Scientific Research Publishing

Journal of Computer and Communications, 2021, 9, 35-56
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Semantically Sensitive Privacy Protection Method for Trajectory Publishing

Zhijian Shao, Bingwen Feng, Xingzheng Li

College of Information Science and Technology, Jinan University, Guangzhou, China
Email: shaozhijian1@gmail.com

How to cite this paper: Shao, Z.J., Feng, B.W, and Li, X.Z. (2021) A Semantically Sensitive Privacy Protection Method for Trajectory Publishing. *Journal of Computer and Communications*, **9**, 35-56.
https://doi.org/10.4236/jcc.2021.94003

Received: March 10, 2021
Accepted: April 13, 2021
Published: April 16, 2021

Abstract

Trajectory data set is the indispensable foundation for constructing reliable Internet of Vehicles (IoV) service and location-based service (LBS), while it is likely to be abused by malicious attackers to infer user's privacy. In this paper, we propose a trajectory protection method based on stop points obfuscation, which can confront various privacy attacks and preserve the semantic information to achieve adequate utility. Two strategies for stop point selection are designed, including category-distance priority method and Markov matrix method. Our new method was analyzed and evaluated on a real-world trajectory data set. The experiment result shows that our method can improve the utility of the data set and provide multi-level privacy protection.

Keywords

Internet of Vehicles, Privacy Protection, Trajectory Data, Markvo Matrix

1. Introduction

With the development of the Internet of Vehicles (IoV) and the popularization of intelligent positioning technology, many new in-vehicles service applications are designed to assist transporting [1]. For example, there are smart taxi dispatching service and vehicle hiring service like Uber, real-time location recommendation service for searching destination and traffic forecast [2], as well as the high-profile auto-driving technology [3]. The trajectory data sets are indispensable for developing these in-vehicles services, especially when it involves the prevalent machine learning method, which makes the data sets could directly affect the quality of service. Therefore, the status of trajectory data set is becoming more and more important.

However, if the raw data collected from the vehicles or user's personal terminal device is published without proper treatment, it might lead to serious privacy

risk. A malicious attacker can query the public data set and extract specific victim user's personal information, including occupation, working location, lifestyle, and even interpersonal network [4]. Because of such privacy leakage, the victim might suffer from annoying advertising and fraud, or even encounter life safety threats in extreme cases. This issue has received much attention from the academia, and researchers have proposed various approaches to protect the trajectory data, including dummy-based approach [5] [6], trajectory synthesis for k-anonymity [7] [8], suppression approach [9] and machine learning approach [10].

While existing approaches have contributed a lot based on the statistical features, researchers usually gave less consideration on the semantic trait, *i.e.*, the users' movement and daily activity patterns carried by trajectory. Though forward-mentioned approaches can generate fake trajectories with high entropy to obfuscate the data set, they inevitably lead to semantic loss and significantly reduce the utility. Specifically, the data set could not reflect the correct and effective user movement patterns after desensitization. If such a data set is used in machine learning applications, it will produce a defective model which gives biased results.

From the perspective of pattern recognition and privacy attack, the semantic information of trajectory data is mainly carried by stop points, where users spent longer time [11]. Because a long staying time implies the purpose of visiting specific locations and can further deduce the user's life habits and privacy information. On the contrary, short-stays middle points only reflect the fact that the user has passed by, but could not provide other sensitive information. Thus, designing a protection algorithm for stop points can not only achieve semantic sensitivity but also can reach reliable protection results with lower computational complexity.

Based on this observation, we proposed a semantic sensitive privacy protection algorithm for trajectory data. First, we extract the stop point information from every trajectory data entry, and determine their Point-of-Interest (POI) description by querying LBS database. Second, the stop points will be categorized into multiple types, and we compute the Markov matrix among these stop point categories, which represents the probability of the transition from one category of stop point to another. Then the matrix can be used to guide the obfuscation process. For each stop point on a trajectory, we can select the obfuscated stop point with two strategies: category-distance priority method or Markov matrix method. After the stop points are obfuscated, the algorithm will generate the remaining middle points according to the shape of origin trajectory. At last, the algorithm will verify whether the new trajectory shows high-fidelity statistical trait.

The main contribution of this paper is summarized as three points:

- We propose a semantically sensitive privacy protection algorithm for trajectory data set. The new algorithm can cloak the users' personal information and confront known privacy attacks, while simultaneously preserving the es-

sential semantic information.

- We design two stop point obfuscation strategies: category-distance priority method and Markov matrix method. The advantages and applicable scenes of each are studied. Data providers can adapt one of the strategies flexibly according to specific scenarios to provide heterogeneous protection levels.
- We evaluate the new algorithm on a real-world data set and compare the results with the existing dummy-based algorithms. The experiment results show that our algorithm can better serve the utility requirements, meanwhile provide multi-level protection for trajectory data.

The rest of this paper is organized as follows. In Section 2, we discuss the related works. Preliminaries are presented in Section 3. Section 4 is where we explain the new algorithm in detail and Section 5 contains the evaluation of the algorithm. We discuss the limitation of our work and a few interesting findings during the research in Section 6 and finally conclude the paper in Section 7.

2. Related Work

2.1. Privacy Risks on Trajectory Data

The privacy risks on public trajectory data set have drawn much attention from researchers. Re-identification attack is a kind of privacy attack that could identify a specific user from a large data set. In this threat model, the attacker already obtains enough background knowledge about the victim, and he could re-identify the victim's daily trajectory by querying the public data set in a targeted manner. Pellungrini *et al.* [4] concluded a privacy risk assessment model and proposed various attacks kind according to the background knowledge requirement. They categorized the background knowledge into three kinds: location visit fact, location visit frequency, and location visit probability.

A more sophisticated attack called semantic attack was proposed by Sui *et al.* [12] in 2016. In this threat model, the attacker can query the semantic information from the trajectory data and infer the victim's behavior when combining POI distribution on the map. This attack could reveal victim's life habits and even interpersonal network, thus lead to severer privacy disclosure.

2.2. Privacy Protection Methods on Trajectory Data

Many methods have been proposed for data protection, including: K-anonymity approach, suppression approach, machine learning approach, and dummy-based approach. K-anonymity [13] is a widely used privacy protection model first proposed in 2002, this model ensures that the unique information for a data entry is indistinguishable from at least other k-1 data entries, but it can only confront the re-identification attacks when a few concomitant policies are followed. In 2007, Machanavajjhala *et al.* [14] concluded that with adequate background knowledge, attackers can break the K-anonymity protection and cause privacy disclose, they examined the weakness of previous principle and proposed a new protection model call L-diversity. L-diversity model ensures l well-represented

values for the sensitive attributes, but well-represented criteria could be difficult to define in certain use cases. So this method is not universally applicable. Nergiz *et al.* [7] redefined the K-anonymity concept for the field of trajectory data. Based on the new definition, they also designed an anonymization algorithm. However, the computational complexity of the trajectory grouping process in this algorithm is relatively high, which makes it difficult to apply to a large data set. Niu *et al.* [8] proposed a Dummy-Location Selection (DLS) algorithm on user location data sets, it can be used to ensure K-anonymity for the POI querying service. Though their research did not cover the trajectory protection topic, their dummy location selection idea made significant impacts on later researches. In general, K-anonymity approaches synthesis new trajectories from a couple of similar trajectories. These methods usually involve clustering techniques, which require extensive computing resources thus hard to run on large data sets.

Suppression is another typical idea for privacy protection, Sweeney [15] first mentioned general data sets can achieve K-anonymity with proper generalization and suppression operations. Suppression refers to the operation of removing high-risk information from the data sets to reduce the overall privacy risk. Pellungrini *et al.* [4] analyzed the privacy risk in human mobility data sets and proposed a series of attacks that could lead to serious privacy disclosure. As countermeasures, they removed the data entry from the data set with higher risk and then examine the data set with various collective measurements. The evaluation results suggested that the suppression could make the data set present higher entropy with little overall distortion, but this method inevitably drops the unique information of individuals. Chen *et al.* [16] proposed a more advanced suppression algorithm, they first introduce the concept of violating sequence, which is used to detect whether particular parts of a trajectory contribute to higher privacy risk. So that one can remove the violating sequence to hinder the threats. This method could reduce the privacy risk with lower information loss, but it did not concern the semantic information of the trajectory data. Zhao *et al.* [9] suggested a suppression method based on trajectory frequency, they designed two algorithms for minimizing the information loss caused by suppression. Suppression-based approaches are simple and universally applicable, but they ignore the semantic information carried by the trajectory data so the anonymization will cause significant utility loss.

More recently, Shaham *et al.* [10] conducted a research about how to protect the trajectory data with machine learning techniques. They designed a new privacy protection framework called MLA, which accepts the original data set and a parameter of the privacy metric k. The framework is consists of three procedures: generalization, alignment and clustering, and the output will be an anonymized data set. A state-of-art technique call generalization hierarchy trees (DGH) is used in the generalization model for processing spatiotemporal trajectory data sets, and they designed multiple algorithms for the clustering process. The experiment was conducted on a 1 km × 1 km area with dense trajectory data. Since alignment and clustering algorithms are time-consuming when running on long

trajectories with plenty of stop points, this machine learning method is not suitable for the sparse and large data set.

Apart from the aforementioned methods, many dummy-based privacy protection approaches were proposed, which require less computing resources and can run smoothly on a large data set. Kato *et al.* [17] designed an algorithm to synthesis high fidelity dummy trajectory on top of reachable area and pause positions. The algorithm can generate massive dummy trajectories to protect real user data. They discussed a lot of spatiotemporal features of dummy trajectories, but the algorithm did not take the semantic information into account. Lei *et al.* [6] also designed a dummy-based protection algorithm on top of the Dummy-Location Selection (DLS) [8] method, this algorithm is easy to implement and can run efficiently, but it also only focus on the geometric features of the new trajectory but ignore the semantic elements. More recently, Zhao *et al.* [18] improved the dummy-based approach by securing start-points and end-points, and combined the bidirectional dummy trajectory generation idea to design a new algorithm. Overall, dummy-based approaches have lower computational complexity and also can be adapted on sparse or dense trajectory data sets. With these methods, data providers can easily generate a great number of indistinguishable dummy trajectories to protect the privacy of the real user. It is promising to introduce semantic information and develop a better protection scheme with the dummy-generation strategy.

In summary, the previous protection methods could be configured to meet the privacy requirement of many application scenarios, but most of them did not consider the semantic information carried by the trajectory data, therefore the utility of data sets were seriously lost after the protection process. Also, the approaches which involve clustering operation is not suitable for the sparse and gigantic data set. After realizing the previous shortcomings, our new method is designed with the care of the semantic information and the operating efficiency.

3. Preliminaries

3.1. Trajectory Data

Definition 1: Trajectory Data

A trajectory data entry is consist of a sequence of spatiotemporal 3-dimensional tuples and user identifier, which can be represented as follows:

$$T = \{(x_1, y_1, t_1), (x_2, y_2, t_2), \cdots, (x_n, y_n, t_n), U_i\} \quad (1)$$

where (x_i, y_i) is the location point and t_i is the timestamp; U_i is the user identifier. A trajectory data set is consists of a number of trajectory data entries, noted that there can be many trajectory data entries linked to a unique user.

Definition 2: Location Semantic Data

In our context, the semantic information of a location point refers to the point-of-interest (POI) description, which can be queried from a LBS database or a LBS open API. We assume that each location (x_i, y_i) correspond to a

unique POI description. All POI information can be classified into multiple levels of categories, corresponding to semantic information of different granularities. For example, here is a POI description from the LBS database:

(*Jeff's cuisine*, *Chinese restaurant*, *Catering*).

The first element in the tuple is the description of this POI, the second element is second-level category information, the third element is the first-level category information. It is a hierarchical directory structure where many second-level categories locate under the first-level. For example, under the *Catering* category, there is second-level category of *Chinese restaurant*, *foreign restaurant*, *coffee shop*, *bar*, and so on. The hierarchic category information makes it convenient to quickly locate the target POI and we can leverage this feature to achieve multiple levels of privacy protection.

3.2. Related Concepts

Definition 3: Markov Matrix

A Markov matrix can be used to represent the stops in a Markov chain [19]. It is used to describe the probability of each event depends only on the state attained in the previous event [20]. For instance, following representation shows that the probability of transfer from state 1 to state 2 is $P_{1,2}$ and so on.

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} \tag{2}$$

Markov matrix has been widely used in location-based machine learning tasks, like activity classification [21], future locations prediction [22] [23]. We will leverage Markov matrix to guide the obfuscation procedure in the new method.

Definition 4: Trajectory Movement Similarity

The movement similarity between two trajectories can be calculated by the sum of rotation angles on each pivot point [18], where pivot points refer to the locations on a trajectory other than the starting point and ending point. The movement similarity is defined as below:

$$\sigma_{a,b} = \sum_{j=1}^{k} \left| \theta_a^i - \theta_b^i \right| \tag{3}$$

where k is the number of pivot point, and θ_a^i represents the angle of ith pivot point on trajectory a. The pivot angle θ can be calculated by:

$$\theta = \arccos\left(\frac{\boldsymbol{L}_{i-1}\boldsymbol{L}_i \cdot \boldsymbol{L}_i\boldsymbol{L}_{i+1}}{\left|\boldsymbol{L}_{i-1}\boldsymbol{L}_i\right| \cdot \left|\boldsymbol{L}_i\boldsymbol{L}_{i+1}\right|} \right) \tag{4}$$

in which L_i is ith location point on the trajectory, so $\boldsymbol{L}_{i-1}\boldsymbol{L}_i$ and $\boldsymbol{L}_i\boldsymbol{L}_{i+1}$ represent a pair of consecutive direction vector.

Definition 5: Semantic Utility

The semantic utility metric is used to measure the utility of the dummy tra-

jectory, it is defined as:

$$w = \frac{\sum_{i=1}^{k} f\left(S_i^{dummy}, S_i^{real}\right)}{k} \tag{5}$$

where S_i^{dummy} is the POI description of ith stop point on the dummy trajectory and S_i^{real} is the corresponding one on real trajectory. f is a matching function to determine the similarity between two POI descriptions. It can be configured for multiple category levels discussed in Definition 2:

$$f_j\left(S_1, S_2\right) = \begin{cases} 1, & S_1.cat_j = S_2.cat_j \\ 0, & S_1.cat_j \neq S_2.cat_j \end{cases} \tag{6}$$

We use the notation $S_i.cat_j$ represent the jth level category attribute of S_i. Following is an example for explaining how the matching function works. Assume there are two POI description S_1 and S_2:

S_1 = (Jeff's cuisine, Chinese restaurant, Catering);

S_2 = (Starbucks, Coffee shop, Catering).

If the matching function is configured to match first-level category, then $f_1\left(S_1, S_2\right) = 1$, because both first-level POI description is *Catering*. But if it is configured for second-level category, then $f_2\left(S_1, S_2\right) = 0$. This design allows data provider to make balance between the utility and protection level by configuring which matching function to use.

4. Semantically Sensitive Trajectory Protection Algorithm

In this section, we are going to show the detail of the new trajectory protection algorithm. There are three main procedures in the algorithm: preprocessing, dummy trajectory synthesis, and trajectory correction.

4.1. Preprocessing

The purpose of the preprocessing step is extracting the semantic information from the trajectory data set and builds a comprehensive model to guide the synthesis. The preprocessing step involves trajectory data of every user by default, so it can extract the commonness of human behavior patterns from the daily movement. Alternatively, it can be configured to run on a small group of users to enhance the protection level for certain members.

4.1.1. Stop Point Detection

Stop point detection, as known as stay point detection, is a well-studied research topic [24]. Li *et al.* [25] proposed a robust stop point detection algorithm. In our context, the algorithm accepts a trajectory T, with two parameters for distance threshold and stopping time threshold, named *distThreh* and *timeThreh* respectively. The algorithm returns a list of location point $\left\{\left(x_1, y_1, t_1\right), \left(x_2, y_2, t_2\right), \cdots, \left(x_n, y_n, t_n\right)\right\}$, where the user at least spent *timeThreh* around each one. **Figure 1** demonstrates the result of stop detection on an example trajectory.

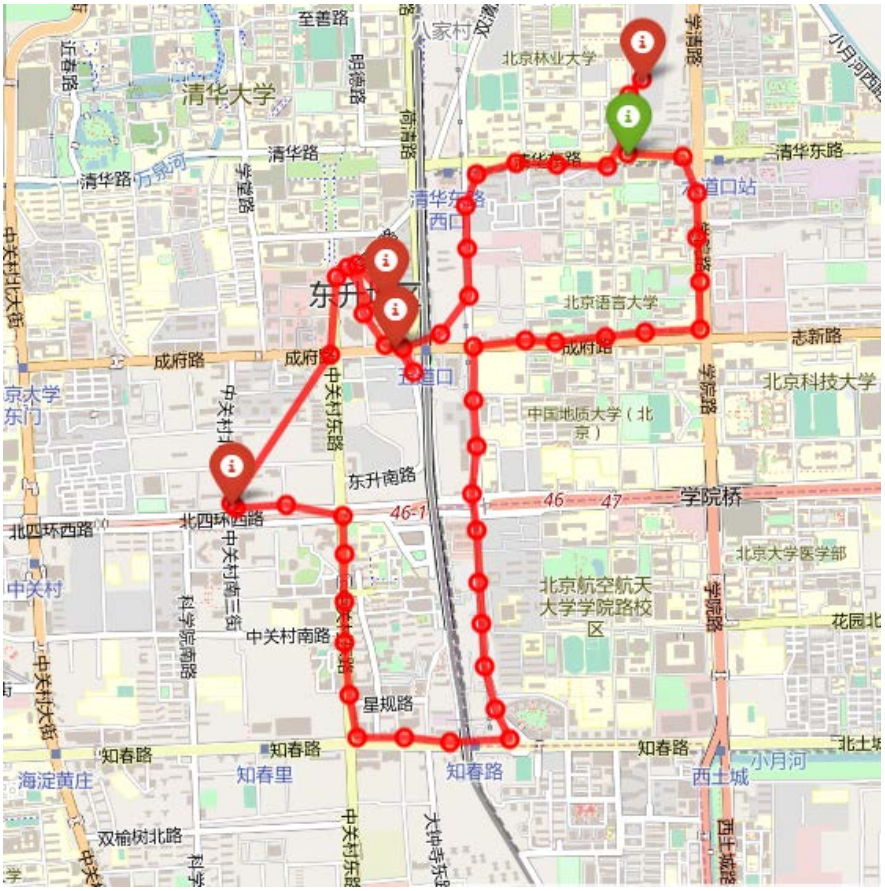

Figure 1. Stop point detection result on a example trajectory. The markers indicate the detected stop points and the red circles are non-stop points, *i.e.*, middle points.

4.1.2. POI Description Processing

As we have discussed in Section 1, stop points can reveal the visiting purpose and users' behavior patterns. In this step, it will link each detected stop point to corresponding POI information, which could be accomplished by querying a public LBS database or LBS API.

Most LBS databases and LBS APIs organize the POI data in hierarchy structure for better utility. In this procedure, the data provider can follow the existing category model of the LBS database, or adopt a new classification model based on specific application scenarios. The aim is categorizing the POI description and attach them to each detected stop point in following format: $\left(x, y, t, desc, cat^1, cat^2\right)$, where $desc$ is the detailed name of POI, cat^1 and cat^2 is first-level category and second-level category title respectively. Table 1 shows the first-level category and second-level category organization from the public Baidu LBS API [26].

4.1.3. Markov Matrix Calculation

The introduction of Markov matrix is to establish a simple but effective semantic-sensitive model to guide the obfuscation of stop points. We assume that users choose their next stop point only depending on their current location so that we

Table 1. Example category model of Baidu LBS API.

First-level category	Second-level category
Catering	Chinese restaurants, Foreign restaurants, Fastfood restaurants, Dessert shops, Coffee shops Bars, Others
Estate	Office building, residential areas, Dormitories, Internal buildings, Others
Company	Company, Office park, Agriculture and gardening, Factory and mines, Others
Shopping	Shopping mall, Store, Supermarket, Convenience store, Home building material, Digital home appliance, Agricultural market, Others

can compute the transition probability among different POI categories to extract users' behavior patterns. For the ease of demonstration, we use first-level category to explain the procedure. Assume that all stop points can be divided into k first-level category: $c_1, c_2, \cdots, c_k$, hence the size of transition matrix M is $k \times k$. A trajectory is decomposed into a set of movements among stop points, so the stop points on the trajectory can be reorganized into source-destination pairs:

$$src: \left(x_i, y_i, t_i, desc_i, cat_i^1, cat_i^2\right) \rightarrow dst: \left(x_{i+1}, y_{i+i}, t_{i+1}, desc_{i+1}, cat_{i+1}^1, cat_{i+1}^2\right).$$

Each item in the matrix M can be computed by:

$$M_{ij} = Pr\left(dst = C_j \mid src = C_i\right) = \frac{Pr\left(dst = C_j, src = C_i\right)}{Pr\left(src = C_i\right)} \tag{7}$$

The value on M_{ij} refers to the probability that, a user is currently stay on a stop point of C_i and he will move to a C_j stop point on next moment. Such a Markov matrix provides data support for selecting appropriate stop points and contributes to configurable privacy protection level in the later trajectory synthesis procedure.

Figure 2 is the visualization of a generated matrix, because most of the user's activities are carried out with the residence as the origin, so almost every row has the highest probability with the "*estate*" category. This feature has important impact on our algorithm design, which will be discussed in Section 4.2.

4.2. Dummy Trajectory Synthesis

This part is the core procedure of our new protection method. The dummy trajectory data generated from this step can replace the origin data, or append to the data set to achieve K-anonymity. Two major steps are designed for the generation, the first step is *stop points obfuscation* and the second step is *middle point generation*.

4.2.1. Stop Points Obfuscation

Most of the semantic information from a trajectory is carried by the stop points,

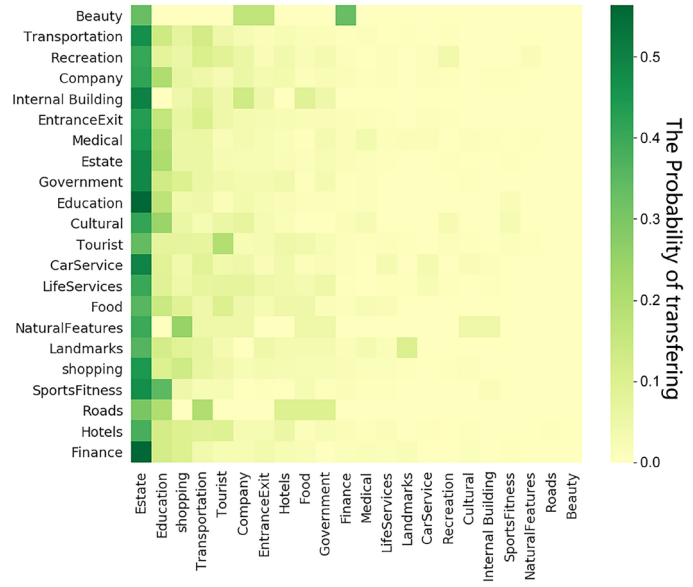

Figure 2. The POI category transition matrix from Geolife data set.

which are also considered to be the backbone of the trajectory data. Therefore how to obfuscate the stop points for privacy protection while not losing too much utility is a crucial but challenging problem. Our design is illustrated in listing **Algorithm 1**.

The workflow first obfuscates the start points with category-distance priority (CDP) method (line 2), then detect the stop point on the trajectory and perform obfuscation for each one with pre-configured method (line 4 - 10). The principle of two obfuscation methods and their details are discussed as follow:

Category-distance Priority (CDP) Method: The majority of trajectory-related machine learning tasks for IoV service, e.g. activity recognition [21], traffic forecast [27] and taxi dispatch [28] do not require precise address information. But for the malicious attacker, it is essential to obtain certain location descriptions to launch a privacy inference attack. For example, if a user is heading to restaurant C in B street block from No.10 building in A community. For normal data analysis tasks, it does not much care about whether the user is set off from No.10 building or No.12 building, and it also does not concern the name of restaurant. The key information is the fact that a user from A community is dining at B street block. Popular machine learning algorithm can detect the pattern behind the movement and generate a data model for specific tasks. On the other hand, an attacker needs to link certain location description with the victim's trajectory to perform attacks. Based on this observation, we proposed the category-distance priority method. It works as **Algorithm 2**.

Firstly, we determine the initial radius parameter r, and the category level j is chosen based on the privacy protection level (line 1). Secondly, the original stop point p is used as the center and r as the radius to search all POIs within the circle area from the LBS database (line 2). Thirdly, the result POIs will be sorted based on category-distance, the POI from the identical category has a higher

Input: Trajectory data $T = \{p_1, ...p_n\}$
Output: Obfuscated Trajectory data T_{dummy}
$T_{dummy} \leftarrow \varnothing$
$T_{dummy} \leftarrow \cup\, CDP(p_1)$
for $i \leftarrow 2$ to n **do**
  if (is_stop_point(p_i)) **then**
    if use MM **then**
      $T_{dummy} \leftarrow T_{dummy} \cup MM(p_{i-1}, p_i)$
    else if use CDP **then**
      $T_{dummy} \leftarrow T_{dummy} \cup CDP(p_i)$
    end if
  else
    $T_{dummy} \leftarrow T_{dummy} \cup p_i$
  end if
end for
return T_{dummy}

Algorithm 1. Stop point obfuscation.

Input: A Stop point with POI description:
$p = (x, y, t, desc, cat^j)$
Output: Obfuscated stop point p_{dummy}
1: $r \leftarrow r_{init}$
2: **while** $r < r_{max}$ **do**
3: $pois \leftarrow$ search_pois(p, r)
4: $pois \leftarrow$ sort_pois_by_category_distance($pois$)
5: $candidates \leftarrow \varnothing$
6: **for** pp in $pois$ **do**
7: **if** $pp.cat = cat^j$ **then**
8: $poi \leftarrow candidates \cup pp$
9: **end if**
10: **end for**
11: **if** length($candidates$) > 0 **then**
12: **return** $candidates[0]$
13: **else**
14: $r \leftarrow r + \Delta r$
15: **end if**
16: **end while**
17: **return** $pois[0]$

Algorithm 2. Category-distance priority method (CDP).

rank. If a candidate of the same kind is found, then it will be output as result (lines 4 - 12). If no suitable POI is found within the range, then radius r will be increased, and perform another search (line 14). Finally, if $r \geq r_{\max}$ but still could not find a POI of the same kind, we directly choose the POI with minimum distance from p as result (line 17).

This method is straight-forward and effective, it is able to retain the semantic information to the maximum extent, but it does not rule out the case that an attacker can deobfuscation with abundant external background knowledge. Therefore we designed another method for stronger protection.

Markov Matrix (MM) Method: In the Markov matrix we obtained from the preprocessing procedure, each row represents the probabilities that from the category of the current position to other categories on next position. Based on this feature, a direct idea is to locate the category which has the highest transition probability and search for the next POI from the target category. But from the experiment result, we found that because most of the user's activities are carried out with the residence as the origin, so in the generated Markov matrix, al-

most every row has the highest probability with the "*estate*" category. Figure 2 illustrates this trait.

If we chose the highest probability category directly as the target category for the next POI, then the new trajectory will only contains stop points from residential area, which is obviously inappropriate. In order to simulate the behavior patterns of genuine users, we use a weighted random function to select the category of next POI, which is denoted as weighted_random(C, P). C is the list of all categories and P is corresponding probability for each category, *i.e.*, the probability from each row of the matrix. After running sufficient times, the random number sequence generated by this function will meet the input probability distribution. The Markov matrix method works as described in Algorithm 3.

The workflow is generally similar to Algorithm 2, except that the target category is obtained from the weighted random function (Line 7). Also, the procedure is enclosed by a while loop (Lines 2 - 16). From the experiments, we conclude that it can always generate a POI from weighted_random function and break the loop with appropriate radius r.

4.2.2. Middle Points Generation

Once the stop point's obfuscation is finished, we can obtain a simplified but semantic sensitive dummy trajectory, which preserves most of the semantic information from the original trajectory but the privacy has been desensitized. Our aim is to get a high-fidelity trajectory, so it is also necessary to handle the middle points. In our context, middle points refer to the location points other than stop points on the trajectory. They carry less semantic information, but subtly affect the comprehensive characteristics like the overall shape and movement speed. We adopt an improved version of the algorithm from [6] for this task. Listing Algorithm 4 shows the details of this procedure.

The generation algorithm iterate over every location on the trajectory, if the location is a stop point, that means it has been secured by the obfuscation in last step, so we skip it (lines 2 - 4). As it is showed in Figure 3, for every middle point, it will generate a circle with p_{i-1} as center and $d = \left(\left|\boldsymbol{p}_{i-1}\boldsymbol{p}_i\right| + random\right)$

Input: Last stop point p_{i-1}, Current stop point p
Output: Obfuscated stop point p_{dummy}

```
1:  r ← r_init
2:  while True do
3:    pois ← search_pois(p, r)
4:    pois ← sort_pois_by_category_distance(pois)
5:    candidates ← ∅
6:    j ← cat_{i−1}
7:    target_cat = weighted_random(C, [M_j1 M_j2 ... M_jk])
8:    for pp in pois do
9:      if pp.cat = target_cat then
10:       poi ← candidates ∪ pp
11:     end if
12:   end for
13:   if length(candidates) > 0 then
14:     return candidates[0]
15:   end if
16: end while
```

Algorithm 3. Markov Matrix Method (MM).

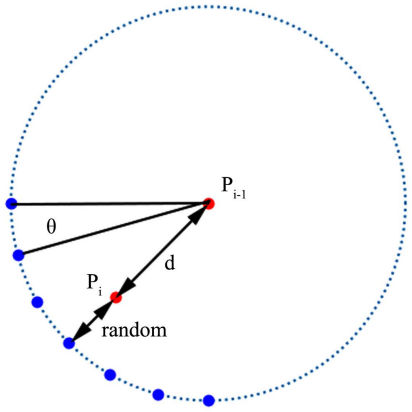

Figure 3. Middle points generation.

Input: Trajectory $T = \{p_1, ..., p_n\}$ output from Algorithm 1.
Output: New trajectory T_{dummy}

```
1: T_dummy ← ∅
2: for i ← 2 to n do
3:   if is_stop_point(p_i) then
4:     T_dummy ← T_dummy ∪ p_i
5:   else
6:     candidates ← ∅
7:     d = sqrt((p_i.x − p_{i−1}.x)^2 + (p_i.y − p_{i−1}.y)^2)
8:     d ← d + random
9:     for j ← −k to k do
10:      candidates ← candidates∪ dest(p_{i−1}, d, j · θ)
11:    end for
12:    T_dummy ← T_dummy∪ random_choice(candidates)
13:  end if
14: end for
```

Algorithm 4. Middle points generation.

as radius, and then pick a location to add to candidate set for every θ degree. $\text{dest}(p,d,\theta)$ is a function that determine a destination location from origin p with distance d and rotation angle θ.

In this procedure, θ and *random* can be configured to adjust the privacy protection level. The larger these values are, the dummy trajectory will have less similarity to the original one, thus better protection performance can be achieved.

4.3. Trajectory Correction

The last step is correcting the generated trajectory. First, we need to check the location number and update the timestamps on the dummy trajectory. Then we will compare whether the movement trend of the dummy trajectory is consistent with the origin.

The correction procedure will first ensures that the location point numbers of T^o and T^d are identical (line 1). Then it adds up the original timestamps with *random_shift* and assigns them to corresponding location point on T^d (lines 2 - 4). The *random_shift* should locate in a certain range which is appropriate for the configured protection level. Line 5 and 6 are computing the slope, *i.e.*, the movement trend of two trajectories, if the difference of slope is larger than a

threshold, then the dummy trajectory should be dropped and re-generate from step *B-2: middle points generation*. The **assert** keyword in Algorithm 5 represents a condition check, if the condition fails, it should return back to the *middle points generation* step.

5. Evaluation

5.1. Experiment Environment and Data Set

The experiments were conducted on a desktop with AMD Ryzen7 1700 CPU with 16GB RAM, which runs Ubuntu Linux LTS 18.04. We implemented the algorithm in Python3 with Jupyter Notebook.

Geolife data set (version 1.2.2) was collected by Microsoft Research Aisa and published in 2016 [29] [30] [31]. It contains 17,621 trajectories of 182 users from April 2007 to August 2012. The collected trajectories are mainly concentrated in Beijing urban area, Figure 4 shows the heat map of the data set.

In the new algorithm, it requires a LBS database or LBS API for querying the POI description. Since Geolife data set only contains the location information (latitude, longitude, timestamp), we chose the Web service API provided by Baidu LBS SDK [32] to provide additional POI information.

Input: Origin trajectory T^o, and dummy trajectory T^d output from Algorithm 4.
Output: Final trajectory T_{dummy}
assert length(T^o) = length(T^d)
for $i \leftarrow 1$ to length(T^o) **do**
  $p_i^d.t \leftarrow p_i^o.t + random\_timeshift$
end for
$l^o = \frac{\sum_{i=1}^{n}\left(x_i^o-\overline{x^o}\right)\left(y_i^o-\overline{y^o}\right)}{\sum_{i=1}^{n}\left(x_i^o-\overline{x^o}\right)^2}$
$l^d = \frac{\sum_{i=1}^{n}\left(x_i^d-\overline{x^d}\right)\left(y_i^d-\overline{y^d}\right)}{\sum_{i=1}^{n}\left(x_i^d-\overline{x^d}\right)^2}$
assert $|l^d - l^o| < l_{threh}$
return l^d

Algorithm 5. Trajectory correction.

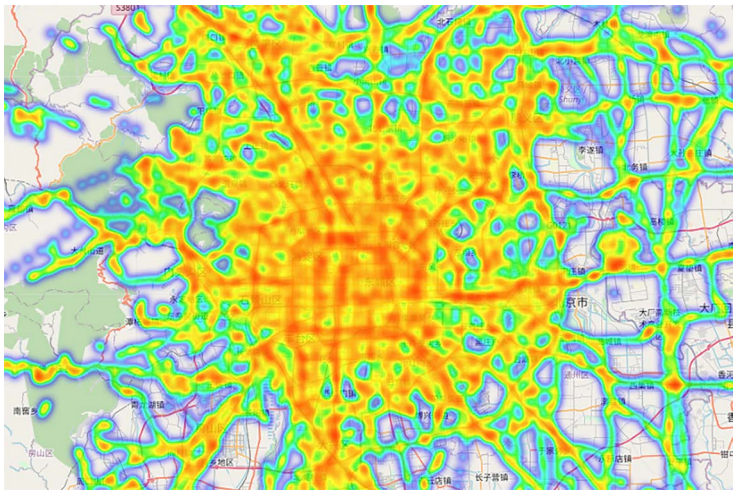

Figure 4. The heat map of Geolife data set around Beijing urban area.

5.2. Utility Loss Evaluation

From the definition of semantic utility (Equation (5)), we define the utility loss as:

$$u = 1 - \frac{\sum_{i=1}^{k} f\left(S_i^{dummy}, S_i^{real}\right)}{k} \tag{8}$$

Therefore, utility loss u will be valued between 0 and 1, suggesting the percentage of how much semantic information has lost for the new trajectory compared to the original data entry from the database. For comparison purpose, we select a typical dummy-based trajectory protection method proposed in [6], named *Dummy Spatiotemporal Correlation (DSC)*, to show our method is superior in preserving semantic information and able to achieve higher utility. We run the new algorithm against every trajectory in the data set with two methods: category-distance priority method and Markov matrix method, and we also record the result of DSC method as a baseline. We configured the θ parameter to 3˚ and the *random* value in the range of $(0,50]$ for the DSC method as well as the middle points generation procedure in our algorithms.

Figure 5 shows the probability density distribution of the experiment results. Figure 5(a) is the result of DSC method, the utility loss has the highest density at 0.5, and also shows a great number of occurrence at 1.0. If the utility loss is 0.5, that means half of the semantic information is lost after applying protection; if it reaches 1.0, that means the semantic information has almost damaged

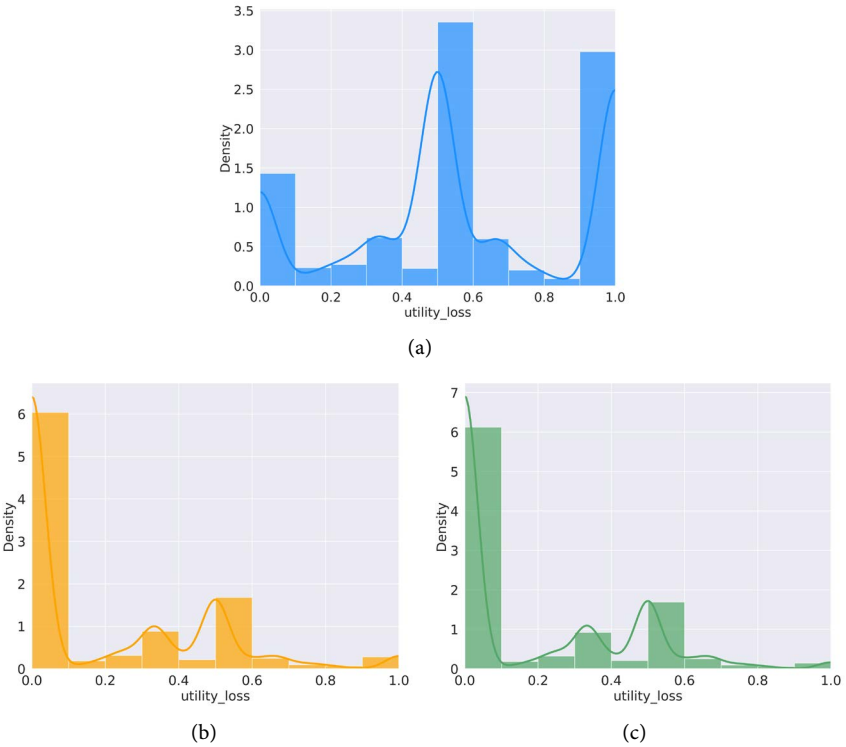

Figure 5. The probability density distribution of utility loss. (a) DSC method [6]; (b) CDP method; (c) Markov matrix method.

completely. This figure suggests the data set suffers from significant utility loss after the DSC protection treatment, because it does not consider the POI descriptions. **Figure 5(b)** and **Figure 5(c)** shows the results of our new methods. It shows that the probability shift to the left side, which means the utility loss has reduced in a considerable extent. Especially for the probability of higher utility loss, the probability of 0.5 drops from over 3.0 density to around 1.5, while the probability of 1.0 decreases to under 0.2. Meanwhile, the probability of 0 utility loss raise from 1.5 to over 6, suggesting that it can completely preserve the semantic elements for a large portion of trajectories. In terms of the comparison between category-distance priority method and Markov matrix method, they has almost identical distribution, except that Markov matrix method shows slightly lower density on the utility loss greater than 0.5.

As a conclusion for this evaluation, our new algorithm can preserve most semantic information after the privacy protection process. It can reduce the utility loss significantly when comparing to previous methods. Moreover, the category-distance priority method and the Markov matrix method have almost identical effect on semantic preservation.

5.3. Similarity Evaluation

Another important evaluation criterion for privacy protection algorithm is similarity, data provider usually need to make a trade-off between utility and similarity. A metric for comparing the similarity between two trajectories is defined by Equation (3) in Section 3.2. While there are many other methods to measure the similarity, this equation computes the difference of rotation angle on every pivot point. The larger the similarity value σ means less similarity between two trajectories, and it is more difficult for the attackers to recover the original trajectory.

In this evaluation, we chose the trajectories which contain less than 1000 location points from the data set, and generate a dummy trajectory from each of them. Then the similarity is computed between the original trajectory and the dummy trajectory. Other parameters for the algorithms are set to the same as we discussed in Section 5.2. The results are properly processed and drawn into box plots in **Figure 6**.

As it is shown in the graph, the trajectories which contain fewer location points have a wider range of similarity. **Figure 6(a)** shows the results of the DSC method, it is more concentrated and the median number is below 25 for all categories of location number. The results of our new methods show a similar trend on the similarity with respect to location points number, but the boxes in the graph have longer lengths, which suggests the data is more sparse. More importantly, the median values of all categories are larger than the ones in the DSC method, which are all above 25. Like the result from utility loss evaluation, the performance of the category-distance priority method and the Markov matrix method is quite similar. The only difference we concluded from the graphs is

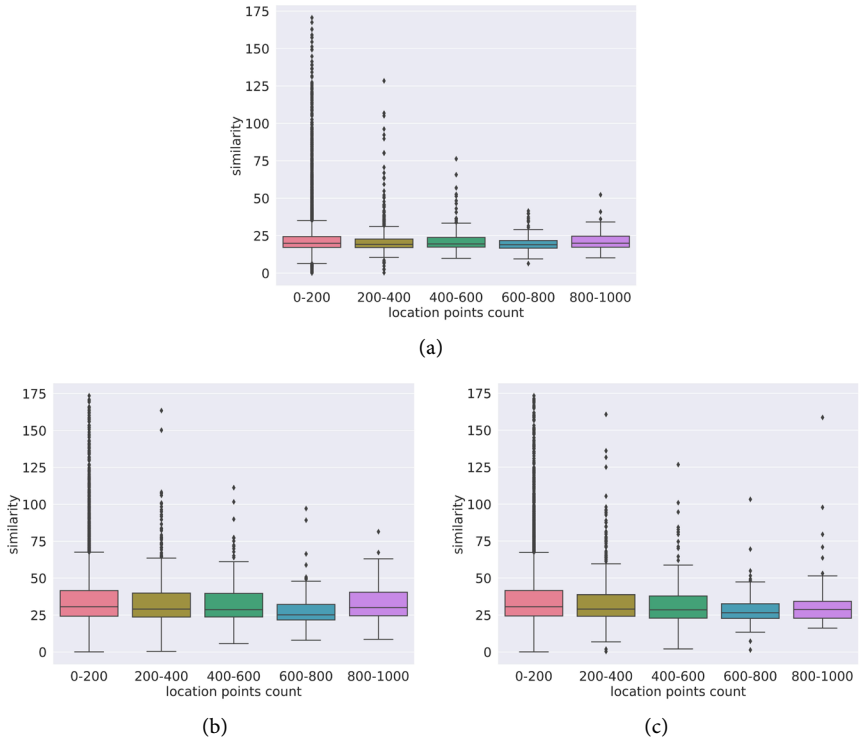

Figure 6. The box plots of similarity against various location number. (a) DSC method [6]; (b) CDP method; (c) Markov matrix method.

that the Markov matrix method can generate more condensed data for the trajectories with larger location points number (above 600).

In general, our new method can generate dummy trajectories with a higher σ value, and the performance is guaranteed on all lengths of trajectories. This evaluation demonstrated that our methods can provide better protection against the attackers in terms of the trajectories recovery.

5.4. Re-Identification Attack

In this experiment, we simulate the privacy attack method proposed in [4] to examine whether our new method can protect the trajectories data from re-identification attack. We randomly selected 10 users from the data set and applied the Home and Work attack, which is considered the most common and fundamental step to re-identify the trajectories of the victim with background knowledge. **Table 2** shows the result, the column named "Distance (DSC)" records the distance between the sensitive location predicted from original trajectories and the ones processed with the DSC method [6]. Likewise, "Distance (CDP)" and "Distance (MM)" are the results of the category-distance priority method and Markov matrix method. From the table, we can see that our algorithms can obfuscate users' trajectories and make the attack method return incorrect prediction. The distance of obfuscation is ranged from around 10 m to over 1 km, depends on the scope of user activities. One exception is user 22, it shows 0 distance because the Markov method fails to search for a suitable POI to

Table 2. Home and work attack result.

User id	Distance (DSC) [6]	Distance (CDP)	Distance (MM)
2 (Home)	59.9 m	18.5 m	174.2 m
2 (Work)	26.5 m	45.4 m	208.7 m
5 (Home)	17.0 m	22.7 m	21.0 m
5 (Work)	6.3 m	19.9 m	36.3 m
16 (Home)	188.8 m	12.8 m	40.1 m
16 (Work)	167.0 m	44.7 m	9.3 m
22 (Home)	139.6 m	36.2 m	0 m
22 (Work)	59.0 m	219 m	262.4 m
37 (Home)	327.1 m	46.0 m	178.6 m
37 (Work)	37.0 m	60.0 m	48.3 m
43 (Home)	56.3 m	54.5 m	19.3 m
43 (Work)	43.2 m	32.5 m	42.9 m
78 (Home)	54.0 m	229.9 m	73.3 m
78 (Work)	407.5 m	275.8 m	50.8 m
104 (Home)	24.1 m	144.6 m	103.6 m
104 (Work)	35.8 m	163.1 m	105.7 m
111 (Home)	110.2 m	21.3 m	106.8 m
111 (Work)	315.6 m	333.7 m	336.6 m
168 (Home)	75.6 m	68.5 m	1263.3 m
168 (Work)	606.8 m	609.4 m	51.0 m

replace the original one, thus a fallback happens. It is believed that such distances can thwart the re-identification attack based on the accurate location background knowledge.

6. Discussion

In this section, we will discuss a few extended topics, including some limitations of our method. The first topic is how to adapt the new method to various application schemes. The new method is flexible, and data providers can design various protection strategies according to their service features. The data provider can run our new method on every data entry from the data set and obfuscate important stop points. This strategy can preserve major semantic information and obfuscate sensitive privacy details, it is appropriate for machine learning tasks, such as traffic forecast and user activity patterns recognition. The data provider can also run the algorithm on each data entry multiple times to generate $k-1$ dummy trajectories and add to the data set. This strategy can achieve K-anonymity and increase the data volume in the data set. It is effective to prevent re-identification attacks and suitable for the application scheme which pro-

vides an API for users to query the data.

In terms of the limitations, since the new method rely on the LBS database to extract the semantic information of the trajectories, therefore the accuracy of the LBS database will remarkably affect the protection result. If the result from the LBS database is not accurate enough, it will lead to plenty of fallback situations during the stop point obfuscation procedure, in such situations the algorithm will simply choose the original stop point to avoid damaging the crucial semantic information. Likewise, the protection performance for the trajectory in rural areas or the areas with sparse POI information is not as good as those in urban areas, because these areas do not contain enough semantic elements to cloak the genuine movement data. However, because most applications involved the trajectories data set are revolve around city daily lives, so we believe that the new methods can fulfill the protection requirements of most application schemes.

7. Conclusion

In this paper, we survey previous trajectory data privacy protection schemes and conclude that most of the proposed methods did not consider the semantic features of the trajectories data. However, many IoV service involves machine learning heavily rely on the semantic information carried by the trajectories data to generate reliable models. To address this issue, we proposed a new trajectory protection method, which is divided into three main procedures: preprocessing, dummy trajectory synthesis, and trajectory correction. Based on the observation that most semantic information is implied by the stop points, we design two methods for stop points obfuscation, which are category-distance priority method and the Markov matrix method. Among them, Markov matrix method leverages the idea of the transition probability matrix and achieves better protection effects. We evaluate the new algorithm on the real-world data set Geolife, the experiment results show that the new algorithm has great advantages on utility loss and similarity metrics comparing to the previous representative dummy-based protection method. In addition, we perform a simulated re-identification attack, the result shows that the new methods can protect sensitive privacy information, e.g. home address and workplace location with proper obfuscation. We hope that this work can mitigate the conflict between the privacy requirement and data utility of trajectory data sets, and urge manufacturers to pay more attention to privacy protection in IoV services.

Acknowledgements

Zhijian Shao was partially supported by Key R&D Program of Guangdong Province (Grant No. 2019B010136003), Natural Science Foundation of Guangdong Province, China (Grant No. 2019B010137005, 2018A030313387). Bingwen Feng was partially supported by National Key R & D Plan of China (Grant No. 2017YFB0802203), National Natural Science Foundation of China (Grant No.61802145), Science and Technology Program of Guangzhou, China (Grant

No. 202007040004, 201804010428), the Fundamental Research Funds for the Central Universities, the Opening Project of State Key Laboratory of Information Security, the Opening Project of Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Rao, B. and Minakakis, L. (2003) Evolution of Mobile Location-Based Services. *Communications of the ACM*, **46**, 61-65. https://doi.org/10.1145/953460.953490

[2] Wiest, J., Höffken, M., Kreßel, U. and Dietmayer, K. (2012) Probabilistic Trajectory Prediction with Gaussian Mixture Models. 2012 *IEEE Intelligent Vehicles Symposium*, Madrid, 3-7 June 2012, 141-146. https://doi.org/10.1109/IVS.2012.6232277

[3] Li, X., Sun, Z., Cao, D., He, Z. and Zhu, Q. (2015) Real-Time Trajectory Planning for Autonomous Urban Driving: Framework, Algorithms, and Verifications. *IEEE/ASME Transactions on Mechatronics*, **21**, 740-753. https://doi.org/10.1109/TMECH.2015.2493980

[4] Pellungrini, R., Pappalardo, L., Pratesi, F. and Monreale, A. (2018) Analyzing Privacy Risk in Human Mobility Data. 2018 *Federation of International Conferences on Software Technologies: Applications and Foundations*, Toulouse, 25-29 June 2018, 114-129. https://doi.org/10.1007/978-3-030-04771-9_10

[5] Lei, P.-R., Peng, W., Su, I. and Chang, C.-P. (2012) Dummy-Based Schemes for Protecting Movement Trajectories. *Journal of Information Science and Engineering*, **28**, 335-350.

[6] Lei, K., Li, X., Liu, H., Pei, Z., Ma, J. and Li, H. (2016) Dummy Trajectory Privacy Protection Scheme for Trajectory Publishing Based on the Spatiotemporal Correlation. *Journal on Communications*, **37**, 156-164.

[7] Nergiz, M.E., Atzori, M., Saygin, Y. and Güç, B. (2009) Towards Trajectory Anonymization: A Generalization-Based Approach. *Transactions on Data Privacy*, **2**, 47-75.

[8] Niu, B., Li, Q., Zhu, X., Cao, G. and Li, H. (2014) Achieving K-Anonymity in Privacy-Aware Location-Based Services. *IEEE INFOCOM* 2014-*IEEE Conference on Computer Communications*, Toronto, 27 April-2 May 2014, 754-762. https://doi.org/10.1109/INFOCOM.2014.6848002

[9] Zhao, J., Zhang, Y., Li, X. and Ma, J. (2014) A Trajectory Privacy Protection Approach via Trajectory Frequency Suppression. *Chinese Journal of Computers*, **37**, 2096-2106.

[10] Shaham, S., Ding, M., Liu, B., Dang, S., Lin, Z. and Li, J. (2019) Privacy Preserving Location Data Publishing: A Machine Learning Approach. IEEE Transactions on Knowledge and Data Engineering, 1. arXiv:1902.08934 http://arxiv.org/abs/1902.08934 https://doi.org/10.1109/TKDE.2020.2964658

[11] Monreale, A., Trasarti, R., Renso, C., Pedreschi, D. and Bogorny, V. (2010) Preserving Privacy in Semantic-Rich Trajectories of Human Mobility. *Proceedings of the* 3*rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS*

and LBS, San Jose, November 2010, 47-54. https://doi.org/10.1145/1868470.1868481

[12] Sui, K., Zhao, Y., Dapeng Liu, Minghua Ma, Lei Xu, Zimu, L. and Pei, D. (2016) Your Trajectory Privacy Can Be Breached Even If You Walk in Groups. 2016 *IEEE/ACM* 24*th International Symposium on Quality of Service* (*IWQoS*), Beijing, 20-21 June 2016, 1-6. https://doi.org/10.1109/IWQoS.2016.7590444

[13] Sweeney, L. (2002) K-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **10**, 557-570. https://doi.org/10.1142/S0218488502001648

[14] Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M. (2007) *L*-Diversity: Privacy Beyond *K*-Anonymity. *ACM Transactions on Knowledge Discovery from Data*, **1**, 3-es. https://doi.org/10.1145/1217299.1217302

[15] Sweeney, L. (2002) Achieving K-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **10**, 571-588. https://doi.org/10.1142/S021848850200165X

[16] Chen, R., Fung, B. C., Mohammed, N., Desai, B. C. and Wang, K. (2013) Privacy-Preserving Trajectory Data Publishing by Local Suppression. *Information Sciences*, **231**, 83-97. https://doi.org/10.1016/j.ins.2011.07.035

[17] Kato, R., Iwata, M., Hara, T., Suzuki, A., Xie, X., Arase, Y. and Nishio, S. (2012) A Dummy-Based Anonymization Method Based on User Trajectory with Pauses. *Proceedings of the* 20*th International Conference on Advances in Geographic Information Systems*, Redondo Beach, November 2012, 249-258. https://doi.org/10.1145/2424321.2424354

[18] Zhao, Y., Luo, Y., Yu, Q. and Hu, Z. (2020) A Privacy-Preserving Trajectory Publication Method Based on Secure Start-Points and End-Points. *Mobile Information Systems*, **2020**, Article ID: 3429256. https://doi.org/10.1155/2020/3429256

[19] Gagniuc, P. A. (2017) Markov Chains: From Theory to Implementation and Experimentation. John Wiley & Sons, Hoboken.

[20] Oxford Dictionary (2020) Markov Chain.

[21] Bashir, F.I., Khokhar, A.A. and Schonfeld, D. (2007) Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models. *IEEE transactions on Image Processing*, **16**, 1912-1919. https://doi.org/10.1002/9781119387596

[22] Gambs, S., Killijian, M.-O. and del Prado Cortez, M.N. (2012) Next Place Prediction Using Mobility Markov Chains. *Proceedings of the* 1*st Workshop on Measurement, Privacy, and Mobility*, Bern, April 2012, Article No. 3. https://doi.org/10.1145/2181196.2181199

[23] Mathew, W., Raposo, R. and Martins, B. (2012) Predicting Future Locations with Hidden Markov Models. *Proceedings of the* 2012 *ACM Conference on Ubiquitous Computing*, Pittsburgh, September 2012, 911-918. https://doi.org/10.1145/2370216.2370421

[24] Zheng, Y. (2015) Trajectory Data Mining: An Overview. *ACM Transactions on Intelligent Systems and Technology*, **6**, Article No. 29. https://doi.org/10.1145/2743025

[25] Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W. and Ma, W.-Y. (2008) Mining User Similarity Based on Location History. *Proceedings of the* 16*th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Irvine, November 2008, Article No. 34. https://doi.org/10.1145/1463434.1463477

[26] Baidu.com (2020) LBS API Category Organization. http://lbsyun.baidu.com/index.php?title=lbscloud/poitags

[27] Zhao, Z., Chen, W., Wu, X., Chen, P. C. and Liu, J. (2017) Lstm Network: A Deep Learning Approach for Short-Term Traffic Forecast. *IET Intelligent Transport Systems*, **11**, 68-75. https://doi.org/10.1049/iet-its.2016.0208

[28] Yuan, N. J., Zheng, Y., Zhang, L. and Xie, X. (2013) T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. *IEEE Transactions on Knowledge and Data Engineering*, **25**, 2390-2403. https://doi.org/10.1109/TKDE.2012.153

[29] Zheng, Y., Zhang, L., Xie, X. and Ma, W.-Y. (2009) Mining Interesting Locations and Travel Sequences from GPS Trajectories. *Proceedings of the* 18*th International Conference on World Wide Web*, Madrid, April 2009, 791-800. https://doi.org/10.1145/1526709.1526816

[30] Zheng, Y., Li, Q., Chen, Y., Xie, X. and Ma, W.-Y. (2008) Understanding Mobility Based on GPS Data. *Proceedings of the* 10*th International Conference on Ubiquitous Computing*, Seoul, September 2008, 312-321. https://doi.org/10.1145/1409635.1409677

[31] Zheng, Y., Xie, X. and Ma, W.-Y. (2010) Geolife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data*(*Base*) *Engineering Bulletin*, **33**, 32-39.

[32] Baidu.com (2020) Web Service API. http://lbsyun.baidu.com/index.php?title=webapi