Privacy- and Context-aware Release of Trajectory Data

ELHAM NAGHIZADE, LARS KULIK, EGEMEN TANIN, and JAMES BAILEY,
University of Melbourne, Australia

The availability of large-scale spatio-temporal datasets along with the advancements in analytical models and tools have created a unique opportunity to create valuable insights into managing key areas of society from transportation and urban planning to epidemiology and natural disasters management. This has encouraged the practice of releasing/publishing trajectory datasets among data owners. However, an ill-informed publication of such rich datasets may have serious privacy implications for individuals. Balancing privacy and utility, as a major goal in the data exchange process, is challenging due to the richness of spatio-temporal datasets. In this article, we focus on an individual's stops as the most sensitive part of the trajectory and aim to preserve them through spatio-temporal perturbation. We model a trajectory as a sequence of *stops* and *moves* and propose an efficient algorithm that either substitutes sensitive stop points of a trajectory with moves from the same trajectory or introduces a minimal detour if no safe Point of Interest (POI) can be found on the same route. This hinders the amount of unnecessary distortion, since the footprint of the original trajectory is preserved as much as possible. Our experiments shows that our method balances user privacy and data utility: It protects privacy through preventing an adversary from making inferences about sensitive stops while maintaining a high level of similarity to the original dataset.

CCS Concepts: • **Security and privacy → Data anonymization and sanitization**;

Additional Key Words and Phrases: Spatio-temporal databases, trajectory privacy, data publication, semantics

ACM Reference format:
Elham Naghizade, Lars Kulik, Egemen Tanin, and James Bailey. 2020. Privacy- and Context-aware Release of Trajectory Data. *ACM Trans. Spatial Algorithms Syst.* 6, 1, Article 3 (January 2020), 25 pages.
https://doi.org/10.1145/3363449

1 INTRODUCTION

The growth in information technology and its penetration into our daily life along with the widespread use of sensing devices that are connected to the Internet, such as smart phones and wearables has resulted in an ever-increasing amount of personal data being collected. According to a recent report, the are more than 7 billion mobile cellular subscriptions, among which more than 45% are smart phones. Further, GPS-enabled navigation systems and wearables are becoming omnipresent in society, capturing human mobility data at a fine-grained scale resulting in the generation of large, detailed spatio-temporal datasets.

Mining large spatio-temporal datasets provides valuable insights into managing key areas of society, such as transportation and urban planning, health and wellbeing, epidemiology, and natural

Authors' address: E. Naghizade, L. Kulik, E. Tanin, and J. Bailey, School of Computing and Information Systems, The University of Melbourne, Parkville, Melbourne, Australia; emails: {enaghi, lkulik, etanin, baileyj}@unimelb.edu.au.

2374-0353/2020/01-ART3 $15.00
https://doi.org/10.1145/3363449

disasters management [8–10, 27, 28, 35, 39]. As a result, there is an increasing trend in sharing large spatio-temporal datasets among several entities. However, privacy has become a major concern when sharing/releasing large datasets, since such fine-grained data incorporate a large amount of personal, sensitive information, as was recently observed by the Strava incident, where the release of fitness tracking data revealed a number of secret army bases [19].

Consequently, means of preserving location and trajectory privacy have drawn the attention of many researchers [5, 15, 16, 23]. Most of the studies in this area can be generally divided into *online* and *offline* privacy preserving approaches. The former deals with real-time protection of location privacy, for example, when an individual is accessing a location-based service and requires an immediate response [5, 16, 17, 22, 30]. The latter typically addresses the privacy of spatio-temporal databases prior to sharing/publishing or mining the data [2, 18, 20, 21, 25, 31, 32].

Since sharing large spatio-temporal datasets opens up tremendous opportunities to extract useful knowledge and to significantly improve location-based services, we need to ensure the best possible level of *data utility* to enable a high quality of analysis is crucial when preserving privacy. Despite its importance, few studies have explicitly focused on maintaining data utility when preserving trajectory privacy. In this article, we propose an algorithm that preserves privacy while ensuring a high level of utility. To estimate the utility of trajectory data, we present a distance measure that is able to capture the difference between two trajectories in terms of regional proximity, duration, and average speed, and it provides a good means of determining data utility for a large range of queries. To better balance the privacy/utility tradeoff, this article focuses on a key question: *What makes a trajectory sensitive?* Determining the sensitive parts of a trajectory may facilitate estimating the required level of distortion to ensure privacy without giving up too much data utility.

Most studies in the literature aim to preserve the footprint of a trajectory and also emphasize the importance of protecting its start and end point. While the stop and end of the trajectory is discriminative enough to identify individuals with a high probability, in our article, we assume that the identity of the individuals is known and rather the trajectory semantics, e.g., the purpose of the trip, is perceived to be sensitive. In particular, we argue that a trip's semantic can be learned more comprehensively from the stops of a trip that are not the start or end point, i.e., the *intermediate stops*. For instance, a visit to a medical centre followed by a stop at a pharmacy may indicate that the trip is a *health-related* trip, whereas a trip with the same start point but with a stop at a mart may imply that it is a *shopping-related* trip.

As a result, in this article we focus on the intermediate stops as the sensitive parts of a trajectory and propose an approach to protect these sensitive stops as the private parts of the trajectory. Protecting stop points rather than the whole trajectory can be beneficial in two ways. First, it alleviates the amount of distortion that is added to the original trajectory data. In addition, this method of protecting trajectory data is highly effective when an adversary's goal is to make further inferences about an individual, e.g., their health or financial status, rather than their identity.

Recently, some studies [14, 20, 24, 31] also aim to protect specific parts of a trajectory that are considered to be more sensitive rather than protecting trajectory as a whole. For instance, Huo et al. [20] assume that most of an adversary's background knowledge is associated with an individual's visited places, and, hence, preserving such places protects the trajectory. To hide the whereabouts of an individual, they coarsened the location of visited places in the database. However, repeated transitions between coarse and fine granularities in a dataset may help an adversary to infer additional information from an individual's path and hence violate trajectory privacy. As a result, our approach aims to preserve the uniformity of the dataset in terms of granularity. Besides, in our work all stop points are not considered equally private, and their level of sensitivity

is determined according to their location type, e.g., hospital, station, restaurant, as well as other parameters.

A key idea of this article is to change the original sequence of stops and moves in a trajectory to protect sensitive stops while ensuring that data granularity remains uniform. This is achieved by first making a sensitive stop a part of a non-sensitive move episode and then substituting it with a less private stop. Given a certain POI density, our *Flip-flop* approach efficiently preserves trajectory privacy through exchanging sensitive stops with less-sensitive, and possibly varied, types of POIs. This approach also manages to maintain utility, since it selects the POIs that introduce the least distortion to the data. The main contributions of this work are as follows:

- —We propose the idea of exchanging stop and move episodes of a trajectory to circumvent trajectory privacy attacks. This results in the generation of trajectories that are homogeneous and have the same features as real trajectories.
- —We introduce a sensitivity measure for stops based on both spatial and temporal features. This metric considers both the location type and duration of a stop.
- —We develop the Flip-flop algorithm that efficiently exchanges sensitive stops with less-sensitive POIs. The algorithm can be tailored to users' privacy preferences. We also provide a probability model for its performance based on POI density as a key factor.
- —We propose a distance concept that measures the spatial and temporal deviation. Our distance metric amends the Fréchet distance to estimate the amount of introduced distortion. Utilizing this distance measure, the Flip-flop approach chooses POIs such that it balances a high level of privacy and data utility.

In Section 2, we survey related studies concerning privacy in trajectory data and highlight our contributions. In Section 3, we state the problem and discuss the preliminaries for understanding the proposed Flip-flop algorithm. In Section 4, we detail our exchange approach. Section 5 presents our evaluation process followed by future direction in Section 6. We conclude this article in Section 7.

2 RELATED WORK

2.1 Trajectory Privacy

Human trajectory data consist of highly sensitive information and, therefore, carelessly publishing it—even an incomplete part of it—can cause serious privacy issues. In an attempt to highlight the importance of privacy-preserving publication of trajectory data, the authors of Reference [37] point out how "even parts of a trajectory could be used as quasi-identifiers to discover the rest of the trajectory." In their work, they consider a case that a database consists of the location of users' card transactions. The adversaries are supposed to be distinct parties who own a subset of the transaction database. The aim of the article is to suppress the original database in a way that none of the parties can derive an additional point that does not belong to their own address list with a probability more than a threshold.

The authors of Reference [32] and Reference [2] have proposed trajectory anonymizing approaches to preserve privacy based on the concept of k-anonymity, which was first proposed in Reference [36] for generic data privacy. The proposed approach in Refeerence [32] makes a certain trajectory indistinguishable from $k - 1$ other trajectories via a generalization-based algorithm. Consequently, privacy is ensured by releasing only a randomly generated set of representative trajectories. The approach in Reference [2], on the other hand, proposes a cluster-based algorithm that utilizes the uncertainty threshold, which is inherent to the trajectory data, to group k co-localized trajectories within the same time period to form a k-anonymized aggregate trajectory. All these works, though, focus on changing the entire trajectory footprint as a means of preserving

privacy. This leads to a more-than-necessary distorted database, which considerably decreases the quality of querying and mining.

Publishing perturbed trajectory datasets has also been studied in the literature, where the privacy is mainly preserved by adding noise to the original data. The authors of Reference [21] propose a differential privacy approach to perturb ship trajectories. After adding global noise to the trajectory and noise to each point of the trajectory and to each coordinate of the trajectory, the authors employ an exponential mechanism that can bound the noise level through sampling distance and direction of the points in the original trajectory, which leads to a better data utility. However, this approach may not be suitable for car trajectories where the underlying road network is available to an adversary and can be used to refine the trajectory data. In a more recent study, the authors of Reference [6] propose the concept of geo-indistinguishability, which ensures the output of the location obfuscation process is similar for all the points that are geographically close to each other. To achieve this, the mechanism adds a two-dimensional Laplace noise to user's actual location point. They then use the composition theorem [29] to generalize their method to protect trajectories. Similarly to the proposed mechanism in Reference [21], geo-indistinguishability requires the addition of a large amount of noise, which may be removed using a road network structure or the general topology of the space. In contrast to these two approaches, where noise is added directly to the location points, there is another trend in applying the concept of differential privacy to trajectory datasets: building a differentially private model using the original trajectories and releasing a synthetic dataset based on the noisy model, where the main goal is to preserve the aggregated count of trajectories in a region of interest [11, 12, 18, 25].

2.2 Protecting Sensitive Stop Points

However, some recent studies have focused on *stop points* along a path as the most-sensitive parts of a trajectory and proposed some solutions to protect these points. The authors of Reference [14] propose an obfuscation technique for online location-based services. This technique considers geographic context of a stop and provides an uncertainty region based on the POI distribution and users' privacy profile. The authors of Reference [31] model a trajectory as a sequence of stays and the goal is to publish a *c-safe* version of the original dataset where the probability of inferring that a person has visited a sensitive stop along visiting a sequence of non-sensitive stops is less than a safety threshold. Likewise, in Reference [20], the authors employed generalization methods in a way that sensitive places along a user's trip will be replaced by l-diverse zones. In other words, the published dataset contains a set of fine-grained location points along with some cloaked areas representing user's stops. Our work shares the same assumption of these studies in the sense that it regards stop points of the trajectory more vulnerable to privacy breaches.

However, publishing a dataset with different levels of location granularities, where an individual spent a significant time in coarser areas, may cause privacy concerns itself. The adversary can easily infer the number of sensitive stops in a trip along with the time they occurred, which is in fact, a privacy breach itself. The adversary may also find a correlation between place types in consecutive stops along a trip to make further inferences. For instance, if a user has stopped in a zone including a hospital, and later has stopped in another zone containing a pharmacy, the adversary may relate these places to make further inferences. It is also possible to refine the obfuscated rectangle if the user takes different paths to get to the same stop point.

Recently, the authors of Reference [13] have proposed a technique to replace the sensitive stop points in a trajectory with alternative POIs. Their approach builds a stop taxonomy tree, the height of which determines the sensitivity of a stop point. However, this approach does not take the temporal property of a trajectory into account and decides on the sensitiveness of a place only by considering its type. However, the duration of staying in an intermediate stop may play an

important role in its privacy vulnerability. Our proposed sensitivity measurement considers both the type of the visited POI and the duration of the visit to determine the sensitivity of a stop.

2.3 Data Utility

When preserving trajectory privacy, maintaining data utility is of great importance. Most existing privacy preserving approaches do not explicitly discuss data utility, and some of them determine the usability of their preserved outcome against a fixed value [16] or based on the accuracy of querying the privacy-aware database [2, 20, 30]. However, estimating utility based on the accuracy of the query results (i) limits the number of applications and (ii) is highly dependant to the quality of the original data and does not necessarily reflect the utility of privacy preserved data [26]. As a result, we determine data utility through measuring the dissimilarity of the original trajectories to their preserved counterparts. This provides a generic means of expressing data utility in terms of *utility loss*.

2.4 Trajectory Similarity

Measuring the similarity of moving object trajectories has attracted great attention with various applications in indexing and retrieval of trajectories, query processing, and trajectory clustering. A distance measure that has been commonly used to determine the similarity between trajectories is the Hausdorff distance. The Hausdorff distance is the maximum distance among all the distances from any given point in a trajectory to its closest point (minimum distance) in the other trajectory. However, the Hausdorff distance looks for the minimum distance between points regardless of their order, which makes it less effective for trajectories. Time-series approaches, such as Dynamic Time Warping (DTW) has been tailored to estimate trajectory similarity as well [38]. In general, DTW aims to stretch two sequences in time to align them with minimal cost. This makes DTW an unsuitable measure in our case, since we need a measure that is able to capture the distortion both in space and time. Another well-established measure for determining the similarity between two trajectories is the Fréchet distance [3]. The Fréchet distance has been described as the minimum length of a leash to connect a walking man to a dog on two independent trajectories, where none of them is allowed to move backward. When measuring the distance between two trajectories, the Fréchet distance looks for all possible time (speed) parameterizations to find the one that minimizes this distance. This makes it a suitable metric to estimate the distance between two time-independent trajectories while in our case we need to compare trajectories with invariant timestamps.

3 PROBLEM DEFINITION

Our work aims to propose a utility-aware trajectory data exchange in applications where an individual's overall path and identity are not considered as private. Instead, the individual's visited places in a trip is of concern. In this section, we discuss some preliminaries. *Trajectory Dataset:* A trajectory is a function from time to geographical space. A trajectory dataset is given in the form of a set of timestamped location points $(\vec{l}, t)$ along with possibly some additional attributes denoted as a, such as speed, direction, and measurement accuracy. Furthermore, in our scenario, the identity—or the pseudonym—of individuals is assumed to be known. As a result, $\forall u_i \in \mathcal{U}$, where $\mathcal{U}$ is the set of all users in the database, there is a set of m_i trajectories in the form of:

$$\mathcal{T}_{ij} = \{(\vec{l}_{1_{ij}}, t_{1_{ij}}, a_{1_{ij}}), (\vec{l}_{2_{ij}}, t_{2_{ij}}, a_{2_{ij}}), \ldots, (\vec{l}_{n_{ij}}, t_{n_{ij}}, a_{n_{ij}})\},$$

where $1 < i < |\mathcal{U}|$ and $0 < j < m_i$ and $\mathcal{T}_{ij}$ indicates the jth trajectory of the ith user. Note that m_i is the number of distinct trajectories of each user and may vary from one user to the other.

We confine the trajectory model to its spatial and temporal features and do not discuss additional attributes.

Trajectory Episodes: Depending on the movement states of a trajectory, it may be segmented into a sequence of *stays* and *moves* [4]. In this way, a trajectory is represented as a set of stays, $\mathcal{S}$, and moves, $\mathcal{M}$, with the respective start and end time of each segment. As a result, for any user, u_i, her jth trajectory is represented as follows:

$$\mathcal{T}_{ij} = \{(M_{1_{ij}}, t_{s_{ij}}, t_{e_{ij}}), (S_{1_{ij}}, t_{s_{ij}}, t_{e_{ij}}), \ldots, (M_{n_{ij}}, t_{s_{ij}}, t_{e_{ij}})\},$$

where $S_{k_{ij}}$ and $M_{k_{ij}}$, $k \in [1, \frac{n}{2} + 1]$, are subsets of $\mathcal{T}_{ij}$ and represent all the stop and move episodes between the timestamps t_s and t_e timestamps. This representation enables us to focus on stop episodes of a trajectory when preserving its privacy.

We extract stop episodes of a trajectory based on the definition in Reference [39], where a stop is characterised by a set of consecutive points whose distance to each other is less than a distance threshold, δ_d, for a period longer than a temporal threshold, δ_t:

$$S = \{(\vec{l}_q, t_q), (\vec{l}_{q+1}, t_{q+1}), \ldots, (\vec{l}_r, t_r)\},$$

where $\forall p \in [q+1, r]$, $d(\vec{l}_q, \vec{l}_p) \leq \delta_d$, and $|t_r - t_q| \geq \delta_t$.

Stop Episodes: Following the stop extraction process, several subsets of points in the trajectory dataset may be derived as stop points. In other words, we group any set of consecutive points, from t_s to t_e, that meet the required conditions as a single stop point and we have:

$$S_i = \{(\vec{l}_{s_i}, t_{s_i}), (\vec{l}_{(s+1)_i}, t_{(s+1)_i}), \ldots, (\vec{l}_{(e-1)_i}, t_{(e-1)_i}), (\vec{l}_{e_i}, t_{e_i})\}.$$

Therefore, for each trajectory we extract a set of stop points, $\mathcal{S}$, in the form of $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$.

Moreover, we assume every stop point is accompanied with a level of sensitivity that specifies the privacy level of a place. The advantage of assigning a sensitivity level to each stop point is twofold: It provides individuals with a flexible privacy preserving approach that can be modified according to their needs. It can also help to minimise the imposed changes to the original data, since less-sensitive stops may be discarded prior to the exchange phase, decreasing the overall amount of data distortion.

For a sequence of trajectory episodes, we aim to simulate a movement behaviour whenever a sensitive stop occurs in the trajectory. This exchange phase is compensated through creating an identical stop episode at a less-sensitive, realistic place. There are of course different criteria for defining a realistic POI. The approach that we adopted first looks for less-sensitive POIs and then filters them according to their place taxonomy to have a variety of place types as substituted stops. Take a 2-hour trip with three sensitive stops as an example. At this point, the algorithm makes sure that all three selected POIs are not, for instance, "restaurants" but rather three distinct POI types. This strategy, however, could be replaced with other initiative to choose realistic POIs.

Exchanged Trajectory: An exchanged trajectory, $\mathcal{T}^*$, is a privacy-aware version of $\mathcal{T}$, in which any sensitive stop point has been swapped with a less-sensitive POI, S^* (if possible). Figure 1 provides an example of a trajectory, $\mathcal{T}$, with a sequence of $\{(M_1, t_1, t_2), (S_1, t_2, t_3), (M_2, t_3, t_4)\}$ episodes that last for d_1, d_2, and d_3, respectively. Assuming S_1 is sensitive and needs to be protected, our approach manipulates the start time of this stop as if it has occurred later. However, it retains the stop's duration, i.e., d_2 and only changes the duration of moves to d'_1 and d'_3. As a result of this, we obtain an exchanged trajectory, $\mathcal{T}^*$, which not only hides the sensitive stop's location, it also preserves the trajectory footprint and duration. It is noteworthy that such exchange needs to be performed with regard to the road network to ensure S^* is a possible semantic stop, e.g., a restaurant and not a point in the middle of a highway.

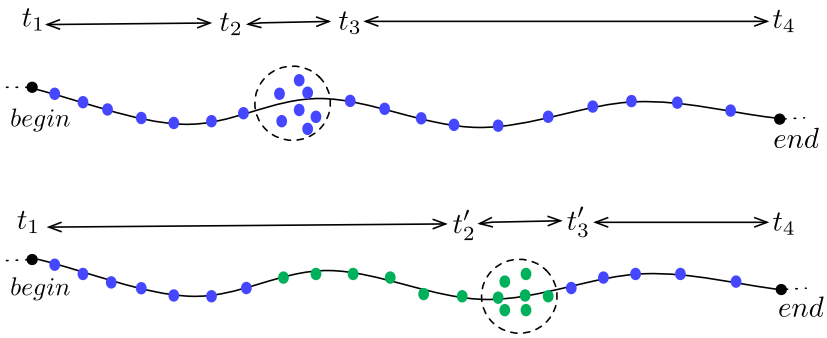


Fig. 1. Protecting stops through exchanging trajectory episodes.

3.1 Adversary Model

In this work, we assume an adversary is any third party with whom the dataset is going to be shared. Through finding places visited that are frequently and/or regularly visited by an individual, the adversary is interested in inferring individuals' personal preferences or habits. We assume the adversary knows that a user's trajectory data may be distorted and is even aware of the privacy preserving process. However, the following aspects are considered as the main barriers to inferring the actual location of original stops:

— *Data Consistency*: Current approaches that aim to preserve intermediate stops of a trajectory focus on cloaking methods, through which the location of stops is coarsened to satisfy a certain privacy requirement. Since a trajectory is presented as a sequence of fine-grained and coarse-grained location data, it is easy for an adversary to distinguish between real trajectories and protected trajectories. Our work utilizes the sequences of stops and moves in a trajectory to protect sensitive stops while ensuring that data granularity remains uniform. This makes it difficult for an adversary to distinguish exchanged trajectories from original trajectories.
— *Individual Preferences*: In the Flip-flop approach, users rank stops based on their type as well as their time of occurrence and duration. The notion of sensitive stops and safe POIs as their substitutes is determined based on these ranks and varies from one user to the other. Since an adversary is not aware about users' personal settings at the first place, it is difficult to infer the original stops.

3.2 Measuring Privacy

As pointed out earlier, in contrast to most prior work that focus on protecting an individual's actual trajectory, our key contribution is to protect an individual's sensitive stops. In general, there are two principal ways of achieving this: first perturbing the trajectory, which leads to the creation of new stops or selecting less-sensitive stops along the original trajectory. In both cases, this leads to utility loss. In this section, we show how to measure the sensitivity of a stop and provide a quantitative to determine of the impact of exchanging a sensitive stop. The aim is to decrease the overall sensitivity of stop(s) along the trajectory.

It is difficult to assign a value to quantify the sensitivity of POIs due to the lack of data. However, some POIs are clearly more sensitive than others. Since individuals have diverging opinion on this sensitivity, we propose to employ a taxonomy where individuals can rank the sensitivity of place types. Without loss of generality, we propose to have five place categories, and we rank the

Table 1. An Example of Sensitivity Measurements for an Original Trajectory with Three Stops and Its Exchanged Match

	S_1	S_2	S_3
r_s	0.7	0.8	0.5
r_s^*	0.35	0.5	0.2

sensitivity of each place, r_p, where $0 < r_p \leq 1$ to reflect this, where a higher r_p corresponds to a more sensitive place.

Other than the place of a stop, temporal properties of a stop are also of great importance when determining its level of sensitivity. Using a temporal factor, we differentiate between a regular stop, for instance, in a hospital, as in the case of a member of hospital workforce or an uncommon stop, as in the case of a patient. This may be done with regard to the start and end time of a stop point. For any uncommon stop, we also presume that the longer the stop point, the more sensitive it becomes. When modelling the stop sensitivity, $r_s \in (0, 1]$, we consider an exponential effect for a stop duration, which can mitigate the temporal factor while incorporating it in the sensitivity measurement.

Having the place sensitivity rank, r_p, the duration of a stop, d_s, and the total duration of a trip, d_t, we compute the overall sensitivity of a stop as:

$$r_s = r_p^{\frac{d_t - d_s}{d_t}},$$

where $r_s = 1$ if the entire trip occurs at a stop point, i.e., the trajectory is comprised of one single stop episode.

We then need to define a set of sensitive stops, $\mathcal{S}_s \subseteq \mathcal{S}$, where $\forall S_i \in \mathcal{S}_s$, its sensitivity rank is greater than a certain threshold. Subsequent to determining all sensitive stops, we need to find a POI as a substitute for each of them. The substitute POI needs to be chosen in such a way that makes it impossible for an adversary to notice any unusual incidence in the trajectory dataset. As mentioned earlier, the sensitivity representation may differ from one application to another and/or based on individuals' preferences. In this case, we can profile users based on their privacy preferences and customise the sensitivity estimation based on their settings. We can also use the same profile when looking for alternative POIs for the exchange process.

Following this exchange, the original stops are not detectable via reverse engineering. Moreover, it is highly unlikely that the adversary distinguishes an altered trajectory from a real one. As a result, we measure the accuracy of our approach in terms of preserving privacy as its ability to minimise the sensitivity level of a trip:

$$Privacy\ Gain = \frac{\sum_{i=1}^{k}(r_{s_i} - r_{s_i}^*)}{max_{sd}} \quad \in [0, 1],$$

where k is the number of stops in a trajectory and max_{sd} is the maximum sensitivity deviation possible if all the stops are exchanged by the k least-sensitive POIs on the route. This means that if the algorithm cannot exchange any of the sensitive stops with a less-sensitive POI, the overall privacy gained is zero, while if all the sensitive stops are exchanged with the least-sensitive POIs, the privacy gained is maximised. Take a three-stop trip as an instance, where all the stops are sensitive. Table 1 shows the sensitivity of the original stops and the substitute POIs. Given there are three POIs with sensitivity level of 0.2, the overall privacy gained is equal to $\frac{(0.7-0.35)+(0.8-0.5)+(0.5-0.2)}{(0.7-0.2)+(0.8-0.2)+(0.5-0.2)}$.

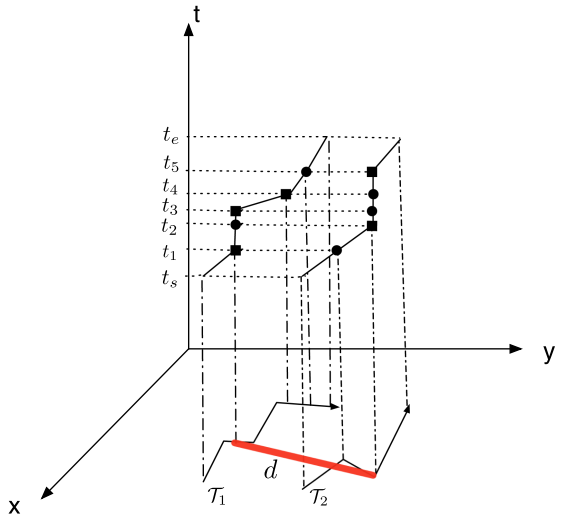


Fig. 2. Estimating the distance between trajectories after matching their timestamps.

3.3 Measuring Utility

As stated before, preserving privacy must be performed with respect to data quality to achieve a reasonable privacy/utility balance. We determine data utility by comparing the original trajectory with its exchanged match and measuring how similar they are. Proposing a general utility measure, instead of limiting it to certain query types and scenarios, helps us determine the performance of our approach regardless of the underlying application.

As for estimating the utility of the exchanged trajectories, commonly used distance measures in the literature (Section 2) are not suitable in our case. We are not only interested in measuring deviations from the original footprint (spatial projection) of trajectories, but we also need to compute temporal displacements caused by our exchange process. However, all existing measures provide a lower bound to the actual distance.

As a result, we adapt a Frećhet-based distance and computes the distance as if a single pair of speed parameterization is available instead of looking for the optimal parameterization that minimises the distance between two trajectories independent of time. To do so, we first need to make sure that for every point in each trajectory $(\vec{l}_i, t_i)$, there exists a corresponding point in the other one with the same timestamp, i.e., $(\vec{l}^*_i, t_i)$. For any t_i in $\mathcal{T}$, if $t_i \notin timestamps(\mathcal{T}^*)$, then we add a respective point in $\mathcal{T}^*$ between $(\vec{l}_j, t_j)$ and $(\vec{l}_{j+1}, t_{j+1})$, where $t_j < t_i < t_{j+1}$, using linear interpolation and vice versa (Figure 2). We then compute the distance as the maximum spatial distance between every temporally coincident pair of points:

$$Distortion = \max_{i \in [1,n]} [d(\vec{l}_i - \vec{l}^*_i)],$$

where n is the number of points in each trajectory, $\vec{l}_i$ and $\vec{l}^*_i$ are the location points of the original trajectory and exchanged trajectory at time i, and d is the Euclidean distance between two points. Estimating data utility, $u \in [0, 1]$, requires the definition of perfect utility and worst utility concepts. In this work, we assume perfect utility occurs if the original trajectory remains unchanged ($u = 1$), but worst utility can be described as a case that given a certain source, destination, and time budget, deviates from the original trajectory as much as possible, hence maximising information loss, i.e., $u = 0$. We then compare the exchanged trajectory against the original trajectory

to determine utility:

$$Utility = 1 - \frac{Distortion(\mathcal{T}, \mathcal{T}^*)}{Distortion_{max}} \quad \in [0, 1].$$

Our proposed utility measure returns one if there is no difference between the original and the perturbed trajectory, i.e., perfect utility. To compute the worst data utility, we need to find the trajectory with *maximum possible distortion* D_{max}, for each trajectory, $\mathcal{T}$. The trajectory with D_{max} can be conceptualised using a space-time prism where all the points inside the prism are reachable within the time budget. However, different trajectories inside the prism have varying degrees of distance from $\mathcal{T}$, and hence we look for the trajectory that maximizes the distance. We describe how to compute the set of trajectories within the space-time prism and finding the trajectory with D_{max} in Section 5.1.

Problem Statement: Given a trajectory, $\mathcal{T}$ with a set of intermediate sensitive stops, the aim is to efficiently find an exchanged trajectory $\mathcal{T}*$ that maximizes the privacy gain by replacing the sensitive stops with less-sensitive POIs while incurring the least amount of distortion to the original trajectory, i.e., minimizing the utility loss.

4 FLIP-FLOP APPROACH

Following a pre-processing phase that mainly deals with finding stop points and determining their level of sensitivity, this work aims to protect retrieved sensitive stops through the exchange strategy described in Section 3. Our exchange strategy aims to preserve the overall characteristics of the trajectory, namely duration, regional proximity, and average speed.

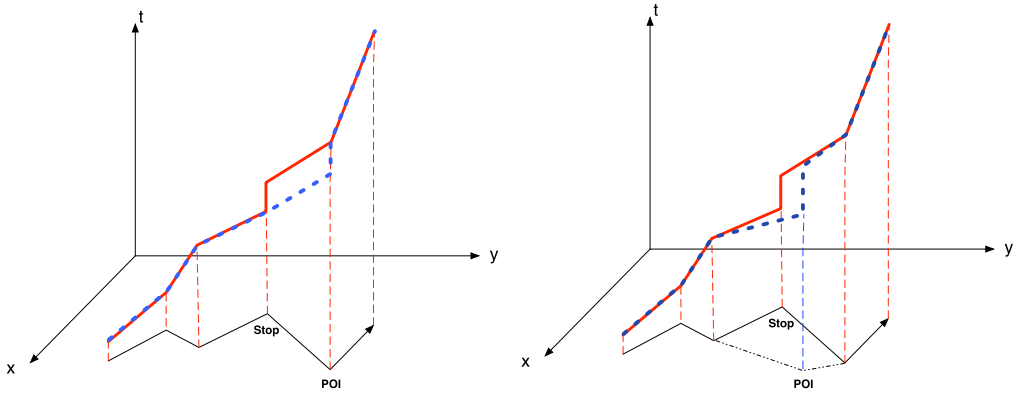
4.1 Stop & Move Exchange

As mentioned earlier, the algorithm utilizes POIs when exchanging the sequences of stops and moves of a trajectory. Depending on the result of searching for POIs, two scenarios may occur, namely replacement and displacement. The algorithm may find a POI on the same route and *replace* the sensitive stop point with it. Otherwise, it needs to look for a close POI that does not belong to the present route and *displace* the sensitive stop with that POI. The replacement process preserves the trajectory footprint completely and only involves temporal modification. However, displacement may cause local changes to both temporal and spatial properties. Nonetheless, this is performed in a way to keep these changes as minimally invasive as possible.

4.1.1 Stop Replacement. Stop replacement mainly concerns exchanging the sensitive stop with a candidate POI on the same route. Here the trajectory is represented in way as if the individual has kept on moving past the sensitive stop but rather stopped at the POI for an equal lapse of time. Figure 3(a) illustrates an original trajectory (solid trajectory) and its exchanged (dashed trajectory) in which the sensitive stop is replaced with a POI.

The algorithm also maintains the pattern of each stop when displacing it. The aim is to preserve the consistency of a trajectory dataset and thus prevent an adversary from noticing a change in the data. The replacement algorithm can be summarised as follows: Having a trajectory, $\mathcal{T}$, a sensitive stop point, $S \in \mathcal{S}$ and a point of interest, $poi \in \mathcal{P}$, the algorithm removes S from $\mathcal{T}$. Then, it finds the point inside $\mathcal{T}$ that corresponds to poi, and following the pattern of S creates a stop, i.e., S^*.

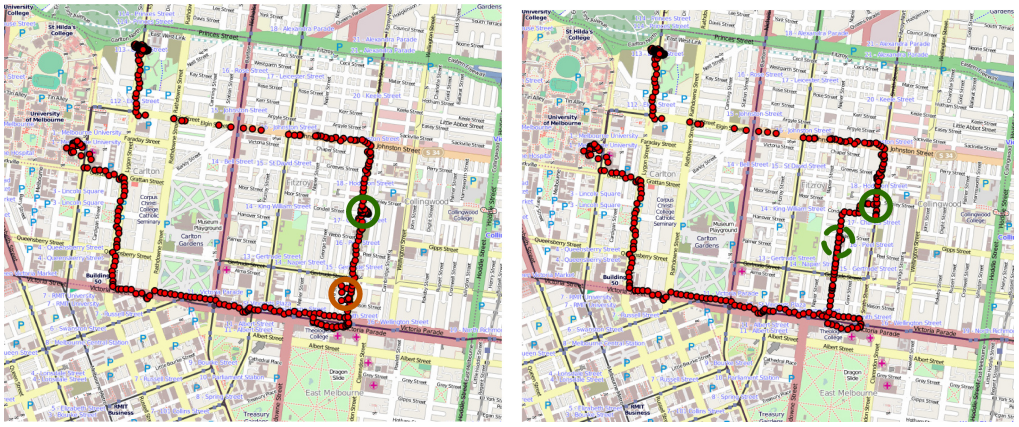
4.1.2 Stop Displacement. In case no POI is found in the previous step, the algorithm needs to search for candidate POIs within close proximity to the original route. The outline of displacing a sensitive stop is depicted in Figure 3(b). To limit the POI search space, we used the properties of an ellipse. Setting any two stop points as the foci of an ellipse, the algorithm is able to look for the POIs inside the ellipse area whose distance from these stops are less than a threshold; this threshold can be increased until a POI is found (Algorithm 1, lines 1—7); however, this increase



(a) Replacement of a sensitive stop.

(b) Displacement of a sensitive stop.

Fig. 3. Strategies to protect sensitive stops.



(a) The original trip.

(b) The preserved trip with a displaced stop.

Fig. 4. A real-world trip in the city of Melbourne and its private match. Green corresponds to non-sensitive stops and orange circle corresponds to the sensitive stops. The solid circles correspond to the actual stops while the dashed circle shows the exchanged non-sensitive stop.

should not exceed the overall boundary of an ellipse that contains all the reachable points from source to destination within the time budget. In this case, the sensitive stop(s) are regarded as non-preservable, and the exchange approach fails to protect them. However, our experimental results show that such case is highly unlikely in real-world scenarios.

The algorithm later finds the shortest path from the first stop to the POI and back to the next stop (Algorithm 1, lines 8 and 9). Apart from limiting the search space, this step ensures data utility by preserving proximity to the original track. Figure 4 shows a real trajectory and its preserved version in which the sensitive stop has been displaced.

4.2 Stop Exchange Process

In our work, we also adopt two methods of searching for POIs. One approach considers a trajectory as a whole and searches for less-sensitive POIs *exhaustively*. The other method compartments

ALGORITHM 1: Stop displacement

Input: Two sensitive stops $S_1, S_2 \in \mathcal{S}$, local trajectory between the two stops, $\mathcal{T}_{loc}$, set of all POIs, P.
Output: A displaced list of points, T_{dis}, with respect to sensitive stop points

$foci_1, foci_2 \leftarrow S_1, S_2$;
$findPOI \leftarrow inEllipse(foci_1, foci_2, P, Axis_{maj})$;
while $length(findPOI) = 0 \ \ AND \ \ Axis_{maj} < \delta$ **do**
  $majA \leftarrow Axis_{maj} * 2$;
  $findPOI \leftarrow inEllipse(foci_1, foci_2, P, Axis_{maj})$;
end
$poi \leftarrow filter(findPOI)$;
$sp1 \leftarrow shortestPath(S_1, poi)$;
$sp2 \leftarrow shortestPath(poi, S_2)$;
$T_{gen} \leftarrow genGPS(\mathcal{T}_{loc}, sp1, sp2)$;
$T_{dis} \leftarrow replaceStop(T_{gen}, S_1, poi)$;

the trajectory into segments, searches for POIs in each section, and performs the Flip-flop exchange.

4.2.1 Exhaustive Exchange. The first approach takes the trajectory as a whole and exhaustively searches for all POIs that are less sensitive than the actual stop (Algorithm 2, lines 2 and 3). We note that a trajectory might not contain less-sensitive POIs to exchange any/all of the sensitive stops in $\mathcal{S}_s$. The sensitive stops are then replaced with less-sensitive POIs if the algorithm can find those along the same route (Algorithm 2, lines 4–13). Assume that following the replacement step, a subset of sensitive stops, $S_d \in \mathcal{S}_s$, where $1 \leq |S_d| \leq |\mathcal{S}_s|$ are remaining. In this case, the algorithm takes the two farthest stop points in $\mathcal{S}_s$ and assigns them as the foci of an ellipse. It then searches for all POIs inside the ellipse area and displaces the stops with the retrieved POIs through the displacement algorithm (Algorithm 2, lines 14–21): We first rank the retrieved POIs based on their type sensitivity, i.e., r_p, and for all the POIs with the least sensitivity, P_c, where $|P_c| \geq |S_d|$, we permutate over all the combinations of the sensitive stops and candidate POIs through finding the shortest path to get from each sensitive stop to the candidate POI and back to the next sensitive stop and using the remaining stop time to compute the candidate POIs sensitivity, i.e., r_s. We then choose the combination that maximizes the privacy gain. In case we cannot find enough candidate POIs to displace the original stops, we increase the area of the ellipse.

4.2.2 Flip-flop Exchange. The Flip-flop approach segments the trajectory into episodes of stops and moves (Algorithm 3 lines 3—6). It then considers the move episode between every two consecutive stops and looks for POIs (Algorithm 3, line 7). If it manages to retrieve a less-sensitive POI on this route, then it replaces the first stop with it (Algorithm 3, lines 8–11). Otherwise, it searches in an ellipse area, whose foci are the two stop episodes to find a less-sensitive POI within this area. Finally, the algorithm displaces the first stop with the retrieved POI (Algorithm 3, lines 12–14). Algorithm 3 highlights the main steps of the Flip-flop method.

Searching a local space, Flip-flop is expected to find less-sensitive POIs faster than the exhaustive algorithm, which performs a global search. As a result, it may be presumed that with an increase in the number of stops, POIs or trip length, Flip-flop outperforms the exhaustive algorithm in terms of runtime. This is mainly because the average search complexity in the exhaustive algorithm is $|P|^k$, where $|P|$ is the number of POIs and k is the number of sensitive stops, whereas Flip-flop searches for less-sensitive POIs $k|P|$ times.

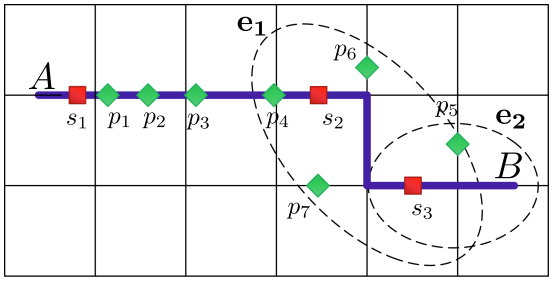


Fig. 5. Effect of POI density on Flip-flop's search space.

ALGORITHM 2: Exhaustive Exchange

Input: Original trajectory dataset, $\mathcal{T}$, list of sensitive stop points, $\mathcal{S}_s$, set of all POIs, P.
Output: An exchanged trajectory, $\mathcal{T}^*$, with respect to sensitive stop points.

$\mathcal{T}^* \leftarrow []$;
$\mathcal{P} \leftarrow filter(routePOI(\mathcal{T}, P))$;
for $poi \in \mathcal{P}$ **do**
  if $length(\mathcal{S}) > 0$ **then**
    $currStop \leftarrow bestMatch(\mathcal{S}_s, poi)$;
    $startIndex \leftarrow \mathcal{S}_s[currStop][ID]$;
    $endIndex \leftarrow \mathcal{S}_s[nextStop][ID]$;
    $\mathcal{T}_{loc} \leftarrow \mathcal{T}[startIndex : endIndex]$;
    $\mathcal{T}^* \leftarrow \mathcal{T}^* + replaceStop(\mathcal{T}_{loc}, currStop, poi)$;
    $\mathcal{S}_s \leftarrow \mathcal{S}_s - currStop$;
end
if $length(\mathcal{S}_s) > 0$ **then**
  while $length(\mathcal{S}_s) > 0$ **do**
    $S_i, S_j \leftarrow farthestStops(\mathcal{S}_s)$;
    $foci_1, foci_2 \leftarrow S_i, S_j$;
    $T_{dis} \leftarrow disStop(\mathcal{T}_{loc}, foci_1, foci_2, P, Axis_{maj})\mathcal{T}^* \leftarrow \mathcal{T}^* + T_{dis}$;
    $\mathcal{S} \leftarrow \mathcal{S} - S_i$;
  end
end

However, this may not always be true in case the increase in the number of stops or trip length does not correspond to the growth in the number of POIs/POI density. Generally, for a trajectory where $|P| < kx$ holds—with x as the average number of extra edges that need to be covered when searching for POIs inside an ellipse—exhaustive search space becomes more constrained and hence the exhaustive strategy becomes faster. Figure 5 illustrates an example where the route from A to B contains three sensitive stops, s_1, s_2, and s_3. The route also has four less-sensitive POIs, i.e., $P_{route} = \{p_1, p_2, p_3, p_4\}$. Clearly, the exhaustive algorithm replaces each stop with one POI from P_{route}. The Flip-flop algorithm, however, chooses a POI from P_{route} and replaces s_1 with it. Therefore, for s_2 and s_3 it does not find any candidate POI on the route between s_2 to s_3 and s_3 to the end. Hence, it needs to widen the search space to displace each of them with a less-sensitive POI within e_1 and e_2.

Considering a subtrajectory as any move episode between two consecutive stops, the probability of having a set of non-empty subtrajectories in terms of less-sensitive POIs is

ALGORITHM 3: Flip-flop Exchange

Input: Original trajectory dataset, $\mathcal{T}$, list of sensitive stop points, $\mathcal{S}_s$, set of all POIs, P.
Output: An exchanged trajectory, $\mathcal{T}^*$, with respect to sensitive stop points.

```
i ← 1;
T* ← [];
while i <= length(S_s) do
    startIndex ← S_s[i][ID];
    endIndex ← S_s[i + 1][ID];
    T_loc ← T[startIndex : endIndex];
    P ← filter(routePOI(T_loc, P));
    if length(P) > 0 then
        poi ← leastSensitive(P) ;
        T* ← T* + replaceStop(T*, S_s[i], poi);
    else
        foci_1, foci_2 ← S_s[i], S_s[i + 1];
        T_dis ← disStop(T_loc, foci_1, foci_2, P, Axis_maj)T* ← T* + T_dis;
    end
    i ← i + 1;
end
```

$$\mathcal{P}_{\neg\emptyset} = \prod_{1 \le i \le k} P_{\neg\emptyset_i}.$$

Defining the probability of finding at least one less-sensitive POI for a subtrajectory, p_s, as the average number of less-sensitive POIs on each edge in the road network, we can define $P_{\neg\emptyset_i}$ using Bernoulli probability:

$$P_{\neg\emptyset_i} = 1 - (1 - p_s)^{l_i},$$

where l_i is the number of edges of the ith subtrajectory.

Finally, we can define the probability that Flip-flop needs to displace sensitive stops in a trajectory, $\mathcal{P}_{dis}$, as the probability of having at least one subtrajectory on which no less-sensitive POI resides:

$$\mathcal{P}_{dis} = 1 - \mathcal{P}_{\neg\emptyset}.$$

In other words:

$$\mathcal{P}_{dis} = 1 - \left(\prod_{1 \le i \le k} 1 - (1 - p_s)^{l_i} \right). \tag{1}$$

The above equation shows that the increase in either l_i or p_s decreases $\mathcal{P}_{dis}$. Moreover, without loss of generality, p_s can be defined as the probability of finding an arbitrary number of POIs in a subtrajectory. This is mainly for cases wherein choosing a certain POI at a stage has a negative effect on the algorithm's ability to find POIs later. In the previous example, if s_2 is displaced with p_5, then Flip-flop is not able to preserve s_3. A similar case may happen if a POI cannot be selected because of speed constraints introduced with choosing previous POIs.

4.2.3 Privacy vs. Utility Optimization. For our proposed Flip-flop approach, two varieties to optimize trajectory privacy or data utility can be considered: In case of having multiple candidate POIs to exchange the sensitive stop, a privacy-aware version (PFF) searches for the least-sensitive

POI as the substitute and a utility-aware version (UFF) that selects a POI that minimizes the distance to the original trajectory when exchanging each POI.

4.3 Privacy Guarantees

As mentioned in Section 3.1, the adversary does not have any knowledge about the category of sensitive stops for an individual and cannot differentiate an actual stop from an exchanged one, since the outcome of our approach has a consistent granularity. Assuming that the adversary has full knowledge about the underlying privacy-preserving scheme, we can estimate the adversary's success in retrieving the original sensitive stops. Without loss of generality, we assume that the POI type is the only parameter when computing the stop sensitivity, and we also assume that the adversary has a binary approach toward an observed stop, i.e., a stop is either sensitive or not sensitive. Note that the adversary does not know what type of POIs are sensitive in this setting. For a given *potentially* perturbed trajectory, the adversary needs to consider a buffer around the trajectory to account for the displacement strategy of Flip-flop, i.e., the actual stops may no longer be on the trajectory any more. Given there are p POIs in the buffer, for any observed stop point, i.e., a real or exchanged stop, on the trajectory the adversary's success to correctly retrieve the pair of stops used to build the ellipse is bound to $\frac{1}{2\binom{p-1}{2}}$. With an increase in p the success of the adversary decays exponentially. Moreover, an increase in the number of observed stop points leads to a considerable uncertainty about the original stop points.

Note that such inference attack potentially returns several pairs as the original stops for a given observation: Depending on the spatial distribution of the points, there may be more than a single pair where the observed stop is within the ellipse formed by that pair as the foci points, hence resulting in a large number of false positive, which has the potential to increase the adversary's uncertainty.

5 EXPERIMENTS

We evaluate the performance of our proposed algorithm in two steps. We first generate synthetic trajectories and discuss the sensitivity of both versions of Flip-flop, i.e., UFF and PFF, with respect to POI density, number of POI classes, as well as the number and duration of stops. We then consider a real trajectory dataset and use Flip-flop to assess the efficacy of our proposed algorithm in providing privacy in a realistic setting.

5.1 Experimental Setup: Synthetic Data

Despite significant efforts to simulate trajectory data [7, 33], existing trajectory data simulators do not create semantic stops, i.e., stops of different types, along a trip. To evaluate the performance of our algorithm in different scenarios such as having trajectories with multiple stops of varying lengths, we developed a simulator to generate synthetic GPS trajectories with different modes of transport and multiple intermediate stops. The simulator first chooses two random points within an area ($\approx$12 km $\times$ 12 km in the city of Melbourne with an underlying road network consisting of 59,680 nodes and 69,534 edges) and computes the shortest path between them. It then finds all the POIs on this path using OpenStreetMap[1] data and randomly selects k POIs as the intermediate stop points, where $1 \leq k \leq 5$.

To reflect a realistic movement behaviour, the algorithm considers the mode of transport when generating data. For a walking behaviour, a constant speed is considered, whereas for driving, the speed is adjusted according to the street type and its speed limit. This results in a set of trajectories

[1] www.openstreetmap.org.

Table 2. Experimental Settings

	Functionality	Range	Default
p_s	POI density	$\frac{1}{2^5} - 1$	$\frac{1}{2^4}$(≈2170 POIs)
$\|S_s\|$	No of Stops	1–5	5
d_s	Stop Duration	1–5	5 (≈60 minutes)
T_p	POI Type	2,5	5

with either a single mode of transport, i.e., car, or up to four switches between walk and car mode. Since the stop duration is used when determining the sensitivity, we also increased the total stop duration of each trip exponentially up to five times, which provides us with 2–60 minutes for each stop. In summary, the generated dataset has a total of 750 distinct trajectories with an average length of 23 km.

To compute the trajectory with worst utility, we look for those points in space whose prism width is maximal: We retrieve the shortest path from all nodes in the network graph to the source and destination of the original trajectory. After preserving a minimum time to stay at stop point(s), we find the furthest node from the source node that can reach the destination within the time budget. Although such trajectory preserves some features of the original trajectory, such as regional proximity, overall duration, actual time of occurrence and the number of stops, it provides the poorest data utility and any deviation from it renders the trajectory data unusable.

To examine the effect of POI density on Flip-flop's performance, we consider two scenarios: We first generate points uniform at random within our region of interest. However, our observations show that most POIs in Melbourne (similar to other urban spaces) are on streets that are classified as territory and secondary roads. As a result, we consider another scenario where the generated POIs are preferentially mapped to those roads rather than distributing them randomly. For varying p_s (average number of POIs per edge), we have between 2,170 and 65,000 POIs in the network (note that some of the randomly generated points cannot be mapped to any edge). Considering the overall length of the network (2,622 km), this POI density creates different urban scenarios, i.e., sparse areas versus more populated areas.

We assign a sensitivity level to each POI according to their type, i.e., T_p. When determining the sensitivity of a POI, we may consider different scenarios. A case where the user classifies POIs as either sensitive or non-sensitive or cases where a user requires more flexibility via categories of less-sensitive to high-sensitive POIs. Through varying the above-mentioned parameters, we conduct a set of experiments to evaluate the performance of our proposed approach (Table 2). The default environment for our experiments is set to reflect the worst cases with a low POI density and high number of long stops. Moreover, the results provided in this section is averaged over 20 runs of experiments. The exhaustive approach searches for potentially sensitive POIs on the route and exchanges any sensitive stop with a POI that minimizes the sensitivity. If it finds multiple alternatives for a stop, then it chooses the POI that best preserves utility. We measure data utility via the distance measure defined in Section 3.3. We did not implement an exhaustive algorithm that maximises data utility as an algorithm that does not change any POI will trivially maintain an optimal data utility equal to 1 (with a distance of zero).

In our experiments, we use the exhaustive approach as a baseline to compare the success of UFF and PFF in preserving privacy, and the result of the exhaustive approach is not further discussed. The PFF and UFF's privacy are evaluated relative to the exhaustive approach (set as 1), which finds the least-sensitive POIs as substitutes (Section 3.2). The utility is measured with respect to the relation provided in Section 3.3, where we employed the generated trajectories with worst utility to estimate the maximum distortion.

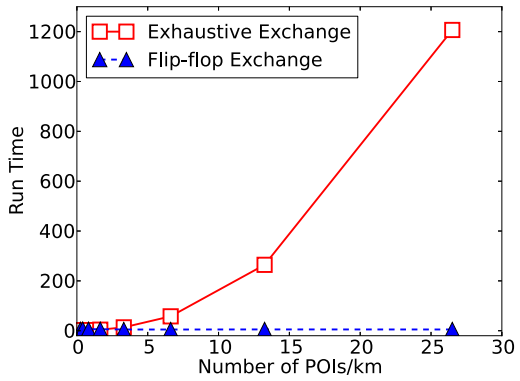


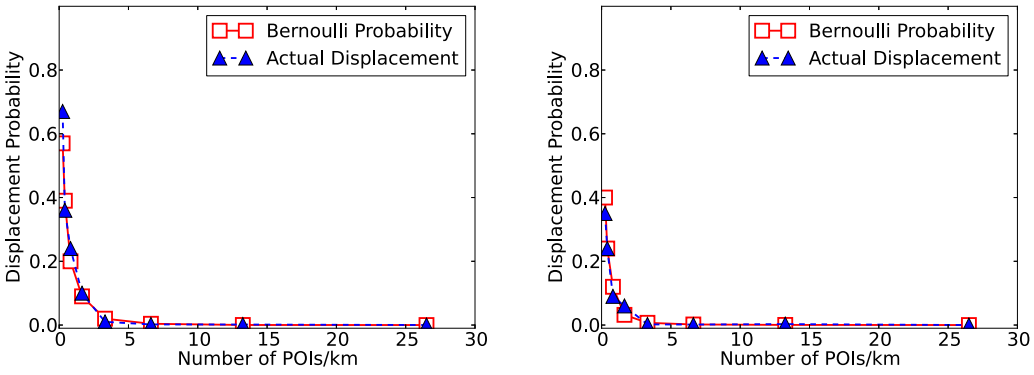
Fig. 6. Effect of POI density on Flip-flop's performance.

5.1.1 Flip-flop Performance. Figure 6 compares the performance of Flip-flop to the exhaustive approach with respect to varying POI densities. The x-axis shows the average number of POIs per kilometer with respect to various p_ss mentioned in Table 2. As can be seen, Flip-flop (bottom line) can cope with any POI density and has an almost constant computation time compared to the exponential growth in time complexity for the exhaustive approach. With lower densities, Flip-flop spends this time on the displacement process, whereas with an increase in POI density, the search for POIs along the original route needs substantial amount of time. For low POI densities, the exhaustive algorithm could be potentially faster than Flip-flop, since a global search increases the probability of finding a POI on the same route and requires less frequent displacements (Section 4.2.2). However, such POI densities (less than 1 POI/km of road), are largely uncommon in real-world scenarios. As for higher densities, it is not possible to run the exhaustive algorithm due to its prohibitive cost, but we further ran Flip-flop for $p_s = 2$ and $p_s = 4$ (50–100 POIs/km), which correspond to POI density at areas such as business districts. The corresponding average runtime of Flip-flop is 7.14 s and 17.99 s, respectively.

5.1.2 Displacement Probability. To provide empirical support for predicting the displacement probability (Equation (1)), we compare it with the actual number of displaced stops in our dataset. We perform this experiment for two scenarios: the case where POIs are distributed randomly, Figure 7(a), and the case where POIs are preferentially mapped to the main roads, Figure 7(b). For the latter case, the Bernoulli probability is computed using two distinct POI ratios, p_s, one for secondary and territory edges of the subtrajectory and one for the other edge types. Both figures, Figure 7(a) and Figure 7(b), show that an increase in the number of POIs corresponds to smaller numbers of displacements along a trajectory. Moreover, for the same POI density, the number of displacements decrease if the POIs are preferentially mapped. Since the number of edges in each subtrajectory follows a skewed distribution, the actual number of displaced stops does not exactly follow the Bernoulli distribution. However, as can be seen in both cases, this probability provides a reasonable estimate of the required number of displacements, which can be used prior to publishing a trajectory dataset: Having the POI density of a region, it can be decided if a certain level of privacy and/or utility can be achieved.

5.1.3 Privacy versus Utility. To evaluate the performance of our approach, we considered four cases where we vary the POI density, the number of POI classes, the number of stops, and the duration of stops, the result of which is presented in Figure 8.

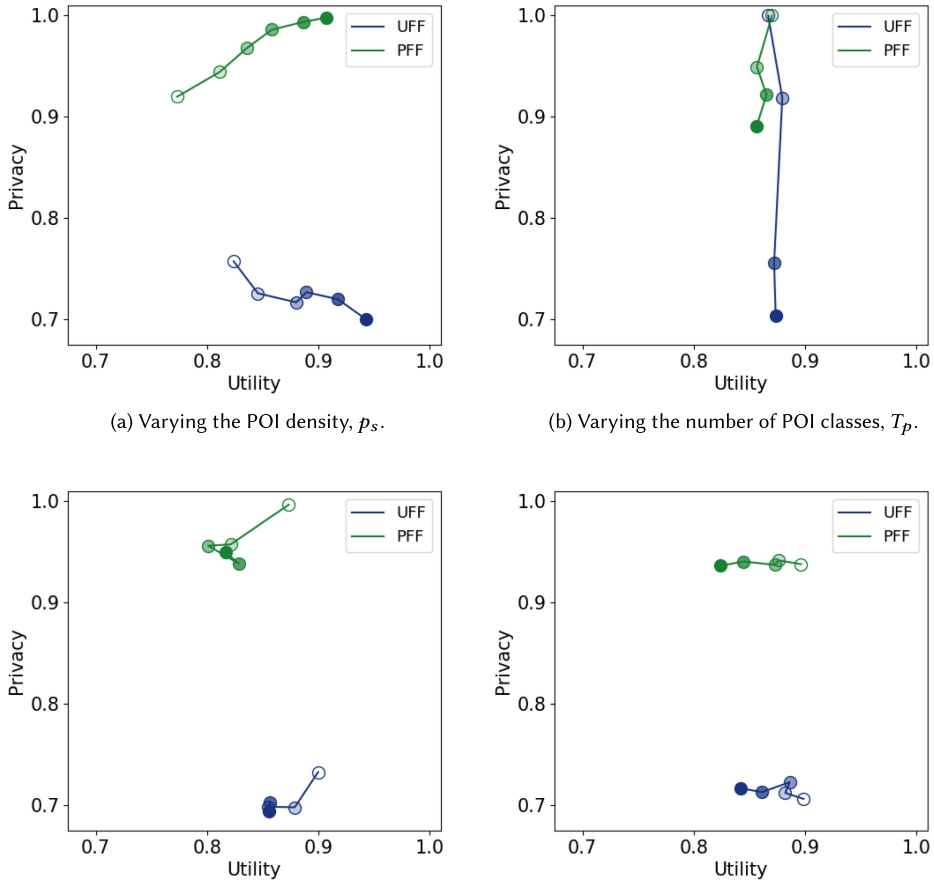
The effect of POI density: Figure 8(a) shows the average privacy and utility in the dataset for varying POI densities, where $p_s = [\frac{1}{2^5}, \frac{1}{2^4}, \frac{1}{2^3}, \frac{1}{2^2}, \frac{1}{2}, 1]$. As expected, low POI densities, i.e., lighter



(a) Uniformly distributed POIs.

(b) Preferentially mapped POIs.

Fig. 7. Effect of POI density on displacement probability on two settings.



(a) Varying the POI density, p_s.

(b) Varying the number of POI classes, T_p.

(c) Varying the number of stops, $|\mathcal{S}_s|$.

(d) Varying the stop duration, d_s.

Fig. 8. The effect of varying the parameters in Table 2 on the privacy/utility balance. The change in the colors from light to dark correspond to an increase in each parameter.

circles, lead to lower data utility for both variants of Flip-flop, because more detours are required from the original trajectory's footprint as less non-sensitive POIs are available along the original trajectory. It also shows that with an increase in POI density, UFF becomes more successful in maintaining data utility; however, the increase in POI density leads to lower privacy levels in UFF, since UFF can find POIs at closer distance but not necessarily with the least sensitivity.

However, it can be observed that PFF distorts the original data more than UFF but achieves significantly higher privacy levels. More specifically, with an increase in POI density, PFF achieves the best possible privacy, i.e., similar privacy levels as the exhaustive approach, although this accuracy comes—as expected—at the cost of a slight loss in utility. As for PFF, an increase in POI density increases the utility level as well. This is mainly due to the ability of PFF to find less-sensitive POIs at closer distances to the original sensitive stop. In a relatively dense area, the maximum privacy is achieved along with a very high level of utility (93% for PFF).

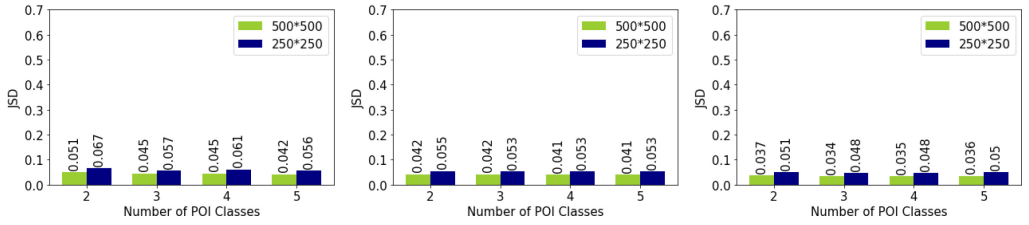
The effect of number of POI classes: As can be seen in Figure 8(b), having two POI types provides better privacy levels for a given POI densities. In fact, in case a user specifies POIs as either sensitive or non-sensitive, it is possible to provide perfect privacy regardless of the emphasis being on privacy or utility, i.e., PFF and UFF varieties of Flip-flop. However, in real-world scenarios users may prefer more flexibility when classifying POIs, which comes at the cost of privacy loss. As can be seen, utility is largely unaffected by the change in the number of POI classes, which may be due to the fact that the data distortion would largely be impacted by POI density and the ratio of POI types rather than the number of distinct POI types.

The effect of the number of stops: We performed Flip-flop on trajectories with a fixed footprint and duration but with varying numbers of stops to determine the role of the number of stop points on the achieved levels of privacy and utility. Figure 8(c) shows the result of PFF and UFF for trajectories with 1 (lightest circles) to 5 (darkest circles) sensitive stops. As expected, an increase in the number of stops correlates with less utility, since the frequency of changes to the original trajectory grows. However, an increase in the number of sensitive stops correlates with privacy loss as well. This is mainly due to the fact that both variants of Flip-flop have to look for less-sensitive POIs within a smaller regions, which decreases their chance of finding the least-sensitive POI. As can be seen when handling a single stop point, PFF almost guarantees finding a POI that can provide maximum privacy, but with an increase in the number of stops, its search space becomes smaller and the achieved privacy drops to an average of 95% (with standard deviation of 5%) for trajectories with five stops.

The effect of stop duration: To demonstrate the effect of an increase in the stop duration on the overall privacy/utility of an exchanged trajectory, we increased the duration of each stop in trajectories with similar footprint and number of stops ($|\mathcal{S}_s| = 5$). As demonstrated in Figure 8(d), longer stops correspond to a decrease in data utility, since a higher proportion of the overall trajectory is changed in the exchange process, i.e., the overall proportion of stop episodes to the entire trajectory grows. However, when varying the stop duration, the overall privacy level remains relatively the same, which highlights the prominent role of POI density and number of POI types on determining the obtained privacy level. Similarly to the previous experiments, PFF results in comparable levels of utility as UFF, while maintaining higher privacy levels.

5.2 Experimental Setup: Real Data

To further evaluate our approach, we used the Porto taxi trajectory dataset [1] that contains GPS trajectories of 442 taxis from July 2013 to June 2014 in Porto. We focused on trajectories that reside within a (≈8 km × 8 km) area in the business district of Porto. We also focused on trajectories with durations longer than 10 minutes and at least 10 coordinates. This has resulted in 5,694 trajectories with an average length of ≈6.5-km distance and ≈16-minutes duration.



(a) Generic sensitivity. (b) Personalised sensitivity. (c) Semi-personalised sensitivity.

Fig. 9. Error in spatial density distribution estimation, $\mathcal{E}_{density}$.

We retrieved the POIs from OpenStreetMaps and removed the points that cannot offer a meaningful stop, e.g., *fountain* and *bench*, and so on. We then used this list to create stops on each trajectory: For each trajectory, we create three versions with one, two, and three stops, where the total duration of stops in all versions is equal to 1 hour.

Following the evaluation of UFF and PFF in the previous section, we set our default setting to PFF in this section. Finally, to determine the stop sensitivity, we adopt three strategies: For varying number of POI types we (i) categorise the POIs in a *generic* manner and apply it to all trajectories, (ii) for each trajectory create a hierarchy of POI classes to imitate *personalised* privacy settings, or (iii) consider certain POIs as safe (sensitive) for everyone and the sensitivity of the remaining POIs is determined in a personalised manner to create a *semi-personalised* scenario. We considered POIs such as *cafes* as safe for every trajectory and places such as *bar* and *pharmacy* as sensitive for all trajectories. In all scenarios, the POI types are randomly categorised into two to five POI sensitivity classes.

In this section, we evaluate the ability of our algorithm to preserve the spatial and spatio-temporal characteristics of the original data using the following scenarios:

5.2.1 Spatial Density Distribution. Computing the spatial density distribution is useful if an application requires an estimate of traffic density in different areas, e.g., distinguishing traffic hot spots in a city. We first partition the space into square grid cells of equal size and construct a z-order space-filling curve [34] to derive the cell IDs, $c_i \in [1, c_{max}]$. We then map each raw GPS coordinate, $\hat{l}$, to the cell ID that contains it. The spatial density distribution is then expressed as $D_{SD} = \{P(c_1), P(c_2), \ldots, P(c_{max})\}$, where $P(c_i) = |\{\hat{l} \in \hat{L} : Map(\hat{l}) = c_i\}|/|\hat{L}|$ and $\hat{L}$ is the set of all GPS coordinates.

For a given set of original trajectories, $\mathcal{T}$, and their respective exchanged set, $\mathcal{T}^*$, we compute the error introduced by our perturbation process as

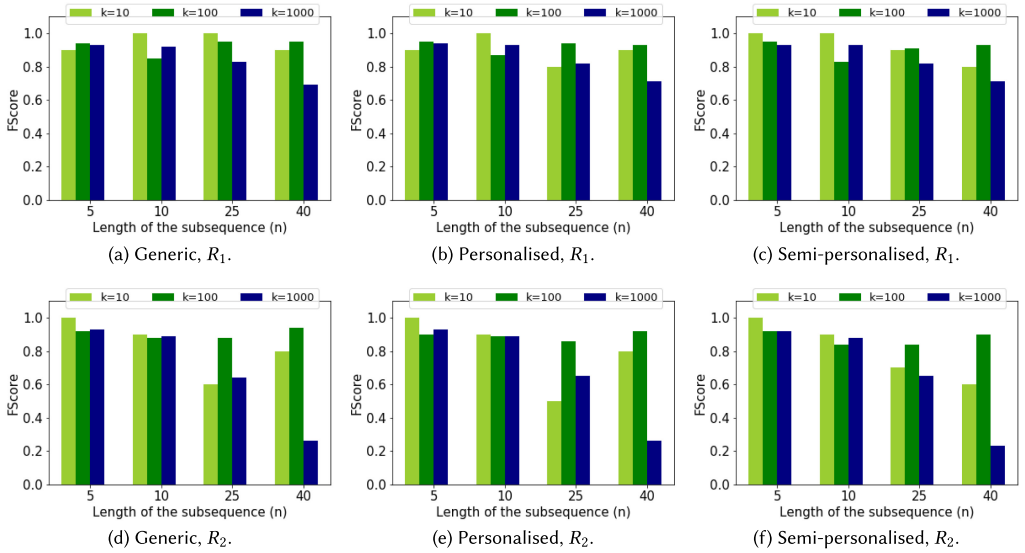
$$\mathcal{E}_{density} = JSD(D_{SD}(\mathcal{T}), D_{SD}(\mathcal{T}^*)) \quad \in [0, ln2],$$

where $JSD(.)$ is the Jensen-Shannon distance. We used two spatial resolutions of our grid cells: R_1 and R_2 correspond to the decomposition of space into 500 m $\times$ 500 m and 250 m $\times$ 250 m grid cells.

Figure 9 shows the density error for the above-mentioned strategies and spatial resolutions. As can be seen, in all scenarios, the error introduced by our perturbation is quite small, while the Semi-personalised strategy introduces the least amount of error. As the number of less-sensitive POIs is generally higher than the other two strategy. We observe that in a more realistic scenario, an increase in the number of POI classes decreases the amount of error, since a larger variety in POI sensitivities leads to a smaller number of replacements in our exchanged trajectories. As expected, the error is higher for the more fine-grained resolution, R_2, since deviations from original trajectories are better captured at this resolution.

Table 3. Utility (Similarity) of Frequent Spatio-temporal Patterns, U_{STP}, for the Top 100 Patterns and the Generic Sensitivity

	$n = 5$				$n = 10$				$n = 25$				$n = 40$			
T_p	2	3	4	5	2	3	4	5	2	3	4	5	2	3	4	5
R_1	0.91	0.94	0.95	0.94	0.82	0.85	0.85	0.85	0.85	0.91	0.94	0.95	0.84	0.92	0.95	0.95
R_2	0.86	0.9	0.89	0.92	0.78	0.89	0.89	0.88	0.78	0.86	0.87	0.88	0.77	0.9	0.95	0.94



(a) Generic, R_1. (b) Personalised, R_1. (c) Semi-personalised, R_1.

(d) Generic, R_2. (e) Personalised, R_2. (f) Semi-personalised, R_2.

Fig. 10. Utility (similarity) of frequent spatio-temporal patterns, U_{STP}, for varying k and n when $T_p = 5$.

5.2.2 Frequent Spatio-temporal Patterns. Frequent spatio-temporal patterns are essential for understanding the traffic flow and general movement behaviour of a population. Similarly to the previous section, we map the raw GPS coordinates to the grid cells; however, this scenario considers the sequence of the grid cells rather than the single points. It is noteworthy that in a mapped trajectory, the same grid cell is usually repeated continuously, which corresponds to the period that it takes for a user to cross that cell. In our dataset, the original and exchanged mapped sequences have an average length of 66.6 ± 39.0 and 66.5 ± 38.0, respectively.

For a set of mapped trajectories, the D_{STP} query outputs the top k frequent sub-sequences with length n in the dataset. We express the utility of our exchanged trajectories compared to the original trajectories as:

$$U_{STP} = F_1(D_{STP}(\mathcal{T}), D_{STP}(\mathcal{T}^*)) \quad \in [0, 1],$$

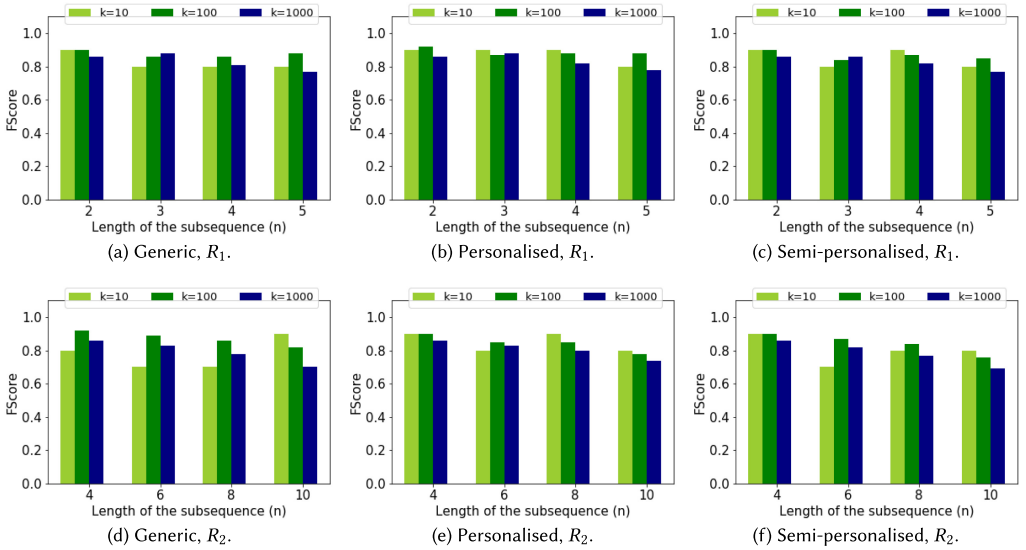
where $F_1(.)$ is the harmonic mean of precision and recall and captures the similarity between the two sets. We vary n between 5 to 40 and k is set to 10, 100, and 1,000 in our experiments.

As can be seen in Table 3, an increase in the number of POI classes, T_p, initially improves the utility in almost all cases, but further increasing this number has a negligible affect on the performance. Hence, we present the rest of our experimental results only for $T_p = 5$.

Figure 10 shows the utility of our exchanged dataset with regard to the frequent spatio-temporal patterns. Using coarse grid cells, i.e., R_1, for all sensitivity strategies, the exchanged dataset exhibits

Table 4. The Length of Mapped Flows in the Original and Exchanged Datasets

	$\mathcal{F}, R_1$	$\mathcal{F}^*, R_1$	$\mathcal{F}, R_2$	$\mathcal{F}^*, R_2$
Mean, std	11.8 ± 5.4	12.8 ± 5.5	20.7 ± 9.1	21.9 ± 9.0



(a) Generic, R_1. (b) Personalised, R_1. (c) Semi-personalised, R_1.

(d) Generic, R_2. (e) Personalised, R_2. (f) Semi-personalised, R_2.

Fig. 11. Utility (similarity) of frequent spatial patterns, U_{SP}, for varying k and n when $T_p = 5$.

high levels of similarity to the original dataset. However, an increase in n, k and spatial resolution results in lower levels of utility. This is expected, since an exchange between the stops and moves episodes in the original trajectories affects the time spent in the cells, which is reflected in the spatio-temporal case with larger sub-sequences. Moreover, while the more personalised strategies toward determining the stop sensitivity have better utility compared to the generic strategy, this difference is not significant.

5.2.3 Frequent Spatial Patterns. This scenario is similar to the previous frequent patterns case except for the fact that the repetition between consecutive timestamps is removed from the sequences to only represent the shifts between the cell IDs regardless of the time it takes to cross them, i.e., the *flow* between the cells is of interest.

Hence, for a set of original mapped flow, $\mathcal{F}$ ($c_i \neq c_{i+1}$ for any given mapped sequence), the utility of frequent spatial patterns, i.e., D_{SP}, in the exchanged flow, $\mathcal{F}^*$ is determined as follows:

$$U_{SP} = F_1(D_{SP}(\mathcal{F}), D_{SP}(\mathcal{F}^*)) \quad \in [0, 1].$$

Since the length of the flow sequences are much shorter than the trajectories (Table 4), we vary n between 2 to 5 for R_1 and between 4 to 10 for R_2. We use similar settings for spatial resolutions and k for this scenario.

As can be seen in Figure 11, the utility of the frequent spatial patterns in our exchanged data is also very high, with an almost-consistent performance for all k and n values. This is because Flip-flop's primary goal is to preserve the footprint of the original trajectory and in case a stop replacement is required, it performs the stop exchange with minimal detour in space. Moreover,

the advantage of the Personalised and Semi-personalised strategies is more pronounced in this case, in particular for smaller sub-sequences.

6 DISCUSSION AND FUTURE DIRECTIONS

Our findings suggest that in regions with a high-POI density, applying PFF results in the best balance between privacy and utility. However, if the POI density in a region is low and utility is paramount, then it is better to adopt UFF. In general, our experiments show that knowing the POI density on a trip may provide individuals with an estimation of how private their trip can be. In other words, if the POI density of a certain trip is known, then the reachable privacy level can be predicted before disclosing the trip. Similarly, given any area with certain POI densities, a service provider may determine if the required data utility can be obtained with regard to individuals' privacy preferences.

Moreover, our experiments in Section 5.2 suggests that a more flexible strategy for determining stop sensitivity preserves the utility better. Such strategy would also make it more difficult for an adversary to make inferences about the individuals' sensitive stop. Nonetheless, we randomly assigned a sensitivity to POI types, which can be further improved by a more realistic strategy.

Currently, our Flip-flop approach classifies non-sensitive POIs based on their *typical duration* and replaces sensitive stops with a POI from a similar category, e.g., a post office is not an appropriate replacement for a bar. In general the definition of stop similarity should be modified according to the underlying application.

In this work, we assume a users' location is determined through a single localization technology, typically through a GPS track. Other localization techniques such as proximity sensing technologies using a mobile phone's WiFi or Bluetooth capability are equally possible. However, if an application has access to several localization technologies simultaneously, then we need to improve our algorithm in a way that it would be able to apply its replacement and displacements strategies uniformly across all available types of location data to ensure private trajectories of high data quality. Moreover, in our experiments, we classified users from those who are willing to share their data and are not concerned with their privacy to those users who are fully concerned about their intermediate stops. We realised that time complexity for Flip-flop increases with stricter sensitivity threshold. The reason is that Flip-flop needs to search more to find a safe alternative for the original sensitive stop. This knowledge can be used by an adversary to obtain an aggregate information about users and their general privacy preferences. Furthermore, an adversary may be able to pinpoint changes in a user's privacy settings, which may be helpful to infer about more sensitive trips. This calls for novel techniques that are able to adjust the perturbation level according to users' preferences while limiting the adversary's ability to (i) make further inferences in the presence of multiple datasets and (ii) differentiate between users.

7 CONCLUSION

We propose an algorithm that adaptively preserves the semantics of a trajectory with regard to the user's privacy preferences. To achieve this, we consider a trajectory as a sequence of stop and move episodes and aim to safeguard intermediate sensitive stop points. Our Flip-flop approach not only exchanges sensitive stops with less-sensitive POIs more efficiently but also results in a high data utility. In summary, we showed that—unlike generic privacy protection methods—a tailored technique results in guaranteed privacy while minimizing utility loss. Our proposed displacement probability estimation in Section 4.2.2, and its corresponding evaluation with different POI densities suggest that knowing the POI density on a trip may provide individuals with an estimation of how private their trip can be. In other words, if the POI density of a certain trip is known, then the reachable privacy level can be predicted before disclosing the trip. Similarly, given any area with

certain POI densities, a service provider may determine whether the required data utility can be obtained with regard to individuals' privacy preferences.

REFERENCES

[1] 2015. Porto taxi trajectory dataset. Retrieved August 9, 2019 from http://www.geolink.pt/ecmlpkdd2015-challenge.

[2] Osman Abul, Francesco Bonchi, and Mirco Nanni. 2008. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 24th IEEE International Conference on Data Engineering*. 376–385.

[3] Helmut Alt and Michael Godau. 1995. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.* 5, 01n02 (1995), 75–91.

[4] Luis Otavio Alvares, Vania Bogorny, Bart Kuijpers, Jose Antonio Fernandes de Macedo, Bart Moelans, and Alejandro Vaisman. 2007. A model for enriching trajectories with semantic geographical information. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*. 22:1–22:8.

[5] S. Madden and J. Gehrke. 2004. Query processing in sensor networks. *IEEE Pervasive Computing* 2, 1 (2004), 46–55. DOI: 10.1109/MPRV.2004.1269131

[6] Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2014. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 251–262.

[7] Thomas Brinkhoff. 2002. A framework for generating network-based moving objects. *GeoInformatica* 6, 2 (2002), 153–180.

[8] Michelle Nicole Burns, Mark Begale, Jennifer Duffecy, Darren Gergle, Chris Karr, Emily Giangrande, and David Mohr. 2011. Harnessing context sensing to develop a mobile intervention for depression. *J. Med. Internet Res.* 13, 3 (2011), e55.

[9] Xin Cao, Gao Cong, and Christian S. Jensen. 2010. Mining significant semantic locations from GPS data. *Proc. VLDB Endow.* 3, 1–2 (2010), 1009–1020.

[10] Basile Chaix, Julie Meline, Scott Duncan, Claire Merrien, Nolla Karusisi, Camille Perchoux, Antoine Lewin, Karima Labadi, and Yan Kestens. 2013. GPS tracking in neighborhood and health studies: A step forward for environmental exposure assessment, a step backward for causal inference? *Health Place* 21 (2013), 46–51. DOI: https://doi.org/10.1016/j.healthplace.2013.01.003

[11] Rui Chen, Gergely Acs, and Claude Castelluccia. 2012. Differentially private sequential data publication via variable-length N-grams. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. 638–649.

[12] Rui Chen, Benjamin C. M. Fung, Bipin C. Desai, and Nériah M. Sossou. 2012. 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*. 213–221.

[13] Yan Dai, Jie Shao, Chengbo Wei, Dongxiang Zhang, and Heng Tao Shen. 2018. Personalized semantic trajectory privacy preservation through trajectory reconstruction. *World Wide Web* 21, 4 (2018), 875–914. DOI: 10.1007/s11280-017-0489-2

[14] Maria Luisa Damiani, Elisa Bertino, and Claudio Silvestri. 2009. Protecting location privacy against spatial inferences: The PROBE approach. In *Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*. 32–41.

[15] Matt Duckham and Lars Kulik. 2005. A formal model of obfuscation and negotiation for location privacy. In *Pervasive Computing*. Lecture Notes in Computer Science, Vol. 3468. 152–170. DOI: https://doi.org/10.1007/11428572_10

[16] Marco Gruteser and Dirk Grunwald. 2003. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*. 31–42. DOI: https://doi.org/10.1145/1066116.1189037

[17] Tanzima Hashem, Lars Kulik, and Rui Zhang. 2013. Countering overlapping rectangle privacy attack for moving knn queries. *Inf. Syst.* 38, 3 (2013), 430–453. https://doi.org/10.1016/j.is.2012.07.001

[18] Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M. Procopiuc, and Divesh Srivastava. 2015. DPT: Differentially private trajectory synthesis using hierarchical reference systems. *Proc. VLDB Endow.* 8, 11 (2015), 1154–1165.

[19] Alex Hern. 2017. Fitness tracking app Strava gives away location of secret US army bases. Retrieved December 12, 2018 from https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases.

[20] Zheng Huo, Xiaofeng Meng, Haibo Hu, and Yi Huang. 2012. You can walk alone: Trajectory privacy-preserving through significant stays protection. In *Database Systems for Advanced Applications*. Springer, 351–366.

[21] Kaifeng Jiang, Dongxu Shao, Stéphane Bressan, Thomas Kister, and Kian-Lee Tan. 2013. Publishing trajectories with differential privacy guarantees. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*. DOI: https://doi.org/10.1145/2484838.2484846

[22] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. 2005. An anonymous communication technique using dummies for location-based services. In *Proceedings of the IEEE International Conference on Pervasive Services*. 88–97.
[23] John Krumm. 2007. Inference attacks on location tracks. In *Proceedings of the Pervasive Computing and Communications (PerCom'07)*. 127–143. DOI: https://doi.org/10.1007/978-3-540-72037-9_8
[24] Byoungyoung Leeand, Jinoh Oh, Hwanjo Yu, and Jong Kim. 2011. Protecting location privacy using location semantics. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1289–1297. DOI: https://doi.org/10.1145/2020408.2020602
[25] Meng Li, Liehuang Zhu, Zijian Zhang, and Rixin Xu. 2017. Achieving differential privacy of trajectory data publishing in participatory sensing. *Inf. Sci.* 400-401 (2017), 1–13. DOI: https://doi.org/10.1016/j.ins.2017.03.015
[26] Tiancheng Li and Ninghui Li. 2009. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 517–526. DOI: https://doi.org/10.1145/1557019.1557079
[27] Qiang Lin, Daqing Zhang, Kay Connelly, Hongbo Ni, Zhiwen Yu, and Xingshe Zhou. 2015. Disorientation detection by mining GPS trajectories for cognitively-impaired elders. *Perv. Mobile Comput.* 19 (2015), 71–85. DOI: https://doi.org/10.1016/j.pmcj.2014.01.003
[28] Yunhao Liu, Yiyang Zhao, Lei Chen, Jian Pei, and Jinsong Han. 2012. Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays. *IEEE Trans. Parallel Distrib. Syst.* 23, 11 (2012), 2138–2149. DOI: https://doi.org/10.1109/TPDS.2011.307
[29] Frank D. McSherry. 2009. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. 19–30. DOI: https://doi.org/10.1145/1559845.1559850
[30] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. 2006. The new casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases*. 763–774.
[31] Anna Monreale, Roberto Trasarti, Chiara Renso, Dino Pedreschi, and Vania Bogorny. 2010. Preserving privacy in semantic-rich trajectories of human mobility. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. 47–54.
[32] Mehmet Ercan Nergiz, Maurizio Atzori, and Yucel Saygin. 2008. Towards trajectory anonymization: A generalization-based approach. In *Proceedings of the ACM SIGSPATIAL 2008 International Workshop on Security and Privacy in GIS and LBS*. 52–61.
[33] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. 2011. On the levy-walk nature of human mobility. *EEE/ACM Trans. Netw.* 19, 3 (2011), 630–643.
[34] Hanan Samet. 2006. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann.
[35] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, and Ryosuke Shibasaki. 2014. Prediction of human emergency behavior and their mobility following large-scale disaster. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, 5–14. DOI: https://doi.org/10.1145/2623330.2623628
[36] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy 1. *Int J Uncertain Fuzz.* 10, 5 (2002), 1–14.
[37] Manolis Terrovitis and Nikos Mamoulis. 2008. Privacy preservation in the publication of trajectories. In *Proceedings of the IEEE 9th International Conference on Mobile Data Management*. 65–72.
[38] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering*, 673–684. DOI: https://doi.org/10.1109/ICDE.2002.994784
[39] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*. 791–800. DOI: https://doi.org/10.1145/1526709.1526816

Received December 2018; revised August 2019; accepted September 2019