e_6	0
V_4	0	0	0	e_6	0	0
V_5	0	0	e_5	0	0	0

FIGURE 6: The adjacent matrix of $G_{example}$.

proposed. The adjacent matrix of the graph is arranged according to the degree of nodes (that is, the number of edges associated with V nodes in the graph) from high to low and in a lexicographic order, and the result is taken as the *Fast CanonicalCode* of the adjacent matrix. Taking the adjacent matrix of graph $G_{example}$ in Figure 6 as an example, the degree of 6 nodes (v_0, v_1, v_2, v_3, v_4 , and v_5) in the graph is 1, 2, 3, 3, 1, and 1, respectively [43]. According to the principle of *Fast CanonicalCode*, the new node ordering is: $v_2(3), v_3(3), v_1(2), v_0(1), v_4(1)$, and $v_5(1)$ as Figure 7 shows. The *Fast CanonicalCode(X)* is $e_4e_200e_5e_30e_60e_1000$. The isomorphism of graphs can be equivalent to the calculation of *CanonicalCode*; the general method is to enumerate all the adjacent matrix, taking Figure 1 as an example; the algorithm complexity is $6! = 720$. By using this strategy, the nonzero elements in the adjacent matrix can appear in front of the subgraph composed of matrix code as much as possible; the

algorithm complexity is $2! \times 3! = 12$, thus accelerating the matching speed of the candidate subgraph coding in the graph transaction coding subgraph. It greatly reduces the search space of graph, and the new sorting strategy can accelerate the calculation of graph isomorphism and support.

Weighted frequent trajectory graph: traditional subgraph pattern mining algorithms treat all frequent subgraph patterns equally, but in real user trajectories, different user trajectory subgraphs have different importance. In order to solve these problems, this paper proposes a frequent subgraph mining algorithm based on edge weight.

Definition 10. Edge weight. The number of paths through edge e_i in the user trajectory graph is the edge weight $W(e_i)$.

Definition 11. The weight of the trajectory subgraph. If the subgraph G consists of m edges $e_1, e_2, \dots, e_m$, then the weight of the trajectory subgraph G is defined as $W(G) = \sum_{k=1}^m W(e_k)/m$.

Definition 12. Average total weighting (ATW). For the weight processing of weighted frequent subgraph mining, there are generally three edge weighting schemes for controlling candidate set generation [44]: (i) average total weighting (ATW), (ii) affinity weighting (AW), and (iii) utility-based weighting (UBW). ATW is more suitable for the actual situation in the mining of weighted frequent trajectory graph. Assume that the graph database GD contains a total of n different edges $e_1, e_2, \dots, e_m$, where the maximum edge weight is $\max_{1 \leq k \leq n} (W(e_k))$ and the minimum edge weight is $\min_{1 \leq k \leq n} (W(e_k))$; then,

	V_2	V_3	V_1	V_0	V_4	V_5
V_2	0	e_4	e_2	0	0	e_5
V_3	e_4	0	e_3	0	e_6	0
V_1	e_2	e_3	0	e_1	0	0
V_0	0	0	e_1	0	0	0
V_4	0	e_6	0	0	0	0
V_5	e_5	0	0	0	0	0

FIGURE 7: The adjacent matrix of *FastCanonicalCode*.

the average weight of this GD is defined as follows: $Meanweight = (\lambda \max_{1 \leq k \leq n} (W(e_k)) + (1 - \lambda) \min_{1 \leq k \leq n} (W(e_k)))$, where λ is the adjustment parameter. Based on many experiments, λ is set as 0.5 in this paper.

This paper mainly considers the mining and analysis of the frequent trajectory graph. In the trajectory graph frequency calculation algorithm (TGFC), the node with the largest degree of the subgraph g is the geographical location most frequently visited. We take the node as the initial query point, as shown in line 2 of Algorithm 2. The number of the degree indicates the importance of the node and the priority of the match. The isomorphic points of the initial query point are found from GD to form *matchnode.set*, as shown in lines 3-7. Finally, starting from the *start_node*, the isomorphism of the subgraph g is found through the comparison traversal of the node degree, label of the isomorphism points, and the edge set containing the isomorphism points, as shown in lines 4-11. The subgraph isomorphism search algorithm (*Iso_search*) judges whether *CanonicalCode(g)* is equal to *CanonicalCode(GD)* at first; if *CanonicalCode(g) = CanonicalCode(GD)*, it indicates that g and GD are isomorphic. Then, it judges that if the node is the last node in *FastCanonicalCode*, if the node is the last node, quit the recursion. Otherwise, traversing all the remaining isomorphism points by the adjacency matrix until all isomorphic subgraphs are found.

5.3. Privacy Protection Based on Weighted Frequent Trajectory Graph. Differential privacy mechanism can effectively protect node and graph structure information mining of users' travel trajectories. When differential privacy protection is adopted in the mining of user trajectory subgraph, if Laplace noise is directly added in the generation of frequent subgraph, a large number of redundant subgraphs will be generated, resulting in low data utility. In this paper, we propose a privacy protection framework based on subgraph edge growth model. In this framework, we set the subgraph average total weight (ATW) according to the definition of weighted frequent trajectory graph; based on the ATW, we get the threshold judgment condition of Laplace noise support, as shown in line 12. Considering that the user travel trajectory graph is nonnumerical data, the exponential mechanism is introduced to screen the candidate frequent subgraphs again. The framework not only ensures the availability and security of the data but also retains the characteristics of the user trajectory travel graph.

The privacy budget for the whole approach is $\epsilon = \epsilon_1 + \epsilon_2$. Firstly, the trajectory subgraph set is arranged according to the number of edges. The 1-graph containing frequent edges is taken as the candidate set of subgraphs. If the subgraph satisfies the canonical code, the child of the subgraph is generated by adding one edge every time. We add Laplace noise to the support of child, if the child's noisy support is less than the threshold θ . Repeat noise perturbation and threshold filtering for child generated by the subgraph to get the candidate subgraph. The above process consumes a privacy budget ϵ_1 . We allocate privacy budget ϵ_2 to introduce the exponential mechanism to screen the candidate subgraphs again. Finally, we find all the frequent subgraph that meets all the conditions.

5.4. Security Analysis of the Proposed Scheme

Step 1. DP-Laplace mechanism privacy analysis: due to the edge growth model which is used to extend the subgraph, after each extension of the difference is only one graph data. The sensitivity of Laplace noise function is $\Delta f(Laplace) = 1$. Adding *Laplace*($1/\epsilon_1$) noise in the subgraph mining process satisfies the ϵ_1 -differential privacy.

Step 2. DP-exponential mechanism privacy analysis: the conditional exponential method of Step 2 satisfies ϵ_2 -differential privacy.

Proof. The utility score we assigned to each candidate subgraph can be considered as

$$u(g, D) = \begin{cases} 0, & \text{nsup}(g) < \theta, \\ \exp\left(\frac{\epsilon_2 * \text{sup}(g)}{2n_i}\right), & \text{nsup}(g) \geq \theta, \end{cases} \quad (7)$$

where $n \text{ sup}(g)$ and $\text{sup}(g)$ are the noisy support and the true support of subgraph g in database D . D_1 and D_2 are two neighboring databases. We add Laplace noise *Laplace*($1/\epsilon_1$) to the support of each candidate subgraph-*gchild*[j]. Based on the definition of Laplace mechanism, we have

$$\begin{aligned} \frac{\Pr(\text{noise} = X)}{\Pr(\text{noise} = X + 1)} &= \frac{\exp(-\epsilon_1 |X|)}{\exp(-\epsilon_1 |X + 1|)} \\ &= \exp(\epsilon_1 |X + 1| - \epsilon_1 |X|) \\ &\leq \exp(\epsilon_1). \end{aligned} \quad (8)$$

Similarly,

$$\frac{\Pr(\text{noise} = X)}{\Pr(\text{noise} = X + 1)} \leq \exp(\epsilon_1). \quad (9)$$

Input: Graph Database GD , Canonical Code of Trajectory Subgraph $code$, the number of edges with n as the node $E(n)$
Output: the number of trajectory subgraph g in graph database GD

1. $num = 0$, trajectory subgraph g , $matchnode.set = null$
2. According to the definition of $FastCanonicalCode$, $FastCanonicalCode[0]$ has the largest node degree, and takes the node with the highest degree as the starting match query point.
3. Each time, select the node with the highest degree from $FastCanonicalCode$ for matching. Update $FastCanonicalCode$ to $FastCanonicalCode_{new}$.
4. **for each** node n in $GD.node$:
5. **if** $n.degree \geq start_node.degree \ \& \ n.label = start_node.label \ \& \ E(n) \geq E(start_node)$: $/* \quad */$
6. $matchnode.set += n$
7. **end for**
8. **for each** node n in $matchnode.set$:
9. $num = Iso_search(GD, start_node, n, num)$
10. **end for**
11. **return** num
12. ————Subgraph Isomorphism Search, $Iso_search()$ ————
13. **if** $CanonicalCode(g) = CanonicalCode(GD)$:
14. $num += 1$
15. **return** num
16. **else if**
17. **if** $start_node$ is the last node of $FastCanonicalCode_{new}$:
18. $num += 1$
19. **return** num
20. **for each** unvisited node n_c in $matchnode.set$:
21. **if** $n_c.degree \geq start_node.child.degree \ \& \ n_c.label = start_node.child.label \ \& \ E(n_c) \geq E(start_node.child)$:
22. traversing all the remaining isomorphism points, obtained isomorphism subgraph by the adjacency matrix.
23. $num += |isomorphism|$:
24. **else**
25. $Iso_search(GD, start_node.child, n_c, num)$:
26. **end for**
27. **return** num

ALGORITHM 2: Trajectory graph frequency calculation (TGFC).

Input: Trajectory $TrDatabase = \{Tr1, Tr2, \dots, Trn\}$, privacy budget $\epsilon = \epsilon_1 + \epsilon_2$, Threshold θ
Output: Frequent i -subgraphs S_i

1. Trajectory Graph Database $GD = \{G_1, G_2, \dots, G_n\} = TGDC(TrDatabase)$
2. **if** $g \neq CanonicalCode(g)$ **then**
3. **return**
4. G is sorted from smallest to largest by the number of edges of GD
5. $set = \emptyset$
6. The number of $g[i] = i$ -graph is n_i ;
7. **for** $j=1$ **to** n_i **do**:
8. enumerate $g[i] \in G \subseteq GD$ and its children $gchild[j]/*gchild[j]$ is $g[j]$'s child with one edge growth in GD */
9. **for each** $gchild[j]$ **do**:
10. $sup(gchild[j]) = TGFC(gchild[j], code(gchild[j]))$;
11. $\theta = Meanweight * min_sup / W(gchild[j])$;
12. **if** $n \sup(gchild[j]) = \sup(gchild[j]) + laplace(1/\epsilon_1) \geq \theta$:
13. $set \leftarrow gchild[j]$;
14. **end if**
15. **end for**
16. $g_j \leftarrow$ select a subgraph g from set such that $Pr\{Selecting \text{ subgraph } g\} \propto \exp(\epsilon_2 * support/2n_i)$;
17. $S_i += g_j$;
18. **end for**
19. **return** S_i ;

ALGORITHM 3: Trajectory weighted graph protection (TWGP).

Based on Equations (8) and (9),

$$\begin{aligned}
 \Pr \left(\sup_{D_1}(g) + \text{noise} \geq \theta \right) &= \int_{\theta - \sup_{D_1}(g)}^{+\infty} \Pr(\text{noise} = x) dx \\
 &\leq \int_{\theta - \sup_{D_2}(g)}^{+\infty} \exp(\varepsilon_1) \Pr(\text{noise} = x - 1) dx \\
 &\leq \int_{\theta - \sup_{D_2}(g)}^{+\infty} \exp(\varepsilon_1) \Pr(\text{noise} = x) dx \\
 &= \exp(\varepsilon_1) \Pr \left(\sup_{D_2}(g) + \text{noise} \geq \theta \right).
 \end{aligned} \tag{10}$$

That is,

$$\Pr \left(n \sup_{D_1}(g) \geq \theta \right) \leq \exp(\varepsilon_1) \Pr \left(n \sup_{D_2}(g) \geq \theta \right). \tag{11}$$

In a similar way, we can prove

$$\begin{aligned}
 \Pr \left(\text{noise} \geq \theta - \sup_{D_1}(g) \right) &= \int_{\theta - \sup_{D_1}(g)}^{+\infty} \Pr(\text{noise} = x) dx \\
 &\geq \int_{\theta + 1 - \sup_{D_2}(g)}^{+\infty} \Pr(\text{noise} = x) dx \\
 &= \int_{\theta - \sup_{D_2}(g)}^{+\infty} \Pr(\text{noise} = x + 1) dx \\
 &\geq \exp(-\varepsilon_1) \int_{\theta - \sup_{D_2}(g)}^{+\infty} \Pr(\text{noise} = x) dx \\
 &= \exp(-\varepsilon_1) \Pr \left(\text{noise} \geq \theta - \sup_{D_2}(g) \right).
 \end{aligned} \tag{12}$$

TABLE 2: Summary of graph data sets.

Data set	Trajectory	Number of vertices	The number of edges	The weight range
DATA1	533	14325	16538	0-5
DATA2	1327	18593	28476	5-15
DATA3	657	15236	20212	15-20

Similarly,

$$\begin{aligned}
 \Pr \left(n \sup_{D_1}(g) \geq \theta \right) &\geq \exp(-\varepsilon_1) \Pr \left(n \sup_{D_2}(g) \geq \theta \right), \\
 \Pr \left(n \sup_{D_1}(g) < \theta \right) &\leq \exp(\varepsilon_1) \Pr \left(n \sup_{D_2}(g) < \theta \right), \\
 \Pr \left(n \sup_{D_1}(g) < \theta \right) &\geq \exp(-\varepsilon_1) \Pr \left(n \sup_{D_2}(g) < \theta \right),
 \end{aligned} \tag{13}$$

because for each candidate subgraph g , we have $-1 \leq \sup_{D_1}(g) - \sup_{D_2}(g) \leq 1$.

$$\exp(-\varepsilon_2) \leq \frac{\exp \left(\left(\varepsilon_2 * \sup_{D_1}(g) \right) / 2n_i \right)}{\exp \left(\left(\varepsilon_2 * \sup_{D_2}(g) \right) / 2n_i \right)} \leq \exp(\varepsilon_2). \tag{14}$$

Therefore,

$$\begin{aligned}
 \Pr(\text{selecting subgraph SG from } G \text{ in } D_1) &= \frac{0 \times \Pr \left(n \sup_{D_1}(SG) < \theta \right) + \exp \left(\left(\varepsilon_2 * \sup_{D_1}(SG) \right) / 2n_i \right) \times \Pr \left(n \sup_{D_1}(SG) \geq \theta \right)}{\sum_{g \in G} \left(0 \times \Pr \left(n \sup_{D_1}(g) < \theta \right) + \exp \left(\left(\varepsilon_2 * \sup_{D_1}(g) \right) / 2n_i \right) \times \Pr \left(n \sup_{D_1}(g) \geq \theta \right) \right)} \\
 &\leq \frac{\exp \left(\left(\varepsilon_2 * \sup_{D_1}(SG) \right) / 2n_i \right) \times \exp(\varepsilon_1) \times \Pr \left(n \sup_{D_2}(SG) \geq \theta \right)}{\sum_{g \in G} \exp \left(\left(\varepsilon_2 * \sup_{D_1}(g) \right) / 2n_i \right) \times \exp(-\varepsilon_1) \times \Pr \left(n \sup_{D_2}(g) \geq \theta \right)} \\
 &\leq \frac{\exp(\varepsilon_2 / 2n_i) \times \exp \left(\left(\varepsilon_2 * \sup_{D_2}(SG) \right) / 2n_i \right) \times \exp(\varepsilon_1) \times \Pr \left(n \sup_{D_2}(SG) \geq \theta \right)}{\sum_{g \in G} \exp(-\varepsilon_2 / 2n_i) \times \exp \left(\left(\varepsilon_2 * \sup_{D_2}(g) \right) / 2n_i \right) \times \exp(-\varepsilon_1) \times \Pr \left(n \sup_{D_2}(g) \geq \theta \right)} \\
 &\leq \exp \left(\frac{\varepsilon_1 + \varepsilon_2}{n_i} \right) \Pr(\text{selecting subgraph SG from } G \text{ in } D_2).
 \end{aligned} \tag{15}$$

Based on the above analysis, Step 1 and Step 2 satisfy ε -differential privacy. By the sequential composition property [44, 45], the privacy protection based on weighted frequent trajectory graph overall satisfies ε -differential privacy.

6. Experiments and Results Analysis

The experiments are conducted on big data platform which is IBM X3650M4 with Intel e5 CPU 16 Core (2.2 GHz)

256 GB RAM and 3TB ROM. All the proposed algorithms are coded by the Python 3.6 programming language in CentOS7.5. We use the desensitized 4G location signaling data of the anonymous user group in Nanjing in October 2019. The data fields are as follows: time, encrypted account number, longitude, latitude, city, and grid number. The location data is randomly divided into three parts according to the same tag attributes crowd for the experiment, and the corresponding graph database is generated

FIGURE 8: The scoring performance (F -score) of each data set under the change of threshold θ . FIGURE 9: The relative error of each data set based on frequency threshold θ .

by the trajectory graph database construction algorithm (GDC) as follows.

In the experiments, we use three encrypted real data sets: DATA1, DATA2, and DATA3. The characteristics of these data sets are summarized in Table 2. Besides, the privacy budget ϵ is set to 15, $\epsilon_1 : \epsilon_2 = 3 : 2$. We employ three widely used metrics in travel trajectory frequent pattern mining based on differential privacy protection (TMDP): F -score, relative error, and run time.

6.1. F -score. $F\text{-score} = (P * R) / (P + R)$, $F\text{-score} \in (0, 1]$, and $P = |U_p \cap U_C| / |U_p|$; P is accuracy and R is recall rate. U_p represents the frequent subgraph mined based on differential privacy and U_C represents the exact frequent subgraph mined. Within the range of 0-1, the larger the F -score value is, the higher the data utility of the algorithm will be.

6.2. Relative Error (RE). $RE = \text{median}_{x \in U_p} (|n \sup(x) - \sup(x)| / \sup(x))$ is used to measure the error with respect to the true supports of frequent subgraphs. The smaller the RE is, the higher the data utility of the algorithm is.

6.3. Run Time. Run time reflects the influence of parameters and data sets on experimental results and intuitively represents the operating efficiency of the algorithm.

In the experiment, we introduce the Basic1 comparison algorithm which introduces Laplace noise into the frequent subgraph mining algorithm of Apriori. The Basic2 algorithm only adds Laplace noise into the frequent subgraph mining process in this paper. This paper will measure the performance of the algorithms (Basic1, Basic2, and TMDP) from three aspects: (1) analysis of F -score based on frequency threshold θ , (2) analysis of relative error based on frequency threshold θ , and (3) analysis of algorithm running time.

(1) Analysis of results based on frequency threshold θ

As shown in Figure 8, with the increase of threshold θ , fewer subgraphs match the threshold θ condition during frequent subgraph mining. Therefore, with the increase of T , the F -score also increases, and the data utility also increases. In the three data sets, the data utility of the TMDP framework is better than algorithm Basic1 and algorithm Basic2,

FIGURE 10: Comparison of running time of each algorithm.

and it is more obvious in the relatively large data set DATA2. When the threshold θ is 0.6, the data utility reaches 0.79, which indicates that the TMDP framework has a better overall performance on the trajectory graph protection algorithms.

(2) Analysis of relative error based on frequency threshold θ

As shown in Figure 9, with the increase of the threshold θ , the value of RE is constantly decreasing. When the threshold is smaller, the number of false frequent graphs will be larger, which makes the relative error of the results larger; that is, the RE will increase. The experimental results of this paper show that when the threshold is selected around 0.5, the RE result performs relatively well. The overall performance of the DMP framework for the three data sets is much better than that of Basic1.

(3) Analysis of algorithm running time run time

Finally, we compared the running time of each algorithm, as shown in Figure 10. In contrast, the TMDP framework consumes more time. Meanwhile, the larger the threshold θ , the less time is consumed, because the larger threshold θ , the fewer frequent subgraphs in the graph set that meet the conditions, and the smaller running time. From the comparison of the three graphs, the larger the data set, the greater the number of graphs, and therefore, the more time consumed.

In the above experiments, travel trajectory frequent pattern mining based on differential privacy protection (TMDP) can realize privacy protection on the premise of ensuring data utility. Although the overall running time and work are increased, it reduces running time and improves data utility. We can see that our TMDP framework achieves better performance.

7. Conclusion

In order to solve the problem of the balance between the utility and security of user location data privacy under the mobile Internet service, we propose a weighted trajectory

graph model scheme, which converts the discrete location data of users into graph data based on weighted frequent graph pattern. The weighted trajectory graph model not only preserves the characteristics of users' trajectories but also facilitates the establishment of privacy-preserving data mining algorithm. In order to ensure the utility of the location data in our privacy protection framework (TMDP), we make double processes of adding noises to the subgraph mining process by the Laplace mechanism and selecting final data by the exponential mechanism. The experiments show that the data utility of the proposed frame is higher and the privacy protection of the frame is effective. However, Internet applications require high real-time and diversity of location data; how to customize efficient privacy protection model and improve data availability under high-dimensional data sources in specific scenarios will be the main research direction in the future.

Data Availability

The location data used to support the findings of this study have not been made available because of trade secrets.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C. W. Lin, and G. Fortino, "Deep learning for pedestrian collective behavior analysis in smart cities: a model of group trajectory outlier detection," *Information Fusion*, vol. 65, pp. 13–20, 2021.
- [2] C. Shahabi, L. Fan, L. Nocera, L. Xiong, and M. Li, "Privacy-preserving inference of social relationships from location data: a vision paper," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–4, New York: ACM Press, 2015.
- [3] P. Cheng, J. F. Roddick, S. C. Chu, and C. W. Lin, "Privacy preservation through a greedy, distortion-based rule-hiding method," *Applied Intelligence*, vol. 44, no. 2, pp. 295–306, 2016.

- [4] S. Shivaprasad, H. Li, and X. Zou, "Privacy preservation in location based services," *Journal of Computers*, vol. 11, no. 5, pp. 411–422, 2016.
- [5] P. Golle and K. Partridge, *On the Anonymity of Home/Work Location Pairs[C]/International Conference on Pervasive Computing*, Springer, Berlin, Heidelberg, 2009.
- [6] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services - MobiSys '03*, pp. 31–42, New York: ACM Press, 2003.
- [7] T. Xu and Y. Cai, "Location anonymity in continuous location-based services," in *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pp. 1–8, New York: ACM Press, 2007.
- [8] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in GPS traces via uncertainty-aware path cloaking," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 161–171, New York: ACM Press, 2007.
- [9] 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.
- [10] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13*, pp. 901–914, New York: ACM Press, 2013.
- [11] A. Belhadi, Y. Djenouri, J. C. W. Lin, and A. Cano, "Trajectory outlier detection: algorithms, taxonomies, evaluation, and open challenges," *ACM Transactions on Management Information Systems (TMIS)*, vol. 11, no. 3, pp. 1–29, 2020.
- [12] Z. Huo and X. F. Meng, "A survey of trajectory privacy-preserving techniques," *Chinese Journal of Computers*, vol. 34, no. 10, pp. 1820–1830, 2011.
- [13] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with privacy grid," in *Proceedings of the 17th International Conference on World Wide Web*, pp. 237–246, New York: ACM Press, 2008.
- [14] L. Liu, "From data privacy to location privacy: models and algorithms," in *Proceedings of the 33rd International Conference on Very Large Data Bases*, pp. 1429–1430, New York: ACM Press, 2007.
- [15] F. Liu, K. Hua, and Y. Cai, "Query l-diversity in location-based services," in *Proceedings of the 10th International Conference on Mobile Data Management*, pp. 436–442, Taipei, 2009.
- [16] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. Di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *Data and Applications Security XXI*, pp. 47–60, Springer, 2007.
- [17] J. Quyang, J. Yin, S. Liu, and Y. Liu, "An effective differential privacy transaction data publication strategy," *Journal of Computer Research & Development*, vol. 51, no. 10, pp. 2195–2205, 2014.
- [18] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *22nd International Conference on Data Engineering (ICDE'06)*, pp. 25–35, Atlanta, GA, USA, 2006.
- [19] R. Wong, J. Li, A. Fu, and K. Wang, " (α, k) -Anonymity: an enhanced k-anonymity model for privacy-preserving data publishing," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, pp. 754–759, Philadelphia, 2006.
- [20] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.
- [21] R. Shokri, G. Theodorakopoulos, C. Troncoso, J. P. Hubaux, and J. Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *In Proceedings of the 2012 ACM conference on Computer and Communications Security*, pp. 617–627, 2012.
- [22] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving user location privacy in mobile data management infrastructures," in *Privacy Enhancing Technologies*, pp. 393–412, Springer, Berlin, Heidelberg, 2006.
- [23] R. Gupta and U. P. Rao, "VIC-PRO: vicinity protection by concealing location coordinates using geometrical transformations in location based services," *Wireless Personal Communications*, vol. 107, no. 2, pp. 1041–1059, 2019.
- [24] J. Freudiger, M. Raya, M. Flegyhazi, and P. Papadimitratos, "Mix-zones for location privacy in vehicular networks," in *In Association for Computing Machinery (ACM) Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, pp. 1–7, Vancouver, British Columbia, Canada: ACM, 2007.
- [25] Z. Xu, H. Zhang, and X. Yu, "Multiple mix-zones deployment for continuous location privacy protection," in *Trustcom/big-dataise/ispa, IEEE*, pp. 760–766, Tianjin, China, 2016.
- [26] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *21st International Conference on Data Engineering Workshops (ICDEW'05)*, p. 1248, Tokyo, Japan, 2005.
- [27] A. Suzuki, M. Iwata, Y. Arase, T. Hara, X. Xie, and S. Nishio, "A user location anonymization method for location based services in a real environment," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*, pp. 398–401, CCM:New York, 2010.
- [28] R. Kato, M. Iwata, T. Hara et al., "A dummy-based anonymization method based on user trajectory with pauses," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, pp. 249–258, ACM:New York, 2012.
- [29] R. Chen, B. C. M. Fung, and B. C. Desai, "Differentially private trajectory data publication," 2011, <https://arxiv.org/abs/1112.2020>.
- [30] H. Jingyu, Y. Gao, and Z. Sheng, "Differentially private publication of general time-gerial trajectory data," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 163–175, Hong Kong, China, 2015.
- [31] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1298–1309, Denver, CO, USA, 2015.
- [32] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "Geo-indistinguishability: a principled approach to location privacy," in *Distributed Computing and Internet Technology*, pp. 49–72, Springer, Cham, 2015.
- [33] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "RescueDP: real-time spatio-temporal crowd-sourced data publishing with differential privacy," in *IEEE INFOCOM 2016*

- The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, USA, 2016.
- [34] R. Gupta and U. P. Rao, “Preserving location privacy using three layer RDV masking in geocoded published discrete point data,” *World Wide Web*, vol. 23, no. 1, pp. 175–206, 2020.
 - [35] R. Dewri and R. Thurimella, “Exploiting service similarity for privacy in location-based search queries,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 374–383, 2013.
 - [36] R. Gupta and U. P. Rao, “An exploration to location based service and its privacy preserving techniques: a survey,” *Wireless Personal Communications*, vol. 96, no. 2, pp. 1973–2007, 2017.
 - [37] T. P. Hong, C. W. Lin, and Y. L. Wu, “Incrementally fast updated frequent pattern trees,” *Expert Systems with Applications*, vol. 34, no. 4, pp. 2424–2435, 2008.
 - [38] C. Dwork, “Differential privacy,” in *Automata, Languages and Programming*, pp. 1–12, Springer, Berlin, Heidelberg, 2006.
 - [39] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, “Discovering frequent patterns in sensitive data,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*, CAM: New York, 2010.
 - [40] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Theory of Cryptography*, pp. 265–284, Springer, Berlin, Heidelberg, 2006.
 - [41] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, Providence, RI, USA, 2007.
 - [42] A. Inokuchi, T. Washio, and H. Motoda, *An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data[C]//European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, Berlin, Heidelberg, 2000.
 - [43] M. Kuramochi and G. Karypis, “An efficient algorithm for discovering frequent subgraphs,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1038–1051, 2004.
 - [44] C. Jiang, F. Coenen, and M. Zito, “Frequent sub-graph mining on edge weighted graphs,” in *Data Warehousing and Knowledge Discovery*, pp. 77–88, Springer, Berlin, Heidelberg, 2010.
 - [45] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Theory of Cryptography*, pp. 265–284, Springer, Berlin, Heidelberg, 2006.