This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2018.2855740, IEEE Transactions on Services Computing

DP-LTOD: Differential Privacy Latent Trajectory Community Discovering Services over Location-Based Social Networks

Changqiao Xu, *Senior Member, IEEE,* Liang Zhu, Yang Liu, *Member, IEEE,*
Jianfeng Guan, and Shui Yu, *Senior Member, IEEE*

Abstract—Community detection for Location-based Social Networks (LBSNs) has been received great attention mainly in the field of large-scale Wireless Communication Networks. In this paper, we present a Differential Privacy Latent Trajectory cOmmunity Discovering (DP-LTOD) scheme, which obfuscates original trajectory sequences into differential privacy-guaranteed trajectory sequences for trajectory privacy-preserving, and discovers latent trajectory communities through clustering the uploaded trajectory sequences. Different with traditional trajectory privacy-preserving methods, we first partition original trajectory sequence into different segments. Then, the suitable locations and segments are selected to constitute obfuscated trajectory sequence. Specifically, we formulate the trajectory obfuscation problem to select an optimal trajectory sequence which has the smallest difference with original trajectory sequence. In order to prevent privacy leakage, we add Laplace noise and exponential noise to the outputs during the stages of location obfuscation matrix generation and trajectory sequence function generation, respectively. Through formal privacy analysis, we prove that DP-LTOD scheme can guarantee ϵ-differential private. Moreover, we develop a trajectory clustering algorithm to classify the trajectories into different kinds of clusters according to semantic distance and geographical distance. Extensive experiments on two real-world datasets illustrate that our DP-LTOD scheme can not only discover latent trajectory communities, but also protect user privacy from leaking.

Index Terms—Location-based Social Networks, Communication Detection, Trajectory Clustering, Privacy Preserving.

1 Introduction

Recently, Location-based Social Networks (LBSNs) (e.g. Foursquare, Facebook Place, Twitter or Geolife, etc.) emerged due to the rapid development of online social networks and physical localization technologies. The characteristic is that people can make use of "check-in" to achieve the sharing and propagation of location-based services in virtual world [1].

As shown in Fig.1, three relational graphs (i.e. ① *user-location*, ② *user-user*, ③ *location-location*) are generated for LBSNs. User-location graph reflects the relation between user and location, in which different locations are visited by different users. User-user graph represents the relation between two users, such as friendship, common interests or preference, etc. Location-location graph shows the relation between two locations, which can explain the physical distance or the similar semantical information among locations.

According to the analysis of graph theory, large numbers of potential social relationship among users may be mined [2]. Meanwhile, trajectory data reflects the sequence of visited locations associated with time. With the collection and storage of large numbers of trajectory data, the users who have similar interest or preference can be clustered into a community, in order to share the common interested contents (e.g. travel routes, restaurants or entertainments, etc.) with each other [3]. Also, trajectory community discovering can offer help for many kinds of applications, such as personalized service recommendation [1], [4], content distribution [5], [6], [7], [8] and intelligent transportation systems [9], [10], and so on. However, the problem of privacy leakage is inevitable when users publish their real trajectory information to LBSNs server. Untrusted third parties may take advantage of the publicly available data to commit malicious activities [11]. For example, attackers can deduce the personalized information (e.g. family address, working place, or living habit) of victims by analyzing the Global Positioning System (GPS) trajectories. If users find out that their privacy may be stolen by the attackers, they will no longer contribute to the location-based services. Therefore, privacy-preserving is important for LBSNs to guarantee good performance.

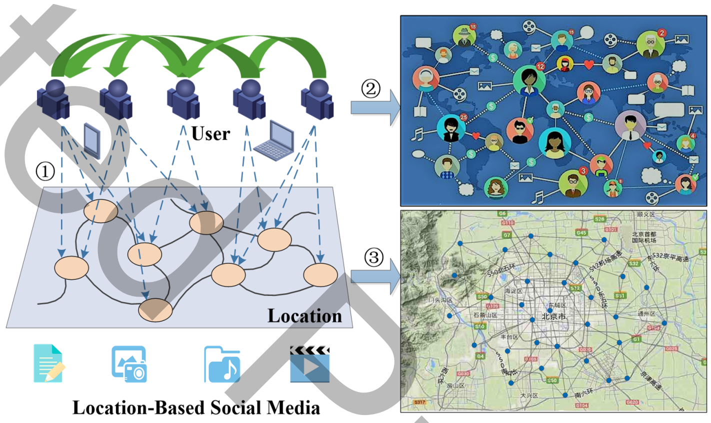


Fig. 1. The architecture of LBSNs.

The common goal of all privacy-preserving methods is that they must protect the user privacy against information leakage, meanwhile provide various of services with high precision. This is a challenging task, particularly for the tradeoff between privacy-preserving and data

- *C. Xu, Y. Liu and J. Guan are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (e-mail: cqxu@bupt.edu.cn; liu.yang@bupt.edu.cn; jfguan@bupt.edu.cn).*
- *L. Zhu is with the School of Computer and Communication Engineering, Zhengzhou University of Light Industry (e-mail: lzhu@zzuli.edu.cn).*
- *S. Yu is with the School of Software, University of Technology Sydney (e-mail: shui.yu@uts.edu.au).*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2018.2855740, IEEE Transactions on Services Computing

utility [12]. Traditional privacy-preserving methods mainly include *k-anonymity* [13], *l-diversity* [14] and *t-closeness* [15]. Take the familiar *k*-anonymity for example, it requires each record in the published datasets to be consisted of at least *k* different participants. As we know, *k*-anonymity achieves the privacy-preserving by shielding or obfuscating some fields of the databases. For linking attack, adversaries can not link any records to the specific participants. However, it only provides limited privacy-preserving because of different level of background knowledge. That is to say, k-anonymity can fail to provide adequate privacy protection if the attacker has enough background knowledge. In practice, attackers can still make use of the additional data sources and theoretical analysis to deduce the sensitive information in databases.

Based on traditional privacy-preserving methods, privacy-preserving in LBSNs can be divided into location privacy-preserving and trajectory privacy-preserving. For location privacy-preserving, the aim is to protect user location information from leaking. Current location privacy-preserving methods mainly utilize Trusted Third Party (TTP) to obfuscate the actual location information. Although the data utility is guaranteed, they depend on the assumption that TTP is fully credible. Therefore, the location information could still be stolen by attacking TTP. Corresponding to location privacy-preserving, trajectory privacy-preserving is to prevent sensitive location information from leaking when attackers acquire the whole trajectory information of victims. However, the data utility is limited for existing trajectory privacy-preserving methods.

In this paper, we focus on Differential Privacy Latent Trajectory cOmmunity Discovering (DP-LTOD), which is a new framework that not only protects user trajectory information from leaking, but also provides effective services for users. Latent Trajectory cOmmunity Discovering (LTOD) aims to cluster the users who have similar interest or preference into a community. By analyzing the uploaded trajectory data, the corresponding interest or preference of each user can be mined. To make LTOD satisfy differential privacy, we firstly transform true locations of users to the obfuscated locations. Then, the obfuscated trajectory sequence of each user can be generated by combining the locations according to time series and distance. Finally, LBSNs server analyzes the obfuscated trajectory data of users to achieve DP-LTOD.

The contributions of our work can be divided into three aspects as following.

1) We formulate the trajectory obfuscation technique to select an optimal trajectory which has the smallest difference with the original trajectory. We prove this problem is a NP-hard, and propose a heuristic trajectory obfuscation algorithm to solve the problem.

2) We present a trajectory clustering technique to classify the trajectories into different kinds of clusters according to semantical distance and geographical distance. Furthermore, a latent trajectory community discovering method (LTOD) by leveraging our trajectory clustering technique is proposed. It can discover the users who have similar interest or preference and cluster them into a community.

3) We design a differential privacy latent trajectory community discovering (DP-LTOD) scheme, which not only protects user privacy from leaking, but also provides effective services for target users. During the stage of location obfuscation matrix generation, noise based on Laplace distribution is added to the numeric outputs. Also, the noise based on exponential distribution is added to the non-numeric outputs at the stage of trajectory sequence function generation.

4) Through formal privacy analysis, we prove our DP-LTOD scheme satisfies ϵ-differential privacy. We conduct an extensive experimental study over two real datasets. The experimental results demonstrate that our DP-LTOD scheme can privately discover latent trajectory community with high accuracy.

2 Related work

In this section, we briefly review the related works on trajectory community discovering, location privacy preserving, trajectory privacy preserving and differential privacy data publishing.

2.1 Trajectory Community Discovering

It involves the research on trajectory clustering according to uploaded trajectory information, which is an important process to find the users with similar movement pattern. Traditional trajectory clustering methods were to take the whole trajectory as a unit to cluster. Gaffney et al. [16] proposed a trajectory clustering-based mixed regression model, which made use of expectation maximization (EM) algorithm to calculate the membership degrees of objective. After that, Chudova et al. [17] proposed a mixed model aimed at translation invariant of curve clustering, which was used to study trajectory clustering according to the space-time trajectory drifting of objective. However, some information may be missed when clustering trajectories as a whole. For example, the similarity of sub-trajectories can also reflect the similarity of the whole trajectories. Han et al. [18] proposed a new partition-and-group framework to discover common sub-trajectories for clustering trajectories. Mining communities based on trajectory-related information (e.g. spatial distance, time, or velocity, etc.) had been an effective method to find similar users. Lee et al. [19] proposed a unifying framework (UT-patterns) to mine trajectory patterns according to the strength of temporal constraints. Zhu et al. [20] made use of sequential probability tree (SP-tree) to discover movement-based communities of users. In [21], Zhao et al. proposed a probabilistic generative model, which made use of different lifestyle-related patterns mined from trajectory data to infer the social strength of users. Although the aforementioned studies offer important insights into this topic, none of them worked on mining latent trajectory communities according to the interest or preference of each user. Our work aims to fill in this gap by taking semantic information and geographical information into account to cluster the users who have similar interest or preference into a community.

2.2 Location Privacy Preserving

In LBSNs, users publish their location-related contents to server in order to acquire personalized service. If the server is vulnerable, attackers may steal user actual location information to do illegal things. Location privacy preserving is important for users to protect their privacy from leaking. Early researches mainly included three aspects: (1) **False location** [22]. The basic idea was to transform the actual location into one or more false locations and publish them to server. (2) **Spatial cloaking** [23]. The techniques in this category generalized the actual location into a cloaked spatial regions which were guaranteed to satisfy the *k*-anonymity. (3) **Encryption** [24]. The main idea was to provide effective location privacy preserving without the assistance of Trusted Third Party (TTP). Due to the high cost of false location method in resource-constrained mobile devices, Liu et al. [25] modeled the process of dummy generation as Bayesian games and proposed a strategy selection algorithm to help users achieve optimized payoffs. However, more redundant results were still be returned from the LBSNs server because of more noises in the query, which made a higher communication cost in mobile client. Spatial cloaking techniques can well reduce the load of mobile client depending on a TTP. Despite this, location-dependent attacks was still happened. Pan et al. [26] proposed an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2018.2855740, IEEE Transactions on Services Computing

incremental clique-based cloaking algorithm by taking the location *k*-anonymity and cloaking granularity as privacy metrics to defend against location-dependent attacks. In this paper, we also utilize an encryption technique without regard to the limitation of TTP. The novelty of our research compared with previous studies is that server generates differential privacy-guaranteed location obfuscation matrix, and mobile terminal utilizes downloaded location obfuscation matrix to transform the actual location into other locations, which have the similar feature with original location.

2.3 Trajectory Privacy Preserving

Trajectory privacy preserving is to protect user trajectory information from leaking. Once one trajectory information is stolen by attackers, the sensitive location information is exposed. Even the attackers may predict the next location of victims according to the revealed trajectory information, which seriously threatens the personal safety of users. Existing studies on trajectory privacy preserving are classified into three categories: (1) **Dummy trajectories** [27]. The techniques in this category generated the dummy trajectories according to original trajectories and published them to server. (2) **Trajectory *k*-anonymity** [28]. Similar with location *k*-anonymity, the actual trajectory information could be concealed by TTP. (3) **Inhibition technique** [29]. The main idea was to forbid the data release of sensitive location information to protect the individual trajectory privacy of users. Aimed to balance the data utility and *k*-anonymity trajectory privacy, Han et al. [30] proposed a semantic space translation algorithm (SST) which provided different levels of privacy protection for different locations. Hwang et al. [31] introduced a novel time-obfuscated technique which breaks the sequence of the query issuing time to protect users' trajectory privacy. In this paper, we utilize trajectory sequence function to generate obfuscated trajectory sequence corresponding to original trajectory sequence. Compared with previous studies, the novelty of our research is that we partition the trajectory into different segments, and select the optimal segment at each time slot according to differential privacy-guaranteed segment selection technique.

Besides the above location privacy-preserving and trajectory privacy-preserving, most existing literatures address trajectory obfuscation based on differential privacy in mobile data publishing have been studied recently. For quantifying the level of privacy-preserving, V. Rastogi et al. [32] firstly proposed differentially private aggregation algorithms (i.e. PASTE) that offer good practical utility without any trusted server. By data analysis on real-life GPS trajectory dataset, A. Monreale et al. [33] proposed an anonymity technique to achieve the conflicting goals of data utility and data privacy. Also, A. Monreale et al. [34] studied a differential privacy-based privacy-preserving model to provide the formal data protection safeguard. In order to achieve differential privacy data release, N. Mohammed et al. [35] proposed a two-party algorithm for vertically partitioned data as two parties according to the definition of secure multiparty computation. After that, T. Zhu et al. [36] proposed a correlated differential privacy solution to decrease most of noise incurred via differential privacy in correlated dataset. In the aspect of trusted recommendation, J. D. Zhang et al. [37] proposed a privacy-preserving location recommendation framework (i.e. PLORE), which utilized differential privacy mechanism to reach a good trade-off between data utility and privacy-preserving. In addition, H. Shin et al. [38] developed a matrix factorization algorithm based on local differential privacy to achieve perturbed data publishing. In this paper, we make use of both differential privacy model and trajectory partitioning algorithm to discover latent trajectory communities under strict privacy-preserving.

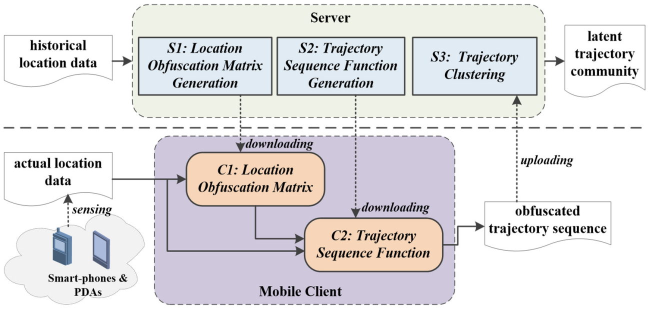

Server
historical location data
S1: Location Obfuscation Matrix Generation
S2: Trajectory Sequence Function Generation
S3: Trajectory Clustering
latent trajectory community
downloading
downloading
uploading
actual location data
sensing
C1: Location Obfuscation Matrix
C2: Trajectory Sequence Function
obfuscated trajectory sequence
Smart-phones & PDAs
Mobile Client

Fig. 2. The work flow of DP-LTOD.

3 Overview of DP-LTOD Scheme

In LTOD, users belong to the same community can share their location-related information with each other. However, the privacy may be revealed when users upload actual trajectory information to LBSNs server.

In Fig. 2, we illustrate the architecture and work flow of DP-LTOD. In a nutshell, it can be divided into servers and mobile clients. On the server side, historical location data will be analyzed firstly to construct the location obfuscation matrix and trajectory sequence function. On the mobile client side, users need to download the location obfuscation matrix and trajectory sequence function from server in advance. When users want to publish their actual trajectory data, procedures *C1* and *C2* are necessary to generate obfuscated trajectory sequence. Then, the mobile client uploads the generated trajectory sequence to server, in order to experience the personalized services from LBSNs. Finally, LBSNs server discovers the latent trajectory community through trajectory clustering (i.e. procedure *S3*).

Based on this design, DP-LTOD is able to classify different users with different interest or preference into different communities. Besides, by making use of differential privacy theory, DP-LTOD can effectively protect user privacy while ensuring the data utility. The detailed workflow of DP-LTOD is summarized as follows.

Step (*S1*): The historical location data which describes the different distribution of geographical locations is taken as training set. In this paper, every location in the training set owns its semantic information, which has been completed by our previous studies [1], [4]. As we know, in order to ensure the data utility, one location should be obfuscated to another location which have the similar semantic information with a high probability. However, the original location may be inferred by making use of the difference of probability. Therefore, we add noise based on Laplace distribution to the numeric outputs to tradeoff the privacy and data utility. According to this rule, the differential privacy-guaranteed location obfuscation matrix can be generated.

Step (*S2*): After that, the work flow moves to select an optimal trajectory which has the shortest geographical distance with original trajectory. Every trajectory can be divided into a number of segments because it is generated by linking the visited locations of users in accordance with time. Therefore, the geographical distance between two trajectories can be converted to the sum of the distance between each two segments. The shorter the distance between candidate segment and original segment is, the higher the probability of being chosen is. However, the next visited location may be inferred by making use of the difference of probability. In order to tradeoff the privacy and data utility, we add noise based on exponential distribution to the non-numeric outputs. According to this rule, the differential privacy-guaranteed trajectory sequence function can be generated.

Step (*C1*): When users want to publish their physical trajectory data to LBSNs server, they should firstly download the location obfuscation matrix. Then the actual location can be obfuscated to a fake location according to the training algorithm.

Step (*C2*): In order to generate the obfuscated trajectory sequence, the trajectory sequence function should be also downloaded from LBSNs server. According to the optimization algorithm, one obfuscated trajectory can be selected corresponding to the original trajectory.

Step (*S3*): LBSNs server receives the obfuscated trajectory sequence of each user. According to semantical distance and geographical distance, the trajectories can be clustered into different communities. Finally, the process of latent trajectory community discovering is completed.

DP-LTOD builds a complete framework with many unique characteristics. It is able to protect users' privacy from leaking and support high data utility. Although the published trajectories are not true trajectories of users, the trajectory communities can be discovered in a certain tolerance range. Besides, DP-LTOD takes the semantic information into account to cluster trajectories, which can well discover the users who have similar interest or preference.

To further explain these main features in DP-LTOD, we detailedly describe the scheme designs and algorithms in the next two sections.

4 DP-LTOD Scheme Designs

In this section, we give a description of latent trajectory community discovering, and formulate the optimization problem of trajectory obfuscation, in order to select the optimal obfuscated trajectory sequence for original trajectory sequence. For ease of the following presentation, Table 1 lists the relevant notations and definitions.

4.1 Latent Trajectory Community Discovering

Different with traditional trajectory community discovering methods, LTOD clusters the users whose trajectories have similar semantic information and geographical information into a community. We define $\langle lon_i, lat_i \rangle$ as the two-tuples of longitude and latitude for location loc_i. Based on our previous works [1], [4], the semantic information C_i corresponding to loc_i can be acquired.

Location Obfuscation Matrix. Let $\Pr[loc_i, loc_j]$ denote the transition probability from location loc_i to location loc_j. If loc_i and loc_j have the same type information, namely $C_i = C_j$, they will be obfuscated each other with a higher probability. However, one location loc_i can not be obfuscated to itself. Thus, location obfuscation matrix $\mathbb{M}$ can be trained according to the relationship between location and type in historical location datasets.

Movement Pattern. Let $loc_1 \to loc_2 \to \cdots \to loc_n$ denote the location sequence of user *k* linked by chronological order in one day. The corresponding type sequence is represented as $C_1 \to C_2 \to \cdots \to C_n$. Movement pattern can well reflect the interest or preference of users in a certain period. It means a series of activities that users usually do. Thus, the movement pattern of one user is acquired by extracting the frequent subsequences from his type sequences.

In this paper, we take semantical distance and geographical distance into account to cluster the users with similar interest or preference. The semantical distance between two users is short when they have similar movement pattern. Geographical distance means the distance between movement trajectories of two users, and the detailed process of geographical distance computing will be explained in the section 4.2.

TABLE 1
Notations and Definitions

Notations	Descriptions
loc_i, L_{ij}, T_i	Actual location, segment, trajectory
loc_i^*, L_{ij}^*, T_i^*	Obfuscated location, segment, trajectory
C_i	Type information of location loc_i
$\mathcal{R}$, $\mathcal{L}$	Set of actual locations, segments
$\mathcal{R}'$, $\mathcal{L}'$	Set of obfuscated locations, segments
$\mathbb{D}$	Query dimension of function
$\mathbb{R}$	Real space of mapping
$\mathcal{O}$	Output domain of function
Δf, Δu	Global sensitivity of function
u	Score function
$\Pr[loc_i, loc_j]$	Probability of loc_i obfuscated to loc_j
$\pi(loc_i)$	Priori probability distribution of loc_i
$\sigma(loc_i)$	Posterior probability distribution of loc_i
M^{u_i}	Mobile status matrix of one user
$Dist\left(L_{ij}, L_{ij}^*\right)$	Distance between L_{ij} and L_{ij}^*
$SD(T_i, T_j)$	Semantic distance between T_i and T_j
$GD(T_i, T_j)$	Geographical distance between T_i and T_j
$LD(T_i, T_j)$	Latent distance between T_i and T_j

4.2 Trajectory Obfuscation

Trajectory obfuscation is to find one trajectory which is similar with the original trajectory. In order to well describe the geographical distance between two trajectories, the first step is to divide the trajectories into several segments. Han et al. [18] have made a quantitatively analysis on the information loss of different partition methods. In this paper, we suppose the trajectories can be divided according to access time of each location. So the original trajectory consists of k segments L_{ij} is defined as T, and the obfuscated trajectory corresponding to T is defined as T^* which consists of k obfuscated segments L_{ij}^*. As shown in Fig. 3, loc_i denotes the true location of one user, and loc_i^* denotes the obfuscated location corresponding to loc_i. Let $\overrightarrow{L_{ij}}$ denote a vector constructed by points loc_i and loc_j. Likewise, $\overrightarrow{L_{ij}^*}$ denotes the vector constructed by points loc_i^* and loc_j^*.

Perpendicular Distance between true segment L_i and obfuscated segment L_i^* is defined as:

$$D_\perp\left(\overrightarrow{L_{ij}}, \overrightarrow{L_{ij}^*}\right) = \frac{d_{\perp 1}^2 + d_{\perp 2}^2}{d_{\perp 1} + d_{\perp 2}}, \tag{1}$$

where $d_{\perp 1}$ and $d_{\perp 2}$ are the distance from points loc_i and loc_j to segment L_{ij}^*, respectively.

Parallel Distance between true segment L_{ij} and obfuscated segment L_{ij}^* is defined as:

$$D_\parallel\left(\overrightarrow{L_{ij}}, \overrightarrow{L_{ij}^*}\right) = \min\left(d_{\parallel 1}, d_{\parallel 2}\right), \tag{2}$$

where $d_{\parallel 1}$ is the minimum of the Euclidean distance from point loc_i' to point loc_i^* and point loc_j^*, $d_{\parallel 2}$ is the minimum of the Euclidean distance from point loc_j' to point loc_i^* and point loc_j^*.

Angle Distance between true segment L_{ij} and obfuscated segment L_{ij}^* is defined as:

$$D_\theta\left(\overrightarrow{L_{ij}}, \overrightarrow{L_{ij}^*}\right) = \begin{cases} \left\|\overrightarrow{L_{ij}}\right\| \times \sin(\theta_i), & if\ \theta \in \left[0, \dfrac{\pi}{2}\right] \\ \left\|\overrightarrow{L_{ij}}\right\|, & if\ \theta \in \left[\dfrac{\pi}{2}, \pi\right], \end{cases} \tag{3}$$

where $\left\|\overrightarrow{L_{ij}}\right\|$ is the length of segment L_{ij}, θ_i is the angle between true segment L_{ij} and obfuscated segment L_{ij}^*. In this paper, we only consider the condition $0 < \theta_i \leq \frac{\pi}{2}$ to generate the obfuscated segments.

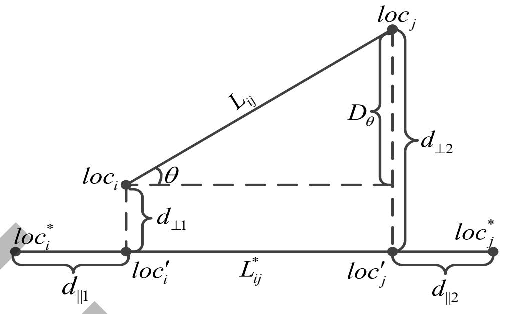


Fig. 3. Geographical distance function between two segments.

Thus, the distance between true segment L_{ij} and obfuscated segment L_{ij}^* is defined as:

$$\begin{aligned} Dist\left(\overrightarrow{L_{ij}}, \overrightarrow{L_{ij}^*}\right) = & \omega_\perp D_\perp\left(\overrightarrow{L_{ij}}, \overrightarrow{L_{ij}^*}\right) + \omega_\| D_\|\left(\overrightarrow{L_{ij}}, \overrightarrow{L_{ij}^*}\right) \\ & + \omega_\theta D_\theta\left(\overrightarrow{L_{ij}}, \overrightarrow{L_{ij}^*}\right), \end{aligned} \tag{4}$$

where $\omega_\perp$, $\omega_\|$ and ω_θ are the weight of *perpendicular distance*, *parallel distance* and *angle distance*, respectively. In practice, the value of $\omega_\perp$, $\omega_\|$ and ω_θ should be determined according to different applications.

In order to mine the obfuscated trajectory sequence which is similar to the original trajectory sequence in geographical space and semantical space. The weights of semantical similarity and geographical similarity can be computed as:

$$c_i = -\Pr\left[loc_i, loc_i^*\right],\ c_{ij} = Dist\left(\overrightarrow{L_{ij}}, \overrightarrow{L_{ij}^*}\right), \tag{5}$$

where $\Pr\left[loc_i, loc_i^*\right]$ denotes the probability of location loc_i obfuscated to location loc_i^*. It can be acquired by the type information of each location (i.e. $\mathbb{P}\left[type\left(loc_i\right), type\left(loc_i^*\right)\right]$).

We define the sets of location loc_i and segment L_{ij} are $\mathcal{R}$ and $\mathcal{L}$, respectively. The corresponding sets of obfuscated location loc_i^* and segment L_{ij}^* are $\mathcal{R}^*$ and $\mathcal{L}^*$, respectively. And there are two decision variables to be used for the following problem formulation.

$$X_i = \begin{cases} 1, & if\, loc_i^* \in \mathcal{R}^* \\ 0, & otherwise \end{cases} \tag{6}$$

$$Y_{ij} = \begin{cases} 1, & if\, arc\left(loc_i^*, loc_j^*\right) \in \mathcal{L}^* \\ 0, & otherwise \end{cases} \tag{7}$$

Formally, the problem of trajectory obfuscation can be formulated as:

$$\begin{aligned} \textbf{minimize} \quad & \sum_{i\in\mathcal{R}} c_i X_i + \sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{V}} c_{ij} Y_{ij} \\ \textbf{subject to} \quad & \sum_{i\in\mathcal{R}} Y_{ij} = X_j,\ j \in \mathcal{V}, \\ & X_i, Y_{ij} \in \{0,1\},\ i \in \mathcal{R}, j \in \mathcal{V}. \end{aligned} \tag{8}$$

The objective function minimizes the total difference between obfuscated trajectory sequence and original trajectory sequence while the first condition is the constraint for sequence continuity. The second condition is to restrict the value of X_i and Y_{ij}. And $\mathcal{V}$ means one set which element i is not included in the set $\mathcal{R}$, namely, $\mathcal{V} = \mathcal{R} - \{i\}$.

The above problem can be essentially treated as a node-weighted steiner tree problem [39], which is proved to be NP-hard. So our basic problem is also NP-hard, which can be solved by a heuristic algorithm. In Section 5, we will explain the heuristic trajectory obfuscation algorithm in detail.

5 DP-LTOD Algorithms

In this section, we first give the definition of trajectory sequence and databases. Then, we construct the privacy model and adversary model. Corresponding to adversary models, we explain two differential privacy-based algorithms in DP-LTOD including location obfuscation matrix generation and trajectory sequence function generation. Finally, we state the trajectory clustering algorithm in DP-LTOD to discover latent trajectory communities.

5.1 Trajectory Sequences and Databases

In this paper, a trajectory sequence is defined as follows.

Definition 1. (Trajectory Sequence). *A trajectory sequence is consisted of the locations according to time series and distance:* $T = loc_1 \xrightarrow{L_{12}} loc_2 \xrightarrow{L_{23}} \cdots \xrightarrow{L_{(|T|-1)|T|}} loc_{|T|}$, *where* $|T|$ *is the length of this trajectory and* $\forall i\left(1 \le i \le |T|\right)$, $loc_i \in \mathcal{R}_T$ *is a discrete location, which is denoted by the latitude and longitude.* $L_{(i-1)i} \in \mathcal{L}_T$ *is a segment linked by two locations.*

Each trajectory sequence represents the mobile behavior of user. Let $\mathcal{R}_T$ denote the set of locations in T, namely, $\mathcal{R}_T = \left\{loc_1, loc_2, \ldots, loc_{|T|}\right\}$, and $\mathcal{L}_T$ represents the set of segments in T, namely, $\mathcal{L}_T = \left\{L_1, L_2, \ldots, L_{|T|-1}\right\}$. The corresponding sets of locations and segments in obfuscated trajectory sequence are denoted as $\mathcal{R}_\mathcal{T}^* = \left\{loc_1^*, loc_2^*, \ldots, loc_{|T|}^*\right\}$ and $\mathcal{L}_\mathcal{T}^* = \left\{L_1^*, L_2^*, \ldots, L_{|T|-1}^*\right\}$, respectively.

For an original trajectory sequence, there are many candidate locations and segments to be selected, in order to generate the obfuscated trajectory sequence. Let $\mathcal{D}_{loc_i^*}$ denote the database of candidate locations, namely, $\mathcal{D}_{loc_i^*} = \{S : S\left(loc\right) \in \mathcal{R}^*\}$. $\mathcal{D}_{L_{ij}^*}$ is the database of candidate segments, namely, $\mathcal{D}_{L_{ij}^*} = \{S : S\left(L\right) \in \mathcal{L}^*\}$. The adjacent databases corresponding to $\mathcal{D}_{loc_i^*}$ and $\mathcal{D}_{L_{ij}^*}$ are $\mathcal{D}'_{loc_i^*}$ and $\mathcal{D}'_{L_{ij}^*}$, respectively.

5.2 Privacy Model

Differential privacy [40] is a theoretical framework of privacy-preserving, which provides strict and theoretical guarantees for the probability distribution of outputs. It not only protects the privacy of published databases against attackers with any priori knowledge, but also requires the outputs of algorithms to be approximately same even if any individuals record in the database is arbitrarily changed. Taking the database of candidate locations for example, the definition of differential privacy is described as follows.

Definition 2. (ϵ-Differential Privacy). *Any two databases* $\mathcal{D}_{loc_i^*}$ *and* $\mathcal{D}'_{loc_i^*}$ *whose difference between them is at most one record (i.e.,* $\left|\mathcal{D}_{loc_i^*} \Delta \mathcal{D}'_{loc_i^*}\right| \le 1$*). A randomized algorithm* $\mathcal{F}$ *satisfies* ϵ*-differential privacy iff for any subsets of output* $Y \subseteq Range\left(\mathcal{F}\right)$*, it satisfies:*

$$\Pr\left[\mathcal{F}\left(\mathcal{D}_{loc_i^*}\right) \in Y\right] \le \exp\left(\epsilon\right) \Pr\left[\mathcal{F}\left(\mathcal{D}'_{loc_i^*}\right) \in Y\right], \tag{9}$$

where $Range\left(\mathcal{F}\right)$ *denotes the value domain of algorithm* $\mathcal{F}$. $\Pr\left[\mathcal{F}\left(\mathcal{D}_{loc_i^*}\right) \in Y\right] / \Pr\left[\mathcal{F}\left(\mathcal{D}'_{loc_i^*}\right) \in Y\right]$ *represents the risk of privacy leakage, which is controlled by the randomness of algorithm* $\mathcal{F}$. ϵ *represents the degree of privacy-preserving, and it is proved that the smaller the value of* ϵ *is, the higher the degree of privacy-preserving is. Usually, its value is as small as 1 or even smaller.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2018.2855740, IEEE Transactions on Services Computing

Definition 3. (Global Sensitivity). *For a given function $f : \mathbb{D} \to \mathbb{R}^d$, the global sensitivity of function f is defined as*

$$\Delta f = \max_{\mathcal{D}_{loc_i^*},\mathcal{D}'_{loc_i^*}} \left\| f\left(\mathcal{D}_{loc_i^*}\right) - f\left(\mathcal{D}'_{loc_i^*}\right)\right\|_1, \qquad (10)$$

for any two databases $\mathcal{D}_{loc_i^*}$ and $\mathcal{D}'_{loc_i^*}$ which differ in at most one record. $\mathbb{D}$ and $\mathbb{R}$ represent the query dimension of function f and real space of mapping, respectively.

According to the above definitions, it can be seen that algorithm $\mathcal{F}$ satisfies ϵ-differential privacy from a theoretical point of view. A standard approach for guaranteeing differential privacy is to add random noise to the real output of algorithm $\mathcal{F}$ based on the databases. Laplace mechanism and exponential mechanism are two common methods for adding noise. And the noise level is related to the global sensitivity, i.e., the maximal change in the outputs of a function when any individual record is changed.

Composition Properties. Generally speaking, the differential privacy algorithm may be utilized repeatedly to address one complex privacy-preserving problem. In order to control the whole privacy-preserving level in the given ϵ range, it is necessary to reasonably allocate ϵ to each steps of the differential privacy algorithm. Sequential composition and parallel composition are two critical composition properties in differential privacy.

Theorem 1. (Sequential Composition). *Suppose k algorithms $\mathcal{F}_1, \mathcal{F}_2 ..., \mathcal{F}_k$ whose privacy-preserving level is $\epsilon_1, \epsilon_2 ..., \epsilon_k$, respectively. For the same database $\mathcal{D}_{loc_i^*}$, it can provide $\sum_{i=1}^{k} \epsilon_i$-differential privacy according to the combined k algorithms $\mathcal{F}\left(\mathcal{F}_1\left(\mathcal{D}_{loc_i^*}\right), \mathcal{F}_2\left(\mathcal{D}_{loc_i^*}\right), ..., \mathcal{F}_k\left(\mathcal{D}_{loc_i^*}\right)\right)$.*

5.3 Adversary Model

In order to evaluate the privacy risk of DP-LTOD, we construct two adversary models by considering location attacks and trajectory attacks, respectively. Suppose attackers have any priori knowledge except actual locations of victims.

1) **Location Attacks**. The adversary is assumed to know some *priori* knowledge about the probability distribution of a victim's actual location loc_i, represented as $\pi\left(loc_i\right)$. In addition, the location obfuscation matrix can be acquired by downloading from the server directly.

According to Bayes' rule [41], the *posterior* probability distribution of the victim's actual location can be predicted as:

$$\sigma\left(loc_i\right) = \frac{\Pr\left[loc_i, loc_i^*\right] \cdot \pi\left(loc_i\right)}{\sum_{loc_j \in \mathcal{R}} \Pr\left[loc_j, loc_i^*\right] \cdot \pi\left(loc_j\right)}, \qquad (11)$$

where loc_j denotes any one location in set $\mathcal{R}$ except loc_i and loc_i^*, $\sigma\left(loc_i\right)$ denotes the posterior probability of loc_j.

Under this adversary model, location privacy-preserving in the process of location obfuscation matrix generation is to limit the improvement of attackers' posterior knowledge over the prior knowledge, i.e., $\sigma\left(loc_i\right)/\pi\left(loc_i\right)$. Intuitively, if two locations loc_i and loc_j have similar probabilities of being obfuscated to location loc_i^*, then an attacker will be unable to distinguish whether the actual location is loc_i or loc_j.

2) **Trajectory Attacks**. The adversary makes the assumption that trajectory sequence of each user can be modeled as a *Markov Chain* (MC) in region $\mathcal{R}$. Then the mobile status M^{u_k} of user k is a transition matrix for the user's MC. The entry $M_{ij}^{u_k}, i, j = 1, ..., n$ of M^{u_k} denotes the probability that user k will move from location loc_i to location loc_j in the next time slot.

Algorithm 1 Differential Privacy Location Obfuscation Matrix

```
Input: A set of locations $\mathcal{R}$, type transition probability matrix $\mathbb{P}$
    // The number of locations is $N$
Output: $N \times N$ differential privacy location obfuscation matrix $\mathbb{M}$
Define $type\left(loc_i\right)$ is the type of each location $loc_i$, $H\left(i,j\right)$ is the probability from $loc_i$ to $loc_j$
Initialize $\mathbb{M} \to \begin{bmatrix} H\left(1,1\right) & \cdots & H\left(1,N\right) \\ \vdots & \ddots & \vdots \\ H\left(N,1\right) & \cdots & H\left(N,N\right) \end{bmatrix}$
For $i \to 1, 2, \ldots, N$ do
    For $j \to 1, 2, \ldots, N$ do
        If $i == j$ then
            $H\left(i,j\right) \leftarrow 0$; // The same two locations can't be obfuscated each other
        Else
            $H\left(i,j\right) \leftarrow \mathbb{P}\left[type\left(loc_i\right), type\left(loc_j\right)\right] + Lap\left(0, \frac{1}{\epsilon_l}\right)$; // Add Laplace noise into the output
        End if
    End for
End for
Normalize the value of $\mathbb{M}$;
Return $\mathbb{M}$
```

According to Markov's rule [42], the objective of attackers is to construct the transition matrix M^{u_k} based on any priori movement information of user k.

Under this adversary model, trajectory privacy-preserving in the process of trajectory sequence function generation is to prevent the attackers from inferring actual segment L_{ij} by making use of the obfuscated segment L_{ij}^* of victims.

5.4 Location Obfuscation Matrix Generation

The objective of location obfuscation matrix is to encode the transition probabilities of obfuscating any one location to another location. In this paper, it protects user location privacy by selecting suitable transition probabilities, which can make it impossible to accurately infer the true location from its obfuscated location, even if adversaries have stolen the location obfuscation matrix. Due to the outputs of location obfuscation matrix are numeric, noise based on Laplace distribution can be added to the outputs, in order to make the generated location obfuscation matrix satisfies ϵ-differential privacy.

Laplace Mechanism. This mechanism can achieve ϵ-differential privacy by adding random noise which follows Laplace distribution into the true outputs.

Theorem 2. [43] *Any one function $f : \mathbb{D} \to \mathbb{R}^d$, whose global sensitivity is Δf. The algorithm $\mathcal{F}$ satisfies ϵ-differential privacy when*

$$\mathcal{F}\left(\mathcal{D}_{loc_i^*}\right) = f\left(\mathcal{D}_{loc_i^*}\right) + < Lap_1\left(0,\lambda\right), ..., Lap_d\left(0,\lambda\right) > . \qquad (12)$$

Meanwhile, the probability dense function of Laplace distribution is denoted as $p\left(x\right) = \frac{1}{2\lambda}\exp\left(-\frac{|x|}{\lambda}\right)$. And λ can be computed by the global sensitivity Δf and the differential privacy parameter ϵ, namely $\lambda = \frac{\Delta f}{\epsilon}$.

For defending against location attacks, **Algorithm 1** shows the pseudo-code of location obfuscation matrix generation. The main idea of this algorithm is to add noise based on Laplace distribution to the support of all possible candidate locations and select optimal locations to be obfuscated according to their noisy supports. In particular, we select the candidate locations according to the order of the original trajectory sequence. Suppose the original trajectory sequence is consisted of k actual locations. In order to generate the location obfuscation matrix, the probability any one location loc_i obfuscated to another location loc_i^* is computed. In this paper, latent trajectory communities could be mined by considering the type information of locations. Therefore, we think the calculation of $\mathbb{P}\left[type\left(loc_i\right), type\left(loc_j\right)\right]$ accesses to sensitive actual information. It is the main privacy requirement but not the only one. For convenience, we define the probability $H\left(i,j\right)$ is 1 when loc_i and loc_i^* belong to the same type. If loc_i and loc_i^* belong to different type, the probability $H\left(i,j\right)$ is 0. Because the optimal location selection is an inquiry function, the global sensitivity of this algorithm is computed as 1 (i.e. $\Delta f = 1$). First, the location obfuscation matrix $\mathbb{M}$ are initialized by adding the value of each element $H\left(i,j\right)$. Then, we compute the noisy support of each element in $\mathbb{M}$ according to the global sensitivity Δf. Finally, the location obfuscation matrix $\mathbb{M}$ can be generated when noisy support of each element exceed the threshold ϵ_l.

Lemma 1. *The optimal location selection technique satisfies ϵ_l-differential privacy.*

Proof. Let Q_i denote the support computations of obfuscated location loc_i^* in initial matrix $\mathbb{M}$. Corresponding to each location in original trajectory sequence, the obfuscated location can be selected from the candidate location database $\mathcal{D}_{loc_i^*}$. And $\mathcal{D}'_{loc_i^*}$ is the adjacent database for $\mathcal{D}_{loc_i^*}$, which means that they are different in at most one record. The amount of noise added to the support of each element is determined by the allocated privacy budget ϵ_l and global sensitivity Δf. Moreover, due to adding (or removing) a candidate location can, in the worst case, affect the support of obfuscated location selection by 1. The global sensitivity of Q_i is 1, namely $\Delta f = 1$. Thus, the Laplace noise $Lap\left(0, \frac{1}{\epsilon_l}\right)$ should be added to Q_i to satisfy ϵ_l-differential privacy. The detailed derivation process can be seen in **Appendix A**.

5.5 Trajectory Sequence Function Generation

The objective of trajectory sequence function is to link the selected locations to generate the obfuscated trajectory sequence. In this paper, it not only guarantees user trajectory privacy, but also achieves good data utility by making use of heuristic trajectory obfuscation algorithm. Due to the outputs of trajectory sequence function are non-numeric, noise based on exponential distribution can be added to the outputs, in order to make the generated trajectory sequence function satisfies ϵ-differential privacy.

Exponential Mechanism. Laplace mechanism can be adapted to the numeric search results. However, there are many search results may be non-numeric in practice, which cause the outputs can't be affected by adding noises. In that case, exponential mechanism had been proposed by McSherry et al. [44] to make the algorithms satisfy ϵ-differential privacy.

Theorem 3. [44] *A given score function $u : \mathbb{D} \times \mathcal{O} \rightarrow \mathbb{R}$, whose global sensitivity is Δf. The algorithm $\mathcal{F}$ satisfies ϵ-differential privacy when*

$$\mathcal{F}\left(D,u\right) = \left\{\Pr\left[r \in \mathcal{O}\right] \propto \exp\left(\frac{\epsilon u\left(D,r\right)}{2\Delta u}\right)\right\}. \quad (13)$$

Thereinto, r represents the selected items from output domain $\mathcal{O}$. And the higher the score is, the bigger the probability it may be selected is.

Algorithm 2 Differential Privacy Trajectory Sequence Function

```
Input: Original trajectory sequence $T$, differential privacy location obfuscation matrix $\mathbb{M}$
Output: Obfuscated trajectory sequence $T^*$
Define n is the length of trajectory $T$, $loc_i$ is the location in $T$, $L_{ij}$ is the segment in $T$, $loc_i^*$ is the location in $T^*$, $L_{ij}^*$ is the segment in $T^*$, $Dist\left(L_{ij}, L_{ij}^*\right)$ is the distance between $L_{ij}$ and $L_{ij}^*$
Initialize $T^* \rightarrow \left\{loc_1^*, loc_2^*, \ldots, loc_n^*\right\}$
For all $loc_i$ in $T$ do
    Select the candidate location $loc_i^*$ from $\mathbb{M}$;
End for
For $i \rightarrow 1, 2, \ldots, n$ do
    For $j \rightarrow 1, 2, \ldots, n$ do
        Compute $Dist\left(L_{ij}, L_{ij}^*\right)$ ;
        $\Pr\left[T^* \leftarrow T^* \bigcup \left\{L_{ij}^*\right\}\right] \propto \exp\left(\frac{\epsilon_s}{2\Delta f} u\left(\mathcal{D}_{L_{ij}^*}, L_{ij}^*\right)\right)$; // Add exponential noise into the output
        Select candidate segment $L_{ij}^*$ according to the computed probability distribution;
    End for
End for
Return $T^*$
```

For defending against trajectory attacks, we utilize exponential mechanism to probabilistically select the segments in database $\mathcal{L}^*$ at every time slot. As we mentioned earlier, there are $k-1$ segments which should be selected to constitute the obfuscated trajectory sequence corresponding to original trajectory sequence. We define a score function $u : \mathcal{L}^* \times \mathcal{D}_{L_{ij}^*} \rightarrow \mathbb{R}$ that assigns a score value for each candidate segment $L_{ij}^* \in \mathcal{D}_{L_{ij}^*}$. According to Eq. (5), the score function can be computed as:

$$u\left(\mathcal{D}_{L_{ij}^*}, L_{ij}^*\right) = \frac{c_{ij}}{\sum_{L_{ij}^* \in \mathcal{D}_{L_{ij}^*}} c_{ij}}. \quad (14)$$

As the change of one record in $\mathcal{D}_{L_{ij}^*}$, the global sensitivity of this function can be computed as:

$$\Delta f = \frac{\max\limits_{L_{ij}^* \in \mathcal{D}_{L_{ij}^*}} \left(c_{ij}\right) - \min\limits_{L_{ij}^* \in \mathcal{D}_{L_{ij}^*}} \left(c_{ij}\right)}{\sum_{L_{ij}^* \in \mathcal{D}_{L_{ij}^*}} c_{ij}} < 1. \quad (15)$$

The score function makes the proposed algorithm prefer to select the obfuscated segment which is closest to the original segment in geographical space. By considering the utility of all candidate segments, the noise based on exponential distribution is added to the output to select one segment $S\left(L\right) \in \mathcal{D}_{L_{ij}^*}$ following the probability as:

$$\Pr[\mathcal{F}(\mathcal{D}_{L_{ij}^*}, u)] = \frac{\exp\left(\frac{\epsilon_s}{2\Delta f} u\left(\mathcal{D}_{L_{ij}^*}, L_{ij}^*\right)\right)}{\sum_{S(L) \in \mathcal{D}_{L_{ij}^*}} \exp\left(\frac{\epsilon_s}{2\Delta f} u\left(\mathcal{D}_{L_{ij}^*}, L_{ij}^*\right)\right)}. \quad (16)$$

Lemma 2. *The optimal segment selection technique satisfies ϵ_s-differential privacy.*

Proof. Let Q_i denote the support computations of selected segment L_{ij}^* in candidate set $\mathcal{D}_{L_{ij}^*}$. Corresponding to each segment in original trajectory sequence, the obfuscated segment can be selected from the candidate segment database $\mathcal{D}_{L_{ij}^*}$. And $\mathcal{D}'_{L_{ij}^*}$ is the adjacent database for $\mathcal{D}_{L_{ij}^*}$, which means that they are different in at most one record. The amount of noise added to the support of each element is determined by the allocated privacy budget ϵ_s and global sensitivity Δf. Thus, the exponential noise should be added to Q_i to satisfy ϵ_s-differential privacy. The detailed derivation process can be seen in **Appendix B**.

Algorithm 2 shows the pseudo-code of privacy trajectory sequence function generation. The main idea of this algorithm is to add noise based on exponential distribution to the support of all possible candidate segments and select which segments to be obfuscated according to their noisy supports. In particular, we select the candidate

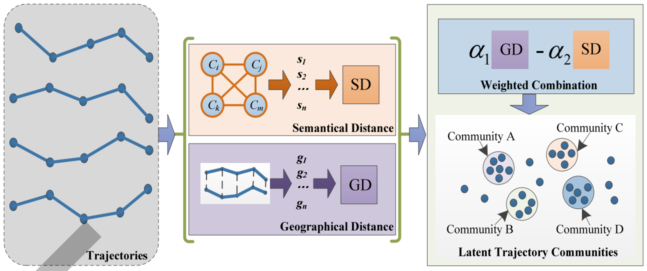


Fig. 4. The process of trajectory clustering for LTOD.

segments according to the order of original trajectory sequence. In order to generate the trajectory sequence function, the first step is to compute the score while any one segment L_{ij} obfuscated to another segment L^*_{ij}. In this paper, the score is computed by considering the geographical distance between segment L_{ij} and segment L^*_{ij}. Then, we compute the noisy support of each segment in set $\mathcal{D}_{L^*_{ij}}$ according to the global sensitivity Δf. Finally, the trajectory sequence function can be generated when noisy support of each segment exceed the threshold.

Lemma 3. *The trajectory sequence function generation algorithm (i.e., Algorithm 2) satisfies ϵ-differential privacy, where $\epsilon = k\epsilon_l + (k-1)\,\epsilon_s$.*

Proof. Suppose the original trajectory consists of k locations, and then it can be divided into $k-1$ segments. The objective of **Algorithm 2** is to generate the obfuscated trajectory which also consists of k locations and $k-1$ segments. The process of trajectory obfuscation can be considered as computing the sum of $Q = \langle Q_1, ..., Q_k\rangle$ and $S = \langle S_1, ..., S_{k-1}\rangle$. Based on the sequential composition property of differential privacy (i.e. **Theorem 1**), this algorithm satisfies ϵ-differential privacy, where $\epsilon = k\epsilon_l + (k-1)\,\epsilon_s$.

5.6 Trajectory Clustering

In our data model, one trajectory consists of a sequence of segments. We take both semantical distance and geographical distance into account to compute the latent distance between two trajectories.

The semantical distance between T^*_i and T^*_j can be computed as:

$$SD\left(T^*_i, T^*_j\right) = \frac{\left|T^*_i \bigcap T^*_j\right|}{\left|T^*_i\right| + \left|T^*_j\right|}, \tag{17}$$

where $|T^*|$ is the number of segments contained in trajectory T^* and $\left|T^*_i \bigcap T^*_j\right|$ is the number of matching segments in trajectory T^*_i and trajectory T^*_j. The two trajectories are more similar if they contain more common sub-sequence in semantical space [1].

The geographical distance between T^*_i and T^*_j can be computed as:

$$GD\left(T^*_i, T^*_j\right) = \sum\nolimits_{k\in\min\left(\left|T^*_i\right|,\left|T^*_j\right|\right)} Dist\left(T^*_{i_k}, T^*_{j_k}\right), \tag{18}$$

where $Dist\left(T^*_{i_k}, T^*_{j_k}\right)$ is the distance between segment $T_{i_k}{}^*$ and segment $T_{j_k}{}^*$.

Based on semantical distance and geographical distance, the latent distance between T^*_i and T^*_j can be computed as:

$$LD\left(T^*_i, T^*_j\right) = \alpha_1 GD\left(T^*_i, T^*_j\right) - \alpha_2 SD\left(T^*_i, T^*_j\right), \tag{19}$$

where α_1 and α_2 denote the weight of geographical distance and semantical distance, respectively.

Algorithm 3 Latent Trajectory Community Discovering (LTOD)

```
Input: A set of trajectories $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$
Output: A set of communities $\mathcal{O} = \{C_1, C_2, \ldots, C_k\}$
Define $SD$ is semantic distance, $GD$ is geographical distance, $LD$
is latent distance, $len$ is the length of each trajectory, $l$ is the number
of common sub-sequence in semantic distance
Initialize $count = 0$
For $i \to 1, 2, \ldots, n$ do
  For $j \to 1, 2, \ldots, (n-1)$ do
    $m \leftarrow \min\left[len\left(T_i\right), len\left(T_j\right)\right]$;
   /* Computing SD */
   For $t \to 1, 2, \ldots, (m-l)$ do
     Compute the number of common sub-sequence $count$;
   End for
     $SD \leftarrow \frac{count}{len(T_i)+len(T_j)}$;
   /* Computing GD */
   For $s \to 1, 2, \ldots, (m-1)$ do
     Compute the distance between two segments $Dist\left(L_s, L_s^{'}\right)$;
   End for
     $GD \leftarrow \sum_m Dist\left(L_s, L_s^{'}\right)$;
   /* Computing LD */
     $LD \leftarrow (\alpha_1 GD - \alpha_2 SD)$;
   /* Trajectory clustering */
   If $i == 1$ then
     $C_1 \leftarrow T_1$;
   Else if $\min_i LD\left(T_i, C_k\right) \leq \theta_t$ then
     Put $T_i$ into $C_k$;
   Else if $\min_i LD\left(T_i, C_k\right) > \theta_t$ then
     Create $C_{k+1}$, put $T_i$ into $C_{k+1}$;
   End if
  End for
End for
Return $\mathcal{O}$
```

To partition the users with different trajectories into k communities based on the latent distance, we present a trajectory clustering method to discover the users who have similar movement pattern and preference. As shown in Fig. 4, we first divide the trajectories into different segments according to the locations. Then, the corresponding semantical distance and geographical distance among trajectories can be computed by Eq. (17) and Eq. (18), respectively. Finally, the latent trajectory communities are discovered by computing weighted combinations of SD and GD.

Algorithm 3 shows the pseudo-code of LTOD. In the phase of trajectory clustering, we firstly select one trajectory T_1 and put it into one community C_1. For the second trajectory T_2, we compute the latent distance between T_1 and T_2. If the distance is no more than the threshold θ_t, the trajectory T_2 is put into C_1. If the distance is more than the threshold θ_t, we create another community C_2 and put T_2 into C_2. For the n-th trajectory T_n, we compute the latent distance between T_n and each trajectory in each community. Through the above iterative process, the n trajectories can be classified into k clusters.

Thus, our DP-LTOD scheme not only discovers the k latent trajectory communities based on the movement pattern of users, but also protects the users' privacy from leaking.

6 Performance Evaluation

In this section, we conduct several experiments from two real-world datasets, in order to evaluate the performance of our DP-LTOD scheme in the aspects of trajectory community detection, effect of privacy budget and effect of key techniques.

6.1 Datasets and Experimental Setup

Datasets. We use two datasets to verify the validity and efficiency of our methods. GeoLife datasets [45] has recorded the GPS trajectory

TABLE 2
20 types of raw POI datasets

Type	Name	Type	Name
1	Food & Beverages Service	11	Motorcycle Service
2	Road Ancillary Facilities	12	Car Service
3	Place Address Information	13	Car Maintenance
4	Scenic Spot	14	Car Sales
5	Public Facilities	15	Commercial Housing
6	Company	16	Life Service
7	Shopping Service	17	Sports Leisure Service
8	Transportation Service	18	Health Care Service
9	Financial Insurance Service	19	Governments Organizations
10	Education Culture Service	20	Accommodation Service

of 182 users with 18670 trajectories in five years (from 2008/10 to 2012/8), which not only includes the daily activity (e.g. going home, working, etc.), but also includes the recreational activity (e.g. shopping, traveling, eating, sports, etc.). Most of the data in GeoLife datasets lies in Beijing and few of them in Europe or USA. Beijing POI datasets [4] includes the location information for all kinds of interest points in Beijing. As shown in Table 2, the raw Beijing POI datasets can be classified into 20 types to well describe the semantic information of trajectories.

Experimental Setup. All experiments are conducted on a computer with Intel i7-3770 3.40 GHz CPU and 8 GB RAM, running 64-bit Windows 10 OS. Because the experiment only considers the GPS trajectory information locating in the region of Beijing, we filter the latitude and longitude in the region of [39.4, 41.1] and [115.4, 117.6], respectively. The total number of sampling points is 20284273 after pre-processing, which include longitude, latitude and time. According to our previous studies [1], [4], different locations have been assigned different semantic information, and different users may have different movement pattern or preference. Fig. 5 shows an example of semantic description of locations for one user. In order to facilitate the further experiments, we pre-process the trajectories to make them have the same number of segments, i.e. $|T_i^*| = \left|T_j^*\right|$.

6.2 Trajectory Community Detection

To evaluate the effect of trajectory community detection, we compare our LTOD method with buddy-based discovering algorithm (BU) [46] and SP-tree [20]. BU algorithm is used to discover the traveling companions for users according to their uploaded streaming trajectories. SP-tree is used to discover the movement-based communities of users, where users in the same community have similar movement behaviors.

Evaluation Metrics. We use precision, recall and F1-score to evaluate the effectiveness while utilizing different algorithms to detect trajectory communities. Since there is no unique method to cluster the users of Geolife datasets into different kinds of communities, we select the test set of mined communities based on user similarity which has been computed by our previous works [1], [4]. This experiment is to compare the performance of trajectory community detection for LTOD, BU and SP-tree.

The definition of each evaluation metric is explained as follows.

- *Precision*: Let $\mathcal{A}$ denote the set of detected communities, and $\mathcal{B}$ represents the set of communities in test set. The precision of trajectory community detection methods can be computed as:

$$Precision@k = \frac{\mathcal{A} \cap \mathcal{B}}{k}, \tag{20}$$

  where k is the number of discovered communities.

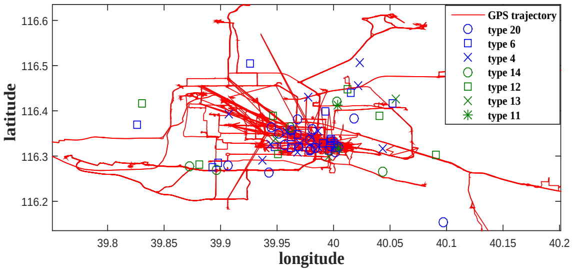


Fig. 5. The semantic description of locations for one user.

- *Recall*: Let $\mathcal{B}'$ denote the set of positive communities in training set. The recall of trajectory community detection methods can be computed as:

$$Recall@k = \frac{\mathcal{A} \cap \mathcal{B}}{|\mathcal{B}'|}, \tag{21}$$

  where $\left|\mathcal{B}'\right|$ is the number of positive communities.
- F_1-*score*: In order to synthetically evaluate the stability of trajectory community detection methods, the metric of F_1-score can be computed as:

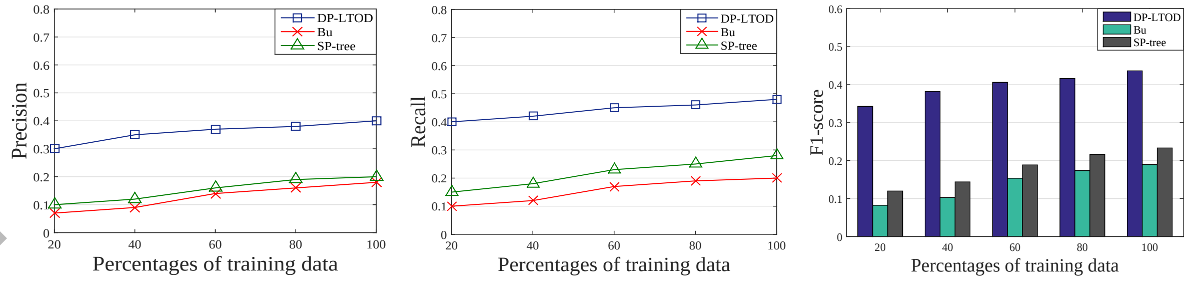
$$F_1(\mathcal{A}, \mathcal{B}) = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{22}$$

According to the evaluation metrics, we compare our DP-LTOD method with BU and SP-tree in the aspects of effect of percentages of training data, effect of number of discovered communities and effect of number of positive communities as follows.

(1) Effect of percentages of training data:

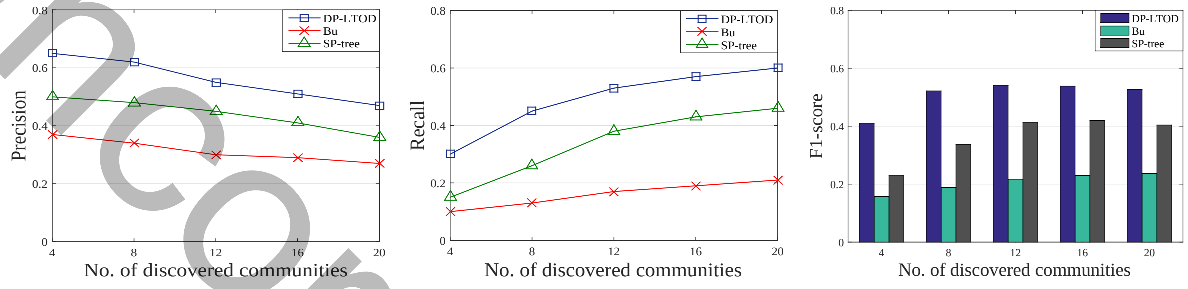
We first compare the performance of these algorithms according to different percentages of training data. Data sparsity is always happened due to user privacy. Especially in LBSNs, users may upload their true locations or trajectories to server in order to have a good quality of experience. However, some sensitive locations should be inhibited to protect users' privacy from leaking. In that case, it is important to efficiently detect the community which includes the users with similar interest or preference for service recommendation. From Figs. 6(a), 6(b) and 6(c), we can see DP-LTOD substantially outperforms BU and SP-tree. For algorithm BU, it uses the density-based clustering, including size, distance, duration and density, to discover traveling companion. However, if the uploaded trajectory datasets is sparse (e.g. inhibited sensitive information publishing), the BU can not effectively discover qualified companions for target user, which inevitably leads to poor performance. For algorithm SP-tree, it utilizes the sequential probability tree to construct user model and mine the movement-based trajectory communities of users. However, only the geographical location sequences are considered in the structure of user model. Thus, algorithm SP-tree does not produce good results when the trajectory data is incomplete and sparse. In contrast, in DP-LTOD, we not only consider the geographical distance among location sequences, but also consider the semantic information of each location. Through mining the movement pattern of each user, we can classify the users who have different interest or preference into different kinds of communities. Since the semantic information is considered instead of actual location information, our DP-LTOD algorithm can well address the issue of data sparsity.

From Fig. 6, we can also see DP-LTOD obtains better performance than BU and SP-tree in terms of Precision. The reasons are explained as follows. Semantic information can well reflect the interest or preference of users. For example, if the common semantic sequence of user A and us-



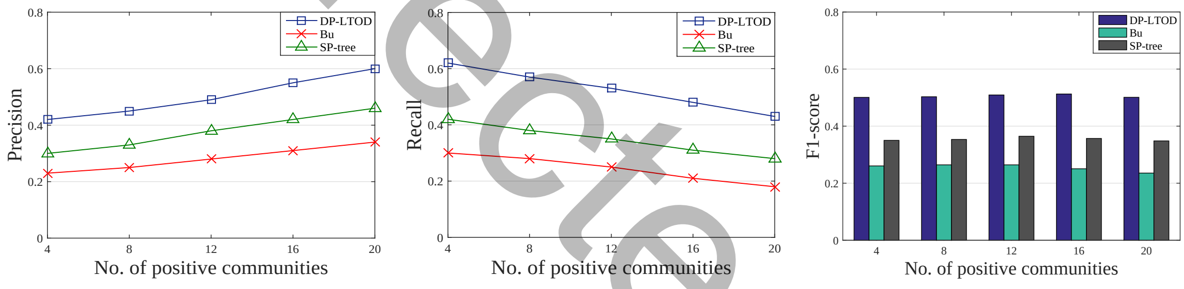
(a) Precision (b) Recall (c) F_1-score

Fig. 6. Effect of percentages of training data



(a) Precision (b) Recall (c) F_1-score

Fig. 7. Effect of number of discovered communities



(a) Precision (b) Recall (c) F_1-score

Fig. 8. Effect of number of positive communities

er B is “$type1 \rightarrow type10 \rightarrow type15$”, they will have the same movement pattern as “$Food\ \&\ Beverages\ Service \rightarrow Education\ Culture\ Service \rightarrow Commercial\ Housing$” according to Table II. Although the actual location sequences of user A and user B are different, they may be the latent friends which have similar interest or preference in semantic space. Thus, user A and user B can be clustered into one community, which effectively increase the precision of trajectory community discovering.

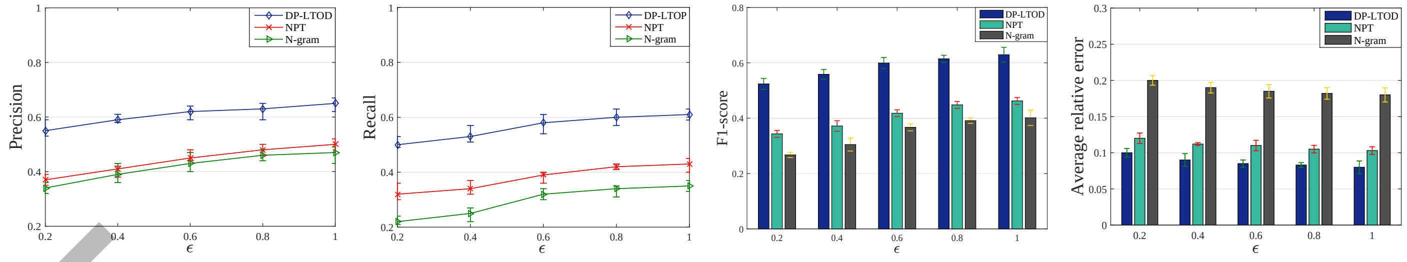
(2) Effect of number of discovered communities:

Second, we compare the performance of the three algorithms according to different number of discovered communities, which reflects the different granularity of trajectory community detection. Coarser-grained business task can well discover the users who have similar mobile behavior (e.g. direction, distance and velocity, etc.). However, fine-grained business task can further mine the similar users who have specific interest or preference (e.g. movement pattern and habit, etc.). In LBSNs, it is important to find the potential friends who have similar interest or preference by fine-grained trajectory clustering [47]. From Figs. 7(a), 7(b) and 7(c), we can see DP-LTOD achieves better performance than the other two algorithms although the precision decreases as the number of discovered communities increases. This is because, our algorithm DP-LTOD aims to mine the interest or preference of users in semantic space, it is well adapted to the fine-grained business task. From the experimental results, we can also see algorithm SP-tree gains comparable performance to DP-LTOD. The reason is that algorithm SP-tree makes use of sequential probability tree to construct user model. Sequential probability tree can discover much frequency information which reflects the activity law of users in geographical space, and thus algorithm SP-tree can achieve better performance than Bu.

(3) Effect of number of positive communities:

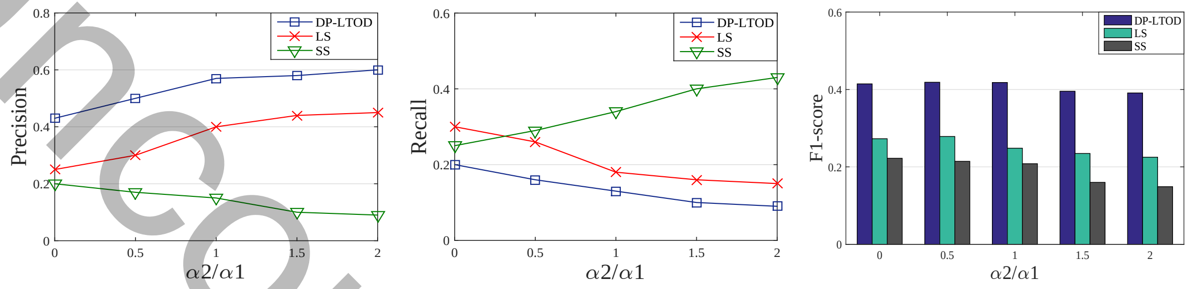
Finally, we compare the performance of the three algorithms according to different number of positive communities. In this experiment, we define positive community as the communities that have existed in the training set. That is to say, we have known some similar users who are clustered into different communities before conducting trajectory community detection. Under such condition, from Figs. 8(a), 8(b) and 8(c), we can see the precision increases as the number of positive communities increases. However, the recall of each algorithm is decreasing. This is because more positive communities training set includes, higher false negatives server has, i.e. Eq. (21). We can also see algorithm DP-LTOD has the best performance of precision in three algorithms. As discussed above, algorithm DP-LTOD takes semantic information of each location into

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2018.2855740, IEEE Transactions on Services Computing



(a) Precision (b) Recall (c) F_1-score (d) Average relative error

Fig. 9. Effect of privacy budget



(a) Precision (b) Recall (c) F_1-score

Fig. 10. Effect of key techniques

account to effectively mine interest or preference of users, and further cluster them into different communities. Since more information is considered than the other two algorithms which only consider geographical information of locations for trajectory clustering, algorithm DP-LTOD can achieves better precision. What's more, from Fig. 8(c), we can see the changing of F_1-score is very small for algorithm DP-LTOD. It reflects that the performance of algorithm DP-LTOD is more stable in spite of the number of positive communities.

From the above experiments, we can see the latent trajectory communities, which have similar user preference, could be well mined by considering geographical information and semantical information of locations. Thus, LBSNs server can provide personalized community discovering services by making use of DP-LTOD algorithm. Next, we will evaluate the effectiveness of privacy budget for different differential privacy trajectory sequence publishing algorithms.

6.3 Effect of Privacy Budget

The aim of this part is to evaluate the data utility of three algorithms when original trajectory sequence is translated to obfuscated trajectory sequence. Privacy budget ϵ can well reflect the strength of privacy-preserving. As ϵ increases, the strength of privacy-preserving will be reduced. Normally, the value of ϵ is not more than 1. Thus, we evaluate the performance with ϵ varies from 0.2 to 1 [48]. Each algorithm is run 20 times and the average is reported. In this part, we utilize proposed trajectory community detection algorithm (i.e. LTOD) to evaluate the data utility of obfuscated trajectory sequences. Furthermore, in order to better complete this experiment, we declare that there is almost no obfuscation mechanisms when the value of ϵ is 1.

As DP-LTOD is the first method that supports the latent trajectory community discovering under differential privacy, we compare the algorithm DP-LTOD with two differential privacy trajectory publishing algorithms, i.e. N-gram [49] and Noisy Prefix Tree (NPT) [50]. In this experiment, the inputs of three algorithms are both the same original trajectory. The number of locations which original trajectory owns is k. The value of k is set as 50. In addition, the values of ϵ_l and ϵ_s are same in default. Algorithm N-gram is able to generate the anonymized trajectories by making use of variable length N-grams, and algorithm NPT can utilize a prefix tree with noise to obfuscate the trajectories. For N-gram and NPT algorithms, we first conduct them on the original trajectories to generate obfuscated trajectories, and then run the non-private LTOD algorithm (i.e. **Algorithm 3**) by using the obfuscated trajectories.

There are three reasons to explain that why we choose N-gram and NPT algorithms as comparison. 1) Our DP-LTOD scheme is also to protect user privacy before users publish their trajectory sequences to LBSNs server. All three methods are designed to protect trajectory information of users from leakage. 2) All three methods utilize differential privacy theory to obfuscate original trajectory sequence, which can not only protect user privacy, but also provide good data utility. What's more, our DP-LTOD method divides original trajectory sequence into a series of locations and segments, and generates obfuscated trajectory sequence by linking the obfuscated locations. 3) This paper not only considers user privacy-preserving, but also considers latent trajectory community discovering. In the aspect of latent trajectory community discovering, DP-LTOD method can cluster the users who have similar interest or preference into a community. Therefore, in the aspect of user privacy-preserving, we compare DP-LTOD method with N-gram and NPT algorithms, and evaluate the performance under different privacy budget ϵ.

According to the evaluation scheme from X. Xiao et al. [51], we utilize relative error (RE) to evaluate the utility of a counting query Q over the sets of obfuscated trajectories $\tilde{\mathcal{T}}$ and original trajectories $\mathcal{T}$. And the relative error with respect to the accuracy of three methods can be computed as:

$$RE = \frac{\left|Q\left(\tilde{\mathcal{T}}\right) - Q\left(\mathcal{T}\right)\right|}{\max\left\{Q\left(\mathcal{T}\right), s\right\}}, \tag{23}$$

where s is a *sanity bound* to effectively mitigates the influence of the queries with negligibly small selectivities.

Fig. 9 shows the performance of DP-LTOD, N-gram and NPT under different privacy budget ϵ. From the experimental results, we find three algorithms behave in a similar way, i.e. the performance of different metrics is improved while ϵ increases. The reason is that as the increasing of ϵ, a lower degree of privacy-preserving is provided and a smaller magnitude of noise is added. We can also see algorithm DP-LTOD constantly achieves better performance with the same level of privacy-preserving. This is because, algorithm DP-LTOD makes use of two steps to add noise. Firstly, Laplace distribution-based noise is added at the stage of optimal location selection, which not only prevents the loss of semantic information, but also ensures ϵ_l-differential privacy. Secondly, exponential distribution-based noise is added at the stage of optimal segment selection, which not only prevents the loss of geographical information, but also ensures ϵ_s-differential privacy. Algorithm N-gram first utilizes a variable-length n-gram model to extract the sensitive information of a trajectory sequence, and then add adaptive noise to the sequential data. Algorithm NPT first constructs the prefix tree for original trajectory sequence, then add noise to each node. As algorithm N-gram and NPT fail to consider the semantic information of locations and the geographical distance of segments, the obfuscated trajectory is different with original trajectory in semantic space and geographical space. However, for algorithm DP-LTOD, since obfuscated trajectory is similar with original trajectory in semantic space and geographical space through the above two processes, the following step of latent trajectory community discovering on server can have a better performance by clustering the obfuscated trajectories. Especially in Fig. 9(d), we observe algorithm DP-LTOD can achieve lower average relative error than the other two algorithms with privacy budget ϵ increases.

6.4 Effect of Key Techniques

In Fig. 10, we show how the optimal location selection technique, optimal segment selection technique and threshold α_1, α_2 affect the performance of algorithm DP-LTOD. In order to compare the performance of key techniques, we set the number of discovered communities to be 8, and the privacy budget to be 0.6. Let LS denote the algorithm which firstly utilizes the optimal location selection technique to select the suitable location, and then randomly chooses the segments to generate obfuscated trajectory. Let SS be the algorithm that randomly selects the obfuscated locations, and then uses the optimal segment selection technique to generate obfuscated trajectory. From Figs. 10(a), 10(b) and 10(c), we can see DP-LTOD achieves better performance than these algorithms with the value of α_2/α_1 increases. We also observe that the precision of algorithm DP-LTOD and LS increases as the value of α_2/α_1 increases but the precision of algorithm SS decreases. This is because, algorithm DP-LTOD and LS consider the semantic information of locations to generate obfuscated trajectory sequences, which can well discover the users who have similar movement pattern and preference. However, algorithm SS only considers the geographical information of segments, which causes the performance decreases. What's more, from Fig. 10(c), we can see the F_1-score for algorithm DP-LTOD is better when the value of α_2/α_1 is set between 0.5 and 1. This result will be helpful for our future study on similar users discovering in LBSNs.

7 Conclusions and Future Work

In this paper, we study the problem of latent trajectory community discovering under the rigorous differential privacy model. First, we introduce Latent Trajectory cOmmunity Discovering (LTOD) which takes users' interest or preference into account to trajectory clustering. Then, we explore the possibility of designing a Differential Privacy Latent Trajectory cOmmunity Discovering (DP-LTOD) scheme which can ensure great performance of privacy and data utility. We found, in DP-LTOD, the feature of obfuscated trajectory sequence is important to improve the accuracy of LTOD. If we could effectively select the optimal obfuscation trajectory, the utility and privacy tradeoff can be significantly improved. To this end, we formulate a trajectory obfuscation problem to select an optimal trajectory which has the smallest difference with the original trajectory. We prove this problem is NP-hard, and propose a heuristic trajectory obfuscation algorithm to solve the problem. In our DP-LTOD scheme, another core is to add the noise required by differential privacy during the stage of trajectory obfuscation. In order to prevent Bayes' attack and Markov's attack, we add Laplace distribution-based noise and exponential distribution-based noise to the outputs during the stages of location obfuscation matrix generation and trajectory sequence function generation, respectively. Through formal privacy analysis, we prove that DP-LTOD scheme satisfies ϵ-differentially private. By experiments, it shows that our DP-LTOD scheme can privately discover latent trajectory community with high accuracy.

For the future work, we will further complete attack model by considering social connection graph or text of content, etc. Deep learning technology will be utilized to train location obfuscation matrix, in order to intelligently perceive users' practical location and select obfuscated location. In addition, we will consider more dimensions (e.g. time or velocity, etc.) to discover latent trajectory communities. It would be interesting to apply discovered latent trajectory communities to recommend personalized service for users in LBSNs.

Acknowledgment

This work was partially supported by National Natural Science Foundation (NSF) China under Grant Nos. 61522103 and 61501405. S. Yu's work is partially supported by NSF China under grant No. 61728201.

Appendix A

Let $\mathcal{D}_{loc_i^*}$, $\mathcal{D}^{'}_{loc_i^*}$ represent any two candidate databases. For any outcome $H(i,j)$ and any transition probability matrix $\mathbb{P}$, we have:

$$\begin{aligned}
\frac{\Pr\left[Alg1\left(\mathcal{D}_{loc_i^*},\mathbb{P}\right)\right]}{\Pr\left[Alg1\left(\mathcal{D}'_{loc_i^*},\mathbb{P}\right)\right]} &= \frac{\prod_{i=1}^{d} p\left(\mathbb{P}_i - Alg1\big(\mathcal{D}_{loc_i^*}\big)_i\right)}{\prod_{i=1}^{d} p\left(\mathbb{P}_i - Alg1\Big(\mathcal{D}'_{loc_i^*}\Big)_i\right)} \\
&= \frac{\prod_{i=1}^{d} \exp\left(-\epsilon_l\left|\mathbb{P}_i - Alg1\big(\mathcal{D}_{loc_i^*}\big)_i\right|\right)}{\prod_{i=1}^{d} \exp\left(-\epsilon_l\left|\mathbb{P}_i - Alg1\Big(\mathcal{D}'_{loc_i^*}\Big)_i\right|\right)} \\
&= \exp\left\{\epsilon_l\Big[\sum_{i=1}^{d}\Big(\Big|\mathbb{P} - Alg1\Big(\mathcal{D}^{'}_{loc_i^*}\Big)_i\Big| \right. \\
&\qquad \left. - \Big|\mathbb{P} - Alg1\big(\mathcal{D}_{loc_i^*}\big)_i\Big|\Big)\Big]\right\} \\
&\le \exp\Big[\epsilon_l\Big(\sum_{i=1}^{d}\big|Alg1\Big(\mathcal{D}^{'}_{loc_i^*}\Big)_i \\
&\qquad - Alg1\big(\mathcal{D}_{loc_i^*}\big)_i\big|\Big)\Big] \\
&\le \exp\left(\epsilon_l\right).
\end{aligned} \tag{24}$$

According to Eq. (9), we prove that the optimal location selection technique satisfies ϵ_l-differential privacy.

APPENDIX B

The probability density function of $S_u^{\epsilon_s}$ is:

$$f_{\mathcal{D}_{L_{ij}^*}}\left(L_{ij}^*\right)=\frac{\exp\left(\frac{\epsilon_s}{2\Delta f}u\left(\mathcal{D}_{L_{ij}^*},L_{ij}^*\right)\right)}{\int_{\mathcal{D}_{L_{ij}^*}}\exp\left(\frac{\epsilon_s}{2\Delta f}u\left(\mathcal{D}_{L_{ij}^*},s\right)\right)d_s}. \tag{25}$$

Let $\mathcal{D}_{L_{ij}^*}$, $\mathcal{D}^{'}_{L_{ij}^*}$ represent any two candidate segment databases, then for any outcome u. We have:

$$\begin{aligned}f_{\mathcal{D}_{L_{ij}^*}}\left(L_{ij}^*\right)&=\frac{\exp\left(\frac{\epsilon_s}{2\Delta f}u\left(\mathcal{D}_{L_{ij}^*},L_{ij}^*\right)\right)}{\int_{\mathcal{D}_{L_{ij}^*}}\exp\left(\frac{\epsilon_s}{2\Delta f}u\left(\mathcal{D}_{L_{ij}^*},s\right)\right)d_s}\\&\leq\frac{\exp\left(\frac{\epsilon_s}{2\Delta f}\left(u\left(\mathcal{D}^{'}_{L_{ij}^*},L_{ij}^*\right)+\Delta f\right)\right)}{\int_{\mathcal{D}'_{L_{ij}^*}}\exp\left(\frac{\epsilon_s}{2\Delta f}\left(u\left(\mathcal{D}^{'}_{L_{ij}^*},s\right)-\Delta f\right)\right)d_s}\\&=\frac{\exp\left(\frac{\epsilon_s}{2}\right)\cdot\exp\left(\frac{\epsilon_s}{2\Delta f}u\left(\mathcal{D}^{'}_{L_{ij}^*},L_{ij}^*\right)\right)}{\exp\left(-\frac{\epsilon_s}{2}\right)\cdot\int_{\mathcal{D}'_{L_{ij}^*}}\exp\left(\frac{\epsilon_s}{2\Delta f}u\left(\mathcal{D}^{'}_{L_{ij}^*},s\right)\right)d_s}\\&=\exp\left(\epsilon_s\right)\cdot f_{\mathcal{D}'_{L_{ij}^*}}\left(L_{ij}^*\right).\end{aligned} \tag{26}$$

Therefore, the probability

$$\begin{aligned}\Pr[Alg2(\mathcal{D}_{L_{ij}^*},u)]&=\int_{\mathcal{D}_{L_{ij}^*}}f_{\mathcal{D}_{L_{ij}^*}}(L_{ij}^*)d_{L_{ij}^*}\\&\leq\int_{\mathcal{D}'_{L_{ij}^*}}\exp\left(\epsilon_s\right)\cdot f_{\mathcal{D}'_{L_{ij}^*}}(L_{ij}^*)d_{L_{ij}^*}\\&=\exp\left(\epsilon_s\right)\cdot\Pr[Alg2(\mathcal{D}^{'}_{L_{ij}^*},u)].\end{aligned} \tag{27}$$

According to Eq. (9), we prove that the optimal segment selection technique satisfies ϵ_s-differential privacy.

REFERENCES

[1] L. Zhu, C. Xu, J. Guan, and H. Zhang, "Sem-ppa: A semantical pattern and preference-aware service mining method for personalized point of interest recommendation," *Journal of Network and Computer Applications*, vol. 82, pp. 35 – 46, 2017.

[2] J. L. Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[3] S. Liu and S. Wang, "Trajectory community discovery and recommendation by multi-source diffusion modeling," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 898–911, April 2017.

[4] L. Zhu, C. Xu, J. Guan, and S. Yang, "Finding top-k similar users based on trajectory-pattern model for personalized service recommendation," in *2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 553–558.

[5] C. Xu, S. Jia, L. Zhong, and G. M. Muntean, "Socially aware mobile peer-to-peer communications for community multimedia streaming services," *IEEE Commun. Mag.*, vol. 53, no. 10, pp. 150–156, October 2015.

[6] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G. M. Muntean, "Performance-aware mobile community-based vod streaming over vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 1201–1217, March 2015.

[7] C. Xu, S. Jia, L. Zhong, H. Zhang, and G. M. Muntean, "Ant-inspired mini-community-based solution for video-on-demand services in wireless mobile networks," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 322–335, June 2014.

[8] S. Jia, C. Xu, J. Guan, H. Zhang, and G. M. Muntean, "A novel cooperative content fetching-based strategy to increase the quality of video delivery to mobile users in wireless networks," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 370–384, June 2014.

[9] J. Cao, S. Wang, and H. Wang, "Detecting communities on topic of transportation with sparse crowd annotations," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 1017–1022, April 2017.

[10] C. Xu, Z. Li, J. Li, H. Zhang, and G. M. Muntean, "Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 7, pp. 1175–1189, July 2015.

[11] Y. Sun, H. Luo, and S. K. Das, "A trust-based framework for fault-tolerant data aggregation in wireless multimedia sensor networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 9, no. 6, pp. 785–797, Nov. 2012.

[12] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," *IEEE Access*, vol. 4, pp. 2751–2763, 2016.

[13] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, pp. 1010–1027, Nov 2001.

[14] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," in *22nd International Conference on Data Engineering (ICDE'06)*, April 2006, pp. 24–24.

[15] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*, April 2007, pp. 106–115.

[16] S. Gaffney and P. Smyth, "Trajectory clustering with mixtures of regression models," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '99. New York, NY, USA: ACM, 1999, pp. 63–72.

[17] D. Chudova, S. Gaffney, E. Mjolsness, and P. Smyth, "Translation-invariant mixture models for curve clustering," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 79–88.

[18] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '07. New York, NY, USA: ACM, 2007, pp. 593–604.

[19] J. G. Lee, J. Han, and X. Li, "A unifying framework of mining trajectory patterns of various temporal tightness," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1478–1490, June 2015.

[20] W. Y. Zhu, W. C. Peng, C. C. Hung, P. R. Lei, and L. J. Chen, "Exploring sequential probability tree for movement-based community discovery," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 11, pp. 2717–2730, Nov 2014.

[21] W. X. Zhao, N. Zhou, W. Zhang, J.-R. Wen, S. Wang, and E. Y. Chang, "A probabilistic lifestyle-based trajectory model for social strength inference from human trajectory data," *ACM Trans. Inf. Syst.*, vol. 35, no. 1, pp. 8:1–8:28, Sep. 2016.

[22] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services," in *2008 IEEE 24th International Conference on Data Engineering*, April 2008, pp. 366–375.

[23] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 12, pp. 1719–1733, Dec 2007.

[24] S. I. Ahamed, M. M. Haque, and C. S. Hasan, "A novel location privacy framework without trusted third party based on location anonymity prediction," *SIGAPP Appl. Comput. Rev.*, vol. 12, no. 1, pp. 24–34, Apr. 2012.

[25] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in location based services," in *2013 Proceedings IEEE INFOCOM*, April 2013, pp. 2985–2993.

[26] X. Pan, J. Xu, and X. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1506–1519, Aug 2012.

[27] 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*, April 2008.

[28] S. Gao, J. Ma, W. Shi, G. Zhan, and C. Sun, "Trpf: A trajectory privacy-preserving framework for participatory sensing," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 6, pp. 874–887, June 2013.

[29] R. Chen, B. C. Fung, N. Mohammed, B. C. Desai, and K. Wang, "Privacy-preserving trajectory data publishing by local suppression," *Information Sciences*, vol. 231, pp. 83 – 97, 2013, data Mining for Information Security.

[30] P. I. Han and H. P. Tsai, "Sst: Privacy preserving for semantic trajectories," in *2015 16th IEEE International Conference on Mobile Data Management*, vol. 2, June 2015, pp. 80–85.

[31] R. H. Hwang, Y. L. Hsueh, and H. W. Chung, "A novel time-obfuscated algorithm for trajectory privacy protection," *IEEE Trans. Serv. Comput.*, vol. 7, no. 2, pp. 126–139, April 2014.

[32] V. Rastogi and S. Nathl, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2010, pp. 735–746.

[33] A. Monreale, G. L. Andrienko, N. V. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel, "Movement data anonymity through generalization," *IEEE Transactions on Network and Service ManagementTransactions on Data Privacy*, vol. 3, no. 2, pp. 91–121, 2010.

[34] A. Monreale, W. H. Wang, F. Pratesi, S. Rinzivillo, D. Pedreschi, G. An-

drienko, and N. Andrienko, "Privacy-preserving distributed movement data aggregation," *Lecture Notes in Geoinformation & Cartography*, vol. 1, pp. 225–245, 2013.

[35] N. Mohammed, D. Alhadidi, B. C. M. Fung, and M. Debbabi, "Secure two-party differentially private data release for vertically partitioned data," *IEEE Trans. Depend. Sec. Comput.*, vol. 11, no. 1, pp. 59–71, Jan 2014.

[36] T. Zhu, P. Xiong, G. Li, and W. Zhou, "Correlated differential privacy: Hiding information in non-iid data set," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 229–242, Feb 2015.

[37] J. D. Zhang and C. Y. Chow, "Enabling probabilistic differential privacy protection for location recommendations," *IEEE Trans. Serv. Comput.*, pp. 1–1, 2018.

[38] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2018.

[39] P. Klein and R. Ravi, "A nearly best-possible approximation algorithm for node-weighted steiner trees," *Journal of Algorithms*, vol. 19, no. 1, pp. 104–115, 1995.

[40] C. Dwork, *Differential Privacy*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[41] J. Spragins, "A note on the iterative application of bayes' rule," *IEEE Trans. Inf. Theory*, vol. 11, no. 4, pp. 544–549, Oct 1965.

[42] E. A. Ghenciu and R. D. Mauldin, "Conformal graph directed markov systems," *Mathematics*, 2007.

[43] R. Sarathy and K. Muralidhar, "Evaluating laplace noise addition to satisfy differential privacy for numeric data," *Transactions on Data Privacy*, vol. 4, no. 1, pp. 1–17, 2011.

[44] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 94–103.

[45] Y. Zheng, X. Xie, and W. Y. Ma, "Geolife: A collaborative social networking service among user, location and trajectory," *Bulletin of the Technical Committee on Data Engineering*, vol. 33, no. 2, pp. 32–39, 2010.

[46] L.-A. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, C.-C. Hung, and W.-C. Peng, "On discovery of traveling companions from streaming trajectories," in *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, pp. 186–197.

[47] Z. Wang, D. Zhang, X. Zhou, D. Yang, Z. Yu, and Z. Yu, "Discovering and profiling overlapping communities in location-based social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 499–509, April 2014.

[48] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1298–1309.

[49] R. Chen, G. Acs, and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pp. 638–649.

[50] R. Chen, B. C. Fung, B. C. Desai, and N. M. Sossou, "Differentially private transit data publication: A case study on the montreal transportation system," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 213–221.

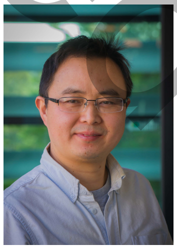
[51] X. Xiao, G. Wang, and J. Gehrke, "Differential privacy via wavelet transforms," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 8, pp. 1200–1214, Aug 2011.

Changqiao Xu received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences (ISCAS) in January 2009. He was an Assistant Research Fellow in ISCAS from 2002 to 2007, where he was a Research and Development Project Manager in the area of communication networks. During 2007 - 2009, he worked as a Researcher with the Software Research Institute at Athlone Institute of Technology, Athlone, Ireland. He joined Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in December 2009. Currently, he is a Professor with the State Key Laboratory of Networking and Switching Technology, and Director of the Next Generation Internet Technology Research Center at BUPT. He has published over 100 technical papers in prestigious international journals and conferences, including IEEE Comm. Surveys & Tutorials, IEEE Wireless Comm., IEEE Comm. Magazine, IEEE TMC etc. His research interests include social networks, wireless networking, and future Internet technology. He serves as a Co-Chair and Technical Program Committee (TPC) member for a number of international conferences and workshops. He is Senior member of IEEE.

Liang Zhu received Ph.D. degree in computer science and technology from BUPT in October 2017. He is currently a lecturer with the Institute of Computer and Communication Engineering at Zhengzhou University of Light Industry, Henan, China. His current research interests include mobile social networks, personalized service recommendation, and privacy preserving.

Yang Liu received the BE degree in electrical engineering and its automation and the ME degree in control theory and control engineering from Harbin Engineering University, Harbin, China, in 2008 and 2010, respectively, and the PhD degree in computer engineering at the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, in 2014. He is currently an associate professor at Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include wireless networking and mobile computing. He is a member of the IEEE and ACM.

Jianfeng Guan received the Ph.D. degree in communications and information system from Beijing Jiaotong University, Beijing, China, in January 2010. He is currently a associate professor with the Institute of Network Technology at Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include around mobile IP, mobile multicast, and next generation Internet technology.



Shui Yu is currently a full Professor of School of Software, University of Technology Sydney, Australia. Dr. Yu's research interest includes Security and Privacy, Networking, Big Data, and Mathematical Modelling. He has published two monographs and edited two books, more than 200 technical papers, including top journals and top conferences, such as IEEE TPDS, TC, TIFS, TMC, TKDE, TETC, ToN, and INFOCOM. Dr Yu initiated the research field of networking for big data in 2013. His h-index is 32. Dr Yu actively serves his research communities in various roles. He is currently serving the editorial boards of IEEE Communications Surveys and Tutorials, IEEE Communications Magazine, IEEE Internet of Things Journal, IEEE Communications Letters, IEEE Access, and IEEE Transactions on Computational Social Systems. He has served more than 70 international conferences as a member of organizing committee, such as publication chair for IEEE Globecom 2015, IEEE INFOCOM 2016 and 2017, TPC chair for IEEE BigDataService 2015, and general chair for ACSW 2017. He is a Senior Member of IEEE, a member of AAAS and ACM, the Vice Chair of Technical Committee on Big Data of IEEE Communication Society, and a Distinguished Lecturer of IEEE Communication Society.