Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Προσωποποιημένη Ανωνυμοποίηση βάσεων δεδομένων κινούμενων αντικειμένων μέσω συσταδοποίησης και σύγχυσης**
Τίτλος Διατριβής (Αγγλικά)	**Personalized Anonymization of moving objects databases by clustering and perturbation**
Ονοματεπώνυμο Φοιτητή	**Θεοδοσόπουλος Βασίλειος του Ευαγγέλου**
Αριθμός Μητρώου	**ΜΠΣΠ10019**
Κατεύθυνση	**ΣΥΣΤΗΜΑΤΑ ΥΠΟΣΤΗΡΙΞΗΣ ΑΠΟΦΑΣΕΩΝ**
Επιβλέπων	**Νίκος Πελέκης, Λέκτορας**

Πανεπιστήμιο Πειραιώς-Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών στα
Προηγμένα Συστήματα Πληροφορικής

Ημερομηνία Παράδοσης **23 Οκτωβρίου 2013**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Νίκος Πελέκης Λέκτορας	Γιάννης Θεοδωρίδης Καθηγητής	Γιάννης Σίσκος Καθηγητής

ABSTRACT

The preservation of privacy when publishing spatiotemporal data is a field that is receiving growing attention. However, while more and more services offer personalized privacy options to their users, few algorithms are able to handle such a high degree of personalization effectively, without incurring unnecessary information distortion. In this paper we study the problem of *Personalized (k,δ)-Anonymity*, which builds upon the model of (k,δ)-Anonymity, while allowing for the fact that each user in the system has his own individual privacy and service quality requirements. We examine how well the *Wait For Me* algorithm handles the problem and propose our own algorithm, built specifically to take advantage of users' personalized privacy settings in order to avoid over-anonymization and decrease information distortion. In addition to taking into account personalized (k,δ) requirements, our approach utilizes dataset-aware trajectory segmentation, in order to examine the results of anonymizing a dataset the trajectories of which have been partitioned into sub-trajectories using privacy-aware criteria.

Furthermore, we study the problem of *Bounded Personalized (k,δ)-Anonymity*, where there is a limit to the acceptable information distortion caused by the anonymization. A novel system is introduced whereby trajectories are assessed and the most demanding ones are edited in terms of their (k,δ) requirements, in order to decrease overall information distortion.

Experimental results show the degree to which personalized anonymization achieves lower information loss than non-personalized algorithms, as well as the degree to which trajectory segmentation affects the process. Further results also demonstrate the effects of demandingness-based trajectory editing on satisfying the criteria for bounded anonymity.

ΠΕΡΙΛΗΨΗ

Η προστασία της ιδιωτικότητας όταν δημοσιεύονται χωροχρονικά δεδομένα είναι ένα πεδίο που τυγχάνει αυξανόμενου ενδιαφέροντος. Παρόλα αυτά, ενώ όλο και περισσότερες υπηρεσίες προσφέρουν επιλογές προσωποποίησης στους χρήστες τους, λίγοι αλγόριθμοι μπορούν να χειριστούν έναν τόσο υψηλό βαθμό προσωποποίησης αποτελεσματικά, χωρίς να επισύρουν αχρείαστη αλλοίωση δεδομένων. Σε αυτή την εργασία μελετούμε το πρόβλημα της Προσωποποιημένης (k,δ)-Ανωνυμίας, το οποίο βασίζεται πάνω στο μοντέλο της (k,δ)-Ανωνυμίας, κάνοντας την παραδοχή πως κάθε χρήστης στο σύστημα έχει τις δικές του απαιτήσεις ανωνυμίας και ποιότητας υπηρεσίας. Εξετάζουμε πόσο καλά διαχειρίζεται αυτό το πρόβλημα ο αλγόριθμος Wait For Me και προτείνουμε τον δικό μας αλγόριθμο, δημιουργημένο συγκεκριμένα για να εκμεταλλεύεται τις προσωποποιημένες απαιτήσεις ιδιωτικότητας και ποιότητας υπηρεσίας των χρηστών ώστε να αποφεύγει την υπερ-ανωνυμοποίηση και να μειώνει την αλλοίωση της πληροφορίας των δεδομένων. Εκτός του να λαμβάνει υπόψη τις προσωποποιημένες προτιμήσεις των χρηστών, η προσέγγισή μας χρησιμοποιεί επίσης κατάτμηση τροχιών βασισμένη στην αντίληψη των δεδομένων, με σκοπό να εξετάσει τα αποτελέσματα της ανωνυμοποίησης δεδομένων όπου οι τροχιές έχουν κατατμηθεί σε υπο-τροχιές με κριτήρια που βασίζονται στη γνώση των δεδομένων.

Επιπλέον, μελετούμε το πρόβλημα της Οριοθετημένης Προσωποποιημένης (k,δ)-Ανωνυμίας, όπου υπάρχει ένα όριο στην αποδεκτή αλλοίωση της πληροφορίας που προκαλεί η ανωνυμοποίηση. Παρουσιάζεται ένα νέο σύστημα όπου οι τροχιές αξιολογούνται και οι πιο απαιτητικές υφίστανται επεξεργασία όσο αφορά στις απαιτήσεις (k,δ) τους, με στόχο να μειωθεί η συνολική αλλοίωση πληροφορίας.

Πειραματικά αποτελέσματα δείχνουν το βαθμό στον οποίο η προσωποποιημένη ανωνυμοποίηση επιτυγχάνει χαμηλότερη απώλεια πληροφορίας από τους μη-προσωποποιημένους αλγόριθμους, και επίσης το βαθμό στον οποίο η κατάτμηση τροχιών επηρεάζει τη διαδικασία. Περαιτέρω αποτελέσματα επιδεικνύουν επίσης τις συνέπειες της επεξεργασίας τροχιών με βάση την απαιτητικότητά τους πάνω στο πλαίσιο της ικανοποίησης των κριτηρίων της οριοθετημένης ιδιωτικοποίησης.

Θα ήθελα να ευχαριστήσω την οικογένεια μου, για τη στήριξη που μου παρείχε στη διάρκεια εκπόνησης αυτής της εργασίας. Επίσης, τους Νίκο Πελέκη και Γιάννη Θεοδωρίδη για τις πολύτιμες συμβουλές και τις παρατηρήσεις τους.

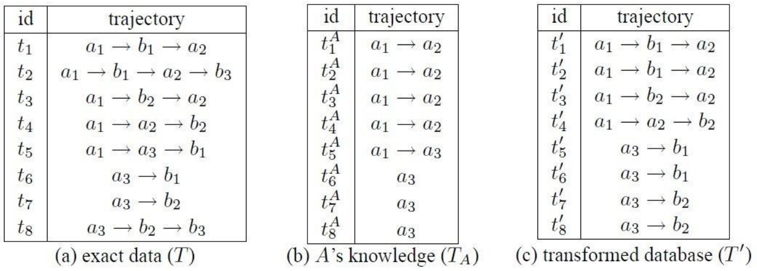
CONTENTS

1. INTRODUCTION

With the rapid development of the information technologies, the advent of mobile computing and the increasing popularity of location-aware services, the volume of mobility data gathered daily by service providers has exploded during the past few years and will very likely continue to do so in the future. Such data on the trajectories of moving objects are analyzed and behavioral patterns extracted from it, so as to support decision-making and strategic planning. Therefore, it is often desired that mobility data is published to facilitate this process.

However the publication of the data creates threats for the privacy of the individuals concerned, because, if combined with other publicly available data, the spatiotemporal traces that users leave behind can reveal their identity as well as other sensitive information about them, such as their place of residence or occupation, their sexual orientation or religious and political beliefs. Thus, it becomes necessary to develop methods providing privacy-preservation in mobility data publishing, where a sanitized version of the original dataset is published, which maintains the maximum possible data utility.

A number of such methods have been proposed so far, most of which adopt the concept of K-anonymity for anonymization, the fundamental principle of which is that every entry of a published database should be indistinguishable from at least K-1 other entries. In [1] and [2] trajectories are grouped into clusters of K members and published not as poly-lines, but as cylindrical volumes which ‘conceal’ the individual trajectories, while in [9] points of trajectories are suppressed so that adversaries with partial knowledge of a trajectory cannot identify it amongst at least K-1 others. A technique which is not based on K-anonymity is [4], where crossing points between trajectories are found or created and then obfuscated in order to decrease an adversary’s chances of successfully following a trajectory.



id	trajectory
t_1	$a_1 \rightarrow b_1 \rightarrow a_2$
t_2	$a_1 \rightarrow b_1 \rightarrow a_2 \rightarrow b_3$
t_3	$a_1 \rightarrow b_2 \rightarrow a_2$
t_4	$a_1 \rightarrow a_2 \rightarrow b_2$
t_5	$a_1 \rightarrow a_3 \rightarrow b_1$
t_6	$a_3 \rightarrow b_1$
t_7	$a_3 \rightarrow b_2$
t_8	$a_3 \rightarrow b_2 \rightarrow b_3$

(a) exact data (T)

id	trajectory
t_1^A	$a_1 \rightarrow a_2$
t_2^A	$a_1 \rightarrow a_2$
t_3^A	$a_1 \rightarrow a_2$
t_4^A	$a_1 \rightarrow a_2$
t_5^A	$a_1 \rightarrow a_3$
t_6^A	a_3
t_7^A	a_3
t_8^A	a_3

(b) A’s knowledge (T_A)

id	trajectory
t'_1	$a_1 \rightarrow b_1 \rightarrow a_2$
t'_2	$a_1 \rightarrow b_1 \rightarrow a_2$
t'_3	$a_1 \rightarrow b_2 \rightarrow a_2$
t'_4	$a_1 \rightarrow a_2 \rightarrow b_2$
t'_5	$a_3 \rightarrow b_1$
t'_6	$a_3 \rightarrow b_1$
t'_7	$a_3 \rightarrow b_2$
t'_8	$a_3 \rightarrow b_2$

(c) transformed database (T')

Figure 1: An example of data suppression w.r.t partial knowledge of adversary A

A significant drawback most of the proposed anonymization methods have is that they operate based on pre-specified privacy requirements, which do not take into consideration the individual users’ preferences and instead assume common, universal settings. This lack of personalization can lead to unnecessary anonymization and data utility loss for users whose privacy requirements are overestimated and to inadequate anonymization and violation of privacy for users whose requirements are underestimated. [4] supports a user-specific maximum perturbation setting and [1], [2] are extensible to use a user-specific radius, but those options are rather limited and do not address the issue adequately. An approach that does offer somewhat more significant personalization is [5], where trajectory-specific privacy requirements are introduced. In contrast, the method we propose uses trajectory-specific values to determine each user’s specific privacy level and service quality requirements, therefore reducing data utility loss and improving service quality.

An additional shortcoming of anonymization methods that use clustering is that they function at the trajectory level. As a result, when dealing with trajectories that are on the whole very different, but have some similar parts, these algorithms fail to recognize these common elements and either assign such trajectories to different clusters or assign them in the same cluster only after considerable spatiotemporal translation. This failure to recognize and make use of similarities between parts of trajectories is counter-intuitive and increases the overall distortion. In our method we have dealt with this problem by utilizing trajectory segmentation in order to discover similar sub-trajectories and use those as the basis of our clustering process.

In this paper we present a method for publishing spatiotemporal trajectory data using personalized (k, δ)-anonymity in order to form trajectory clusters, using the definition of (k, δ)-anonymity as introduced in [1] and [2]. Its novelty compared to existing methods is that it takes into consideration different privacy requirements (k, δ) for each individual, where k dictates the required privacy level and δ functions as a service quality threshold. This personalization of both aspects results in more fine-tuned anonymization that satisfies the requirements of the maximum possible amount of users while incurring the minimum possible information loss. In addition, our method adopts a privacy-aware trajectory segmentation phase, during which trajectories are partitioned into sub-trajectories, which allows the clustering algorithm to discover similarities between them and assign the respective partitions into common clusters, the members of which require no or minimal editing so as to fulfill (k, δ)-anonymity, thus keeping distortion minimal.

Figure 2 illustrates the difference between Wait4Me and our method, assuming a dataset of 5 trajectories, each with its own user-specific privacy (k) requirement. Since W4M uses a single, global value for k, it must use the maximum value of the dataset (k=3) in order to satisfy all users. With that k, and using whole trajectories as operating units, all five trajectories have been assigned into a single cluster, requiring spatial translation for 8 out of 25 total trajectory points. Our method, on the other hand, employs segmentation in order to partition trajectories into similar sub-trajectories, which are then used as our operating unit. The combination of sub-trajectories with the use of user-specific k values means that the same dataset can now accommodate the creation of four clusters instead of one, satisfying every trajectory's privacy requirements without any translation in this case. Data utility is also enhanced, as there is no or little over-anonymization and the finer-made clusters preserve as much of the original information as possible.

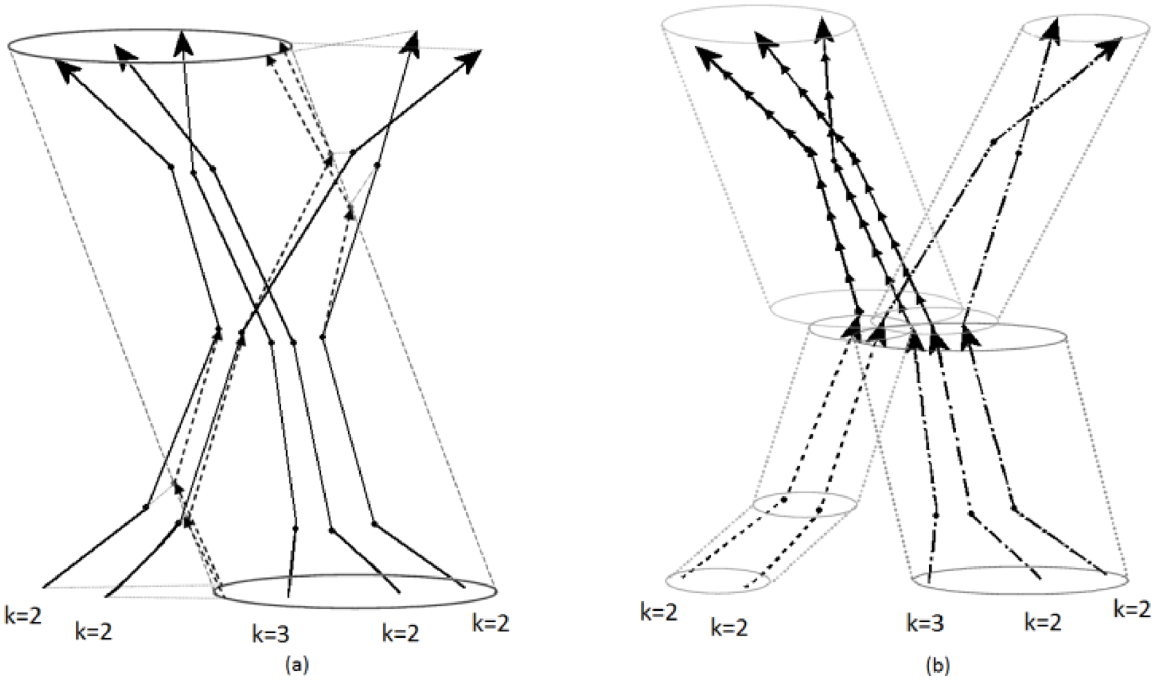


Figure 2: (a) Clustering and translation using W4M with universal (3,δ) values (b) Clustering and translation using PW4M with personalized (k,δ) values and trajectory segmentation

2. RELATED WORK

The methods that have been proposed so far in order to tackle the issue of privacy-reserving mobility data publishing mostly adopt the principle of K-anonymity, which was originally proposed for relational databases. It stipulates that attributes of a table are divided in sensitive attributes, the values of which should be protected and preserved, and quasi-identifiers. In order for a table to be K-anonymized every tuple should be indistinguishable from at least K-1 other tuples, in that their quasi-identifiers should be identical. In order to accomplish this, quasi-identifier attributes can be altered using methods such as generalization or suppression.

In the context of mobility data a dataset is considered K-anonymized if each trajectory in it is indistinguishable from at least K-1 other trajectories. Given the nature of spatiotemporal data, all attributes (x,y,t) are considered both sensitive and quasi-identifiers at the same time. Methods similar to those used for relational data can be employed to achieve anonymization.

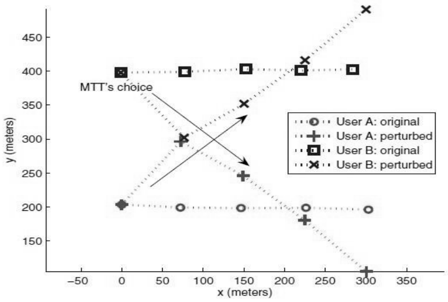


Figure 3: An example of path crossing between two trajectories

Hoh and Gruteser's method [4] is an example of data perturbation with a goal of decreasing an adversary's certainty of correctly identifying a user. To do that, its Path Perturbation algorithm creates fake intersection points between couples of non-intersecting trajectories if they are close enough. The crossing points must be generated within a specific time-window and within a user-specified radius, which indicates the maximum allowable perturbation and desired degree of privacy. A larger radius means increased perturbation and decreased data utility, but also more intersections, and every intersection along a user's trajectory decreases the adversary's chances of successfully following it. Therefore the ability to set a user-specified perturbation radius allows a degree of personalization, but it lacks flexibility since it always causes a trade-off between data utility and privacy. If a large radius is allowed, then privacy is strengthened because the trajectory can intersect a large number of other trajectories, but then the distortion of the trajectory also becomes greater, so the data utility decreases.

Terrovitis and Mamoulis [9] proposed an approach that uses suppression. Trajectories are modeled as sequences of locations where users made transactions and an adversary is assumed to have partial knowledge of users' visited locations and their relative order, therefore an incomplete projection of the dataset. Based on this assumption the algorithm seeks to eliminate the minimum amount of locations from trajectories so that the remaining trajectories are K-anonymous with regards to an adversary's partial knowledge. A greedy algorithm is employed iteratively, under the assumption of multiple adversaries with different projections of the dataset, in order to remove locations with minimal information loss. The approach includes however no element of personalization, since the value of K is universal and application-determined.

Never Walk Alone [2] and its extension Wait 4 Me [1], proposed by Abul et al, follow a clustering-based approach which takes advantage of the inherent uncertainty of a moving object's location introducing the concept of (k, δ)-anonymity. An object's location at a given time is not a point, but a disk of radius δ, and the object could be anywhere inside that, so a trajectory is not a polyline, but a cylinder of consecutive such disks. To achieve K-anonymity, each trajectory is assigned to a group of at least K-1 others using a greedy clustering algorithm. Then the trajectories of each cluster are spatially translated so that they will all lie entirely within the same cylinder (uncertainty area) of radius δ/2.

W4M's main differences to NWA are its usage of the outlier- and time-tolerant EDR as a distance function, instead of the Euclidean distance, during the clustering phase and the use of ST-editing instead of space-translation, during the spatial translation phase. Both NWA and W4M are extensible for user-specific values of radius δ, which offers a limited degree of personalization. That is, a user can adjust his desired level of distortion by setting a maximum radius for his trajectory's uncertainty area, but he does not have any control over his level of privacy, since the value of k is universal and common for all users. It is worth noting that in [10] the authors attempt to show that (k, δ)-anonymity fails to offer trajectory K-anonymity; a conclusion that is in our opinion erroneous, as it is based on an inaccurate modeling of the proposed concept.

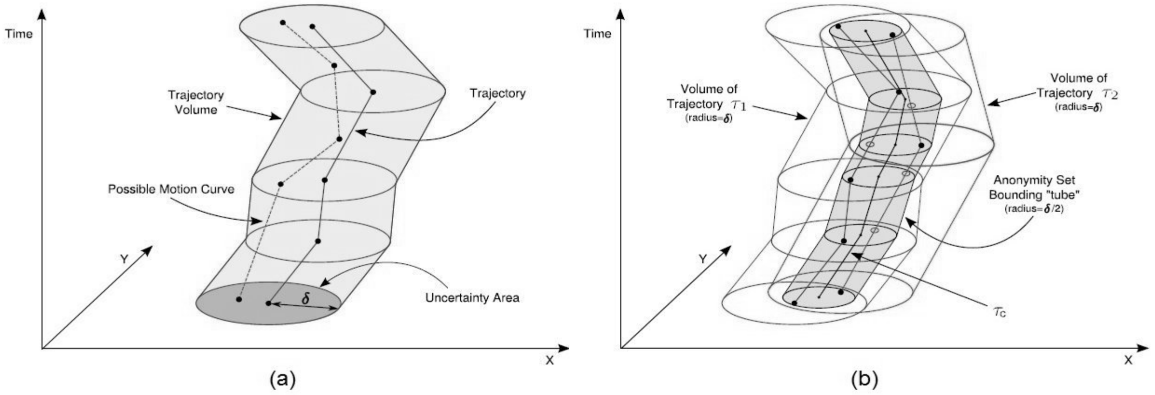
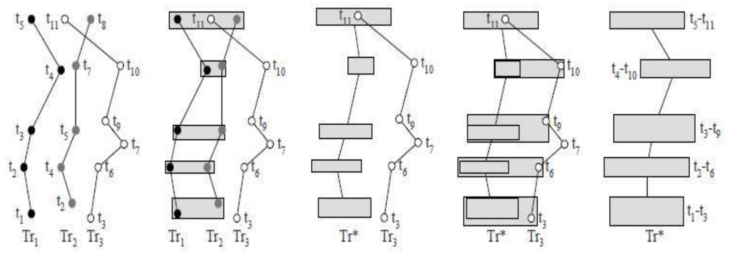

Time
Trajectory Volume
Trajectory
Possible Motion Curve
Uncertainty Area
δ
Y
X
(a)
Time
Volume of Trajectory τ_1 (radius=δ)
Volume of Trajectory τ_2 (radius=δ)
Anonymity Set Bounding "tube" (radius=δ/2)
τ_c
Y
X
(b)

Figure 4: (a) An uncertain trajectory lying inside the cylinder with radius δ, (b) a (2,δ)-anonymity set consisting of two co-localized trajectories

Another clustering-based approach is [5] by Mahdavifar et al, which introduces the idea of non-uniform privacy requirements, whereby each trajectory is associated with its own privacy level indicating the number of trajectories it should be indistinguishable from. Trajectories are first divided into groups depending on their privacy level. Clusters are then created by randomly selecting a centroid and adding to the cluster the trajectories nearest to it, if their EDR distance is lower than a fixed radius, until the max privacy requirement within the cluster is satisfied. If the requirements are not satisfied, groups with lower privacy levels are progressively searched for trajectories to be added to the cluster, until all the privacy requirements have been met. Finally, the trajectories of each cluster are anonymized using a matching point algorithm that generates an anonymized trajectory as the cluster representative. While this approach offers a greater degree of personalization than others, it still leads to a compulsory trade-off between privacy and quality for each user. If a trajectory has a high privacy requirement, it will very likely be part of a large cluster, thus suffering from increased information loss and low data utility, since the user cannot set a 'quality' requirement.

Always Walk with Others [7] is a generalization-based approach, which transforms trajectories into series of anonymized regions, while assuming either an adversary's partial knowledge of a trajectory or full knowledge of it and the desire to disclose sensitive information. To achieve anonymity the algorithm creates groups with representative trajectories and then iteratively adds to them their closest trajectories until they have K members. Every time a trajectory is added to a group, its representative is updated as the sequence of minimum bounding spatiotemporal regions that include itself and the new trajectory. After that step, K points in each anonymized region are randomly selected and connected to points similarly generated in adjacent regions in order to form K new trajectories.



(a) Original trajectories (b) Anonymity group $G = \{Tr_1, Tr_2\}$ (c) Anonymization of G (d) Add Tr_3 into G (e) Anonymization of G

Figure 5: Generalization-based approach: Anonymization Step

Another generalization-based algorithm is PPSG [6] by Monreale et al, operating under the assumption that the adversary has knowledge of the anonymization method, of the existence of a user or of his partial trajectory. The algorithm finds characteristic points of trajectories and applies spatial clustering to them. The centroids of those clusters are then used for Voronoi tessellation of the area covered in the dataset dividing it into cells. Trajectories are made up by segments linking those cells, the thickness of each segment representing the density of trajectories in a cell. If a cell has fewer than K trajectories, then it is merged with an adjacent cell so that it satisfies that requirement, a process facilitated by the use of a prefix tree representation. Both those generalization-based approach offer no personalization capabilities, since their K values are universal.

3. PROBLEM FORMULATION

In this section we present the formal background and description of the problem of Personalized Anonymity and of our solutions for it.

Following the definition adopted by [2] an uncertain trajectory buffer is defined as a cylindrical volume of radius δ centered at an object's expected trajectory.

Definition 1 (Uncertain Trajectory): A trajectory of a moving object is a polyline in three-dimensional space represented as a sequence of spatiotemporal points: (x_1, y_1, t_1), (x_2, y_2, t_2) . . . (x_n, y_n, t_n), $(t_1 < t_2 < \cdot\cdot\cdot < t_n)$. During the time period $[t_i, t_{i+1}]$ the object is assumed to move along a straight line from (x_i, y_i) to (x_{i+1}, y_{i+1}) at a constant speed. Given a trajectory τ between times t_1 and t_n, and an uncertainty threshold δ, the pair (τ\δ) defines an uncertain trajectory. For each point (x, y, t) along τ, its uncertainty area is the horizontal disk (i.e., circle and its interior) with radius δ that is centered at (x, y, t), where (x, y) is the expected location at time $t \in [t_1, t_n]$. The trajectory volume of (τ\δ), denoted Vol(τ\δ) is the union of all such disks for all $t \in [t_1, t_n]$. A possible motion curve of τ is any continuous function $fPMC_τ$: Time→R^2 defined on the interval $[t_1, t_n]$ such that for any $t \in [t_1, t_n]$, the spatiotemporal point $(fPMC_τ(t), t)$ is inside the uncertainty area at time t.

Definition 2 (Co-localized Trajectories): Two trajectories $τ_1$, $τ_2$ defined in $[t_1, t_n]$ are considered co-localized w.r.t. δ, if for each point (x_1, y_1, t) in $τ_1$ and (x_2, y_2, t) in $τ_2$ with $t \in [t_1, t_n]$, it holds that $Dist((x_1, y_1), (x_2, y_2)) \leq δ$, where Dist is the Euclidean distance: $Dist((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. We write $Coloc_δ(τ_1, τ_2)$ omitting the time interval $[t_1, t_n]$.

Definition 3 ((k,δ)-anonymous Set of Trajectories): Given a set of trajectories S, an uncertainty threshold δ and an anonymity threshold k, S is (k,δ)-anonymous if $|S| \geq k$ and $\forall\, τ_i, τ_j \in S$, $Coloc_δ(τ_i, τ_j)$.

A dataset of trajectories D is considered (k,δ)-anonymous if each of its members belongs to a (k,δ)-anonymity set. If D does not meet this requirement, then it must be transformed into a sanitized version, called D^s, which will satisfy the aforementioned condition.

Definition 4 ((k, δ)-anonymity): Given a dataset of trajectories D, an uncertainty threshold δ and an anonymity threshold k, (k, δ)-anonymity is satisfied by transforming D to D^s, such that for each trajectory $τ^s \in D^s$ there exists a (k, δ)-anonymity set $S \subseteq D^s$, $τ^s \in S$, and the distortion between D and D^s is minimized.

One of the possible approaches to the transformation of a dataset to its sanitized version, and the one we are following in this paper, is the spatiotemporal translation of trajectory points. Distortion usually measures the difference between the original and the sanitized data. A trajectory's distortion is defined as the sum of its point-wise distances to its sanitized version, and the total distortion caused by sanitizing the entire database is defined as the aggregation of its individual trajectories' distortion.

Definition 5 (Translation Distortion): Given a trajectory $τ \in D$ defined in the time interval T and its sanitized version $τ^s \in D^s$, the distortion caused by translating τ into $τ^s$ is $TD(τ, τ^s) = \sum_{t \in T} Dist(τ[t], τ^s[t])$. The total distortion caused by translating D into D^s is $TTD(D, D^s) = \sum_{τ \in D} TD(τ, τ^s)$.

The problem introduced in this paper is that of (k, δ)-anonymizing a database of trajectories of moving objects where each object has its own $(k_i, δ_i)$ values, while keeping the distortion caused by the translation minimal.

Problem 1 (Personalized (k, δ)-anonymity): Given a dataset of users D = {u_1, u_2, …, u_n} where u_i = {$τ_i$, k_i, $δ_i$} with $τ_i$ being the trajectory and k_i, $δ_i$ the anonymity preferences of user u_i, find an anonymized version of D called D^s = {τ_1^s, …, τ_m^s}, 1 < m ≤ n, where τ_i^s is a (k_i, $δ_i$)-sanitized version of $τ_i$ and distortion TD(D, D^s) is minimal.

An extension of the above problem is that of (k, δ)-anonymizing a database of trajectories of moving objects where each object has its own (k_i, $δ_i$) values while keeping the translation distortion below a given threshold $Dist_{max}$. A trivial case of the problem is when TD(D,D^s) ≤ $Dist_{max}$, since in that case the solution D^s of Problem 1 is also a solution to this problem. In non-trivial cases, where TD(D,D^s) > $Dist_{max}$, the problem can be solved by relaxing the (k_i,$δ_i$) constraints of members that would cause the highest distortion if anonymized.

Problem 2 (Bounded Personalized (k, δ)-anonymity): Given a dataset of users D = {u_1, u_2, …, u_n} where u_i = {$τ_i$, k_i, $δ_i$} with $τ_i$ being the trajectory and k_i, $δ_i$ the anonymity preferences of user u_i, find an edited version of D called D^b = {u_1^b, u_2^b, …, u_n^b} where u_j^b = {$τ_j$, k_j^b, δ_j^b}, such that there can be found an anonymized version of D^b called D^{bs} = {τ_1^{bs}, …, τ_m^{bs}}}, 1 ≤ l ≤ m, where τ_i^{bs} is a (k_i^b, δ_i^b)-sanitized version of $τ_i$ and distortion TD(D, D^{bs}) ≤ $Dist_{max}$.

4. PERSONALIZED (k,δ)-ANONYMITY

4.1 Baseline Solutions

Given a dataset D with personalized privacy requirements (k_i, δ_i) for each user, we can get a baseline solution to Problem 1 by using standard Never Walk Alone or Wait 4 Me, which use a single, universal value for each of k and δ. In order to satisfy all users' privacy requirements, the maximum k_i and the minimum δ_i of the dataset must be found and their values assigned to the universal k and δ variables respectively. The following algorithm illustrates this simple solution. Trash_max is a variable used by W4M to set the maximum amount of points the algorithm is allowed to trash and not include in the anonymized dataset.

Algorithm 1. Naïve Personalized W4M

```
Input: D, Trash_max
Output: $D^s$
$D^s \leftarrow \emptyset$;
$maxk_i \leftarrow$ GetMaxK(D);
$min\delta_i \leftarrow$ GetMinDelta(D);
$D^s \leftarrow$ Wait4Me(D, $maxk_i$, $min\delta_i$, Trash_max);
return $D^s$;
```

In order to improve this very crude attempt at satisfying personalized (k,δ) values, we propose an approach based on the concept of user-specific privacy requirements, which we name Personalized Wait4Me. PW4M follows the general structure of Wait4Me, consisting of two steps: a Greedy Clustering phase, which has been shown in [1] to have the best effectiveness/efficiency ratio, followed by a Spatiotemporal Translation phase, which uses EDR as a distance function. During the Greedy Clustering phase a pivot is randomly selected, a cluster is formed around it by its k-1 unvisited closest neighbors, and then the unvisited trajectory that is farthest away from previous pivots is selected as a new pivot and the process is repeated until clusters satisfying certain criteria have been created. During the Spatiotemporal Translation phase each cluster formed during the previous phase is transformed into a (k,δ)-anonymity set (as described in section 3)

The most significant difference between PW4M and W4M is that, whereas in W4M a pivot is selected and then invariably grouped along with its k-1 closest neighbors in order to form a cluster, in PW4M each cluster has its own, non-fixed k value. After a pivot is selected, its personalized k value is set as the cluster's k value and the following process is iteratively repeated: the algorithm checks if the cluster's k value is satisfied by its current size. If so, the process ends, the cluster is formed and the next pivot is selected. If the cluster's size does not satisfy its k value, the closest unvisited neighbor of the pivot is added to it, the cluster's k value is updated to be the maximum k found amongst its current members, and the process starts again by checking if the cluster's new size satisfies its updated k requirement. It can easily be seen that this approach results in clusters of non-fixed size ranging between 1 and $maxk_i$. In the same spirit, the spatiotemporal editing phase of PW4M differs to that of W4M in that there is no universal δ applied to all clusters, but each cluster is edited based on its own δ value, which is the minimum δ found amongst its members.

The input required for our algorithm is the database of trajectories and a Trash_max value that bounds the size of trash, which are the outliers suppressed by the clustering algorithm in order to improve the quality of the end result. The personalized anonymity threshold k and uncertainty threshold δ of each trajectory are assumed to be included in the trajectory dataset. The algorithm's output is the personalized (k,δ)-anonymized dataset D^s.

Algorithm 2. Personalized W4M

```
Input: D, Trash_max
Output: $D^s$
$D^s \leftarrow \emptyset$;
$\gamma \leftarrow$ PersonalizedW4M$_{clust}$(D, Trash_max);
for each cluster $C \in \gamma$ do
    $C' \leftarrow \emptyset$;
    let $\tau_c$ be the pivot of C, $\delta_c$ be the $\delta$ value of C;
    for each $\tau \in C$ do
        $\tau' \leftarrow$ STedit($\tau, \tau_c, \delta_c$);
        $C' \leftarrow C' \cup \{\tau'\}$;
    $D^s \leftarrow D^s \cup C'$;
return $D^s$;
```

Algorithm 3 below, named Personalized W4M_clust, shows the exact structure of the clustering step of PWM. It follows the general structure of the respective algorithm of W4M and Greedy Clustering, where W4M is based on.

It iteratively selects pivot trajectories to function as centers of clusters, with pivots being selected at random from amongst the available active trajectories (line 5). A pivot's (k_i,δ_i) values serve as the initial (k,δ) requirements of its candidate cluster (lines 7-8). The algorithm then successively adds to the candidate cluster the nearest unvisited neighbor of the pivot and updates the cluster's k and δ, until the cluster's size is enough to satisfy its k requirement, which equals the maximum k_i value amongst its members (lines 9-13). Once a candidate cluster's k criterion has been met, the candidate is made into an actual cluster, as long as its radius is not larger than a certain limit max_radius (lines 14-17). If a cluster cannot be formed around a pivot, the pivot is deactivated (line 18), so that it will not be used again as a pivot for a candidate cluster -though it remains available as a member of some other cluster- and a new pivot is selected.

Once all possible clusters have been formed, the remaining unassigned trajectories are assigned to the cluster of their closest pivot, on condition that their k_i can be satisfied by the cluster's size (including themselves), their δ_i are not smaller than the cluster's current δ, and their addition will not increase the cluster's radius beyond max_radius (lines 19-22). If a trajectory cannot be added to any cluster without violating a condition, it is moved to the trash (line 23).

If the solution found results in trash with size larger than the Max_Trash threshold, the max_radius constraint is relaxed and the process starts again from the beginning until a solution is achieved that satisfies the trash-size requirement (lines 24-25). As output, the algorithm returns only the clusters formed, excluding the suppressed trajectories implicitly.

Algorithm 3. Personalized W4M$_{clust}$

```
Input: D, Trash_max
Output: $\gamma$
1: initialize(max_radius);
2: repeat
3:     Active $\leftarrow$ D; Clustered $\leftarrow \emptyset$; Pivots $\leftarrow \emptyset$; Trash $\leftarrow \emptyset$;
4:     while Active$\neq$0 do
5:         $\tau_p \leftarrow$ random($\tau$)|$\tau \in$ Active;
```

```
6:        $c_{\tau p}$.size ← 1;
7:        $c_{\tau p}$.k ← $\tau_p$.k;
8:        $c_{\tau p}$.δ ← $\tau_p$.δ;
9:        while ($c_{\tau p}$.k > $c_{\tau p}$.size) do
10:           $c_{\tau p}$←{$\tau_p$}∪{nearest neighbor of $\tau_p$ in D\Clustered};
11:           $c_{\tau p}$.size← $c_{\tau p}$.size + 1;
12:           $c_{\tau p}$.k = max($c_{\tau p}$.k, nearest-neighbor.k);
13:           $c_{\tau p}$.δ = min($c_{\tau p}$.δ, nearest-neighbor.δ);
14:        if $\max_{\tau \in c_{\tau p}}$ Dist($\tau_p$,τ) ≤ max_radius then
15:           Active←Active\$c_{\tau p}$ ;
16:           Clustered←Clustered ∪ $c_{\tau p}$ ;
17:           Pivots←Pivots ∪ {$\tau_p$};
18:        else Active←Active\{$\tau_p$};
19:    for each τ ∈ D\Clustered do
20:        $\tau_p$←$\mathrm{argmin}_{\tau' \in Pivots | c_{\tau' p}.size \geq \tau.k - 1,\ c_{\tau' p}.\delta \leq \tau.\delta}$Dist(τ' ,τ);
21:        if Dist($\tau_p$,τ) ≤ max_radius then
22:           $c_{\tau p}$←$c_{\tau p}$ ∪ {τ};
23:        else Trash←Trash ∪ {τ};
24:    increase(max_radius);
25: until |Trash|≤|$Trash_{max}$|;
26: return {$c_{\tau p}$|$\tau_p$ ∈ Pivots };
```

After forming clusters, each of them is separately processed and transformed into a (k,δ)-anonymity set, with (k, δ) being values specific to the cluster. Here we follow the approach proposed in [1], which achieves that by using the cluster's pivot as reference and editing the other trajectories so that they are co-located with it (see Section 3) and also have the same number of points as the pivot; the only difference being that in [our method] each cluster uses its own value of δ for co-localization instead of a universal value. The co-localization process requires adding or deleting points from trajectories. EDR distance is used for that purpose, which produces a sequence of addition/deletion operations and matches points from the two trajectories into disjoint pairs such that the necessary amount of operations is minimized. The operation sequence dictated by EDR affects both trajectories, but since deleting a point from the pivot can be reversed by adding a corresponding point to the other trajectory, the EDR operations sequence can be translated into a series of actions affecting only the other trajectory, leaving the pivot untouched.

The first step therefore is to get the sequence of operations required for the optimal EDR matching between the pivot and the other trajectory (line 1). The sequence is then parsed (line 4). If a deletion from the pivot is required, then a point is added to trajectory t instead, randomly somewhere within a radius δ/2 from the respective point of the pivot s_j, while the new point's temporal coordinate is that of s_j (lines 5-7). If there is a match between two points s_i and t_i and no deletion or addition is required (lines 9-12), then t_i undergoes the shortest possible spatial translation so that it moves within a radius δ/2 from s_i, if it is not already within that radius, and t_i's temporal coordinate becomes that of s_j. Finally, if a deletion of a point from trajectory t is required, the algorithm does nothing, implicitly deleting that point without any further actions (lines 13-14).

Algorithm 4. STedit

Input: δ and two trajectories t and s (the pivot),

```
Output: t’
edit ← EDR_op_sequence(t,s);
t’ ← ⟨ ⟩;
i,j ← 1;
for each op ∈ edit do
    if op=remove($s_j$) then
        t’.append(⟨random_point_in_circle($s_j$.x, $s_j$.y, δ/2) ,$s_j$.t⟩);
        j ← j+1;
    else
    if op=match($t_i$ ,$s_j$) then
        t’.append(⟨transl($t_i$.x, $t_i$.y,$s_j$.x, $s_j$.y, δ/2), $s_j$.t⟩);
        i ← i+1;
        j ← j+1;
    else // last case: remove($t_i$)
        i ← i+1;
return t’;
```

Two optimization methods are presented in [1], which extend W4M in order to facilitate scaling to large databases in terms of execution time. The first one is the introduction of a linear complexity, time-tolerant distance function, called LSTD, as a replacement for EDR and the second one is the addition of an extra step in the standard W4M algorithm, where the database is divided into chunks, each containing similar trajectories, and then each chunk is treated as a separate database. PW4M can very easily be extended to include these two optimization methods, therefore being also scalable to large databases.

4.2 Personalized (k,δ)-Anonymity with Trajectory Segmentation

In this section we introduce our novel approach to the problem of Personalized (k,δ)-Anonymity, which aims to improve upon the baseline solutions presented in the previous section by implementing trajectory segmentation, in order to increase clustering effectiveness and decrease distortion levels. A shortcoming of the baseline solutions which is common in all clustering methods is that they use the trajectory as the smallest working unit. As a result, when two trajectories have some similar parts, but are on the whole significantly different, the algorithm is unable to discover and make use of those similar elements, leading to an overall increased distortion during clustering. In order to deal with this issue, our approach includes a trajectory segmentation phase, where trajectories are partitioned into sub-trajectories according to a set of privacy-aware criteria. It is these sub-trajectories that are then used as input for the anonymization stage of the algorithm that follows. While this segmentation incurs extra computational cost, it offers a distinct advantage in that it facilitates the discovery of patterns shared between parts of trajectories which are otherwise significantly different on the whole.

Algorithm 5 presents the generic two-step concept that we propose. Given a dataset of trajectories D, the algorithm first applies a trajectory segmentation algorithm on it to produce the dataset of partitioned sub-trajectories D^p. As a second step that dataset is then processed by a personalized anonymization algorithm, the result of which is then returned to the user, an anonymized sub-trajectory-based dataset D^{ps}.

Algorithm 5. Trajectory Anonymization with Segmentation

```
Input: D
Output: $D^{ps}$
$D^{p}$ ← DatasetSegmentation(D);
$D^{ps}$ ← DatasetAnonymization($D^{p}$);
return $D^{ps}$;
```

The concept is very generic, in that it does not strictly define the algorithms used for either of the two steps. Any segmentation and any anonymization algorithm can be used, although sophisticated segmentation algorithms which take neighboring trajectories into account when partitioning will yield better results.

For our approach, Personalized (k,δ)-Anonymity with Trajectory Segmentation (PW4M+TR), we have chosen to use the one put forward in [7]. In contrast to other approaches, it does not require pre-processing, it does not use trajectory simplification, it takes into account the temporal aspect of trajectories and it makes no assumptions with regards to trajectories' patterns. Another point of difference is that, while other methods treat each trajectory individually without taking into consideration the rest of the dataset, our chosen method partitions trajectories based on the *representativeness* of each trajectory segment, which indicates the number of other trajectories' segments in proximity. The main advantage achieved by using representativeness-based partitioning is that the resulting sub-trajectories are internally homogenous with regards to the number of other trajectories co-localized with them, since each sub-trajectory is selected in such a way that its various segments have a similar number of other sub-trajectories' segments near them. This makes the assignment of sub-trajectories to clusters easier and as a result the clusters formed will cause less distortion when spatiotemporally translated. An example of using PW4M on a dataset partitioned using this method can be seen in Figure 1, where segments that are similar in terms of the number and identity of their neighboring trajectories have been identified and grouped into sub-trajectories, which have in turn been assigned to the appropriate clusters.

5. BOUNDED PERSONALIZED (k,δ)-ANONYMITY

In order to deal with the problem of Bounded Personalized (k,δ)-anonymity, where there is a requirement to keep anonymization distortion below a given threshold, we expand on the methods presented in the previous sections by introducing dataset assessment and requirement relaxation. Since distortion is caused by spatiotemporal translation, a naïve approach would be to anonymize a dataset once, identify the trajectories which have undergone the most translation and edit them. However, unless a trajectory is the most demanding in its neighborhood, it is translated not because it has strict (k,δ) values itself, but so that a cluster can be formed which will satisfy the criteria of its most demanding neighbor. Therefore, in order to decrease overall distortion it is the most demanding trajectories that must be identified and edited.

Since high k and low δ values make a trajectory more demanding and increase the difficulty of assigning it to a cluster, the following formula provides a simple metric for a trajectory's *demandingness*.

Definition 6 (Trajectory Demandingness): Given a trajectory τ ∊ D with privacy requirements (k,δ), its demandingness is τ.dem = τ.k/τ.δ.

However the above formula does not take into account at all the rest of the dataset and the location of the trajectory in regards to other trajectories. This makes it potentially inaccurate, since a trajectory with demanding (k,δ) values might be very close to a number of other trajectories, which makes it easy to assign to a cluster, while a trajectory with undemanding (k,δ) might have no or few neighbors to be clustered with. Therefore we need a variable that includes information on a trajectory's neighborhood.

For that purpose we have chosen to use the *representativeness* value of a trajectory, as defined in [7], which we define as the average *representativeness* value of all the segments of a trajectory, functioning as an estimation of a trajectory's number of neighbors. Since representativeness shows nearby trajectories, we can use it to estimate the degree to which a trajectory's k value can be satisfied by its neighborhood. This k-satisfaction degree is then used instead of k, in order to calculate its dataset-aware demandingness.

Definition 7 (Dataset-aware Trajectory Demandingness): Given a trajectory τ ∊ D with privacy requirements (k,δ) and representativeness r, its demandingness is τ.dem = (τ.k/τ.r)/τ.δ.

Once each trajectory's demandingness has been calculated, they are sorted by its value and a percentage of the most demanding ones are then edited. The goal of editing them is to make them equally demanding to the most demanding trajectory of those that will not be edited, which we call the *threshold trajectory*. Since a trajectory's representativeness (r) is fixed, demandingness reduction can be achieved by editing k and δ. First, we examine k to see if it's satisfied by the trajectory's r. If it is, then it does not need to be edited. If it is not, then we decrease it until its new value either equals r, it results in a demandingness score equal to the threshold's score, or is equal to 2 (since k=1 violates anonymity). Once editing k is finished, or if k is already satisfied by r, we proceed to increase δ, until its new value results in a demandingness score equal to the threshold's score.

The goal of editing a number of the dataset's trajectories is to decrease anonymization distortion. In [1] and [2] anonymization distortion is affected not only by the total spatiotemporal distortion of anonymized trajectories, but by sending some trajectories to the trash, with each trashed point treated as causing distortion equal to the maximum point-wise translation that occurred during the anonymization. Something similar is needed for trajectories which have their (k,δ) requirements edited. In contrast to trashing a trajectory, which is an absolute action that always produces the same result, editing a trajectory is an action that can be performed to different degrees. Therefore, we need a measure that indicates how much or little a trajectory has been edited. For the purpose we define *edit cost* as the ratio of the demandingness difference between itself and the threshold trajectory's to the demandingness difference between the most demanding trajectory and the threshold, that is the ratio of editing magnitude required for the particular trajectory compared to the editing magnitude required for the most demanding one.

Definition 8 (Trajectory Edit Cost): Given the trajectories $τ$, $τ_{thr}$ and $τ_{max} \in D$, with $τ_{thr}$ being the threshold trajectory and $τ_{max}.dem \geq τ_i.dem\ \forall\ τ_i \in D$, its edit cost is $τ.editcost = (τ.dem-τ_{thr}.dem) / (τ_{max}.dem-τ_{thr}.dem)$.

Once anonymization has been completed, the distortion caused can be calculated. In our approach we follow a similar logic to the one used in [1]. The total distortion is given by the sum of total spatiotemporal translation, the distortion caused by trashed points and the distortion caused by editing trajectories that were not trashed. Each edited trajectory causes distortion equal to the number of its points multiplied by the maximum dataset translation, multiplied by edit_fc, which indicates how significant is the act of editing compared to trashing a trajectory, multiplied by the trajectory's edit cost, which indicates how significantly the trajectory was edited compared to the maximally edited one. .

Definition 9 (Edited Trajectory Distortion): Given an edited trajectory $τ \in D|τ \notin Trash$, with $τ.n$ the number of its points, $τ.editcost$ its edit cost, edit_fc the editing significant factor and $Ω$ the maximum translation occurring during the anonymization, the trajectory's contribution to the overall distortion cost is $τ.dist = τ.n * Ω * edit\_fc * τ.editcost$.

Algorithm 6 below shows the generic concept we propose for tackling the Bounded (k-δ)-Anonymity problem. With the trajectory database and a distortion threshold given as input, the data are first assessed in order to calculate each trajectory's demandingness score and edit cost (lines 1-2) and the number of trajectories that will be edited in the first attempt is initialized (line 3). Based on the demandingness scores previously calculated, the (k, δ) values of the most demanding trajectories are edited, with Edit_size determining the amount of editable trajectories (line 5). The edited dataset is then anonymized (line 6) and the resulting distortion calculated, using the edit_costs previously calculated in order to determine the distortion caused by editing each trajectory (line 7). If the total distortion is below the maximum threshold, the algorithm ends and the anonymized dataset returned, otherwise the number of trajectories to be edited is increased (line 8), and the editing and anonymization phases are repeated, this cycle continuing until the distortion requirement is satisfied or the entire dataset has been edited.

Algorithm 6. Bounded Anonymization

```
Input: D, Dist_max
Output: $D_b$
1: dem_scores ←DatasetAssessment(D);
2: edit_costs ←CalculateEditCosts(dem_scores);
3: initialize(edit_size);
4: repeat
5:     $D_e$ ←DatasetEdit(D, dem_scores, edit_size);
6:     $D_b$ ←DatasetAnonymization($D_e$);
7:     dist ←CalculateDistortion($D_b$, edit_costs);
8:     update(edit_size);
9: until (dist ≤ $Dist_{max}$ || edit_size ≥ |D|);
10: return $D_b$;
```

Since the distortion caused by the anonymization of a dataset is heavily dependent on the original data and the dataset's privacy/quality requirements, it is possible that there will be combinations of strict distortion requirements and very demanding datasets that prohibit the discovery of a solution.

Algorithm 7 below describes in detail our approach to the generic method presented above, called Bounded Personalized W4M, including the integration of the assessment and editing steps with the clustering algorithm presented in the previous section. The required input is the database of trajectories, which we assume to already include information on the representativeness value of each trajectory, calculated using [7], as well as the thresholds for maximum allowed trashed trajectories and maximum allowed distortion. It is worth noting that the method is valid for datasets of both whole trajectories and segmented sub-trajectories. Therefore, the same algorithm can be used in combination either with PW4M or with PW4M+TR.

Since providing a sensible absolute number as distortion threshold would require a difficult a-priori estimation from the user running the algorithm, we have chosen to express $Dist_{max}$ as the desired improvement percentage over the distortion caused by anonymizing an unedited dataset. Therefore, giving a value of 0.1 would indicate that the desired $Dist_{max}$ is 10% smaller than the distortion caused when running the algorithm without any editing.

The first step is to calculate the score of each trajectory and find the maximum among them (lines 2-5). The scores are an estimation of the difficulty to anonymize trajectories and used to compare trajectories to each other. After calculating the scores the trajectories are sorted, to facilitate the next steps (line 6). Edit_size, which indicates the portion of the dataset's trajectories that will be edited, is initialized to 0, so that the algorithm runs the first time without any editing (line 7). When there are trajectories to be edited, the highest-scoring one that is not included in them is set as the dataset's threshold. The goal of the editing process will be to edit the 'costliest' trajectories so that their score will become equal to the threshold trajectory's score (lines 8-9).

Starting with the highest-scoring trajectory and continuing until Edit_Size has been reached, the cost of each trajectory is calculated, which indicates the ratio of editing required in order to equalize the current trajectory's cost to the threshold's cost compared to the editing required in order to do the same for the costliest trajectory. It follows that editing costs' values range from 0 to 1. Once the editing cost is calculated, if the trajectory's k value is higher than its representativeness, it is decreased until one of the following happens: it reaches the minimum allowed value of 2, the new k value is the lowest possible value that does not make the trajectory's score lower than the threshold score or it becomes equal to the trajectory's representativeness value, which means there are enough neighboring trajectories to cover its k requirement (lines 15-16). Next, the trajectory's delta is increased until the trajectory score becomes equal to the threshold score (line 17). The trajectory is then marked as 'edited', the edit-counter is updated and the next-highest-ranking trajectory selected (lines 18-20).

After the editing phase is complete the edited dataset D is given as input to the clustering algorithm, which produces an anonymized dataset D' (line 21). The distortion caused by the anonymization is then calculated (lines 23-31). If the total distortion is higher than the given threshold, the portion of the dataset that is marked for editing is increased (line 31) and the editing process starts again after resetting the dataset to its former state. When the first execution of the clustering algorithm is complete, without any editing, the resulting distortion is saved as 'initial distortion' and Edit_size is initialized to 1, which means 1% of the dataset will be edited for the next iteration of the algorithm (lines 32-34). For all subsequent iterations, the improvement caused by the previous editing is compared against the best improvement up to that point, and if it is better, the Edit-size value that resulted in it stored as 'optimal edit size'; then it is increased by 1 in preparation for the next round of editing (lines 35-39). The algorithm ends when a solution has been found that achieves the required distortion reduction or no solution has been found by editing 99% of the dataset, in which case the algorithm does one final round of editing using the optimal size before ending (lines 40-43).

Algorithm 7. Bounded Personalized W4M

```
Input: D, Trash_max, Edit_max, Dist_impr
Output: $D_b$
max_score ← 0;
for each τ ∈ D do
    τ.score ← (τ.k/τ.r)/τ.δ;
    max_score ← max(τ.score, max_score);
SortByScore(D);
Edit_size ← 0;
$τ_{thres}$ ← (|D|- Edit_size)-th highest scoring trajectory;
score_thres ← $τ_{thres}$.score;
repeat
    ResetTrajectories(D);
    Edited←Ø; Trashed←Ø; Dist ← 0; Edit_count ← 0;
    τ ← highest scoring trajectory;
    while Edit_count < Edit_size do
        τ.cost←(τ.score-score_thres)/(max_score-score_thres);
        if (τ.k > τ.r) then
            τ.k ← max(2, τ.r, ⌈score_thres*τ.r*τ.δ⌉);
        τ.δ ← (τ.k/τ.r)/score_thres;
        Edited← Edited ∪ {τ};
        Edit_count ← Edit_count + 1;
        τ ← next trajectory;
    D’ ← Personalized W4M(D, $Trash_{max}$);
    max_trans← 0;
    for each τ ∈ D’ do;
        for each point ∈ τ do
            Dist ← Dist + point.trans;
            max_trans ← max(point.trans, max_trans);
    for each τ ∈ D|τ ∉ D’ do
        Trashed← Trashed ∪ {τ};
        Dist ← Dist + τ.points_n * max_trans;
    for each τ ∈ Edited \ Trashed do
        Dist ← Dist+τ.points_n*max_trans*τ.cost*edit_mdf;
    if (Edited = Ø) then
        Dist_init ← Dist; best_impr←0;
        Edit_size ← 1; opt_edit_size←0; final_edit←0;
    else
        improvement ← 1 – (Dist/Dist_init)
        if (improvement > best_impr) then
            opt_edit_size ← Edit_size;
        Edit_size←Edit_size+1;
    if (Edit_size = 100) then
        Edit_size←opt_edit_size;
        final_edit←1;
until (Dist_impr≤impr || (final=1 && Edit_size>opt_size))
return D';
```

6. EXPERIMENTS

In this section, we evaluate the effectiveness of our Personalized (k-δ)-Anonymity algorithm PW4M. We describe the experimental data and environment in Section 6.1. We make a base comparison between *Wait For Me* and our method in section 6.2, while we briefly discuss the effects of parameter values in Section 6.3.In section 6.4 we examine the results of using W4M and PW4M with our without having first partitioned the trajectories of the dataset into sub-trajectories using dataset-aware criteria. In section 6.5 we examine the results of using trajectory editing to relax demanding trajectories' requirements and decrease anonymization distortion.

6.1 Experimental Setting

We use one synthetic and one real and dataset to evaluate the performance of the examined algorithms. The synthetic dataset called Synth contains 25 trajectories (1.392 points) of moving vehicles covering an area of 91 km^2 during a 30-minute period. The trucks_revised dataset is real, containing 1100 trajectories (94.000 points) of trucks covering an area of 2.500 km^2 during a 40-day period.

6.2 Base Comparison

We first compare *Wait For Me* to PW4M, in order to prove the validity of our personalized our approach. The dataset used for this experiment is trucks_rev, with randomly generated (k,δ) requirements for each trajectory, k ∊ [2,25], δ ∊ [500,1000]. In order to satisfy all users, W4M's universal (k,δ) values are set to 25 and 500 respectively. 'PW4M base' is a version of our algorithm that finds the maxk_i/minδ_i values in the dataset and uses them for all trajectories, ignoring their individual requirements, essentially replicating the way W4M works. PW4M does not take universal (k,δ) values as input either, it parses each trajectory's specific (k_i,δ_i) requirements from the dataset and uses them throughout the process.

Table 1: Comparison between W4M, PW4M (base) and PW4M anonymizing the trucks dataset with the same parameters (k=25, δ=500)

Algorithm	*Discernibility*	*Created Points*	*Deleted Points*	*Mean Spatial Transl.*	*Mean Temp Transl.*	*Total Distortion (x10^6)*
W4M	27500	22522	2145	8110.54	392529	37699
PW4M (base)	27500	22522	2145	8110.54	392529	37699
PW4M	26920	25932	437	7914.42	283915	27461

Table 1 shows the results of running the experiment previously described. We observe that the base version of PW4M unsurprisingly performs exactly as W4M. The 'standard' version of PW4M, on the other hand, produces a very different result. The dataset's discernibility is slightly decreased, meaning there is a slight drop in data quality, but total distortion is also significantly decreased, indicating a substantial reduction of information distortion, due to the more effective assignment of trajectories to clusters.

6.3 Effects of (k,δ) variation

In this section we examine the effects of using varying combination of (k,δ) values in regards to the total information distortion caused by the anonymization. For this experiment we use PW4M with the trucks_rev dataset. Each trajectory's (k,δ) requirements are randomly generated, with k ∈ [2,k_{max}], δ ∈ [δ_{min}, δ_{max}]. The δ_{max} variable has been set to 1% of the dataset bounding rectangle's diameter. The k_{max} and δ_{min} variables change with each iteration, with δ_{min}'s value expressed as a percentage of δ_{max}.

Table 2 shows the total distortion caused by PW4M for different combinations of (k_{max},δ_{min}), while Figure 6 provides a visual representation of the results. We observe that the distortion curve is not monotone in regards to either variable. The effect of varying δ_{min} however seems to be negligible compared to that of k_{max}, which significantly affected the results.

Table 2: Total Distortion caused by PW4M for different (*k_{max},δ_{min}*) combinations

δ_{min}/δ_{max}	*10%*	*20%*	*30%*	*40%*	*50%*
k_{max}=5	18289	18275	18254	18243	18233
k_{max}=10	28610	28256	28316	28264	28192
k_{max}=15	30137	30103	30148	30011	29993
k_{max}=25	49081	49051	49022	48999	48973
k_{max}=35	26255	26225	26196	26166	26143
k_{max}=50	46488	46408	46380	46350	46345

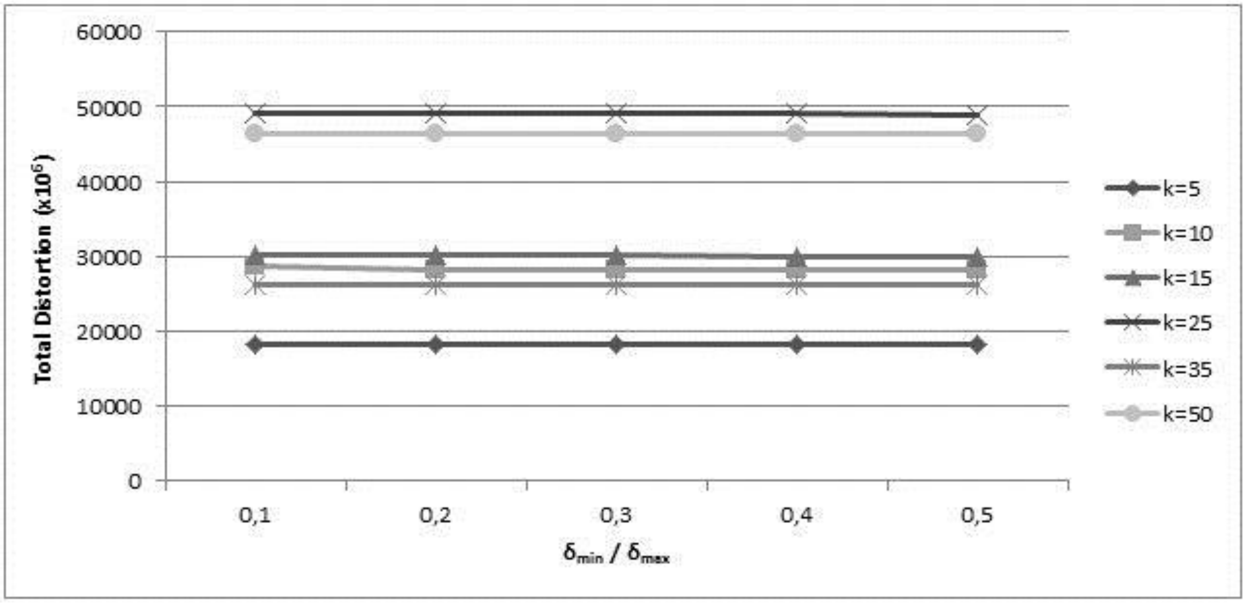


Figure 6: Distribution of total distortion for different combinations of (*k_{max},δ_{min}*)

6.4 Effects of Trajectory Partitioning

In this section we examine the effect of trajectory partitioning, by comparing the results of applying PW4M on two different versions of the Synth dataset, one of them containing the trajectories as they originally were and one containing the same trajectories, but each partitioned into sub-trajectories. We also use W4M on the dataset, as a comparison basis. The synthetic dataset is constructed so that there are three trajectory groups, with trajectories within each group being close and similar to each other. Group A and group B start in relative proximity, while group C starts at a distance to them, but all three groups converge to the same area mid-way during the simulation, and follow a similar path from then on. Simple partitioning is applied to the trajectories, breaking them down in two sub-trajectories each, with the partition happening within the converging area. The values for (k,δ) were randomly generated with k ∈ [2, 5] and δ ∈ [100,1000]. The radius within which two points are considered co-localized was set to 500 for all x, y and t, while Trash_max was set to 0, in order to avoid any distortion caused by trashing trajectories.

Table 3: Comparison between W4M, PW4M and PW4M+TR anonymization of the Synth dataset

Algorithm	*Trajectories*	*Clusters*	*Created Points*	*Deleted Points*	*Mean Spatial Translation*	*Mean Temp Translation*	*Discernibility*	*Total Distortion (x10^{3})*
W4M	25	5	32	39	620	1.1	125	865
PW4M	25	5	38	30	441	0.9	125	615
PW4M+TR	50	11	119	50	340	1.4	254	475

Table 3 shows the results of applying to the dataset first basic (k,δ)-anonymity (W4M), then personalized (k,δ)-anonymity over the original data (PW4M) and finally personalized (k,δ)-anonymity over segmented trajectories (PW4M+TR). We can see here that segmentation resulted in much better results. Not only data quality improved by 104% due to the higher number of clusters, but data distortion decreased by 45% when using both personalized anonymity and segmentation, compared to 29% that was the improvement of using just personalized anonymity. These results indicate that trajectory segmentation has a substantial effect and significantly improves anonymization by increasing data quality and decreasing distortion, thereby validating our approach with PW4M+TR.

6.5 Effects of Trajectory Editing

In this section we examine the effects of trajectory editing based on the algorithms outlined in section 5. Firstly, we apply the Bounded PW4M algorithm to two different versions of the trucks_rev dataset, using a different range of randomly assigned (k,δ) values, in order to examine the effect of edit_size on the final result and how privacy requirements can influence it. Secondly, in order to examine the effects of trajectory editing on datasets of whole trajectories and on datasets consisting of segmented sub-trajectories, we apply Bounded PW4M and Bounded PW4M+TR to the synthetic dataset. In both experiments, each dataset is processed multiple times, each time with a different percentage of the total amount of trajectories being edited.

Firstly, we apply the PW4M+bounded algorithm to two different versions of the trucks_rev dataset, using a different range of randomly assigned (k,δ) values, in order to examine the effect of edit_size on the final result and how privacy requirements can influence it. Secondly, in order to examine the effects of trajectory editing on datasets of whole trajectories and on datasets consisting of segmented sub-trajectories, we apply PW4M and PW4M+TR to the synthetic dataset. In both experiments, each dataset is processed multiple times, each time with a different percentage of the total amount of trajectories being edited.

Figure 7 illustrates the effects of editing varying percentages of the dataset each time, for two versions of the same dataset, each using a different range of (k,δ) values. We observe that more demanding (k,δ) values do not necessarily result in higher distortion. Furthermore, not only distortion changes in a non-monotone way as edit_size increases, but we also observe edit-size values that can actually increase it. That is due to the fact that each edited trajectory incurs a distortion penalty, so this penalty grows proportionately to the edit-size. However, the distribution of demanding trajectories across the clusters and the distribution of (k,δ) values in the dataset significantly influence the degree to which relaxing additional trajectories' requirements affect the clustering and anonymization phases. Therefore, higher percentage of edited trajectories does not guarantee decreased distortion, indicating that there exists an ‘optimal’ edit-size value, where distortion is the minimum possible. That value, however, appears to depend both on the dataset and on the ‘edit_factor’, which measures the distortion cost of editing a trajectory compared to trashing it. This unpredictability makes the application of a heuristic method for the purpose of smart and efficient optimal edit-size identification difficult, necessitating a brute-force approach of multiple iterations with progressively increasing edit-size in order to find it.

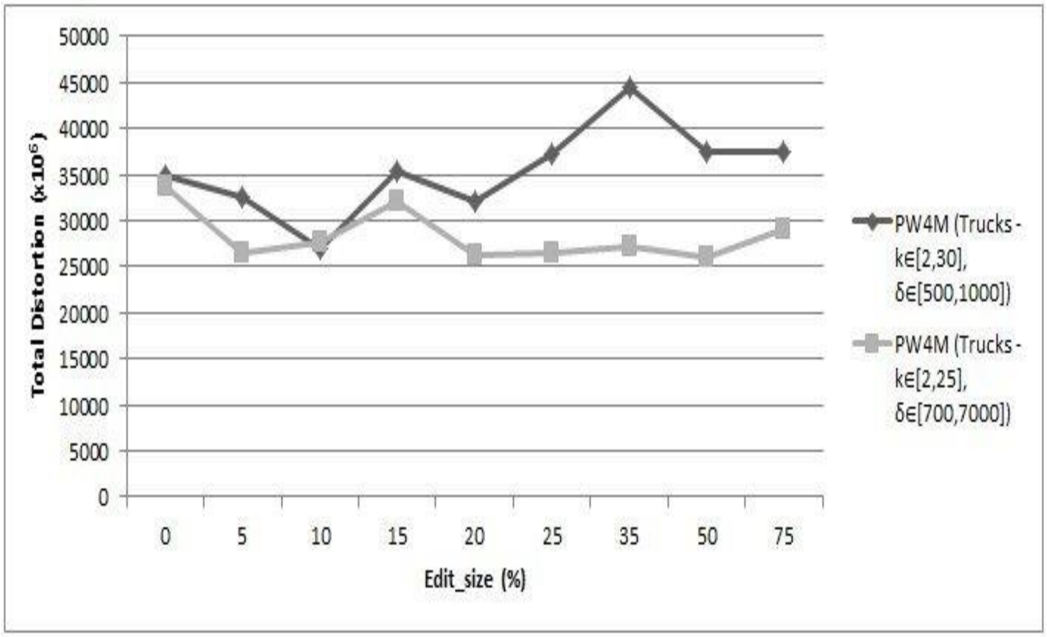


Figure 7: Distortion distribution for varying edit-size values

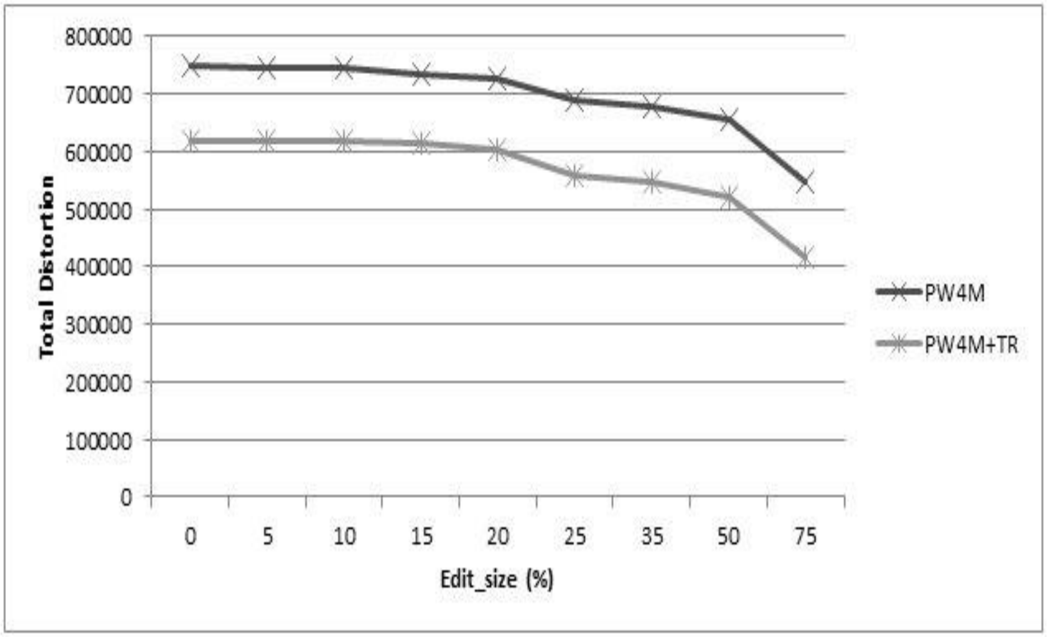


Figure 8: Comparison between trajectory editing applied to a dataset consisting of whole or segmented trajectories

Figure 8 illustrates the difference between applying trajectory editing to whole and segmented trajectories. We observe that the curves of PW4M+TR+bounded and PW4M+bounded are very similar, indicating that trajectory editing has the same effect on datasets of either whole or segmented trajectories. It is also worth noting that, in contrast to the previous experiment, the distortion curves here are monotone in regards to Edit-size, meaning that unless there was an a threshold limiting the amount of trajectories that could be edited, we could get optimal results by setting Edit_size to the maximum possible value.

7. CONCLUSIONS

In this paper, we have proposed a novel approach to anonymizing trajectories, Personalized (k,δ)-Anonymity, using user-specific privacy requirements. Based on this framework, we have developed the trajectory clustering algorithm PW4M, which takes advantage of user-specific (k,δ) requirements in order to assign trajectories to clusters of minimal size, so as to avoid over-anonymization, increase data quality and decrease distortion. Expanding upon that framework, we made use of dataset-aware Trajectory Segmentation, in order to further improve our approach's effectiveness, by partitioning trajectories to sub-trajectories that are more easily assignable to clusters. Additionally, we examined the concept of Bounded (k,δ)-Anonymity, whereby there is a threshold to the acceptable distortion caused by the anonymization process, and proposed methods for Trajectory Assessment and Trajectory Editing, so as to achieve that goal by relaxing the requirements of the most demanding trajectories without editing the spatiotemporal data.

To show the effectiveness of our methods, we have performed experiments using one synthetic and one real dataset: car movement data over Milan and truck movement data over Athens respectively. Our personalized anonymity approach has been shown to significantly increase the overall quality of the anonymized datasets, while it has also been demonstrated that trajectory segmentation can improve data quality even further. Experimental results also show that our trajectory assessment and editing algorithms perform very well towards the goal of decreasing data distortion without altering the trajectories' spatiotemporal information itself.

Overall, we believe that we have provided a novel approach in mobility data anonymization. Data analysts are able to preserve the quality of anonymized datasets taking advantage of user-specific privacy requirements combined with methods such as segmentation and trajectory editing. We also believe that there is a number of points, such as sensitivity to (k,δ) values distribution, replacement of greedy clustering with a more sophisticated clustering method, sensitivity to segmentation method and alternative trajectory assessment and editing methods, which warrant further study in order to expand and improve upon the framework presented here.

8. REFERENCES

[1] Abul, O., Bonchi, F., & Nanni, M. (2010). Anonymization of moving objects databases by clustering and perturbation. *Information Systems*, *35*(8), 884-910.

[2] Abul, O., Bonchi, F., & Nanni, M. (2008, April). Never walk alone: Uncertainty for anonymity in moving objects databases. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on* (pp. 376-385). IEEE.

[3] Chow, C. Y., & Mokbel, M. F. (2011). Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter*, *13*(1), 19-29.

[4] Hoh, B., & Gruteser, M. (2005, September). Protecting location privacy through path confusion. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on* (pp. 194-205). IEEE.

[5] Mahdavifar, S., Abadi, M., Kahani, M., & Mahdikhani, H. (2012). A clustering-based approach for personalized privacy preserving publication of moving object trajectory data. In *Network and System Security* (pp. 149-165). Springer Berlin Heidelberg.

[6] Monreale, A., Andrienko, G. L., Andrienko, N. V., Giannotti, F., Pedreschi, D., Rinzivillo, S., & Wrobel, S. (2010). Movement Data Anonymity through Generalization. *Transactions on Data Privacy*, *3*(2), 91-121.

[7] Nergiz, M. E., Atzori, M., & Saygin, Y. (2008, November). Towards trajectory anonymization: a generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS* (pp. 52-61). ACM.

[8] Panagiotakis, C., Pelekis, N., Kopanakis, I., Ramasso, E., & Theodoridis, Y. (2012). Segmentation and sampling of moving object trajectories based on representativeness. *Knowledge and Data Engineering, IEEE Transactions on*, *24*(7), 1328-1343.

[9] Terrovitis, M., & Mamoulis, N. (2008, April). Privacy preservation in the publication of trajectories. In *Mobile Data Management, 2008. MDM'08. 9th International Conference on* (pp. 65-72). IEEE.

[10] Trujillo-Rasua, R., & Domingo-Ferrer, J. (2012). On the privacy offered by (k, δ)-anonymity. *Information Systems*.