algorithms

MDPI

Article

Storage Efficient Trajectory Clustering and *k*-NN for Robust Privacy Preserving Spatio-Temporal Databases

Elias Dritsas *, Andreas Kanavos, Maria Trigka, Spyros Sioutas and Athanasios Tsakalidis

Computer Engineering and Informatics Department, University of Patras, 26504 Patras, Greece; kanavos@ceid.upatras.gr (A.K.); trigka@ceid.upatras.gr (M.T.); sioutas@ceid.upatras.gr (S.S.); tsak@ceid.upatras.gr (A.T.)

* Correspondence: eldritsas@gmail.com; Tel.: +30-2610-996959

Received: 30 October 2019; Accepted: 7 December 2019; Published: 11 December 2019

check for updates

Abstract: The need to store massive volumes of spatio-temporal data has become a difficult task as GPS capabilities and wireless communication technologies have become prevalent to modern mobile devices. As a result, massive trajectory data are produced, incurring expensive costs for storage, transmission, as well as query processing. A number of algorithms for compressing trajectory data have been proposed in order to overcome these difficulties. These algorithms try to reduce the size of trajectory data, while preserving the quality of the information. In the context of this research work, we focus on both the privacy preservation and storage problem of spatio-temporal databases. To alleviate this issue, we propose an efficient framework for trajectories representation, entitled DUST (DUal-based Spatio-temporal Trajectory), by which a raw trajectory is split into a number of linear sub-trajectories which are subjected to dual transformation that formulates the representatives of each linear component of initial trajectory; thus, the compressed trajectory achieves compression ratio equal to $M : 1$. To our knowledge, we are the first to study and address *k*-NN queries on nonlinear moving object trajectories that are represented in dual dimensional space. Additionally, the proposed approach is expected to reinforce the privacy protection of such data. Specifically, even in case that an intruder has access to the dual points of trajectory data and try to reproduce the native points that fit a specific component of the initial trajectory, the identity of the mobile object will remain secure with high probability. In this way, the privacy of the *k*-anonymity method is reinforced. Through experiments on real spatial datasets, we evaluate the robustness of the new approach and compare it with the one studied in our previous work.

Keywords: Hough transformation; *k*-anonymity; privacy preservation; trajectories compression

1. Introduction

The research area of moving object databases has become an emerging technological discipline, and has consequently gained a lot of interest during the last decade due to the development of ubiquitous location-aware devices, such as PDAs (Personal Digital Assistant), mobile phones, GPS-enabled (Global Positioning System) mobile devices, and RFID (Radio Frequency Identification), or road-side sensors. The technological achievements and advances in sensing and communication/networking, along with the innovative technological design features (thin and light) of computing devices and the development of embedded systems have enabled the recording of a large volume of spatio-temporal data. Mobile object trajectories are among the wide variety of spatio-temporal data that are especially important to scientists. Actually, they help them in discovering movement patterns (individual or group) and knowledge which, in recent literature, have been established as trajectory or mobility mining [1]. Also, the technology of databases is evolving to support the querying and representation of the trajectory of moving objects (e.g., humans, animals, vehicles, natural phenomena). Hence, the main parts of trajectory data-mining include pre-processing, data management, query processing, trajectory data-mining tasks, and privacy protection [2].

Real-life applications, such as the analysis of traffic congestion, intelligent transportation, animal immigration habits analysis, cellular communications, military applications, structural and environmental monitoring, disaster/rescue management, as well as remediation, Geographic Information Systems (GIS), Location-Based Services (LBS), and other domains have increased the interest in the area of trajectory data-mining and efficient management of spatio-temporal data.

It should be noted that the explosive growth of social media has produced large-scale mobility datasets whose publication puts people's personal lives at severe risk. Indeed, users get used to sharing their most-visited or potentially sensitive locations, such as their home, workplace, and holiday locations that are easy to obtain through social media. Nowadays, the amount of spatio-temporal data has been growing exponentially. Therefore, there is an urgent need to develop efficient methods for storing and managing this large amount of information. A plethora of studies have been conducted for handling mobile objects' trajectory data. More precisely, several of them attempt to reduce the storage size [3–5], while others investigate the privacy preservation of trajectory data [6,7]. Nowadays, not only are storage-efficient spatio-temporal transformation schemes needed, but also secure querying on large-scale spatio-temporal data [8]. An accurate capture of a moving object trajectory usually needs a high sampling rate to collect its location data. Thus, massive trajectory data will be generated, which is difficult to fit into the memory for utilizing data-mining algorithms. A common idea is to compress the trajectory data to reduce the storage requirements while maintaining the utility of the trajectory. In the context of this work, we present the storage efficiency of dual methods and experiment on data from the SMaRT system, through which the data of moving object trajectories are generated and used as input to our methods in order to evaluate the security level they offer. As already stated, the privacy of the *k*-anonymity method recommended in [9] is reinforced. More specifically, we summarize the main contributions of our paper as follows:

1. We compare the proposed methods on addressing *k*-NN queries on moving objects' trajectories data, which are stored both in dual and native dimensional space. Our implementation shows that the innovative method of Dual Transformation constitutes a practical solution that can provide secure *k*-NN queries.
2. We conduct an extensive experimental evaluation that studies various scenarios that can affect the vulnerability of the *k*-NN queries and proceed to a comparative analysis of the underlying methods. We prove the efficiency of our solution using real data drawn from SMaRT.
3. We recall two protocols for Pseudonyms Recovery and Registration with the aim of reinforcing the individuals' privacy in the released data. An individual cannot be re-linked to specific users with a high degree of certainty, as it is described in Section 3.7.

The rest of this paper is organized as follows: In Section 2, previous related works are presented in relation to our approach. The following are described in Section 3: (a) the dual transformation methods used; (b) the problem definition; (c) the problem formulation (d) the privacy-preserving analysis; (e) the experimental environment, and source of datasets. Section 4 presents the graphical outcomes gathered from experiments, while Section 5 evaluates experimental results in relation to the pros and cons of the proposed methods. Finally, Section 6 records the conclusions in terms of the studied problem and future directions of this work.

2. Related Work

In this section, we review existing related works in the domain of secure querying on spatio-temporal databases. Our discussion includes privacy-preserving approaches for trajectory-based queries.

In recent years, trajectory databases have constituted an important research area that has received a lot of interest. Most researchers have focused on the querying of moving objects and their trajectory. The so-called trajectory-based queries are also gaining much interest. The queries based on trajectory data require the knowledge of the whole, or at least a part of the mobile objects' trajectory to be processed. Such queries may provide useful information about an object's average speed, travelled distance, and so forth. In [8], three common mechanisms in privacy-preserving trajectory publishing are described. Generalization and suppression are the most common ones used to implement the k-anonymity. However, the main drawback of these mechanisms is that they suffer from a high possibility of information loss—thus, perturbation techniques based on randomization (e.g., adding noise) may be utilized as an alternative.

Actually, the problem of secure querying on spatio-temporal data in combination with k-anonymity has gained much attraction among researchers. Indeed, authors in [10] describe the historical k-anonymity based on each mobile user's trajectory data history, known as Personal History Locations (PHL). According to PHL anonymity, a user, U is camouflaged by $k-1$ users whose PHLs have a common part with its own, rendering him/her indistinguishable among them. Privacy preservation is enforced as the generalization method has been applied. More specifically, by trying to preserve historical k-anonymity, authors increased the uncertainty related to the user's real location data at the time of the query by modifying the spatio-temporal information of the query. More precisely, in [11], by employing the k^l anonymity privacy model, authors ensure that an intruder, who has knowledge of any sub-trajectory TS of size l of a user's trajectory T^j, cannot distinguish their one among $k-1$ trajectories that protect them with probability, based on TS, at most $\frac{1}{k}$.

In a more recent work [9], the authors investigated the privacy-preserving problem based on real spatio-temporal data. That paper employed the k-anonymity method and formed the anonymity set based on motion vectors with the aim of executing secure spatial k-NN queries. More specifically, the problem of k-anonymity from a dimensionality perspective and the impact of used dimensions on the vulnerability of suggested methods was investigated. The experiments presented the effectiveness of the proposed method, such as the clustering under particular attributes combination, and observed that it benefited from attributes suppression during the k-anonymity set computation. Authors in [12] suggested a novel spatio-temporal Mysql ReTrieval framework based on the MySQL and PostgreSQL database management system. In the context of that work, authors employed Hough-X transformation so as to evaluate the efficiency of range queries on nonlinear two-dimensional trajectories of mobile objects. Indeed, they demonstrated that the Hough-X dual approach, in combination with the range-tree variant, was quite efficient.

Generally, the trajectory of a mobile user is non-linear. However, it can be approximated by a discrete number of linear sub-trajectories with the use of a trajectory segmentation application. Each partition is represented by a line segment between two consecutive partition points, and is expected to provide an effective and efficient way to obtain insights into motion characteristics and behavioral preferences of mobile objects. Our approach performs low-rate sampling and considers linear interpolation between successive sampled points, where each line segment represents the continuous moving of the object during sampled points. The duality transformation of line segments operates as a pre-processing step and aims at increasing the security level and reinforcing the privacy of k-NN queries, which is the main subject of this work. Also, we have in our disposal linear components of the initial trajectory, as well as storage of the first and last spatial point in order to represent that line along with the dual representative, that is, the Hough-X (and/or Hough-Y) dual points. Lastly, this step will turn out to be useful from a storage perspective in Big Data applications, and will render the proposed methods a strong candidate for efficient querying on massive data, in combination with the appropriate indexing method.

3. Materials and Methods

3.1. Dual Transform for Moving Objects

In general, the geometric dual transform maps a hyper-plane h from R^m to a point in R^m, and vice versa. In this section, we briefly present how the duality transformation operates in a one-dimensional case. A line from the plane (t, y) or (t, x) is mapped to a point on the dual plane (see Figure 1).

1. **Hough-X**: The equation $y(t) = ut + a$ is mapped to a point (u, a), where axes u, a represent the slope (that is, velocity) and intercept of an object's trajectory, respectively. Thus, we get the dual point (u, a), the so-called Hough-X transform.
2. **Hough-Y**: The equation $y(t) = ut + a$ is rewritten as $t = \frac{1}{u}y - \frac{a}{u}$, a different dual representation, the so-called Hough-Y transform. The point in the dual plane is represented as (b, c), where $b = \frac{-a}{u}$ (the intersection with the line $y = 0$) and $c = \frac{1}{u}$.

It is worth mentioning that the Hough-X transform cannot represent vertical lines, while horizontal lines cannot be represented using the Hough-Y transform. Nonetheless, both transforms are valid, since in our setting, velocity is bounded by $[u_{min}, u_{max}]$, and thus lines have a minimum and maximum slope.

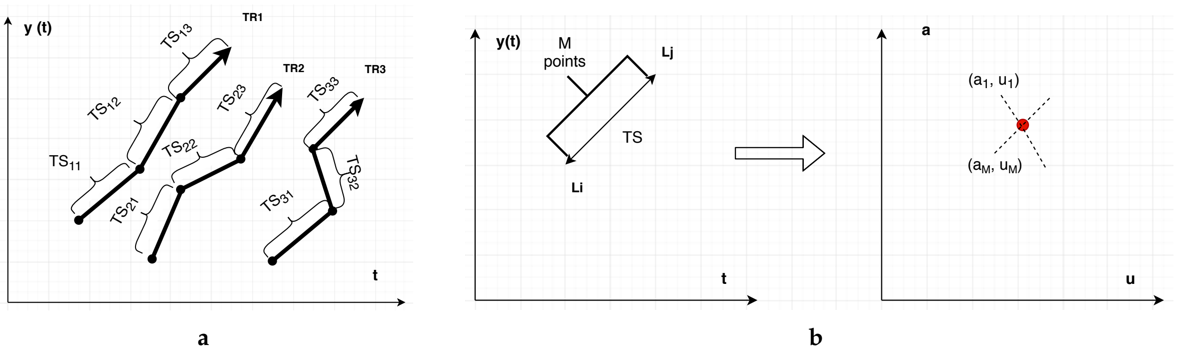


Figure 1. An overview of trajectory segmentation and Hough-X transformation for a linear trajectory segment (TS), which consists of M points. The dual points of M points in TS are the same, for example, $a_1 = \ldots = a_M, u_1 = \ldots = u_M$, where (**a**) shows the y(t) line and (**b**) shows the Hough-X points.

3.2. kNN Classification and Clustering in Dual Space

Here, we consider points in dual space $\mathcal{P}$. Given two dual points dp_1 and dp_2, we define as $dist(dp_1, dp_2)$ the distance between dp_1 and dp_2 in $\mathcal{P}$. In the context of this work, we utilize the Euclidean distance metric, which is defined as

$$dist(dp_1, dp_2) = \sqrt{\sum_{i=1}^{p} \left(dp_1[i] - dp_2[i] \right)^2},$$

where $dp_1[i]$, $dp_2[i]$ denote the values of dp_1, dp_2 along the i dimension in $\mathcal{P}$. For example, in Hough-X space, the distance between the dual points $dp_1 = (u_1, a_1), dp_2 = (u_2, a_2)$ is computed as $dist(dp_1, dp_2) = \sqrt{(u_1 - u_2)^2 + (a_1 - a_2)^2}$.

Definition 1. *DukNN: Given a dual point dp, a data-set of dual points Y and an integer k, the k nearest neighbors of dp from Y, denoted as $DukNN(dp, Y)$, is a set of k points from Y such that $\forall l \in DukNN(dp, Y)$ and $\forall q \in \{Y - DukNN(dp, Y)\}$, $dist(l, dp) < dist(q, dp)$.*

Definition 2. *DukNN Classification: Given a dual point dp, a training dual points data-set Y, and a set of classes Cl_Y where the dual points of Y belong, the classification process produces a pair (dp,cl_{dp}), where cl_{dp} is the majority class to which dp belongs.*

Definition 3. *Clustering: Given a finite data-set of dual points $\mathcal{DP} = \{dp_1, dp_2, \ldots, dp_N\}$ in R^p, and number of clusters K, the clustering procedure produces K partitions of $\mathcal{DP}$ such that among all K partitions (clusters) $C_1, C_2, \ldots, C_K$ find one that minimizes $\arg\min_{C_1, C_2, \ldots, C_K = \mathcal{P}} \sum_{c=1}^{K} \sum_{dp \in C_c} \left\| dp - \frac{1}{|C_c|} \sum_{dp_j \in C_c} dp_j \right\|^2$ where $|C_c|$ is the number of dual points in cluster C_c.*

Note that the aforementioned dual methods act as a feature extraction technique. More specifically, they extract the dual point of each of the x, y coordinates of a mobile user trajectory. The k nearest neighbors algorithm is then applied on dual points features and allowed to return dual points, whose distance from the query dual point is less than the distance from the rest of the training dual points. Considering the Hough-X transformation of attribute x or y, the search area is a circle with the center being the query point and a radius such that k nearest neighbors exist. If we assume Hough-X of (x, y) attributes, the k nearest neighbor search area is four-dimensional (u_x, a_x, u_y, a_y) with complex hypercube geometry.

3.3. Problem Definition

Here, we consider a database that records the location information of mobile objects in the two-dimensional space on a finite area. Also, we assume that objects move with small velocities that lie in the range $[u_{min}, u_{max}]$ starting from a specific location at a specific time-stamp and which move along a non-linear trajectory. In order to be able to store and handle queries in an efficient way, a mobile object's trajectory is approximated with a series of linear ones, as depicted in Figure 2.

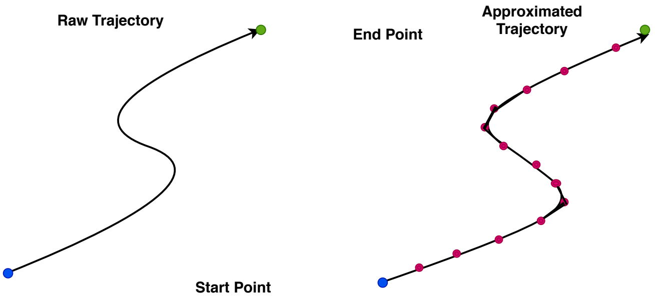


Figure 2. A raw trajectory approximation with a discrete number of R linear sub-trajectories. In the dual dimensional space, each one is represented as a dual point—for example, the linear sub-trajectory $[l(t_0), l(t_1)]$ is represented as a dual point dp_1, and the linear sub-trajectory $[l(t_1), l(t_2)]$ is represented as a dual point dp_2.

Definition 4. *A linear trajectory is a straight line that an object keeps track of, starting from a location $l(t_0) = [x_0, y_0]$ at time t_0. Then, its location for $t > t_0$ will be $l(t) = [x(t), y(t)]$, or $l(t) = [x_0 + u_x(t - t_0), y_0 + u_y(t - t_0)]$, where $u = (u_x, u_y)$ is the object's velocity in each plane [12].*

Definition 5. *A trajectory partition or sub-trajectory segment is a line segment $L_i L_j$, where for $i < j$, both points belong to the same trajectory and are connected in order to form a partition denoted by TS_i [13].*

Definition 6. *Characteristic points are the points where the trajectory changes rapidly.*

Definition 7. *The dual points array constitutes a set containing points of a trajectory that are represented in the dual space.*

Definition 8. *A compressed trajectory path is a subset of the trajectory's points that indicate a significant change in the motion characteristics, that is, the speed or direction of a moving object.*

Definition 9. *Given a trajectory T of size $|T|$ and a compressed trajectory T_c of T with size $|T_c|$, the Compression Ratio (CR) is* $\frac{|T|}{|T_c|}$.

Authors in [14] claim that the compression ratio constitutes a common metric for evaluating the effectiveness of compression algorithms that can accurately reflect the change of a trajectory's data size. It is influenced by the original signal's data-sampling rate, as well as the quantization accuracy.

3.4. Problem Formulation

In the context of this study, the problem of privacy preservation when dealing with spatio-temporal databases goes one step further, and is related to the work [9]. The spatio-temporal data is the location data of a number of mobile users along with the time-stamp of each position, as shown in Table 1. Through the SMaRT system, we have in our disposal offline trajectory data that give us information about Hough-X, as well as Hough-Y of spatial data (x, y). Hence, for each database record per time-stamp, that is, the mobile user trajectory point, we can consider the values of four attributes (x, y, θ, u) (as in Table 1) along with the values of an additional eight attributes' $(U_x, a_x, U_y, a_y, b_x, w_x, b_y, w_y)$ (as in Table 2).

Table 1. An overview of an original spatio-temporal database.

ObjId	Timestamp	TimeToNextPoint	x	y	Angle	Velocity
1	2013-03-09 10:00:01	0	21,082	56,436	1.23	0
1	2013-03-09 10:00:04	3	21,099	56,432	1.16	4.5
1	2013-03-09 10:00:11	7	21,221	56,484	1.51	14.6
1	2013-03-09 10:00:19	8	21,331	56,524	1.95	11.3
1	2013-03-09 10:00:21	2	21,402	56,495	0	29.5
2	2013-03-09 10:00:03	0	35,587	59,829	−2.76	0
2	2013-03-09 10:00:08	5	35,568	59,782	2.94	7.8
2	2013-03-09 10:00:16	8	35,580	59,723	−2.07	5.8
2	2013-03-09 10:00:25	9	35,530	59,668	−1.52	6.4
2	2013-03-09 10:00:34	9	35,476	59,671	−2.85	4.6

Table 2. An overview of the transformed spatio-temporal database.

ObjId	Timestamp	U_x	a_x	U_y	a_y	b_x	w_x	b_y	w_y
1	2013-03-09 10:00:01	4.37	22,242,219.9	1.03	4,800,692.9	0.23	−5,093,637.76	0.97	−4,645,833.3
1	2013-03-09 10:00:04	13.4	22,242,156.2	5.83	4,800,651.2	0.075	−1,659,862.40	0.17	−82,3641.2
1	2013-03-09 10:00:11	10.58	22,242,287.4	3.79	4,800,713.7	0.0946	−2,103,289.59	0.26	−1,267,515.2
1	2013-03-09 10:00:19	27.3	22,242,427.4	11	4,800,762	0.04	−814,740.93	0.09	−436,432.91
1	2013-03-09 10:00:21	27.3	22,242,427.4	11	4,800,762	0.04	814,740.9	0.09	−436,432.91
2	2013-03-09 10:00:03	2.92	22,256,723.4	7.32	4,804,052.4	0.3425	−7,622,165.55	0.14	−656,291.3
2	2013-03-09 10:00:08	1.15	22,256,709.8	5.75	4,803,996	0.87	−19,353,660.69	0.17	−835,477.56
2	2013-03-09 10:00:16	4.27	22,256,692.6	4.64	4,803,941.2	0.23	−5,216,411.92	0.22	−1,034,341.51
2	2013-03-09 10:00:25	4.6	22,256,639.5	0.23	4,803,925.9	0.22	−4,826,741.21	4.29	−20,588,283.27
2	2013-03-09 10:00:34	1.5	22,256,625.5	5.2	4,803,925.8	0.67	−14,837,750.3	0.19	−923,831.89

So, we have chosen to anonymize dual point attributes by employing the k-NN method, which enables us to form the k-anonymity set of each mobile object per time-stamp, as depicted in Table 3. The data anonymization is handled both as a clustering and a no-clustering problem. In both approaches, the anonymity set is formed again by the k nearest neighbors IDS. For each mobile user i and per time-stamp l, we compute its k nearest neighbors IDS and keep them in a vector with form $knns_{il} = [id_{il1} id_{il2} \ldots id_{ilk}]$ for $l = 1, 2, \ldots, L$. In Table 3, an example of such sets for N mobile users' dual points is presented. For each user, we measure the number of the k nearest neighbors dual points that remained the same from one time-stamp to another.

Table 3. Dual points k-anonymity sets for N mobile users in $L = 5$ time-stamps.

Objid	Time Moment	knns Indexes
1	1	$[id_{111}, id_{112}, \ldots, id_{11k}]$
1	2	$[id_{121}, id_{122}, \ldots, id_{12k}]$
1	3	$[id_{131}, id_{132}, \ldots, id_{13k}]$
1	4	$[id_{141}, id_{142}, \ldots, id_{14k}]$
1	5	$[id_{151}, id_{152}, \ldots, id_{15k}]$
2	1	$[id_{211}, id_{212}, \ldots, id_{21k}]$
2	2	$[id_{221}, id_{222}, \ldots, id_{22k}]$
2	3	$[id_{231}, id_{232}, \ldots, id_{23k}]$
2	4	$[id_{241}, id_{242}, \ldots, id_{24k}]$
2	5	$[id_{251}, id_{252}, \ldots, id_{25k}]$
...	...	...
...	...	...
N	1	$[id_{N11}, id_{N12}, \ldots, id_{N1k}]$
N	2	$[id_{N21}, id_{N22}, \ldots, id_{N2k}]$
N	3	$[id_{N31}, id_{N32}, \ldots, id_{N3k}]$
N	4	$[id_{N41}, id_{N42}, \ldots, id_{N4k}]$
N	5	$[id_{N51}, id_{N52}, \ldots, id_{N5k}]$

By employing the dual transformation methods as described in Section 3.1, the k-anonymity set of mobile users is formulated based on their dual points. Hence, an alternative definition for the k-anonymity is as follows:

Definition 10. *(kDUST-anonymity). A transformed database record is k-anonymous with respect to Hough-X dual points—that is, velocity and intersection attributes* (U_x, a_x) *or* (U_y, a_y)*, if* $k - 1$ *discrete records in the same specific time-stamp* τ *at least have the same dual point attributes so that no record of k is distinguished from its* $k - 1$ *neighboring records.*

Remark 1. *As we already mentioned in [9], k-anonymization intuitively hides each individual among* $k - 1$ *others. This means that linking cannot be performed with confidence greater than* $\frac{1}{k}$*. Nevertheless, k-anonymity may not protect users against the unveiling of dual point attributes.*

3.5. System Model

Here, we consider a spatio-temporal database with N records—that is, N moving objects in the xy plane. Each record (x_i^j, y_i^j) represents the spatial coordinates of the mobile user j in time-stamp t_i^j, or point i of its trajectory j [15]. From the location coordinates (x, y), we can extract the corresponding dual points by employing the methods described in Section 3.1. Suppose a trajectories database $T = \{T^1, \ldots, T^N\}$ of equal length L in which each trajectory is represented via a sequence of L triples, that is, $T^j = \{(x_1^j, y_1^j, t_1^j), (x_2^j, y_2^j, t_2^j), \ldots, (x_L^j, y_L^j, t_L^j)\}$.

For each point i in trajectory j, we define in four-dimensional space a vector $DP_i^j = (U_{x_{ij}}, a_{x_{ij}}, U_{y_{ij}}, a_{y_{ij}})$ which denotes the dual points array. Hence, we can redefine and store the trajectory j as $T^j = \{DP_1^j, DP_2^j, DP_3^j, \ldots, DP_L^j\}$.

The privacy preservation of k-NN query in trajectory databases is addressed with the use of two different methods. The first one is entitled dual-based k-NN ($DukNN$) which applies k-NN directly onto dual points, while the second one is called dual-based clustering k-NN ($DuCLkNN$). The main difference between these two methods lies in the fact that the latter is applied in clustered dual point data. In addition, the operations involved in addressing a k-NN query are thoroughly described in Algorithms 1 and 2, respectively.

Algorithm 1 DukNN

```
input The number of $k$ nearest neighbors
input The number of mobile users $N$
input The dual points array of $N$ users in $L$ time-stamps
output $k$ nearest neighbors indexes of $N$ users in $L$ time-stamps
for $i = 1$ to $L$ do
    for $j = 1$ to $N$ do
        Apply $k$-NN for the dual points of all users in order to identify the set of $k$-NN indexes $I_i^j$ of user $j$ in time-stamp $i$
    end for
end for
```

Algorithm 2 DuCLkNN

```
input The number of $k$ nearest neighbors
input The number of mobile users $N$
input The dual points array of $N$ users in $L$ time-stamps
output $k$-NN indexes of $N$ users in $L$ time-stamps
Apply $K$-Means of dual points $(U_x, a_x)$ of $N$ users for the $L$ time-stamps
for $i = 1 : L$ do
    for $j = 1 : N$ do
        Apply $k$-NN method between the dual point of user $j$ and the dual point of users inside the cluster $\mathcal{C}_i^j$ of user $j$ in time-stamp $i$ and find the set of $k$-NN indexes $I_i^j$
    end for
end for
```

In the case of employing the Algorithm 1 in order to run a k-NN query, we must focus on a specific time-period during which we will have in our disposal the dual point of all users' locations. Given that each user stands in the same sub-trajectory during the study period, the privacy is preserved in that segment since the k nearest neighbors remain unchanged. On the other hand, in the case of employing Algorithm 2, the clustering step is ahead; we can again claim that the clusters composition remains the same, since the clustering method is applied in dual space and mobile users have the same dual point. As a result, the k nearest neighbors inside the cluster will remain the same. Hence, without loss of generality, in both cases, the privacy is piecewise preserved, except for the points of discontinuity (known as characteristic points) where the motion characteristics may change.

3.6. Vulnerability and Storage Efficiency

In this paper, we assume the mobile users' trajectory on a real map with small velocities; thus, we use the Hough-X transform, since an object's motion is mapped to the (U, a) dual point. To answer a k-NN query, the following steps are performed:

1. Decompose the k-NN query into $1D$ queries for the (t, x) and (t, y) projection.
2. For each projection, get the dual k-NN query by using a Hough-X transform.
3. Return the anonymity set, which contains the trajectories IDS that satisfy the dual k-NN query in each projection.

In following, the analysis is focused on the robustness estimation of the proposed approach based on Hough-X. Specifically, the ensuing steps are followed:

1. Split the initial trajectory into a number of linear sub-trajectories, each of which consists of the same number of M spatial points.
2. Apply Hough-X in each part.

Suppose that M is the number of points of the $1D$ trajectory, which a dual point represents, and D is the number of dual points, which describe the $1D$ trajectory projection (t, x) or (t, y) in dual space.

Therefore, the whole trajectory has a length equal to DM spatial points, for which $M \gg D$ should hold. In the following, we camouflage a mobile user who keeps track of a linear trajectory $x(t)$ or $y(t)$ or its corresponding dual point with the k nearest neighboring dual points, which is very probable to remain the same in the next timestamp. Actually, while users move onto the linear sub-trajectory, which relates to the same dual point, the k-NN set will remain intact. Therefore, for as long as it happens, we can claim that the k-anonymity holds. Indeed, the privacy preservation is reinforced by a factor M that formulates the so-called vulnerability level to $\frac{1}{kM}$.

We recall the spatial data security metric that we have already defined in [9] for the quantification and measure of the robustness of our methods. Again, the vulnerability remains equal to $\frac{1}{k}$ in dual point space. Nonetheless, the definition of vulnerability in the initial dataset is measured as the following. Since the points inside a sub-trajectory are protected by the same dual points, it is obvious that their vulnerability is considerably reduced to $\frac{1}{Mk}$; this aspect entails that with a probability equal to $\frac{1}{Mk}$, an intruder can distinguish the identity of a mobile user. The same holds for all sub-trajectories. Hence, the vulnerability in each projection is defined as:

$$\begin{aligned} V_x &= \frac{1}{Mk} \\ V_y &= \frac{1}{Mk} \end{aligned} \tag{1}$$

where V_x and V_y is the vulnerability measure based on Hough-X in projection (t, x) and (t, y), accordingly.

Next, the vulnerability in each projection is combined, and the total vulnerability is written as in the following equation:

$$V_{total} = V_x V_y \binom{2}{M} = \frac{1}{(Mk)^2} \binom{2}{M}, \tag{2}$$

where $\binom{2}{M}$ represents all combinations of M points that correspond to 2 dual points of the initial trajectory.

Several trajectory compression approaches have been proposed aiming at reducing the trajectory's size. An initial discrimination classifies the compression methods either as offline (after trajectory generation) or online (instantly as objects move). The data compression constitutes a method that decreases the size of the data in order to limit the memory space and ameliorate the efficiency of storage, processing, and/or transmission without loss of information. Various trajectory compression algorithms exist in literature that try to balance the tradeoff between accuracy and storage size. We refer to some major ones—namely, distance-based, velocity-based, semantic, similarity-based, and priority queue [4]. The proposed Hough-X based approach achieves trajectory compression suitable for either a single or multiple trajectory set. Without loss of information, Hough-X maps each linear sub-trajectory spatial point to its representative dual point.

Compression can be achieved by applying dimensionality transformation to increase the storage efficiency of the data. Suppose we reduce three-dimensional data (x, y, t) to Hough-X space of (t, x), that is, (U_x, a_x). Storage space-saving is achieved through the number of available dual points D, being less than the number of points M in the corresponding linear sub-trajectory; hence, achieving in the whole trajectory $CR = \frac{MD}{D}$ or $CR = M : 1$, where for example, M spatial points correspond to one dual point, as shown in Figure 1. This conserves space and achieves more compression, as depicted in Figure 3, and thus it is expected to have a greater impact on large-scale spatio-temporal databases.

Potentially, by employing a dual method based on Hough-X, we could generate a trajectory codebook by applying Hough-X transformation to all linear sub-trajectories of a given set of trajectories in a map region. In the training step, dual points that stem from the same linear part are similar and must be grouped into the same cluster; also, each cluster is assigned to a single representative vector, called a dual code-vector. Hence, each trajectory inside the codebook is represented by its dual points.

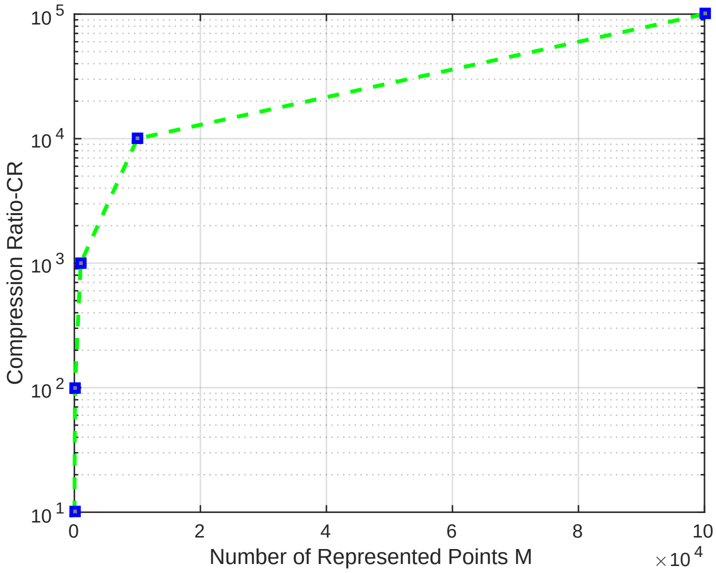


Figure 3. Theoretical curve of compression ratio for M = [10 100 1000 10,000 100,000].

At this point, we should note that the Hough method acts as a clustering one. Actually, *K*-means is a popular method for both clustering and codebook design. In the coding step, each input dual points vector is compressed to the nearest dual code-vector referenced by a simple index. The index of the matched code-vector in the codebook is then transmitted to the decoder over a channel and is used by the decoder in order to retrieve the similar trajectory dual points from an identical codebook. The key operation is that it is stored and transmits the index of the dual code-vector, rather than the entire code-vector.

As a result, the recommended schema is space-compressed because of the duality, and also more robust in comparison with the suggested methods in previous works [9,12].

3.7. Privacy Preservation Analysis

Privacy relates to individual data protection and the human right to be able to determine the information about themselves that is to be hidden. Privacy-preserving data management includes *k*-anonymity, a noted method for data anonymization before publication, which has also been studied in the context of trajectory data. Authors in [16] claim that given a set of trajectories, the objective of the data publication is to transform them into some *k*-anonymized form in order to prevent original data publication, putting at risk the privacy of individuals related with the data. In addition, they mention that an intruder, who knows a sub-trajectory of the original trajectory of an individual, may utilize it with the aim of extracting the whole trajectory of that person based on the published data. Finally, they recognize an upper bound for the re-identification probability of the whole trajectory within the released data, namely $1/k$, where the parameter k reflects the expected level of privacy.

Our solution transforms the original spatial point into the dual-point one using bijective mapping, such as Hough. This technique allows for a *k*-NN search directly on the transformed points, thus providing stronger location privacy. Assuming an insecure Transformed Database Management System (TDBMS) possibly located at a third party (e.g., a service provider in the cloud), an attacker sees its environment. In particular, the attacker has access to the transformed database, to the queries upon the transformed data, as well as to the results. Also, we suppose that the attacker is aware of the dual transformation scheme and aims to retrieve the original database executing Hough-X and/or Hough-Y algorithms with respect to the size of the database. Nonetheless, in our proposed paper, we aim to prevent an attacker from obtaining the original database, as it is possible that they may occupy extra knowledge about this original database. To better evaluate the power of the transformation scheme, we taxonomize the attacks into different levels based on the possessed knowledge.

1. **Level 1**: The attacker only observes the transformed database.
2. **Level 2**: Except the transformed database, the attacker is familiar with a set of plain tuples of the original database, but does not know the corresponding encoded values of those tuples in the transformed database.
3. **Level 3**: Apart from the transformed database, the attacker observes a set of tuples in the original database, and thus knows the corresponding encoded values of those tuples.

A few cryptography-based approaches, such as homomorphic encryption (HE), verifiable computation (VC), and secure multi-party computation (MPC) have been designed in order to provide secure big-data processing in the Cloud [17]. However, other approaches, such as Asymmetric Key Cryptography and trusted Public Key Infrastructure have been developed over the years in order to support privacy preservation in the spatio-temporal domain. The basic idea behind these techniques is to encrypt the identity of the user prior to sending it to the service provider. In this way, the service provider does not have any knowledge about the real identity of the individual who initiated the *k*-NN query. To prevent an external adversary from linking queries to the same mobile object, its pseudonym has to be secure. For this reason, we are concerned about pseudonyms' recovery, as well as registration protocols consisting of three entities, namely, Users (U), Identity Provider (IP), and Service Providers (SP). Recall that they are based on Brand's credentials and have been suggested by Brand in the context of "The New System" with the aim of making the communication more reliable and secure. We believe that the adoption of these protocols will reinforce the identity privacy of mobile objects and the spatio-temporal databases at large. For the sake of completeness, the main steps of these protocols, along with the privacy preservation properties they offer, are presented.

The mobile user, U performs the following protocol in order to retrieve a set of pseudonyms with the identity provider (IP):

Initially, user U chooses random values $r_{(1,1)}, r_{(1,2)}, \ldots, r_{(1,m)}, e \in Z_q$ where e is known only to user U, then computes the quantity $t_1 = g_1^{r_{(1,1)}} g_2^{r_{(1,2)}} \ldots g_m^{r_{(1,m)}} g_{m+1}^{e} \in Z_q$ and finally sends it to IP $(g_1, g_2, \ldots, g_{m+1} \in G_q)$.

Secondly, IP recovers the quantity t_1, collects random quantities $r_{(2,1)}, r_{(2,2)}, \ldots, r_{(2,m)}$ and computes the product $t = t_1 \cdot t_2$, where $t_2 = g_1^{r_{(2,1)}} g_2^{r_{(2,2)}} \ldots g_m^{r_{(2,m)}}$.

Thirdly, user U creates the r_i's according to the equation $r_i = r(1,i) + r(2,i)$ for $i = 1,2,\ldots,m$ and computes the quantity $t = g_1^{r_1} g_2^{r_2} \ldots g_m^{r_m}$. Hence, the corresponding user creates m pseudonyms $(P_i, sign(P_i))$ and values $s_i \in Z_q$, such that $P_i = (tf_0)^{s_i}$ for $i = 1,2,\ldots,m$.

A mobile user U registers a pseudonym $(P_i, sign(P_i))$ with a service provider SP_i presenting the pseudonyms $(P_i, sign(P_i))$ and uncovering the value r_i encoded in P_i. The user with the service provider SP_i performs the following proof of knowledge, provided that $P_i \neq 1$, so as the tuple $(P_i, sign(P_i))$ will be a valid one.

$$PK(\delta_1, \delta_2, \ldots, \delta_{i-1}, \delta_{i+1}, \ldots, \delta_m, \epsilon, \varsigma) : (\delta_1, \delta_2, \ldots, \delta_{i-1}, r, \delta_{i+1}, \ldots, \delta_m, \epsilon, \varsigma) = rep_{(g_1, \ldots, g_m, g_{m+1}, P_i)} f_0^{-1} \quad (3)$$

Then, the service provider SP_i stores the tuple (P_i, r_i) and associates it with either a new or an existing user account. Through this protocol, it is demonstrated that the user owns the pseudonyms and proves that the disclosed value r_i is actually the value encoded in P_i.

The privacy preservation lies in the following facts:

1. The service provider cannot find out any additional information about the quantities encoded in P_i, except for the disclosed value r_i.
2. The random set $(r_1, r_2, \ldots, r_m, e)$ is created so that nobody (user, identity provider) can control its end value.

3. The e is randomly selected from the user so that it can remain unknown to the IP. The user also computes a secret key $(r_1 s_i, r_2 s_i, \ldots, r_m, es_i)$, one for each pseudonym $(P_i, sign(P_i))$; this proves that the user is aware of it without unveiling it.
4. The assumption of the Discrete Logarithm in a group G_q of prime order q along with the values $r_i \in Z_q = \{1, 2, \ldots, q\}$ $(i = \{1, 2, \ldots, m\})$ ensures that any malicious user MU, irrespective of the level of knowledge they possess about the Original and Transformed Database, even if they engage a pseudonym recovery protocol to the IP and obtain a valid pseudonym $(P, sign(P))$, has negligible probability that it is the value encoded in the public key P.

Suppose there is a mobile user who has initiated a discrete number of k-NN queries with different pseudonyms for each one. The unlinkability that the aforementioned protocols provide relates with the service provider's incapability to connect to the IP with the different pseudonyms to that mobile user and validate that they belong to the same user. Thus, the privacy preservation of that user identity is achieved.

3.8. Experimental Data and Environment

The experimental data used in this paper were utilized from the SMaRT Database GIS Tool (http://www.bikerides.gr/thesis2/). The experiments were based on trajectory datasets of bike riders in the area of Corfu, Greece. For each trajectory point, the Hough-X and Hough-Y dual points, that is, the values of (U_x,a_x), (U_y,a_y), (b_x,w_x), and (b_y,w_y), were available for L time-stamps. The environment where the experiments were carried out had the following characteristics: Intel(R) Core(TM) 2 Duo CPU E8400 @ 3.00 GHz CPU, 16 GB of memory, 64-bit Operating System, x64-based processor, and Matlab 2018a.

4. Results

In this section, we conducted several experiments based on a real dataset with parameters and relevant values presented in Tables 4–6, as well as the results in Figures 4–7. Our aim was to evaluate the performance of Algorithms 1 and 2 in terms of vulnerability. We experimented on a dataset of size $N = \{87, 995, 1000\}$.

Table 4. Parameters for the experiment of using only Hough-X of x and Hough-X of x, y, for $N = 1000$ trajectories, $L = 10$ time-stamps (Figure 4a,b) and $N = 995$ trajectories, $L = 100$ time-stamps (Figure 4c,d).

K	k	Clustering Attributes	k-NN Attributes	Clustering Attributes	k-NN Attributes
5	10	(U_x, a_x)	(U_x, a_x)	(U_x, a_x, U_y, a_y)	(U_x, a_x, U_y, a_y)
5	20	(U_x, a_x)	(U_x, a_x)	(U_x, a_x, U_y, a_y)	(U_x, a_x, U_y, a_y)
5	30	(U_x, a_x)	(U_x, a_x)	(U_x, a_x, U_y, a_y)	(U_x, a_x, U_y, a_y)
10	20	(U_x, a_x)	(U_x, a_x)	(U_x, a_x, U_y, a_y)	(U_x, a_x, U_y, a_y)

Table 5. Parameters for the experiment using Hough-X of x and suppressing Hough-X of y (Exp1) for $N = 1000$ trajectories, $L = 10$ time-stamps (Figure 5a,b) and using Hough-X of y and suppressing Hough-X of x (Exp2) for $N = 995$ trajectories, $L = 100$ time-stamps (Figure 5c,d).

K	k	Clustering Attributes	k-NN Attributes (Exp1)	k-NN Attributes (Exp2)
5	10	(U_x, a_x, U_y, a_y)	$(U_x, a_x, *, *)$	$(*, *, U_y, a_y)$
5	20	(U_x, a_x, U_y, a_y)	$(U_x, a_x, *, *)$	$(*, *, U_y, a_y)$
5	30	(U_x, a_x, U_y, a_y)	$(U_x, a_x, *, *)$	$(*, *, U_y, a_y)$
10	20	(U_x, a_x, U_y, a_y)	$(U_x, a_x, *, *)$	$(*, *, U_y, a_y)$

Table 6. Parameters for the experiment using (x, y) for clustering and Hough-X for k-NN for $N = 87$ trajectories and $L = 100$ time-stamps (Figure 7c,d).

K	k	Clustering Attributes	k-NN Attributes
3	8	(x, y)	x
3	8	(x, y)	(x, y)
3	8	(x, y)	(U_x, a_x)
3	8	(x, y)	(U_x, a_x, U_y, a_y)

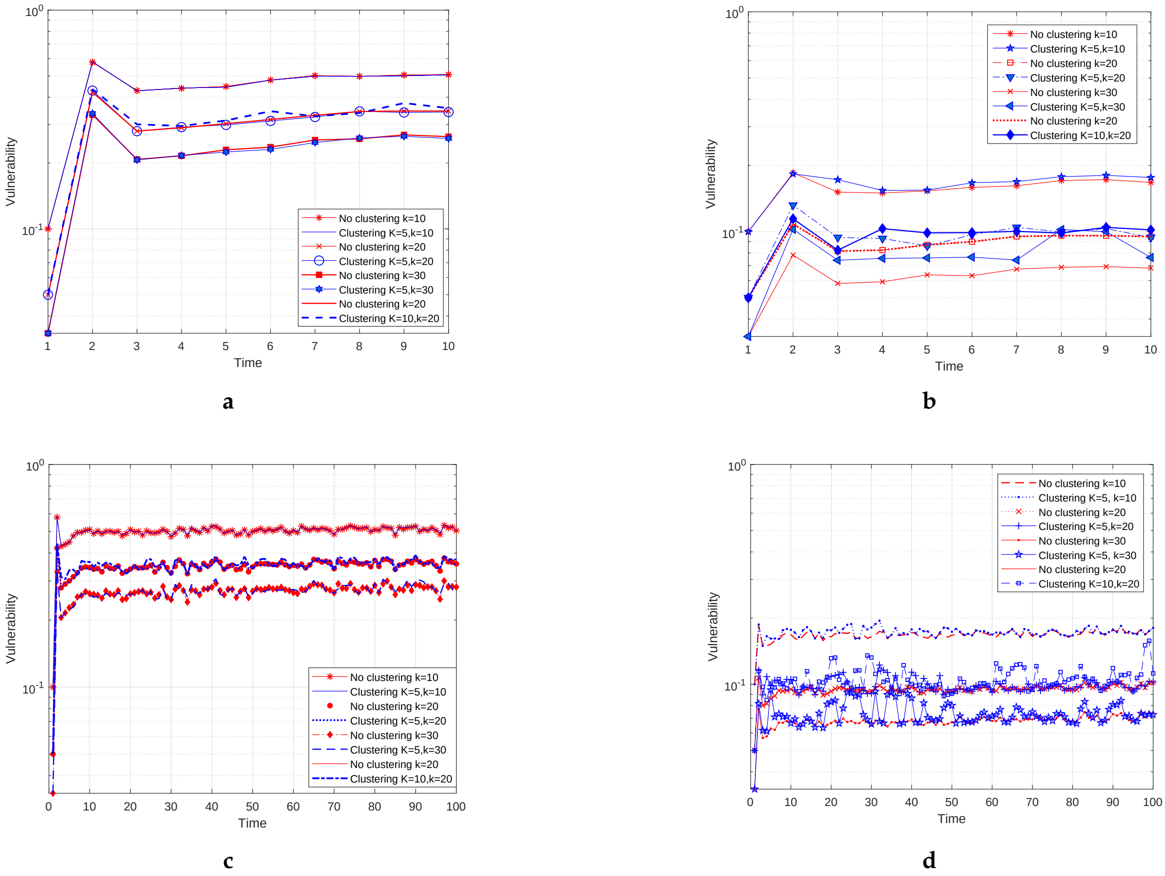


Figure 4. Both clustering and k-NN: (**a**) (U_x, a_x) and (**b**) (U_x, a_x, U_y, a_y) for $N = 1000$ trajectories, $L = 10$ time-stamps, (**c**) (U_x, a_x) and (**d**) (U_x, a_x, U_y, a_y) for $N = 995$ trajectories, $L = 100$ time-stamps.

4.1. Vulnerability Evaluation in Hough Space

In the context of proposed work, we focused on k-anonymity from a different perspective as we employed Hough-X transformation of (x, y) spatial data to formulate the anonymity set. The number of clusters is denoted by K, while k refers to the number of nearest neighbors in terms of Euclidean distance. In Figures 4 and 5, both approaches achieve similar performance that can be improved as the number of k increases. Although for low values of k, the vulnerability remains relatively high, the more the nearest neighbors are utilized to form the anonymity set, the lower the vulnerability becomes. Actually, Figure 4b,d depicts that the use of Hough-X transformation of y attribute considerably improved the performance in both cases. To ameliorate the classification accuracy, we considered the Hough-X attributes of y as well.

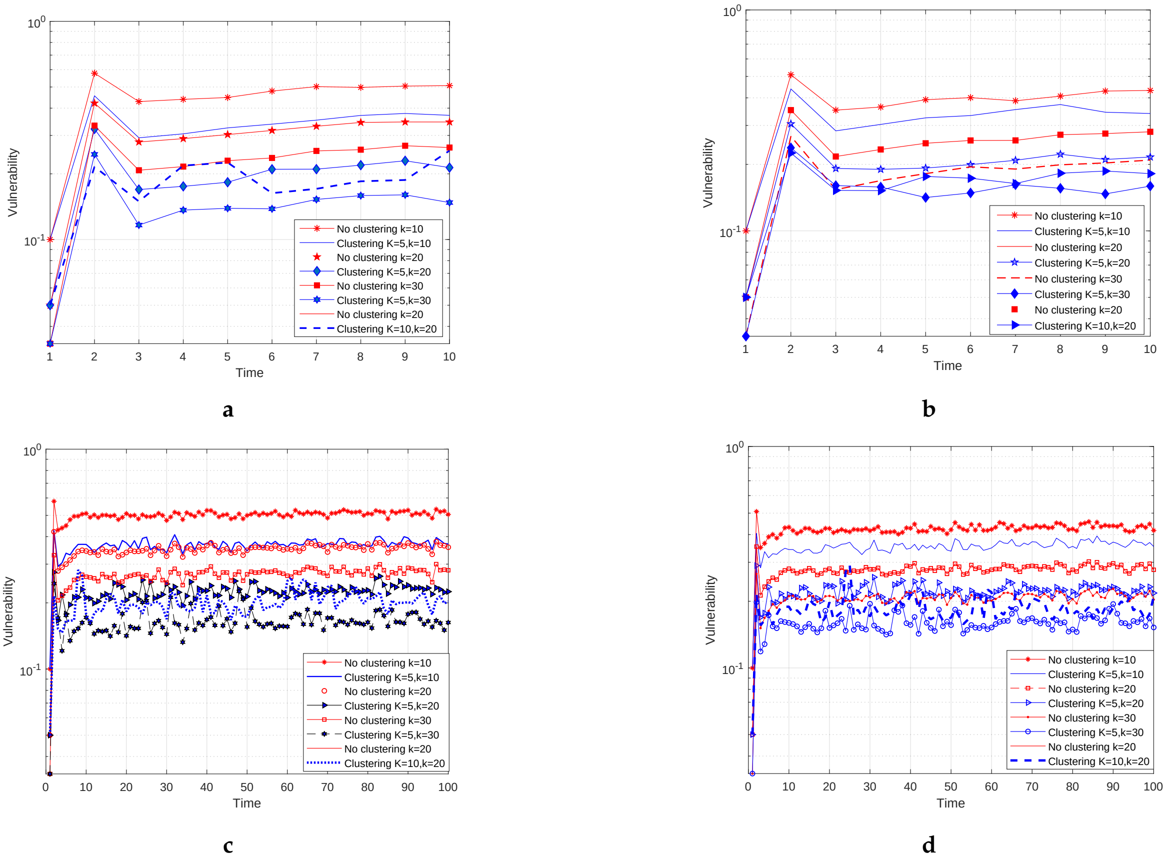


Figure 5. Clustering with (U_x, a_x, U_y, a_y) and suppressing k-NN: (**a**) $(U_x, a_x, *, *)$ and (**b**) $(*, *, U_y, a_y)$ for $N = 1000$ trajectories, $L = 10$ time-stamps, (**c**) $(U_x, a_x, *, *)$ and (**d**) $(*, *, U_y, a_y)$ for $N = 995$ trajectories, $L = 100$ time-stamps.

We should note that the information (U_x, a_x) in combination with (U_y, a_y) increased the robustness of both methods for the same number of nearest neighbors. This entails that the performance of the k-NN classifier improved, and thus the anonymity set shows less time variation from one time-stamp to another. In the following, we employ the suppressing k-anonymity method for the composition of the k anonymity set. In particular, we applied K-Means clustering that takes advantage of (U_x, a_x, U_y, a_y), while k-NN is applied either on (U_x, a_x) or (U_y, a_y). Here, the clustering method presents much better performance than the non-clustering one. However, for the same number of k, the performance of both methods is worse than in the first case, where we based it on attributes (U_x, a_x, U_y, a_y) for both clustering and k-NN computation. The experimental results in Figure 5 present the performance of attribute suppression in terms of k anonymity set computation.

Subsequently, a scenario with synthetic data from the real trajectory dataset was considered. More specifically, for each dual point, a number of copies M were generated, which shows that these dual points correspond to the same linear sub-trajectory of the trajectory. Figure 6a,b shows that DukNN (non-clustering) and DuCLkNN (clustering) methods have identical performance for $M = 5$, $k = 10$ and $K = 5$, and we verify that vulnerability is piece-wise preserved, except for the characteristic points.

In Figure 6c, we compare vulnerability in the Hough-X space of x and y attributes with the one in native dimensional space of x and y. We observe that the results in native space are better by almost 5% than the ones in Hough-X space. This may relate with the linear dependency of dual space of the native one.

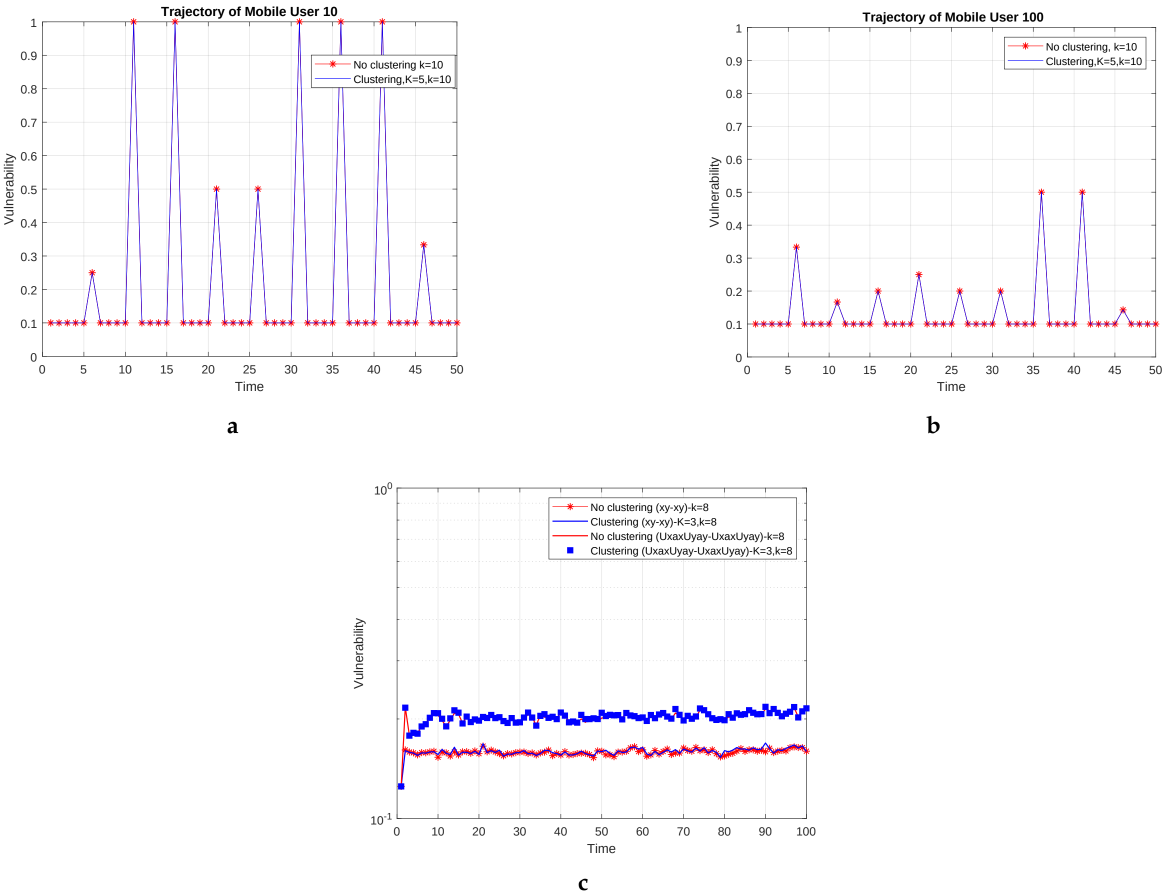
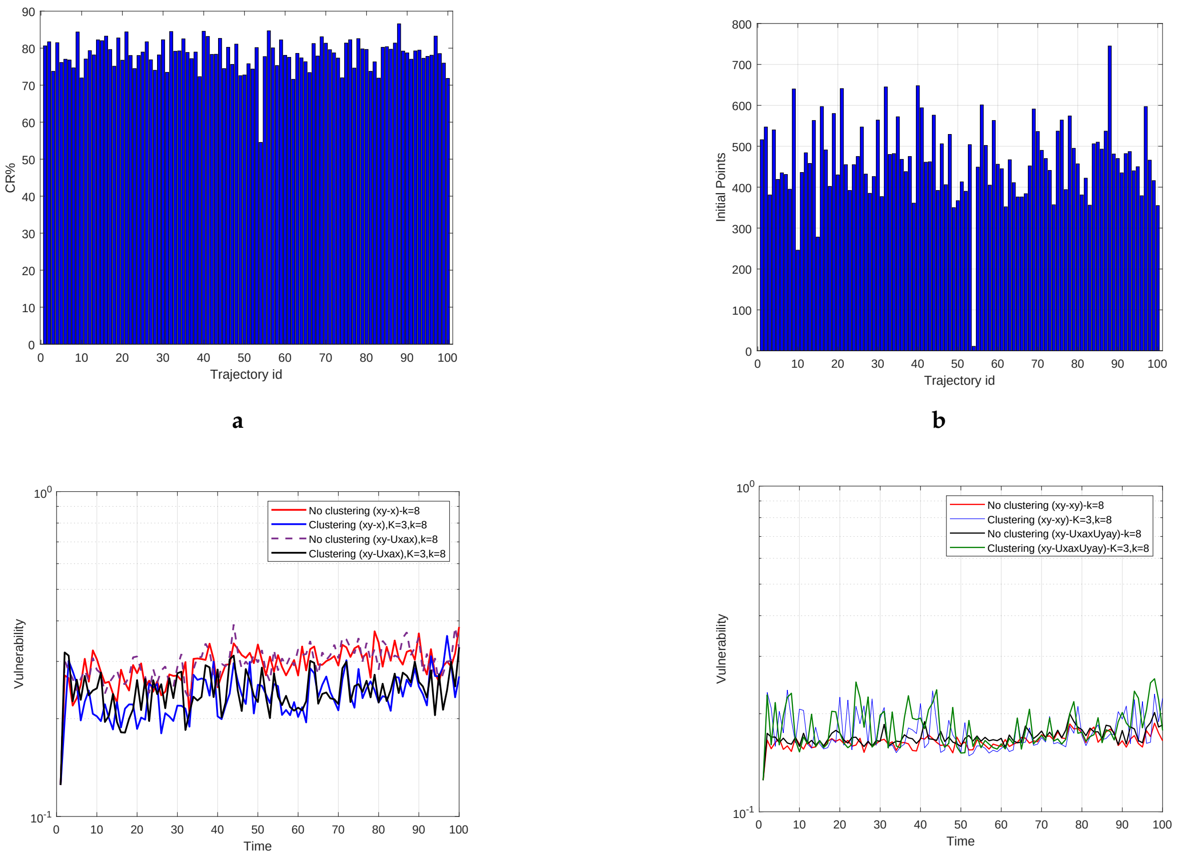


Figure 6. Clustering with (U_x, a_x) and k-NN with (U_x, a_x): (**a**) Mobile User 10 and (**b**) Mobile User 100 for $N = 995$ trajectories, $L = 50$ time-stamps, (**c**) Vulnerability measure in dual Hough-X and native dimensional space of (x, y).

4.2. Vulnerability Evaluation in Hybrid Space

In this subsection, we consider the fact that clustering takes place in spatial coordinate space while the k-NN query is in Hough space. In this case, the dataset concerns the compressed version of mobile users' trajectories as derived from SMaRT. Figure 7a,b depicts information about the initial trajectory's length and the selected points, as well as the compression ratio per trajectory ID. Note that the compression ratio in the system is computed as $(1 - \frac{SelectedPoints}{InitialPoints}) \times 100\%$, where the selected points are 100. From the dataset, we exclude 13 trajectories whose length is much less than 100, such as the average length of compressed trajectories.

Another observation is introduced in Figure 7c,d, where the vulnerability in hybrid space has similar behavior and performance with the one in spatial coordinates space. This relates to the fact that Hough-X constitutes a linear transformation on spatial coordinates. Again, employment of the suppressing method, as shown in Table 6, makes the vulnerability of the k-NN query with the clustering method even more secure.



a **b**

c **d**

Figure 7. (**a**) Initial points per trajectory and (**b**) compression ratio for $N = 87$ trajectories, $L = 100$ time-stamps. Clustering with (x, y) and k-NN: (**c**) x and (U_x, a_x) and (**d**) (x, y) and (U_x, a_x, U_y, a_y) for $N = 87$ trajectories, $L = 100$ time-stamps.

5. Discussion

Hough Transform is a robust method used in Image Analysis and Computer Vision. The core idea is to map data onto the dual parameter space and then interpret it through classification and clustering. The major role of the Hough Transform is to detect straight lines and compute their representatives, that is, dual points. The term "representative" is strictly connected with Clustering Using Representatives (CURE), an efficient data-clustering algorithm for large-scale databases. Compared with K-means clustering, which has already been addressed in our previous work, it is more robust to outliers and able to identify clusters having non-spherical shapes and size variances. Trajectory Clustering (TRACLUS) [18,19] is a characteristic algorithm which has been designed for partitioning and applying clustering among partitions of different trajectories, and finally finding the representative sub-trajectory for each cluster, as presented in Figure 8. Several representative spatio-temporal clustering methods have been reviewed in a more recent work [20]. Nevertheless, in our work, we made use of this term with a different meaning. The notion of the representative points of a trajectory sample points is defined. Thus, the representative point of a number of spatial points that belong to the same line segment is the dual transformation point. Our framework can be combined with any existing clustering algorithm. As a preliminary approach, we chose K-Means, which applied on-line clustering on dual-point data for a number of mobile objects' trajectories at specific time-stamps.

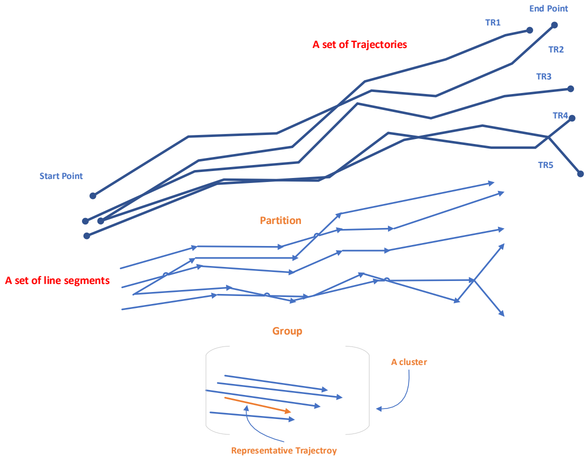


Figure 8. Trajectory partition, grouping, and representatives.

This work investigated the impact of Hough-X, which has already been applied in range queries, to the robustness of the methods proposed in [9] for addressing secure k-NN queries. The experimentation with the number of clusters K, which should be known in advance, obviously only affected the method utilizing clustering; there, it is obvious that vulnerability behaves a little worse as the cluster number increases. Indeed, when adding features, the data cluster density decreases, where the model becomes more sparse, and hence the clustering task becomes even more difficult. A usual phenomenon and important part in Machine Learning is the reduction of a higher dimensional space into a lower dimensional one in order to avoid the Curse of Dimensionality.

An important property of Hough is its robustness to low quality or uncertain data (either due to non-uniform sampling or noise) [21]. Therefore, even if a trajectory is represented by different sample points in 2D Euclidean space, in Hough space it may have the same dual points. Under this condition, Hough space reflects mobility patterns better than the original trajectory spatial data (x, y), leading to more homogeneous clusters and improving the k-NN performance. As experimental results verify in Figure 6a,b, the above properties have a positive impact on vulnerability which is pair-wise preserved, showing that the clusters occur within cluster space-time similarity. The authors in [22] provide an efficient scheme for representative clustering on uncertain data. Finally, assuming feature suppression, the method with clustering demonstrates higher robustness or lower vulnerability, which is the main issue in k-anonymity, and thus in privacy preservation. It is a case which shows the superiority of the method with clustering in terms of vulnerability. Indeed, when the mobile users are protected by k nearest neighbors based on lower dimensionality data than the ones used in clustering, it is more difficult for an attacker, who has access to history data, to link the public information of the $k - 1$ nearest neighbors (that is, unlinkability).

6. Conclusions

In conclusion, we carried out research on privacy preservation based on real spatio-temporal data, through which we demonstrated the impact of parameters k and K in terms of the vulnerability of the proposed methods. We observed that the increase of k benefits both methods, verifying that the security of a mobile user is more robust when the latter is protected by a high number of nearest neighbors. This paper proposes the application of k-NN queries based on dual points of Hough-X projection with the aim of reinforcing anonymity of k-NN queries and decreasing storage requirements. More specifically, we investigated the problem from the perspective of dual point attributes. The experimental results indicate that although the outcomes of the Hough-X based vulnerability are not optimal in comparison with spatial coordinates space, the difference between them is less than 5%, which still makes Hough-X an appropriate choice for storage-efficient privacy preservation.

The SMaRT framework approximates users' non-linear trajectories with linear ones from time-stamp to time-stamp, and the current results a concerned with the low data-sampling rate. A challenging and open issue is experimentation on the impact of the data sampling rate (low and high) in the described procedures and transformations. Also, we plan to extend and/or enhance proposed methods to be applicable to 3D (x, y, z) trajectories to represent the real situation of, for example, tracing the GPS trajectories of observed birds with devices or drones. In such a case, the dual methods can be applied in z projection in the same way as x, y ones. Additionally, we intend to evaluate the efficiency and scalability of the suggested approaches on big spatio-temporal databases in a distributed environment, that is, in the cloud, and compare its performance with appropriate indexing methods. Our aim is to make SMaRT suitable for supporting k-NN queries based on the proposed methods.

Ultimately, it will be useful to evaluate the time of some transactions (e.g., roll-back where the end user lost himself and decides to come back the same way to a certain point or look for another way), that is, how long the end-user will take to receive the answer from the Database Management System (DBMS) with the aforementioned implemented procedures compared to those used nowadays.

Author Contributions: Methodology, E.D., A.K., M.T., S.S. and A.T.; Software, E.D., A.K., M.T., S.S. and A.T.; Writing–original draft, E.D., A.K., M.T., S.S. and A.T.; Writing–review and editing, E.D., A.K., M.T., S.S. and A.T.

Funding: This research work is funded by General Secretariat for Research and Technology (GSRT), Hellenic Foundation for Research and Innovation (HFRI), and supported by University of Patras.

Acknowledgments: The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved both content and presentation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Körner, C.; May, M.; Wrobel, S. spatio-temporal Modeling and Analysis—Introduction and Overview. *Künstliche Intell. KI* **2012**, *26*, 215–221. [CrossRef]
2. Feng, Z.; Zhu, Y. A Survey on Trajectory data-mining: Techniques and Applications. *IEEE Access* **2016**, *4*, 2056–2067. [CrossRef]
3. Gudmundsson, J.; Katajainen, J.; Merrick, D.; Ong, C.; Wolle, T. Compressing Spatio-temporal Trajectories. *Comput. Geom.* **2009**, *42*, 825–841. [CrossRef]
4. Han, Y.; Sun, W.; Zheng, B. COMPRESS: A Comprehensive Framework of Trajectory Compression in Road Networks. *ACM Trans. Database Syst. TODS* **2017**, *42*, 11. [CrossRef]
5. Song, R.; Sun, W.; Zheng, B.; Zheng, Y. PRESS: A Novel Framework of Trajectory Compression in Road Networks. *PVLDB* **2014**, *7*, 661–672. [CrossRef]
6. Hasan, A.S.M.T.; Qu, Q.; Li, C.; Chen, L.; Jiang, Q. An Effective Privacy Architecture to Preserve User Trajectories in Reward-Based LBS Applications. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 53. [CrossRef]
7. Peng, T.; Liu, Q.; Meng, D.; Wang, G. Collaborative Trajectory privacy preservation Scheme in Location-based Services. *Inf. Sci.* **2017**, *387*, 165–179. [CrossRef]
8. Ye, H.; Cheng, X.; Yuan, M.; Xu, L.; Gao, J.; Cheng, C. A Survey of Security and Privacy in Big Data. In Proceedings of the 16th International Symposium on Communications and Information Technologies (ISCIT), Qingdao, China, 26–28 September 2016; pp. 268–272.
9. Dritsas, E.; Trigka, M.; Gerolymatos, P.; Sioutas, S. Trajectory Clustering and k-NN for Robust privacy preservation spatio-temporal Databases. *Algorithms* **2018**, *11*, 207. [CrossRef]
10. Verykios, V.S.; Damiani, M.L.; Gkoulalas-Divanis, A. Privacy and Security in spatio-temporal Data and Trajectories. In *Mobility, Data-Mining and Privacy*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 213–240.
11. Poulis, G.; Skiadopoulos, S.; Loukides, G.; Gkoulalas-Divanis, A. Distance-Based kˆm-Anonymization of Trajectory Data. In Proceedings of the 14th IEEE International Conference on Mobile Data Management (MDM), Milan, Italy, 3–6 June 2013; pp. 57–62.
12. Gerolymatos, P.; Sioutas, S.; Nodarakis, N.; Panaretos, A.; Tsakalidis, K. SMaRT: A Novel Framework for Addressing Range Queries over Nonlinear Trajectories. *J. Syst. Softw. JSS* **2015**, *105*, 79–90. [CrossRef]
13. Mao, Y.; Zhong, H.; Qi, H.; Ping, P.; Li, X. An Adaptive Trajectory Clustering Method Based on Grid and Density in Mobile Pattern Analysis. *Sensors* **2017**, *17*, 2013. [CrossRef] [PubMed]

14. Sun, P.; Xia, S.; Yuan, G.; Li, D. An Overview of Moving Object Trajectory Compression Algorithms. *Math. Probl. Eng.* **2016**, *2016*, 1–13. [CrossRef]
15. Yuan, G.; Sun, P.; Zhao, J.; Li, D.; Wang, C. A Review of Moving Object Trajectory Clustering Algorithms. *Artif. Intell. Rev.* **2017**, *47*, 123–144. [CrossRef]
16. Basu, A.; Monreale, A.; Corena, J.C.; Giannotti, F.; Pedreschi, D.; Kiyomoto, S.; Miyake, Y.; Yanagihara, T.; Trasarti, R. A Privacy Risk Model for Trajectory Data. In Proceedings of the 8th IFIP International Conference on Trust Management, Singapore, 7–10 July 2014; pp. 125–140.
17. Yakoubov, S.; Gadepally, V.; Schear, N.; Shen, E.; Yerukhimovich, A. A Survey of Cryptographic Approaches to Securing Big-Data Analytics in the Cloud. In Proceedings of the IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 9–11 September 2014; pp. 1–6.
18. Lee, J.; Han, J.; Whang, K. Trajectory Clustering: A Partition-and-Group Framework. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, 11–14 June 2007; pp. 593–604.
19. Panagiotakis, C.; Pelekis, N.; Kopanakis, I.; Ramasso, E.; Theodoridis, Y. Segmentation and Sampling of Moving Object Trajectories Based on Representativeness. *IEEE Trans. Knowl. Data Eng. TKDE* **2012**, *24*, 1328–1343. [CrossRef]
20. Shi, Z.; Pun-Cheng, L.S.C. spatio-temporal Data Clustering: A Survey of Methods. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 112. [CrossRef]
21. Li, X.; Zhao, K.; Cong, G.; Jensen, C.S.; Wei, W. Deep Representation Learning for Trajectory Similarity Computation. In Proceedings of the 34th IEEE International Conference on Data Engineering (ICDE), Paris, France, 16–19 April 2018; pp. 617–628.
22. Züfle, A.; Emrich, T.; Schmid, K.A.; Mamoulis, N.; Zimek, A.; Renz, M. Representative Clustering of Uncertain Data. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014.