FADBM: Frequency-Aware Dummy-Based Method in Long-Term Location Privacy Protection

Jiabang Liu[1,2], Xutong Jiang[1,2], Song Zhang[1,2], Hao Wang[3], Wanchun Dou[1,2,*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, P. R. China
[2]The Department of Computer Science and Technology, Nanjing University, Nanjing, P. R. China
[3]Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjøvik, Norway
Email: {liujb,jiangxt,songzhang}@smail.nju.edu.cn,hawa@ntnu.no,douwc@nju.edu.cn

***Abstract*—With the rapid usage of location-based services (LBSs), protection of location privacy has become a significant concern. Existing dummy-based methods mainly consider generating dummies in continuous queries, which neglects the fact that the user would launch queries in a frequent region(e.g., home), resulting in privacy disclosure in a long-term period (i.e., more than 30 days). To solve this problem, we propose a method which generates dummy frequent regions, and make dummies locate in these dummy regions as far as possible. Compared with other methods, evaluation based on real-world dataset shows that the proposed method can reduce the ratio of recognized dummies and restored trajectory in a long-term situation.**

***Index Terms*—privacy protection, location-based services, k-anonymity, dummy generation, trajectory privacy**

I. Introduction

With the widespread usage of wireless networks technology and mobile phones featuring global positioning system (GPS), more and more location-based services (LBS) have emerged providing various services for people's work and daily life needs. In an LBS, a mobile user sends a request containing his/her location and interests to a location service provider (LSP). The LSP returns the point of interests (POIs) near the user's current location. For example, visitors can send POI queries to the LBS servers. By submitting LBS queries, users can enjoy the convenience provided by LBS.

However, every coin has two sides as such convenience might come at the price of user's privacy leakage. The LSP has the potential to violate the user's privacy [1]. By collecting the user's queries, an untrustworthy LSP can infer personal information about the user [2], such as his/her location, preferences and possibly his/her state of health. Even worse, the LSP could disclose the user's private information to third parties for financial or other business advantage. Consequently, it is essential to pay more attention to protect users' location privacy.

To address the location privacy issue, numerous research efforts [3]–[13] has been attracted over the past several years. These efforts employ well-known privacy metrics such as k-anonymity [14] and rely on a trusted third-party server. Among these efforts, the dummy-based method [15] is one of the popular solutions. This method works by generating a group of dummies aside with the real location for each user's request, and all of these locations are then submitted as the service request. In this way, the user's real location cannot be identified. Existing dummy-based method is mainly concerned with a short period(e.g., a single query or a whole journey within twenty-four hours). In practice, however, long-term requests still exist [8], [12]. For example, a user makes requests at home for up to 30 days. When most of existing dummy-based methods generate dummy for the user's home, the sparsely distributed dummies surround home, which forms a surrounding's frequency significantly lower than the home. Thus, the adversary has a high confidence that the user's real location is in the region with high frequency, and the existing dummy-based methods cannot protect the user's location privacy completely in long-term period.

In this paper, based on the existing dummy-based methods, a frequency-aware dummy-based method (FADBM) is proposed to achieve long-term k-anonymity for users in LBS. Different from existing approaches, FADBM carefully selects dummy locations in consideration of that long-term requests may be disclosed to adversaries, and make sure that the selected dummy locations around frequent regions are satisfied with time reachability. The major contributions of this paper are as follows:

- Use existing dummy-based methods directly in long-term requests (more than 30 days), and the correct ratio for adversary to infer some dummies is more than 25% by means of hypothetical adversary method based on frequent regions. This shows that these methods cannot protect user's location privacy completely.
- To guard against this hypothetical adversary method, an FADBM method is proposed to achieve k-anonymity by carefully choosing dummy locations in frequent regions, and considers both frequent regions and time reachability to ensure that the selected dummy regions are reasonable as far as possible.
- The proposed dummy generation methods is evaluated by taking real geographical information into account. Experimental results demonstrate the effectiveness of protecting the user's long-term privacy.

The rest of this research paper is organized as follows: Section II and Section III present related work and motivation respectively. Section IV provides a general overview of the

[*] Corresponding author: Wanchun Dou (e-mail: douwc@nju.edu.cn)

hypothetical adversary method. Section V presents details of the proposed protection method against this hypothetical adversary method, and an evaluation of the method's performance-efficiency is presented clearly in Section VI. Finally, Section VII concludes this paper.

II. Related work

Considerable efforts have been achieved to protect users' location privacy over the past several years. Among these efforts, the dummy-based method is one of the popular solutions. In some early work, Kido et al. [16] introduced dummies into location privacy protection. In their scheme, the user utilized the random walking model to generate a group of dummies aside with the real location for each request of the user, and all of these locations are then submitted as the service request. In this way, the user's real location cannot be identified. Lu et al. [17] proposed two dummy location generating schemes called CirDummy and GridDummy, which achieve k-anonymity for considering the privacy-area.

However, these above-mentioned methods may not work well because they ignore that the adversary might have side information. For example, these methods generate dummies which are in a lake or a rugged mountains, and these dummies would be identified by the adversary. Niu et al. [3] pointed out it, and their method carefully selects dummy locations considering that side information may be exploited by adversaries. They chose these dummy locations based on the entropy metric, and then enhanced the algorithm by making sure that the selected dummy locations are spread far away. Additionally, to reduce the risk of privacy leakage further, Niu et al. [7] cached the service data obtained for both the real location and dummy locations of the current query, and used the cached data to answer future queries so as to reduce the queries sent to the LBS server. In this way they reduced the number of queries sent to the LBS server for protecting users' privacy.

Gradually, researchers began to think temporal and spatial continuity between dummies. Liu et al. [4] presented an solution which firstly adopts the underlying dummy-based schemes to generate initial candidate dummies, and then analyzed the spatiotemporal correlation between neighboring location sets submitted from three aspects, namely time reachability, direction similarity and indegree/out-degree.

Unfortunately, the existing dummy-based schemes mainly focus on protecting the user's location privacy in a single query or a whole journey within twenty-four hours. Some researchers have already considered long-term location privacy protection [8], [12], but these methods may still have some shortcomings that a region is the user's high frequency active area, not just a location. When the user adopts these schemes to protect his location privacy in consecutive requests, and we utilize the hypothetical adversary method in Sec IV to attack, some dummies can be inferred with no less than 25% correct ratio.

III. Motivation

Existing dummy-based method may fail in a new situation. This situation is that the adversary has the user's queries data whose time span is a long period, more one month, instead of a short period(e.g., a single query or a whole journey within twenty-four hours). Here are some instructions on why existing dummy-based method failing.

Considering the user's mobile pattern, Gonzalez et al. [18] has indicated that humans follow simple reproducible patterns in a long period. So the user's queries are instinctively regular within a long-period period. That is, regions where user's queries were issued are in a high level of similarity. This kind of region is named as frequent regions. When the user's queries are not in frequent regions, the adversary can identify some dummy with a high confidence.

The problem is described as: When the adversary collects the user's long-period queries on consideration of the above content, frequent regions can be recognized by an adversary using clustering method. Not only that, if these regions are stamped with time identification, an adversary can use time reachability between regions to restore user's common trajectory. Problems are illustrated in the following scenario.

Above-Mentioned problem is detailed with a concrete example. Our 3-anonymity example is constructed by using a small portion of the processed real trajectory dataset. Fig. 1 shows that adversary can recognize the user's frequent region R_T during a time period T marked in red with a long-period period. When the user queries again in R_T on T, existing dummy-based methods generate two random dummy locations(orange-coloured triangle). The adversary considers that the users' real location is in R_T probably, thereby, he/she can identify some dummies with a high confidence, or even obtaining the user's real location directly, as shown in Fig. 1.

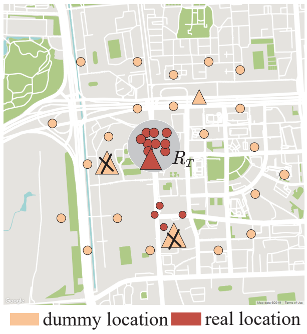


Fig. 1. Real location disclosed by using existing 3-anonymity method

Further than that, when the adversary using clustering method direct at multiple time period $\{T_1, T_2, \ldots, T_i\}$, he/she can recognize the frequent region over different time periods(e.g.,R_{T_i},$R_{T_{i+1}}$), as shown in Fig. 2. Then, the adversary checks the reachability within time frame ΔT, $\Delta T = T_{i+1} - T_i$. The trajectory between R_{T_i},$R_{T_{i+1}}$) can be identified with a high probability, or even be disclosed, as shown in Fig. 2.

The general idea of our solution is to generate the dummy frequent regions, which is illustrated in Sec. V-A.

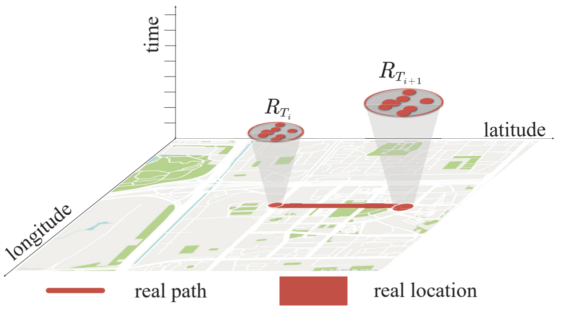


Fig. 2. Real trajectory disclosed by using existing 3-anonymity method

IV. Hypothetical Adversary Method

A. Long-period Query Database

An LBS user sends her current location accompanied with a query asking for her interest, and the LSP returns a set of points as query result. The query database records are in the form of (ID, LOC, POI_{LOC}), where ID represents a user. LOC consists of $\{\{loc_1, t_1\}, \{loc_2, t_2\}, \dots, \{loc_N, t_N\}\}$, and loc is locations which the query contains, t is the time stamp of the query.

When this query database contains more than 30 days of query information, we call this kind of database as long-period query database.

B. Adversary Model

An adversary is any spiteful third party with whom this long-period query database is disclosed. We assume that this adversary has the following background knowledge about a transport network:

Path Weight: We assume that the adversary will assign a higher weight to more frequently traveled streets for large crowds. Generally, main roads are more likely to receive higher weights.

Maximum Velocity Bound: The adversary may also assume that there is a maximum velocity with which a user can travel between two subsequent time stamps, which can be used in time reachability. The velocity of a user at a given time can be estimated based on the maximum speed limit of a road.

C. Adversary Algorithm

The adversary can utilize the long-period query database to recognize the user's frequent regions by using the clustering method, and then infer the real location of each user's query from frequent regions. Not only that, an adversary can use background knowledge between regions to restore user's common trajectory.

Initially, the clustering method is chosen to recognize the frequent regions. Due to the irregular shape of the frequent regions, density-based spatial clustering method is chosen from the mainstream clustering algorithm. And then, Ordering Points To Identify the CluStering(OPTICS [19]) is taken into account because of the difference in query density between the frequent regions and common regions. One thing to note is that loc in the query database is the real location in real world, and calculating the distance between them can not simply use Euclidean Metric or Manhattan distance. To fit this scene better, some parameters of the original OPTICS algorithm are redefined. Actual traffic distance is replace the distance in the original OPTICS algorithm. That is, the distance between loc_i and loc_{i+1} is gained by asking Google Maps API. Note that the distance used in **Algorithm 1** is actual traffic distance.

Secondly, the adversary makes effort to obtain the frequent regions in a certain period. A day can be divided into multiple time periods such as $T_1 = 0:00 - 0:15$ by the time interval. Construct the set $\mathcal{T} = \{T_1, T_2, \dots, T_N\}$. The shorter the time interval, the more accurate frequent region in T_i which the adversary obtains. The excessively short time interval may result in non-existent frequent regions during this period.

Points$\{loc, t\}$ are extracted from the long-period query database, whose t is in a certain time periodT_i. Then, construct a new set $D_{T_i} = \{\{loc_1, t_1\}, \{loc_2, t_2\}, \dots, \{loc_N, t_N\}\}$. Enhanced-OPTICS requires two parameters:ε, which describes the maximum distance to consider, and M, describing the number of points required to form a cluster. $d(loc_i, loc_{i+1})$ is the distance between loc_i and loc_{i+1}. $N_\varepsilon(loc)$ is the points whose distance to loc is less than or equal to ε. loc is a core point if at least M points are found within its ε-neighborhood $N_\varepsilon(loc)$. The two following concepts are used in the enhanced-OPTICS:

1) Each point p is assigned a core-distance that describes the distance to the M-th closest point:
$$cd(p) = \begin{cases} UNDEFINED, & if|N_\varepsilon(p)| < M \\ d(loc, N_\varepsilon^M(p)), & otherwise \end{cases} \quad (1)$$
$d(p, N_\varepsilon^M(p))$ describes the M-th smallest distance to p in $N_\varepsilon(p)$.
2) The reachability-distance of another point o from a point p is either the distance between o and p, or the core distance of p, whichever is bigger:
$$rd(o,p) = \begin{cases} UNDEFINED, & if|N_\varepsilon(p)| < M \\ max\{cd(p), d(o,p)\}, & otherwise \end{cases} \quad (2)$$

$\{orderlist_i\}_{i=1}^N$ is the ordered array of the D_T. $\{rd_i\}_{i=1}^N$ is the reachability-distance of i-th point, $\{cd_i\}_{i=1}^N$ is the core-distance of i-th point, $i = 1, 2, \dots, N$. $\{orderlist_i\}_{i=1}^N, \{rd_i\}_{i=1}^N, \{cd_i\}_{i=1}^N$ would be abbreviated as $\{orderlist_i\}, \{rd_i\}, \{cd_i\}$ without ambiguity. The following algorithm solve the above-mentioned value. Additionally, since the optics algorithm is very mature, the idea of the algorithm is briefly describe, and we focus on some of our adjustments.

The adversary gains $\{orderlist_i\}$, $\{rd_i\}$, $\{cd_i\}$, then generates marked array$\{m_i\}$. In this process, a parameter $\overline{\varepsilon}$ needs to be provided to represent the number of containing points for the different ClusterID, and $\overline{\varepsilon}$ can be adjusted according to the actual situation. Then, the adversary can get the points of different ClusterID by using $\{m_i\}$ limited by $\overline{\varepsilon}$, $SameCluster_i = \{\{loc_1, t_1\}, \{loc_2, t_2\}, \dots, \{loc_N, t_N\}\}$. Take $SameCluster_1$ as an example. Utilize loc in $SameCluster_1$ to calculate the center point of these points, and then calculate the shortest radius of circle containing these

Algorithm 1 Enhanced-OPTICS

```
Input: $D_T, \varepsilon, MinPts$
Output: $\{orderlist_i\}, \{rd_i\}, \{cd_i\}$
function PREPROCESSING($D_T, \varepsilon, MinPts$)
    $initialization(core\_points, \{cd_i\})$
    $\{rd_i\} = \{orderlist_i\} = \varnothing$
    for unprocessed point $p$ in $core\_points$ do
        $N = getNeighbors(p, \varepsilon)$
        mark $p$ as processed
        output $p$ to $\{orderlist_i\}$
        if $|N| \geqslant MinPts$ then
            $Seeds = \phi$
            UPDATE($N, p, Seeds, \{cd_i\}, \{rd_i\}$)
            for $q$ in $Seeds$ do
                $N^{'} = getNeighbors(q, \varepsilon)$
                mark $q$ as processed
                output $q$ to $\{orderlist_i\}$
                if $N^{'} \geqslant MinPts$ then
                    update($N^{'}, q, Seeds, \{cd_i\}, \{rd_i\}$)
    return $\{orderlist_i\}, \{rd_i\}, \{cd_i\}$
```

points. This circle is the frequent region of T, $R_{T_{i_1}}$. Then, the adversary calculates the core point of $\{t_1, t_2, \dots, t_N\}$, and makes this core point as the time stamp of $R_{T_{i_1}}$. The following function is the description of how generating $R_{T_i} = \{\{R_{T_{i_1}}, t_1\}, \{R_{T_{i_2}}, t_2\}, \dots, \{R_{T_{i_N}}, t_N\}\}$. In this function, the procedure $calculate\_core()$ is one of the most popular clustering algorithms K-Means in which K is 1. Execute the above-mentioned **Algorithm 1** and **Algorithm 2** for each $T \in D_{\mathcal{T}}$, and get the frequent regions $\mathcal{R} = \{R_{T_1}, R_{T_2}, \dots, R_{T_i}\}$, $R_{T_i} = \{\{\{R_{T_{i_1}}, cp_1, t_{cp_1}\}, \dots, \{R_{T_{i_N}}, cp_N, t_{cp_N}\}\}\}$.

Algorithm 2 Enhanced-OPTICS

```
Input: $\{orderlist_i\}, \{rd_i\}, \{cd_i\}, \overline{\varepsilon}, range$
Output: $\{\{R_{T_{i_1}}, cp_1, t_1\}, \dots, \{R_{T_{i_N}}, cp_N, t_N\}\}$
function CLUSTER($\{orderlist_i\}, \{rd_i\}, \{cd_i\}, \overline{\varepsilon}$)
    $ClusterID = -1$
    $k = 1$
    for $i = 1, 2, ..., N$ do
        $j = orderlist_i$
        generate ClusterID for each $m_j$
    for $same\_id\_points \in \{m_j\}$ do
        $cp = calculate\_core(loc_{i=1}^{N})$
        $radius = max\{euclidean\_dis(cp, \{loc\}_{i=1}^{N})\}$
        if $radius < range$ then
            $R_T = Circle(cp, radius)$
            $t = calculate\_core(\{t\}_{i=1}^{N})$
    return $R_{T_i} = \{\{R_{T_{i_1}}, cp_1, t_1\}, \dots, \{R_{T_{i_N}}, cp_N, t_N\}\}$
```

Finally, after the $R_{\mathcal{T}}$ is obtained by the adversary, the privacy protection expectation of dummy-based method would be likely to fall. To be special, attacking methods can be divided into recognizing the dummy locations from the continued query and restoring the frequent trajectory used by users.

Here's a look at the adversary how recognizing the dummy locations. When the user continues to ask a query $\{loc, t\}$,$loc = \{real\_loc, dummy_1, \dots, dummy_{k-1}\}$, the adversary can find out t belonging to which of the $\mathcal{T}$(e.g.,T). Then, judge these points by using the following formula:$false\_loc = \{point \mid point \notin R_T \cap t \in T\}$,$true\_loc = loc - false\_loc$. $false\_loc$ is excluded because of that the user is more likely to initiate a query in the frequent regions $R_{\mathcal{T}}$.

The following is the adversary how restoring the frequent trajectory used by users. Each cp of the R_{T_i} queries path to each cp of the R_{T_i} by using GoogleMap API, which is limited by the time reachability between the time stamp t of $cp \in R_{T_i}$ and $cp \in R_{T_{i+1}}$, and this path is called as PartPath. The API result is the Latitude and longitude points of this path. When all PartPath are obtained, permutation and combination of these paths can be done as candidate paths. To select top k path from the set of generated candidate paths. A weighting function is derived to rank the paths based on edge centrality, and then the top k ranked candidate path is selected as the path representing the user's frequent trajectory.

The weighting function counts the weight of each candidate path. The weight for each candidate path is calculated by the summation of each PartPath length divided by the length of the whole candidate path which is then multiplied by the weight of each PartPath. Note that this weight of each PartPath is supposed to know by the adversary.

$$w_{CandPath} = \sum_{n}^{i=0} w_{PartPathWei_i} \times \frac{PartPathLength_i}{totLength} \quad (3)$$

The weighting function returns the path that contains the greatest overlap with other paths in the candidate path set. Therefore, edges that are more commonly used in a set of candidate paths are favored over edges that are not. Provided users take fairly direct routes to their destination, this weighting function works well.

D. Attack Effect Metrics

Two metrics are introduced to respectively evaluate the effectiveness of adversary recognizing the dummy locations and restoring the frequent trajectory.

To begin with, Rec_Dummy is a metric about how many dummies locations are recognized within the user's continued queries. Rec_Dummy is calculated by the recognized dummy locations belonging to $dummy\_loc$ divided by the all dummy locations in n queries.

$$Rec\_Dummy = \frac{1}{n} \sum_{i=0}^{n} \frac{|false\_loc \in dummy\_loc|}{|dummy\_loc|} \quad (4)$$

The next is the metric about how accurate the frequent trajectories are restored. The points of top k ranked candidate path are encoded as GeoHash. Geohash is a public domain geocoding system, which encodes a geographic location into a short string of letters and digits. It is a hierarchical spatial data structure which subdivides space into buckets of grid shape. Utilize this GeoHash technique, and the trajectory can be shaped in a grid.

$$Res\_Tra = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{1}{k}\frac{|cand\_path_j \cap daily\_path_i|}{|daily\_path_i|} \quad (5)$$

When we calculate this metric Res_Tra, we can calculate the repetition number of the grid between the top k ranked candidate path and i-th daily path, $|cand\_path_j \cap daily\_path_i|$ which is divided by the grid number of i-th daily_path, $|daily\_path_i|$. Do it for each i-th daily path.

V. Our proposed Method

A. Basic Idea

To address the problem in Section III, our basic idea is to generate dummy region which is similar as frequent regions. If a query is issued in a frequent region, a dummy location is generated around a fixed location; if not, a dummy location is generated to meet accessibility with the last query, otherwise, this dummy location is as close as possible to the next frequency region.

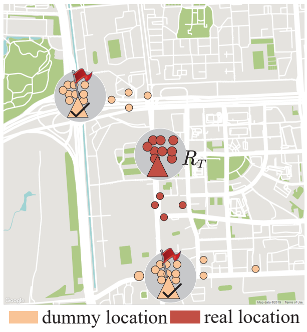


Fig. 3. Dummy locations chosen considering long-period information.

Fig. 3 illustrates our basic idea. When the user queries in R_T on T, our approach is to generate dummy locations around the flag as illustrated. After using our method, adversary gets three regions by using clustering methods to recognize user frequent regions R_T. When the user continue to ask a query in R_T on T, two dummy locations are generated in other two dummy frequent regions. The adversary can not recognize the real location from three locations by utilizing the frequent region, thereby, the expectation of 3-anonymity method can be met.

However, the above-mentioned method only takes into account the situation within a period of time T. Considering the continuity of time, our approach can be improved to protect the user's real trajectory privacy.

As shown in Fig. 4, when the adversary check time reachability of the user's frequent regions($R_{T_i}, R_{T_{i+1}}$), the adversary can get the path satisfied time reachability.

This path may be the user's real trajectory which is marked with red line. When we want to protect the user's real trajectory privacy, our improved approach is to generate dummy paths to conflate reality with dummy. When our approach chooses dummy frequent regions, the time reachability of these regions should be checked. Wrong demonstration is that two regions can not be reachable within time frame ΔT, $\Delta T = T_{i+1} - T_i$. The path of these regions is marked with gray line in the diagram. Right demonstration is the path marked with orange-coloured line. In this situation, the adversary is hesitant about picking out one from two paths which consist of one dummy trajectory and one real trajectory. However, the best situation is that two dummy trajectory is satisfied with time reachability.

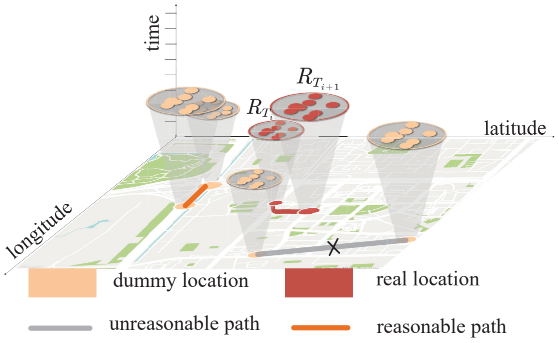


Fig. 4. Dummy regions chosen by considering time reachability

B. A Frequency-Oblivious Baseline Method

Before our advanced dummy-based method described, we first introduce a baseline method for comparison purpose.

Entropy-based metrics [20] and distortion-based metrics [15]. Entropy is an uncertainty of recognize the real location from all the locations, and it has been widely used in the literature. We choose the same entropy as Niu et al. [7] , the map is divided as N×N cells. Each cell has a probability of being queried (called query probability) in the past. Let q denote this probability. Then we have $\sum_{i=1}^{N^2} q_i = 1$. For the k locations (i.e,cells) contained in a query which one real location and $k-1$ dummies, each location has a conditional probability of being the real location. Let $p_i(i = 1, 2, ..., k)$ denote the probability that the i-th location is the real location. Then $p_i = \frac{q_i}{\sum_{j=1}^{k} q_i}$, and obviously $\sum_{i=1}^{k} p_i = 1$ The entropy H of identifying the real location out of the anonymity set is defined as:

$$H = -\sum_{i=1}^{k} p_i \cdot log_2 P_i \quad (6)$$

In this method, dummies are selected to maximize the entropy for the current query only, without considering the frequent regions' effect.

C. Frequency-Aware Dummy Selection Method

1) Dummy locations chosen for protecting location privacy: Similar to the work of the predecessors [21], we also assume that the anonymity service is managed by some cellular service provider (absolutely trusted), through which mobile users have access to wireless communications. The cellular service provider offers anonymity services as a value-added feature to their clients, and supplies the initial long-period locations database (more one month). The location samples in the database may be collected from clients' regular phone calls. When we have the long-period locations database, the R_T and time stamp t can be owned by using **Algorithm 1** and **Algorithm 2**.

Our main idea is to select a set of realistic dummy locations $\{dum\_loc_1, dum\_loc_2 \ldots, dum\_loc_i\}$. Ensure high entropy for the current query and at the same time provide more contributions, which possibly satisfies as more as dummy locations in dummy frequent regions. In order to generate the dummy frequent regions on T, points are required to be chosen as core points of dummy frequent regions. The selection of these core locations $core\_loc$ is similar to the selection of the dummy points, which can be referred on the previous researchers' work [3]. When the core point has been chosen, the user should specify customized radius ϵ and required privacy level $\kappa()$ on T. Then, how generating dummy locations in these dummy frequent regions on T is a problem that need to be solved.

The user's real location may be in R_T or not, thereby, the process of choosing dummy locations should consider two situation. When the user's real location is in a frequent region R_T, the dummy location is also available in the dummy frequent region. Otherwise, the user's dummy location needs to be made further arrangement. In this situation, our thought is that the dummy locations is generated around the dummy frequent regions, and the distance between dum_loc and its corresponding $core\_loc$ is Less than the $d(real\_loc, cor\_core\_loc)$. At the same time, the equation $(max\{H\})$ should be met.

Algorithm 3 Frequency-Aware Dummy Selection Method

```
Input: $q$(each cell's query probability), $c_r$(real location on $T$),
    $R_T$(frequent regions on $T$), $\epsilon$(customized radius ),
    $k$(privacy level)
Output: $C_{dummy}$
1: function GENERATE DUMMY LOCATIONS(Input)
2:     generate dummy frequent regions $\{R_{T\,dummy}\}_{i=1}^{k}$ by the specifing privacy level $k$ and customized radius $\epsilon$
3:     sort cells based on their query probability $q$
4:     for each $R_{T\,dummy} \in \{R_{T\,dummy}\}_{i=1}^{k}$ do
5:         choose $k$ cells ($\frac{k}{2}$ cells are right before $core\_loc$ and $\frac{k}{2}$ cells are right after $core_{loc}$ in the sorted list)
6:         randomly select $\frac{k}{2}$ cells out of them as the candidate set $\widehat{C}$
7:         compute each $loc$in $\widehat{C}$ and select $loc$ which satisfies $|d(loc, core\_loc) - d(c_r, core\_loc_{real})| \leqslant \psi$
8:         add this $loc$ into $C_{dummy}$
9:     return $C_{dummy}$
```

2) Dummy regions chosen for protection trajectory privacy: When these regions are stamped with time identification, some frequent regions are excluded because of no reachable trajectories with other $k-1$ common areas and the user's common trajectory may be recognized. Therefore, when generating dummy frequent regions on T_{i+1} for the user, we will perform a time reachability check with R_{T_i}.

When generate frequent regions on T, compute the core location of each R_T on T and get the result $cl = \{cl_1, cl_2, ..., cl_k\}$. When we generate dummy frequent regions for the next time period, a directed graph $G_{T+1} = (V_{T+1}, E_{T+1})$ is used to express the rest core locations and dummy trajectories filtered by the judgment of time reachability on $T+1$. The $loc \in G_T$ should meet these following requirements :

$$\begin{cases} |V_{T+1}| \geqslant k \\ \forall cl_i \in \{cl\}_{T+1}, \exists cl^{'} \in \{cl\}_T : \Delta t \leqslant real\_time\{cl_i, cl^{'}\} \end{cases} \tag{7}$$

$real_time$ is the real time from cl_i to $cl^{'}$ by querying GoogleMap API. $E_T = \{< cl_i, cl^{'} >\}$ is the set of the movement trajectory between the location sets cl_{T+1} and cl_T.

Algorithm 4 Frequency-Aware Dummy Selection Method

```
Input: $q$(each cell's query probability),$c_r$(real location on $T$),$\{R_T, t\}$(frequent regions on $T$ and time stamps $t$),$\{R_{T+1}, t\}$,$\epsilon$(customized radius),$k$(privacy level),$C_{dummy_T}$
Output: $C_{dummy_{T+1}}$
1: based on the formula (7) to generate dummy frequent regions $\{R_{T+1\,dummy}\}_{i=1}^{k}$ by the user's specific privacy level $k$ and customized radius $\epsilon$
2: if $C_{dummy_T}\ IN\ R_T$ then
3:     if $c_r \in R_{T+1}$ then
4:         run line 2-8 in Algorithm 3;
5:     else
6:         sort cells based on their query probability $q$ in the range of $\varsigma d(c_r, core_loc_{real})$ and $\tau d(c_r, core_loc_{real})$
7:         run line 2-7 in Algorithm 3
8:         based on the formula (7) to check these cells
9:         randomly select one and add into $C_{dummy_{T+1}}$
10: else
11:     if $c_r \in R_{T+1}$ then
12:         run line 2-8 in Algorithm 3
13:     else
14:         similar to line 6-9 in algorithm 4
15: return $C_{dummy_{T+1}}$
```

VI. Evaluation

A. Experiment Setup

We adopt (Microsoft Research Asia) Geolife project [18], [22], [23] to pick out a user's LBS data over the real road map of Beijing. This Geolife is one of the most popular data generators utilized in LBS privacy protection researches. We selected the dataset whose duration lasts more than 75 days and processed it. The first 30 days as the initial data, the last 35 days as the test data.

The specific method is that pick a data set from this project that has been recorded for more than 60 days and change his acquisition frequency and increase the acquisition frequency from $1-5$ seconds to 10 minutes. The time period of the entire data set is in $7:00-13:00$. The whole area of road map covers $10km \times 10km$, which is split into 40000 equal-sized grids, where the size of each grid is $0.05km \times 0.05km$, and we calculate the grid which each point belongs to. For simplicity, the path between the two points is simplified to the square through which their real path passes. At the same time, we assume that the user has the same privacy protection requirement in all requests. The next step is that execute the above-mentioned **Algorithm 1** and **Algorithm 2** for initial data, and get the frequent regions $\mathcal{R}$. Then this $\mathcal{R}$ is as a background to our protection method. Additionally, our experiments are performed on a PC wiht Intel Core i7-8550U, 16GB DDR3-1600 RAM and Microsoft Windows 10-64bit operating system.

In our following experiments, k is related to k-anonymity, which means the degree of anonymity, ε is the maximum distance to consider, and M, describing the number of points required to form a cluster.

B. An Intuitive Comparison by Using Thermogram

Considering that we are working on frequent areas and being able to visually demonstrate our results, we use 3D thermograms to illustrate the effects of long-term protection. For each input, we performed one test with k = 2. In this test, queried locations are, respectively, recorded for genuine location group and each dummy group.

In order to generate a heat map, the heat of each grid needs to be calculated. Firstly, the number of times t that locations appear in each grid is counted up. Then, calculating weight of each grid of $l_{i,j}$ is that $w_{m,n} = \frac{dis((m,n),(i,j))-min(dis)}{max(dis)-min(dis)}$

Based on these statistics, the heat value of each grid is calculated by following equation:

$$h_{i,j} = \sum_{m=1}^{length} \sum_{n=1}^{width} (w_{m,n} \times t_{i,j}) \tag{8}$$

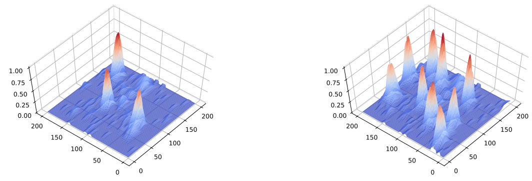
After data processing, thermogram were created according to heat value matrices. Fig. 5 shows two thermogram maps generated for different methods using the same input and parameter $k = 2$. Fig 5(a) is corresponding to the original data(i.e., the user's real location) using the baseline method, while Fig 5(b) is corresponding to 2-anonymity of our method. In thermogram maps, x and y are location coordinates and z represents heat.

As we can see, there are three high peaks in the data using baseline method in Fig 5(a), and which means that the user may have a higher possibility of appearing there. When the user initiates the continued request, the adversary can have a high confidence that the user is in the region which has a high peak, thereby, dummy locations generated for the user could be invalid. When using our method, more dummy areas are generated for the user, and these regions also have high peaks, as shown in Fig 5(b). Adversaries cannot distinguish the genuine one from others because they all look similar to each other intuitively.

Results in Fig 5 indicate that the dummy-based baseline method cannot protect the user's location privacy completely.

C. Quantifiable results

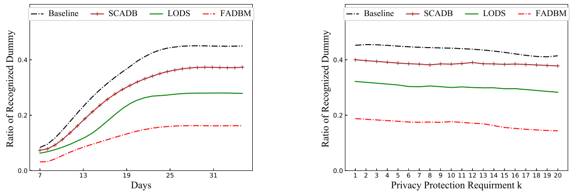
In this group of experiments, we compared our proposed FADBM with three existing methods, SCADB which represents the dummy selection method [4], and LODS [8]. Additionally, these two methods cannot be applied to our data directly, thereby, we keep the core ideas of these methods while some changes were made to accommodate our data. The baseline method described in Sec. V-B is also compared to better understand our method. These five methods are processing in the test data. Then, they are attacked by the hypothetical adversary method mentioned in the Sec. IV, and we used the corresponding indicators for evaluation. $\varepsilon = 7$, and $M = \frac{Days}{1.5}$.



(a) 3-anonymity baseline method (b) our 3-anonymity method

Fig. 5. Comparison of effects after using our method

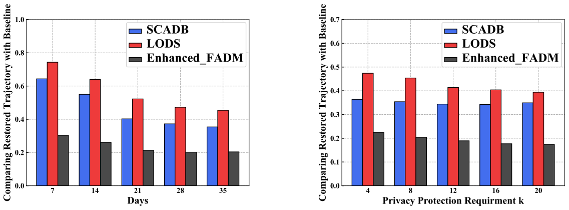
1) Recognized Dummy: Recognized_Dummy is a metric about how many dummy locations are recognized within the user's continued queries. Rec_Dummy is calculated by the recognized dummy locations belonging to $dummy\_loc$ divided by the all dummy locations in n queries. Specific calculation method is in Sec. IV-D. This metric is considered from two aspects, the change of $Days$ and k. Fig. 6 shows the effects of $Days$ and k on the value of Rec_Dummy. In Fig. 6(a), we chose to change $Days$ and remain $k = 8$. There is a a considerable increase occurred from $Days = 7$ to $Days = 25$, and the rate of increase slow down form from $Days = 25$ to $Days = 35$. The curve eventually flattens out. The maximal values of Rec_dummy in SCADB and baseline are at the top of the figure since they do not consider frequent regions at all. LODS performs better than SCADB due to part of using frequent regions, more precisely, this method is thinking about a location, not an region. Our FADBM performs much better than all of them and their maximal values are 0.16 and 0.13. The reason is that it carefully selects the dummy which have higher contribution to generate dummy frequent regions. The results can be explained in Fig. 6(b) using the same reason. These results show that our method makes a good utilization of background knowledge $\mathcal{R}$ and thus decrease Rec_Dummy.



(a) Changes over days with $k = 8$ (b) Changes over k with $Days = 30$

Fig. 6. Metric of recognized dummy

2) Restored Trajectory: The restored trajectory is about how accurate the frequent trajectories are restored. This metric can be analyzed in two parts, the $Days$ and k. To better understand these methods, all values of other methods are compared to the baseline's value. Fig. 7(a) shows the effects of $Days$ on the restored trajectory. Obviously, the privacy degree of LODS is the worst since it does not use time reachability. SCADB performs better since it checks time accessibility for adjacent points. However, it does not take accessibility between frequent areas over time into account. Our FADBM has much higher privacy degree than all those methods due to our more advanced design of frequent-aware dummy selection. In particular, FADBM performs better than others due to consideration of accessibility between frequent areas over time in selecting dummy locations. Fig. 7(b) further explains how k affects Res_Traj. LODS [11] does not consider caching, and hence its effect of protecting trajectory is not satisfactory and around 43%. Since SCADB considers short-term accessibility, its Res_Traj stays at a lower level around 35%. While in our methods, dummy selection optimizes protecting trajectory as one objective, and such frequent-aware dummy selection achieves a much lower restored trajectory than other methods.



(a) Changes over days with $k = 8$ (b) Changes over k with $Days = 30$

Fig. 7. Metric of restored trajectory

VII. Conclusion

In this paper, we proposed frequency-aware dummy-based method to protect user's location privacy against hypothetical adversary method. The method achieves k-anonymity effectively by selecting dummy locations considering frequent regions and time reachability to ensure that the selected dummy locations are reasonable as far as possible. Evaluation results indicate that the proposed methods can significantly improve the privacy level.

Acknowledgment

This research is supported by the National Key Research and Development Program of China (no. 2017YFB1400601), the National Science Foundation of China under Grant no. 61672276, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University.

References

[1] C. Hu, W. Li, X. Cheng, J. Yu, S. Wang, and R. Bie, "A secure and verifiable access control scheme for big data storage in clouds," *IEEE Transactions on Big data*, vol. 4, no. 3, pp. 341–355, 2017.

[2] A. Quattrone, E. Naghizade, L. Kulik, and E. Tanin, "Tell me what you want and i will tell others where you have been," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 1783–1786.

[3] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 754–762.

[4] H. Liu, X. Li, H. Li, J. Ma, and X. Ma, "Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

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

[6] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving privacy with probabilistic indistinguishability in weighted social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1417–1429, 2016.

[7] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2015, pp. 1017–1025.

[8] F. Li, Y. Chen, B. Niu, Y. He, K. Geng, and J. Cao, "Achieving personalized k-anonymity against long-term observation in location-based services," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[9] X. Wang, A. Pande, J. Zhu, and P. Mohapatra, "Stamp: Enabling privacy-preserving location proofs for mobile users," *IEEE/ACM transactions on networking*, vol. 24, no. 6, pp. 3276–3289, 2016.

[10] S. Zhang, G. Wang, Q. Liu, and J. H. Abawajy, "A trajectory privacy-preserving scheme based on query exchange in mobile social networks," *Soft Computing*, vol. 22, no. 18, pp. 6121–6133, 2018.

[11] D. Wu, Y. Zhang, and Y. Liu, "Dummy location selection scheme for k-anonymity in location based services," in *2017 IEEE Trustcom/BigDataSE/ICESS*. IEEE, 2017, pp. 441–448.

[12] F. Tang, J. Li, I. You, and M. Guo, "Long-term location privacy protection for location-based services in mobile cloud computing," *Soft Computing*, vol. 20, no. 5, pp. 1735–1747, 2016.

[13] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, "User-defined privacy grid system for continuous location-based services," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2158–2172, 2015.

[14] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[15] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*. IEEE, 2005, pp. 194–205.

[16] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *ICPS'05. Proceedings. International Conference on Pervasive Services, 2005.* IEEE, 2005, pp. 88–97.

[17] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*. ACM, 2008, pp. 16–23.

[18] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *nature*, vol. 453, no. 7196, p. 779, 2008.

[19] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.

[20] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *International Workshop on Privacy Enhancing Technologies*. Springer, 2002, pp. 41–53.

[21] T. Xu and Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 547–555.

[22] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

[23] Y. Zheng, X. Xie, W.-Y. Ma *et al.*, "Geolife: A collaborative social networking service among user, location and trajectory." *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.