Politecnico di Torino

Politecnico di Torino

Corso di Laurea Magistrale in Cybersecurity

Ottobre 2025

A greedy generalization algorithm for anonymizing Origin-Destination matrices

Relatori:
Prof. Luca Vassio
Nikhil Jha, Ph.D.

Candidato:
Pietro Armenante

Abstract

Mobility data, describing the locations and movements of individuals within a geographic area, are a key resource for analysing and managing transportation systems. These data are frequently summarised in the form of origin–destination (OD) matrices, where each entry represents the number of trips between a specific origin and a specific destination. While OD-matrices are a valuable tool for modelling travel demand and managing transport networks, they also raise important privacy concerns, as they may allow the identification of individuals or sensitive travel patterns when published at high spatial resolution.

In this work, a k-anonymisation algorithm specifically designed for OD-matrices is presented: the ODkAnon algorithm. Unlike traditional approaches that apply uniform spatial generalisation to both origin and destination cells, this method dynamically determines, for each flow that does not meet the k-anonymity threshold, whether to generalise the origin or the destination. This selective strategy aims to minimise the loss of spatial precision while ensuring that all flows satisfy the k-anonymity requirement. ODkAnon algorithm creates homogeneous geographical areas. This means that the final OD matrix will comprise non-overlapping areas.

The proposed approach is applied to three real-world mobility datasets: GPS trajectories from the NetMob25 challenge in Paris (France), car-sharing trips in Turin (Italy), and one year of taxi trips in Porto (Portugal). To assess the performance of the proposed method, its results are compared with those obtained from three well-established algorithms in the literature (Mondrian, ATG, and OIGH), evaluating each approach in terms of both privacy protection and data utility. ATG is creating overlapping hexagons, while OIGH is homogeneous, but it is also uniform: it is not creating hexagons of different sizes, but it is giving a unique size to each hexagon.

Experimental results show that ODkAnon achieves competitive performance compared to existing methods. OIGH is the fastest, followed by ATG and then ODkAnon (while Mondrian's runtime varies by dataset). Unlike ATG, ODkAnon automatically balances generalisation between origins and destinations without parameter tuning. In terms of utility, ODkAnon is generally comparable to ATG, in some cases even better, and consistently superior to OIGH. Mondrian often provides higher utility, but this comes at the cost of not enforcing hierarchical consistency or homogeneity in the resulting partitions.

Moreover, using the Paris dataset, the ODkAnon algorithm has also been used to study and create different OD matrices both for the participants and for the population. Protecting the population produces a different anonymization. Even more interesting, when the protection is applied to the population, the participants'

OD matrix loses k-anonymity. The viceversa is also true. Moreover, it is possible to observe amplified differences when segmenting the population by sex, age and work. Consider the example of sex (men and women), with the same k thresholds. The two segments have a similar total number of trips. When protecting the participants, it is shown that protecting men is more challenging, as it requires very coarse hexagons, while for women the resulting hexagons remain much finer. Furthermore, when applying protection to the population, the difference becomes even more pronounced.

Acknowledgements

I would like to thank you my supervisors Luca Vassio e Nikhil Jha for everything they have done during this journey. You have always been by my side, supporting me and guiding me whenever things became difficult to understand. Thank you to the whole SmartData group, where I found not only a stimulating and inspiring environment to work in, but also a warm community of beautiful people.

Finally, thank you to Turin. You embraced me with open arms, and if I am the person that I am today, it is also thanks to you. You made me independent, you helped me discover who I am, what I love, and how I want to live my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Context and Motivation

Mobility data have become increasingly accessible due to the evolution of data collection processes and the diversification of their sources. While in the past they were mainly obtained through ground surveys, the widespread adoption of GPS devices and mobile phones has enabled the collection of massive volumes of data. Among the most notable sources are Call Detail Records (CDRs) and passive Network Signalization Data (NSD), both generated from mobile phone usage.

Human mobility refers to the study of how people move within cities, for instance by characterizing patterns such as commuting to work, returning home, or using public transportation. A thorough understanding of these patterns is fundamental in several domains, including epidemic control [1, 2], urban planning [3, 4, 5], traffic forecasting systems [6, 7, 8], as well as mobile and network applications [9, 10, 11].

The analysis of trajectory data has therefore emerged as a significant research field, given its wide range of practical applications [12]. Processing mobility data can improve people's daily lives, by supporting navigation apps and route recommendations, while also providing useful insights for decision-making in both the public and private sectors. The widespread use of personal devices such as smartphones and wearables, along with modern navigation systems, has allowed the collection and analysis of these data with remarkable precision. Combined with recent technological improvements, this has led to an unprecedented growth in their use [13].

Generally speaking, trajectories are sequences of timestamped locations (such as GPS coordinates). These, at first sight, may appear innocuous to users' privacy, but trajectories can reveal exact home locations and even accurate behavioral patterns [14]. They readily tell you when and for how long a particular individual does what. Exploiting this, a malicious person can infer circumstances and trends

that affect sensitive aspects of an individual's life, such as health status, religious beliefs, social relationships, or sexual preferences [13]. The uniqueness of human traces implies that, with little background knowledge about data subjects (such as their place of residence or work), adversaries can attack seemingly protected data with ease [15][16]. In this context, research shows that knowing only four spatio-temporal points at low resolution is enough to uniquely identify 95% of the individuals in a given database of large scale [17]. Furthermore, we can recover an original, seemingly sanitized trajectory within an obfuscated area using auxiliary public information, like road maps, speed limits, or simple spatio-temporal correlation models [18][19]. All this ultimately leads to poor privacy. Even though many solutions have been proposed in the literature, most suffer from limitations: some are vulnerable to relatively simple attacks, while others significantly reduce data utility by discarding valuable information or even producing unrealistic trajectories. Moreover, many applications of trajectory data require repeated computations, as they are often used to continuously monitor dynamic conditions such as traffic [13]. However, regularly publishing updated versions of a database in a privacy-preserving way makes the challenge even harder. The main reason is that each publication leaks some information about the individuals contained in the database, and it is not simple to ensure that combinations of published private data will not compromise privacy at any moment.

The simplest indicator we can extract from a set of trajectories is an Origin–Destination (OD)-matrix, describing the flows between origins o and destinations d. Although they represent a dramatic simplification compared to trajectories, they are still a crucial indicator of mobility. Like mobility data, OD-matrices may have small and isolated flows, and origins and destinations can be among sets of up to thousands of areas. This makes OD-matrices harder to anonymise than regular relational data [20].

While data analysis can generate significant economic and societal benefits, tensions regarding privacy risks are growing[1][2]. Protecting data subjects and reducing possible harm inflicted upon them hence gains importance. Consequently, legal frameworks in the European Union and other regions explicitly limit personal data collection, processing, and sharing. It is clear that mobility data contain potentially personal information and so it is crucial to handle it with a special care.

[1]Steve Lohr, *Just Collect Less Data, Period*, The New York Times, July 15, 2020. Available at: `https://www.nytimes.com/2020/07/15/technology/just-collect-less-data-period.html`, accessed on October 11, 2025

[2]Nick Srnicek, *We need to nationalise Google, Facebook and Amazon. Here's why*, The Guardian, March 14, 2018. Available at: `https://www.theguardian.com/technology/2018/mar/14/tech-big-data-capitalism-give-wealth-back-to-people`, accessed on October 11, 2025

The GDPR is defining personal information as any piece of information which can be linked to an particular individual. Therefore, ensuring strong privacy protection when analysing location trajectories is thus not only a technical challenge but also a legal obligation.

1.2 Contribution

In this work, a new methodology to make OD-matrices anonymous is proposed: the ODkAnon algorithm. To this end, we used the widely spread criterion of k-anonimity [21]. We propose a novel k-anonymisation algorithm tailored for large-scale OD-matrices using the H3 hexagonal spatial indexing system. Unlike traditional uniform spatial generalisation techniques, this method applies an adaptive approach that dynamically decides, for each flow not meeting the k-anonymity threshold, whether to generalise the origin or the destination hexagon. To efficiently handle very large and sparse OD-matrices, the algorithm leverages sparse matrix representations and precomputed hierarchical relationships within the H3 hexagonal index trees. This allows for rapid identification of sibling hexagons and their parents to speed up the generalisation process and balancing the matrix dimensions dynamically to avoid excessive loss of spatial resolution on either origin or destination side. The OD matrices obtained will be compared by several traditional methods of anonymization to reach k-anonymity by generalizing geographical areas. A k-anonymous dataset is a dataset where every item (in this case, a trip) cannot be distinguished from at least $k-1$ other ones. These include generalization over a hierarchy (ATG and OIGH) and the classical Mondrian. Moreover, this novel approach is benchmarked against three well-known generalisation algorithms, evaluating both individual-level and population-level privacy through weighted mobility data. This comprehensive analysis demonstrates how significant differences may exist when considering population-protecting for OD-matrices anonymization, rather than survey participant-protecting ones. These differences can even be amplified across socio-demographic segments.

All the obtained results are reproducible using our open-source code available in a GitHub repository.[3]

1.3 Research Questions

In order to guide the development and evaluation of this work, we have formulated a set of research questions. These questions focus on the challenges of spatial

[3] `https://github.com/SmartData-Polito/ODkAnon`, accessed on October 11, 2025.

anonymization in mobility data and on the potential of the proposed approach to address them. They serve both as a framework for the design of the methodology and as a reference for interpreting the results.

Research Question 1 How can the H3 hexagonal spatial indexing system be used to partition geographic areas in a different way than the traditional rectangular approaches, such as the Mondrian algorithm?

Traditional spatial partitioning techniques, such as those based on rectangular grids, can introduce distortions and inefficiencies, creating irregular geographic areas. In several approaches, such as Mondrian, there is not even a hierarchy structure but just a division of the space rectangles. Such rectangles can be both very large and very small: if they are very large the approximation can be too wide. The H3 hexagonal grid can better shape geographic areas, providing more accuracy and consistent spatial aggregations for mobility data.

Research Question 2 Can OD-matrices be generalised adaptively by applying different levels of spatial aggregation to origins and destinations, in order to achieve k-anonymity while minimising information loss?

Standard generalisation approaches often apply the same aggregation level to both origins and destinations, potentially leading to unnecessary loss of spatial detail. An adaptive strategy that determines independently which dimension to generalise may better preserve data utility while still meeting privacy requirements. The novel algorithm proposed creates hexagons of varying sizes in a strictly homogeneous manner, avoiding overlaps and ensuring a more reliable representation of mobility data.

Research Question 3 How does the proposed approach perform in terms of privacy protection when evaluated both at the individual level and at the population level using weighted mobility data?

Privacy risks can be different for individual trajectories and for aggregated population movements. Evaluating both perspectives provides a more comprehensive understanding of the privacy–utility trade-off. The algorithm highlights how protecting one or the other involves substantially different challenges, leading to promising insights. Furthermore, segmenting the population into different groups or ranges provides an even stronger representation of what it means to safeguard certain subsets of the population over others.

1.4 Thesis Structure

The remainder of this thesis is structured as follows. Section 2 outlines the legal framework related to trajectory anonymisation, introduces different types of trajectories, discusses relevant privacy notions, and reviews related work on data anonymisation. Section 3 presents the details of the novel methodology proposed. Section 4 describes the datasets used, benchmarks the algorithm against well-known approaches, and introduces the indicators employed to compare anonymisation methods. Section 5 reports the final results, while Section 6 concludes the thesis.

Chapter 2

Related work

2.1 Legal framework

Privacy is a fundamental human right. As stated in Article 12 of the Universal Declaration of Human Rights:

> *No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.*[1]

However, there has always been an asymmetry between the benefits of computerized databases and the rights of individual data subjects [22]. In an effort to recover such an asymmetry, the Code of Fair Information practices was published as a central part of the report of the Committee of the Secretary of Health, Education, and Welfare, Records, Computers, and the Rights to Citizens (USA) [23]. It enunciates five fundamental principles to properly keep records which are:

- the prohibition of secret databases,
- data subjects must be allowed to inspect their records and how are they used,
- the data obtained for one purpose may not be used for other purposes without the consent of the data subject,
- the data subject must be able to correct or amend their records,
- the data must be kept reliable and secure.

[1]Source: `https://www.un.org/en/about-us/universal-declaration-of-human-rights`, accessed on October 11, 2025

According to the European data protection law, the processing of personal data is legitimate if:

- the individual whose personal data are being processed (the data subject) has unambiguously given consent,
- or processing is necessary for the performance of a contract, for compliance with a legal obligation, for protecting vital interests of the data subject, for the performance of a task carried out in the public interest,
- or for the purposes of legitimate interests pursued by the data processing entities except when such interests are overridden by the fundamental rights and freedoms of the data subject.

In [24], based on the legal framework, authors defined and explained the following eight privacy by design strategies: Minimise, Hide, Separate, Aggregate, Inform, Control, Enforce and Demonstrate. These strategies are saying that

- the amount of personal information processed should be minimal,
- the data should be hidden from plain view,
- the processing should be done in distributed fashion whenever possible,
- personal information should be processed at the highest level of aggregation with the least possible detail in which it is still useful,
- data subjects should be informed whenever personal information is processed,
- a privacy policy compatible with legal requirements should be enforced,
- and the data controller should be able to demonstrate compliance with the privacy policy and legal requirements.

It is crucial at this point to define the two main kind of data defined by the GDPR. The Article 4(1) states that:

> ***personal data*** *means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.*[2] "

[2] *Source:* `https://gdpr-info.eu/art-4-gdpr/`, *accessed on October 11, 2025*

It is possible to define **non-personal data** as data other than personal data as defined in Article 4(1) of the GDPR. While personal data is protected by the GDPR, non-personal data are not.

The principals of data protection defined by the GDPR do not apply to anonymous information. **Anonymity** refers to a state where a data subject can no longer be identified or singled out from the data. In other words, during an anonymization process the data must be irreversibly processed in such a way that it can no longer be used to identify a natural person by using "all the means likely reasonably to be used" by any party. On the other hand **pseudonymisation** means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person (Article 4(5)).[3] Unlike data that is pseudonymized, anonymized data guarantees that the individual person cannot be identified when all available additional information on the subject is considered.

However, *if the case of a re-identification attack is possible, this goes under the GDPR*. To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly. To ascertain whether means are reasonably likely to be used to identify the natural person, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments (Recital 26).[4]

In case of data processing for statistical purposes (e.g. average number of cars in an area, percentage of vehicles on a given street, etc.), the output is called aggregate data (Recital 162)[5]. Again, if those data do not permit re-identification, they are not subject to GDPR. Moreover if aggregate and/or anonymous data are subject to re-identification and they disclose sensitive information about racial or ethnic origin, political opinions, religious or philosophical beliefs, they are subject to more strict measures (Article 9).[6]

Considering the nature of respondent-created spatial information and the possibility to identify an individual according to the GDPR, [25] identifies three main

[3]Source: `https://gdpr-info.eu/art-4-gdpr/`, accessed on October 11, 2025

[4]Source: `https://gdpr-info.eu/recitals/no-26/`, accessed on October 11, 2025

[5]Source: `https://gdpr-info.eu/recitals/no-162/`, accessed on October 11, 2025

[6]Source: `https://gdpr-info.eu/art-9-gdpr/`, accessed on October 11, 2025

types of Public Participation Geographic Information System (PPGIS) spatial data, namely primary personal spatial data, group-level spatial data, thematic spatial data. These classes relate to different types of mapping tasks differing on whether the respondent may be identified from the spatial data itself or from the spatial data in conjunction with other personal information. These data types and recommendations for their treatment during an anonymization process are introduced in Figure 2.1.

Table 2.1: PPGIS data and potential personal information (before anonymization).

Data type	**GIS entity type**	**Likelihood of individual identification**	**Recommendations for data anonymization**
1. Primary personal spatial data Residential location(s), second homes	Point	**Very likely**: In areas with low residential density, an individual or the individual's household could be identified from non-anonymized point data. **Likely**: Increased risk of identification when spatial data is linked to other individual-level variables. **Unlikely**: In areas with high residential density, an individual may be recognized on the level of street address.	Always recommended. Increased need for anonymization when the residential location is situated in rural areas or urban areas with low population density, or when the amount of other individual-level variables increases (gender, age, occupation, etc.).
2. Group-level spatial data Locations identifiable to a limited group of individuals (e.g., place of work, university, child's kindergarten)	Point, polyline	**Unlikely**: If data is presented as such. **Likely**: Increased risk of identification when spatial data is linked to other individual-level variables.	Recommended when spatial data is linked to other individual-level variables.
3. Thematic spatial data Locations with no direct connection to the individual (e.g., environmental perceptions, places related to behavior in public or private spaces visited by many people such as shopping centers, parks, etc.)	Point, polyline, polygon	**Very unlikely**: If data is presented as such. **Likely**: Increased risk of identification when spatial data is connected to other individual-level variables that can be used to infer individual behavior patterns (e.g., activity spaces).	Anonymization is rarely needed. Recommended in specific cases when spatial data is connected to other individual-level variables and patterns derived from thematic spatial data that can be used to identify the individual.

2.2 Risk of re-identification

The protection of mobility data is challenging because location traces are inherently rich in information and often unique to each individual. Even when explicit identifiers (such as names or phone numbers) are removed, the spatio-temporal patterns describing where and when a person moves can still make them identifiable. At an abstract level, this section focuses on the concept of re-identification: the risk that anonymized data can be linked back to specific individuals. When this idea is translated into the context of geographic locations, it means that a person's home, workplace, or other frequent places can function as quasi-identifiers, enabling adversaries to match them with external information.

The main goal of trajectory privacy is to protect against risks and threats when unauthorized actors get access to the data. An adversary can gather sensitive information of individuals within or across the datasets. It is possible to classify existing attack models on trajectories into two categories: linkage and probabilistic. Linkage attack models refer to *which* sensitive data is inferred, and are categorized depending on such information, while the probabilistic attack models quantify *how much* knowledge is revealed by accessing the dataset [26].

To show the privacy risks in human traces, the following paragraphs expose some possible attacks and threats of the literature. The distinction of the models and their explaination correspond to the classification of [26] with the extension of the reconstruction and prediction attack of [13].

2.2.1 Linkage Models

Depending on the attack target, linkage models are categorized into record linkage (i.e., inferring individual identity), attribute linkage (i.e., inferring personal profile such as health condition), table linkage (i.e., inferring personal data through the presence of a known individual in the dataset), and group linkage (i.e., inferring social relationships).

Record Linkage An adversary with some background knowledge (e.g., exposed locations, origin and destination locations, and social relationships) can attempt to identify the record of a known victim (i.e., run a re-identification attack). Re-identification attacks are the simplest form of this type [27]. They utilize auxiliary information, i.e., information exposed through other means and thus available to the adversary. In particular, personal-context-linking attacks use known information about a victim (e.g., they have been to a coffee shop) to discover their trajectory in the database. While these attacks are based on trajectory microdata (i.e., raw trajectory locations), aggregated trajectory data (e.g., the number of users within an area) also poses privacy issues. In [28], authors exploit the uniqueness and

regularity of human mobility (e.g., night and daytime mobility behaviors) to recover individual trajectories from aggregated mobility data without any prior knowledge.

Attribute Linkage If sensitive values frequently occur within similar trajectories, an adversary can uncover sensitive information even though cannot unequivocally isolate single trajectories. Despite value diversity can be ensured, if distinct sensitive values sharing a semantic similarity occur frequently within trajectories, an adversary can still cause a privacy breach (i.e., perform an attack based on similarity).

In the mobility domain, this often involves *points of interest* (POIs), such as shops, workplaces, or recreational facilities. Revealing the POIs can cause a privacy breach as such data may be sensitive (e.g., frequent visits to a hospital suggest potential diseases). Examples of POIs are home, work, religion or political parties' locations. In [29], given a dataset of location check-ins, authors use spatiotemporal knowledge and the regularity of human mobility to classify demographics attributes such as gender, age, education, and marital status based on the individual's POIs extracted from check-in dataset. Another example is a Reddit user who was able to identify Muslim taxi drivers in New York City by integrating anonymized taxi trips to the daily praying time. By uncovering which taxi drivers are inactive at such time, it is possible to infer sensitive information such as religion[7].

Many attacks belongs to the family of attribute linkage. Reconstruction attacks aim at rebuilding trajectories in the database. For example, [18] introduces a reconstruction algorithm that can construct trajectories closer to the original data than the perturbed one. Similarly, filtering attacks [30] also aim at reducing noise added. Finally, the possibility of predicting a user's locations (prediction attacks) is also a threat, since attackers can discover the user's destination, probably even before they arrive.

Table Linkage The inference of an individual's presence in a private dataset can also leak sensitive information. For instance, knowing that a victim is part of a dataset of hospital patients implies that she suffers from some disease [31]. Learning merely the presence or absence of an individual in a trajectory database can be a direct privacy threat. Although techniques can reduce the attack success ratio, it may yield a significant utility loss.

Group Linkage The analysis of trajectory data can leak social relationships between individuals in the published dataset. In [32], authors investigate the

[7]Source: `https://mashable.com/archive/redditor-muslim-cab-drivers`, accessed on October 11, 2025

influence of social relationships on human mobility, showing that social relationships can explain about 10% to 30% of all human movement. In other words, individuals tend to group in communities (e.g., family and colleagues) where community members share some traits with other members stronger than with non-members.

2.2.2 Probabilistic Models

A probabilistic attack quantifies how much information an adversary can gather by accessing the dataset rather than focusing on exactly what records, attributes, or tables the adversary can link to a target victim [33]. Intuitively, access to a trajectory dataset should not reveal substantially more information than what is already known by the adversary. In this sense, probabilistic attacks can be seen as a generalization of attribute linkage [34]: while attribute linkage aims at uncovering a specific sensitive attribute (e.g., whether a person has visited a hospital), probabilistic attacks aim at reducing the overall uncertainty of the adversary about an individual, even without identifying a particular attribute.

2.3 Protection of locations

Trajectories correspond to a path or trace generated or drawn by a moving object, usually referred to as an individual or user [13]. Protecting a trajectory can involve protecting data at different levels of granularity:

- protection of a single location,
- protection of OD-matrices,
- protection of trajectories.

2.3.1 Protection of a single location

Several studies focus on the anonymization of individual points. This scenario may involve considering each point as indipendent from the others, segmenting a trajectory as if its points are not related each other, or considering them as Points of Interests (POIs), a point which also has a semantic meaning (e.g. “coffee shop” or “work”). In the literature different algorithms have been proposed for this task. Even if created for a different purpose, one of this is the Mondrian algorithm [35]. Another possibility is given by [36]: this study is defining what a stay point is and what an individual interesting location is to protect the areas which are more dense in terms of point. This work will better explain those studies later in 2.5.1.

2.3.2 Protection of OD-matrices

A trajectory can also be represented in a simplified form as a pair of points indicating the origin and destination of a trip. In particular, an OD-matrix can be defined as a matrix with origins O_n on the vertical axis, and destinations D_m on the horizontal one, or vice versa, where n is the number of origins and m is the number of destinations. Each origin and destination can represent either a single precise point or a larger area. The entries of the matrix $v_{n,m}$ are the values associated to the couple (O_n, D_m). This value represent the number of trips from O_n to D_m. Even if a dataset does not explicitly contain an OD-matrix, it is straightforward to derive one from a collection of trajectories. An OD-matrix essentially condenses a trajectory into just its start and end points, representing each trip as a movement from an origin to a destination. For this reason, a two-feature dataset with origin and destination is enough to create and work with an OD-matrix, along with a measure such as the number of trips between them.

$$\mathbf{OD} = \begin{array}{c|cccc} & D_1 & D_2 & \cdots & D_m \\ \hline O_1 & v_{1,1} & v_{1,2} & \cdots & v_{1,m} \\ O_2 & v_{2,1} & v_{2,2} & \cdots & v_{2,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O_n & v_{n,1} & v_{n,2} & \cdots & v_{n,m} \end{array}$$

Even if this representation is a dramatic simplification compared to trajectories, they remain a crucial tool for mobility analysis [20]. By condensing trips into their essential origin–destination pairs, they provide exactly the information needed to estimate travel demand, plan transportation infrastructures, manage congestion, and assess socio-economic interactions between areas. They also serve as a practical compromise between data utility and privacy, since they preserve the key structure of mobility flows without exposing detailed individual trajectories .

In the literature, several studies are addressing the problem of the anonymization of OD-matrices from different perspectives. Some algorithms are proposed by [37] and [20]. The former is proposing a different version of the classic k-anonymity assuming to know which is the maximum knowledge of an adversary. This new privacy model has been called k^m-anonymization. The latter is proposing an algorithm for protecting OD-matrices using the concept of k-anonymity. Another possibility is to apply the Mondrian algorithm, extending it protect multiple attributes at once. These and other relevant works will be discussed in more detail later in 2.5.1 and 2.5.2.

2.3.3 Protection of trajectories

The definitions of trajectories used in this section follow those provided by [13].

Different types of trajectories exist. The most basic form is the raw trajectory, defined as an ordered sequence of spatio-temporal points $T =< p_1, \ldots, p_m >$ where $|T| \doteq m$ denotes the length of T and $p_i = (x_i, y_i, t_i)$ represent the position (x_i, y_i) at timestamp t_i. Moreover, trajectories respect the temporal order $(t_{i+1} > t_i)$, ensuring that movements never go back in time, and that no one is in two different locations simultaneously.

A more expressive representation is the semantic trajectory, where each spatio-temporal point contains additional information such as a name or description (e.g., "coffee shop" or "work"), or further attributes like opening hours and visitor counts. As discussed in 2.3.1, these locations are referred to as Points of Interest (POIs). In addition to this, it is possible to have even much more complex trajectories called *multiple aspect trajectories*. They can contain any possible type of recordable information, like weather variations, transportation mode, or the current heart rate or emotions of individuals [38]. To reduce complexity, simplified trajectories are sometimes used: these omit timestamps and focus on the order of locations, such as $T = (x_i, y_i), \ldots, (x_m, y_m)$.

Once the different types of trajectories have been defined, it is time to understand how to handle a set of them. Trajectory databases consist of one or multiple trajectories from individuals, usually over a shared region. They can be represented them as collections of rows, where each row contains the data of a single individual:

$$D = \begin{cases} T_1 : & p_1^{(1)} & p_2^{(1)} & \cdots & p_{m_1}^{(1)} \\ T_2 : & p_1^{(2)} & p_2^{(2)} & \cdots & p_{m_2}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_r : & p_1^{(r)} & p_2^{(r)} & \cdots & p_{m_r}^{(r)} \end{cases}$$

where T_i represents a trajectory belonging to user i. The length of each trajectory is given by m_i and depends by the single user. There are cases in which the same user can contribute multiple trajectories to the database. In this case, i is just a label of the trajectory and does not necessarily relate to a user.

As for trajectories, there are differences in structure between such databases. Some consist only of trajectories of equal length, and others assume that trajectories are periodically recorded (i.e., every trajectory has a spatio-temporal point for every time interval defined). Further types include those with irregular recordings, such as with points only included when the user is in a relevant location. A particular scenario in trajectory publishing is the data-stream scenario, where a flow of information is received and published periodically. Therefore, a streaming database can be seen as a sequence $D = \{S_1, \ldots, S_t, \ldots, S_i\}$, where each update

S_i represents the information corresponding to time i:

$$D = \begin{cases} & S_1 & S_2 & \cdots & S_{m_j} \\ T_1: & p_1^{(1)} & p_2^{(1)} & \cdots & p_{m_1}^{(1)} \\ T_2: & p_1^{(2)} & p_2^{(2)} & \cdots & p_{m_2}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_r: & p_1^{(r)} & p_2^{(r)} & \cdots & p_{m_r}^{(r)} \end{cases}$$

The database at time t is denoted $D_t = \{S_1, \ldots, S_t\}$ and called a *stream prefix*. Since some databases consist of non-periodically recorded trajectories, gaps in this representation are possible. For this reason, T_i may not have a location for time t, and remain empty in row i of S_t.

The structure of trajectory data and databases makes their protection particularly challenging. The high sparsity and uniqueness of trajectories can easily lead to re-identification [13]. In addition, the semantic information associated with trajectories introduces further risks, as it may reveal personal habits and individual preferences. High sparsity refers to the fact that individuals are scattered across a vast geographical space. This makes it difficult to form sufficiently large anonymity sets without excessive generalization. Merging trajectories often requires aggregating them into very large geographical areas, leading to a significant loss of data utility. Moreover, a few anchor points, such as home and work locations, often form a unique fingerprint that can re-identify an individual in a large dataset. This uniqueness prevents individuals from "hiding in the crowd," as no natural crowd of identical trajectories exists.

In this scenario several algorithms have been proposed. In 2.5.3 the most significative will be discussed, including the grid-based anonymization, dummy trajectories (both using the Classic Random Scheme and Rotation Pattern Scheme), the Mix Zones, K-δ anonymity (NWA) and the DPTD.

2.4 Privacy notions

It is possible to categorize privacy models for the release of anonymized trajectory data as formal and ad-hoc models [26]. Formal models are independent from the data type, and extend the existing principles (e.g., k-anonymity, l-diversity, t-closeness, and differential privacy) to trajectories. Ad-hoc models are specific to spatiotemporal data and mobility features (e.g., road network constraints). This last category will be presented later in 2.5.3.

Attributes of a dataset can be classified as Identifiers, QuasiIdentifiers, Confidential, and Non-confidential. The easiest step to have anonymization is believed to be only removing all the Identifiers, that is, attributes that with no doubt can identify

an individual (such as Social Security Number, Passport, Name-surname). However, exploiting unique combinations of attribute values can reidentify unambiguously an individual in a database even without explicit identifiers. Those attributes are called quasi-identifiers (QIs). Thus, since QIs can be used to relate anonymized records to external non-anonymous databases, this may lead to re-identification. And by reidentifying an individual in a database his Confidential (sensitive) attributes may be revealed (e.g., Salary, Medical conditions, etc.) Therefore, anonymization techniques must deal also with QIs.

The two main models for privacy protection, from which many others have been developed, are k-anonymity [21] and ϵ-differential privacy [22].

2.4.1 K-anonymity

A dataset is k-anonymous if each record is indistinguishable from at least other $k-1$ records within the dataset, when considering the values of its QIs. This guarantees that the individuals cannot be re-identified by linking attacks with probability less than $\frac{1}{k}$.

Name	**Age**	**ZIP code**	**Disease**
Marco	34	20100	Diabetes
Luca	34	20100	Cancer
Anna	42	00100	Flu
Paolo	42	00100	Flu

Table 2.2: Example of a 2-anonymous table.

Consider the example shown in Table 2.2. Names are present to better clarify the rows we are discussing, but they should, of course, be removed. The table satisfies 2-anonymity because there are two entries for each group: one composed by Marco and Luca, the other is composed by Anna and Paolo (the age and the ZIP code are the equal respectively). This "group" is called Equivalence Class (EQ) and represents the set of entries indistinguishable each other. Indeed each record in the EQ is indistinguishable from at least k-1 records (the other record). This means that Marco and Luca are indistinguishable and the same for Anna and Paolo.

l-diversity

However, when the sensitive attributes on a group of k-anonymous records, have low variablity (e.g., when they are all equal), there is no need of reidentification to disclose the value of the sensitive attribute of a record. This remark was done in [39], who proposed the model of l-diversity for solving this issue. A k-anonymous

data set is said to be l-diverse if, for each group of records sharing quasi-identifier values, there are at least well-represented values for the sensitive attribute.

Imagine that some anonymization has been done on Table 2.2 generating Table 2.3. Considering to know that either the age or the ZIP code of one of the two individuals in the third or fourth entry, the attacker can find out that his victim has the flu, no matter understanding which is the correct entry of the victim. This happens because there is low (in this case zero) variability in the QIs.

Age	**ZIP code**	**Disease**
30-40	201xx	Diabetes
30-40	201xx	Cancer
40-50	001xx	Flu
40-50	001xx	Flu

Table 2.3: Example showing that even if k-anonymous, a table can still leak information.

On the other hand, considering the Table 2.4 the problem do not occur anymore: even if the attacker know to which group the patient belong to, he cannot deduce with certainty the disease because there are at least 2 different values. This means that the Table 2.4 is 2-diverse for the definition of distinct l-diversity.

Age	**ZIP code**	**Disease**
30-40	201xx	Diabetes
30-40	201xx	Cancer
40-50	001xx	Cancer
40-50	001xx	Flu

Table 2.4: Example of a l-diverse table.

t-closeness

Later, in [40] it was shown that the model of l-diversity does not prevent attribute disclosure when the overall distribution of the sensitive attribute is skewed. Hence, they proposed the t-closeness model. A k-anonymous data set is said to have t-closeness if, for each group of records sharing quasi-identifier values, the distance between the distribution of each sensitive attribute within the group and the distribution of the attribute in the whole data set is no more than a threshold t.

Consider the Table 2.5. It contains one single individual with HIV and all the others with Flu. Even if multiple values exist for the sensitive attribute, an attacker may infer information based on the overall dataset distribution. Even though the

first group has $l = 2$ (HIV and Flu), HIV is rare in the general population. If an attacker knows a person belongs to the first group, they can assume with 33% probability (more than the average) that the victim has HIV.

Age	ZIP code	Disease
30-40	201xx	HIV
30-40	201xx	Flue
30-40	201xx	Flue
40-50	001xx	Flue
40-50	001xx	Flue
40-50	001xx	Flue

Table 2.5: Example of a table with a skewed sensitive distribution.

2.4.2 Differential privacy

Imagine a database with information about people. Differential privacy ensures that whether any single individual's data is included in the database or not, the results of any analysis will be almost identical. This means an attacker can't learn much about any specific person by looking at the analysis results. It can be considered as adding carefully calibrated "noise" to the data or query results. This noise is random enough to protect individuals but small enough that overall patterns and statistics remain accurate.

The following definitions of this section are provided by [41].

Given a randomized algorithm A, the algorithm A satisfies ϵ-differential privacy if for two neighboring datasets D and D', and all the possible outputs O ($O \in$ *Range (A)*), *Range (A)* represents the output range of A:

$$Pr[A(D) = O] \leq e^{\epsilon} \times Pr[A(D') = O] \tag{2.1}$$

where the Pr[·] denotes the probability of a user's privacy leakage. This means that any neighboring datasets D and D' that have the same data structure and only have one record difference between them. The parameter ϵ is the private budget that controls the degree of privacy protection. A smaller ϵ corresponds to stronger privacy protection, and vice versa.

For a query function $f : D \to R^d$ and any neughboring datasets D and D', the global sensitivity of the function f is:

$$\Delta f = \max_{D,D'} ||f(D) - f(D')||_p \tag{2.2}$$

where R is the real number field mapped by dataset D, d denotes the query dimension of function f, and p is used to measure the norm distance of Δf, and generally, $p = 1$.

Let $f : D \rightarrow R^d$ denote a query function over a dataset D, then a random algorithm A satisfies ϵ-differential private if its output is

$$A(D) = f(D) - Lap(\Delta f/\epsilon) \tag{2.3}$$

where Δf is the global sensitivity of the function, $Lap(\Delta f/\epsilon)$ is a random variable sampled from the Laplace distribution, and the density function of the Laplace distribution is as follows:

$$p(x) = \frac{\epsilon}{2\Delta f} e^{-|x|\epsilon/\Delta f} \tag{2.4}$$

The Laplace distribution has a mean of 0 and $2(\Delta f/\epsilon)$ variance. The amount of noise is proportional to the f and inversely proportional to the private budget ϵ; that is, if f is fixed, the smaller the ϵ, the more the noise injected and the higher the degree of privacy, and vice versa.

Differential privacy has two essential properties: sequential composition and parallel composition. Sequential composition prescribes that if a sequence of computations is performed on the same data, each part provides differential privacy independently, then the privacy guarantee of the entire sequence is accumulated. Parallel composition stipulates that if a sequence of computations is carried out on disjoint subsets of data, the entire sequence provides the worst privacy guarantee.

Level of granularity

When applying Differential Privacy (DP), it is crucial to define precisely what information is being protected. This depends on the concept of neighboring databases considered. For this reason, different adaptations of the concept of neighborhood have been suggested in the literature.

In trajectory data the concept of granularity is particularly relevant. The neighborhood definition directly impacts the privacy guarantee offered. [13] explore the most common granularity notions. In the following paragraphs a full explanation is given.

User-level privacy Consider two databases D and D', they are user-level neighboring if they differ only in the data of a single user. For example, if each user contributes a single trajectory, then two databases D and D' differ when one trajectory is removed, added, or replaced. When users contribute multiple trajectories, the definition extends to all the trajectories of that user.

Event-level privacy This notion aims to hide the presence or absence of a single event in a user's data. Two streaming databases D and D' are event-neighboring

if they differ by exactly one spatio-temporal point. In practice, changing a single point of a trajectory makes the two databases event-neighbors.

$$D = \begin{cases} & S_1 & S_2 & \cdots & S_m \\ T_1: & p_1^{(1)} & p_2^{(1)} & \cdots & p_m^{(1)} \\ T_2: & p_1^{(2)} & \mathbf{p_2^{(2)}} & \cdots & p_m^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_r: & p_1^{(r)} & p_2^{(r)} & \cdots & p_m^{(r)} \end{cases}, \quad D' = \begin{cases} & S_1 & S_2 & \cdots & S_m \\ T_1: & p_1^{(1)} & p_2^{(1)} & \cdots & p_m^{(1)} \\ T_2: & p_1^{(2)} & \hat{\mathbf{p}}_2^{(2)} & \cdots & p_m^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_r: & p_1^{(r)} & p_2^{(r)} & \cdots & p_m^{(r)} \end{cases}$$

The guarantee this notion gives is that each point in the database remains inaccessible to an attacker. However, this notion has some drawbacks.

First, it still allows risks of identity disclosure. Event-level privacy guarantees that if the attacker knows a single spatio-temporal point of a single trajectory, the probability of re-identification is bounded by ϵ. However if an adversary knows $r > 1$ points of a trajectory, the protection decreases to $r\epsilon$ [42]. Since real-world trajectories often contain hundreds of points, the re-identification risk remains high. Second, it does not fully prevent attribute disclosure. For example, if a user visits a hospital multiple times, the fact "visited a hospital" is not hidden. Finally, it is vulnerable to correlation attacks.

***w*-event privacy** This notion can be seen as the one that makes points of the database over w consecutive timestamps undetectable. This definition has been suggested by [43].

This definition protects data where sensitive information is disclosed from a sequence of events of length w. It not only protects the locations visited by a single user over w consecutive timestamps but can also protect those of different users. In terms of privacy, for values of w close to 1, w-event privacy approximates to event-level privacy, while for large values, it converges to user-level privacy. Moreover, this notion protects more information than event-level privacy while allowing less noise addition than user-level, even though some of its deficiencies still exist.

The notion still leaks attributes when these cannot be protected by the same w-window. For example, assume user u_1 in Figure 2.1 (where $w = 3$) is a compulsive gambler and visits the casino (red dot) multiple times a day. The sensitive information that u_1 has been at the casino is not protected as the red dots cannot fit into a unique w-window. The user's identity is still unprotected if the attacker's knowledge exceeds the window.

Given that consecutive spatial points are usually more correlated, this notion is better than event-level privacy against correlation attacks. However, the assumption of w-event privacy that trajectories are periodically recorded, may overestimate the number of consecutive protected locations. For instance, in Figure 2.1, where

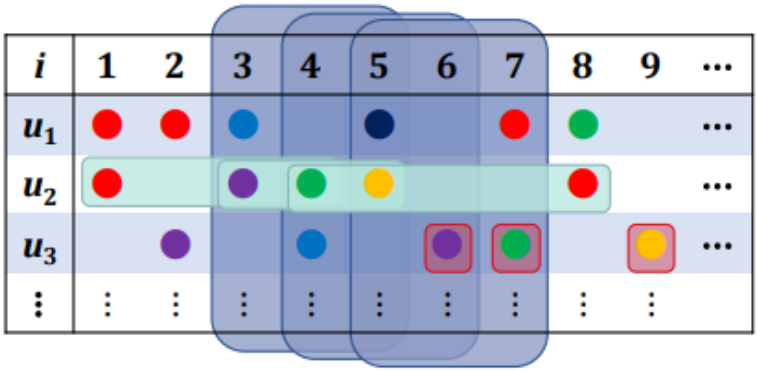

Figure 2.1: An example of a non-periodically recorded streaming database [13], colored dots represent different locations. The rounded boxes represent protection scopes of event-level (red), w-event (blue), and l-trajectory privacy (green), for $w = l = 3$.

non-periodically recorded trajectories are present, the 3-window cannot protect more than two locations of a single user.

***l*-trajectory privacy** In order to overcome the problem of users' trajectories which are not periodically recorded, [44] introduces the concept of l-trajectory privacy.

The goal of the l-trajectory privacy notion is to protect each sequence of l points from the same user independently of the number of timestamps they span. This means that, while in the w-event privacy a fixed rectangle was protecting the users (a sort of time window), the l-trajectory is protecting l points independently from how far their are in time. Varying l allows us to move closer to event-level ($l = 1$) or to user-level privacy ($l \to \infty$).

Although this notion overcomes the problem of w-event privacy of assuming periodically recorded trajectories, it does not address its other deficiencies.

Element-level privacy Finally, [45] propose element-level privacy, designed to prevent disclosure of specific sensitive attributes rather than all events. For instance, in a traffic study, users may accept revealing "owns a car" but wish to hide "visited a hospital."

They model data of a user u as a multiset of values $x^{(u)} = \{x_1^{(u)}, x_2^{(u)}, \ldots, x_{m_u}^{(u)}\}$, where each $x_i^{(u)}$ belongs to the universe of possible values of X. Then it considers a K-partition of the universe X into the clusters $c_1, \ldots, c_K$. These clusters are viewed as the elements to be protected. By definition, each $x_i^{(u)}$ belongs to one cluster c_j.

Ensuring element-level privacy means hiding that each user has elements belonging to the cluster, independently of how many elements it includes. The authors believe that this notion can be adapted to trajectory data: data points can be clustered according to geographical zones and times. And in the case of semantic trajectories, even according to semantic values, e.g., having a cluster for all health-related locations.

Type of privacy	Difference between neighboring databases
User-level	A user's whole trajectories
Event-level	A spatio-temporal point visited by a user (an event)
w-event	A window of events over w consecutive timestamps
ℓ-trajectory	A sequence of ℓ consecutive spatio-temporal points from a single user
Element-level	A user's set of points belonging to the same cluster

Figure 2.2: A summary of the different level of granularity [13].

2.5 Algorithms in the literature

As stated in previous paragraphs, protecting trajectories may mean protecting different part of them (such as one single point, a couple of point or even the whole trajectory). Based on this choice, there are different types of algorithms that are presented in literature.

2.5.1 Algorithms protecting the single location

Mondrian

The Mondrian algorithm [35] is a greedy algorithm based on a multidimensional recording model.

In general a global recording achieves anonymity by mapping the domain of the quasi-identifier attributes to generalized or alter values. Global recording can be further broken down into two sub-classes: the single-dimension global recording and the multidimentional global recording.

Assuming there is a total order associated with the domain of each quasi-identifier attribute X_i, a single-dimensional partitioning defines, for each X_i, a set of non-overlapping single-dimensional intervals that cover D_{X_i} to some summary statistic for the interval in which it is contained. This partitioning model can be easily extended to multidimensional recording defining a multidimensional region as a pair of d-tuples $(p_1, ..., p_d), (v_1, ..., v_d) \in D_{X_i} \times ... \times D_{X_d}$ such that $\forall i, p_i \leq v_i$. Basically,

this means that each region is bounded by a d-dimensional rectangular box. A strict multidimensional partitioning defines a set of non-overlapping multidimensional regions that cover $D_{X_i} \times ... \times D_{X_d}$. ϕ maps each tuple $(x_1, ..., x_d) \in D_{X_i} \times ... \times D_{X_d}$ to summary statistic for the region in which it is contained. In the picture 2.3 it is possible to see the difference between the two partitionings. Assuming that on the horizontal axis there is the ZIP code and on the vertical axis there is the age of the person in the database, a single-dimensional partitioning is dividing the space along one single axis (in this case the ZIP code). On the other hand, the strict multidimensional partitioning has first divided the space along the ZIP code and then along the age.

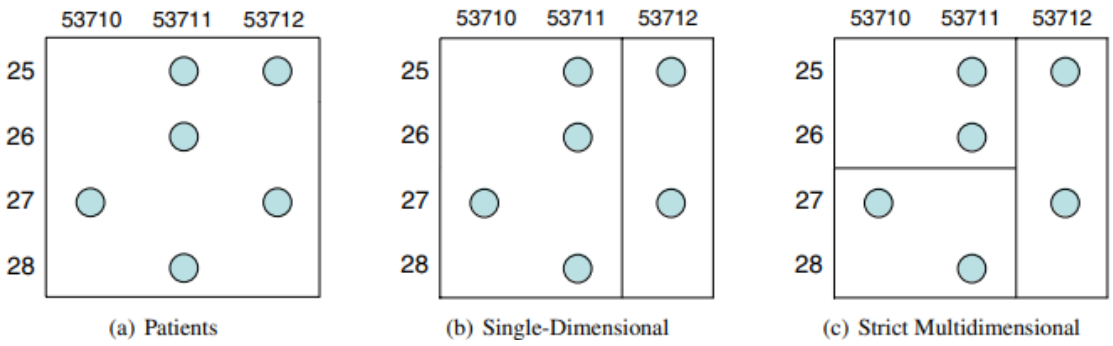

Figure 2.3: Spatial representation of the single dimensional and multidimensional recording [35].

At this point it is important to define what a cut is and when it is admissible. Considering a multiset P of points in d-dimensional space, a cut perpendicular to axis X_i at x_i is allowable if and only if $Count(P.X_i > x_i) \geq k$ and $Count(P.X_i \leq x_i) \geq k$. This means that the first cut done on the space have to satisfy k-anonymity in both the partitions created (both the partitions must have at least k elements). If a new cut is done on the other axis, it is crucial to check the same process: are there at least k elements on the new two partitions created? The process keeps going on until it is no longer possible to create partitions.

The Mondrian algorithm is leveraging those definitions to obtain k-anonymity. Given a d-dimensional space:

1. the algorithm is checking the spans of each dimension, which means obtaining the intervals of the d dimensions as the difference between its maximum value and its minimum value.

2. The algorithm picks the dimension with the biggest span and it splits it on the median, checking that the partitions created contains at least k elements.

3. At this point the algorithm is computing again the spans on the still-non-divided dimensions and on the new partitions created. It will pick the one with the biggest span and will divide it on the median.

4. The process will keep going on until the algorithm is not able to find anymore new partitions that contains at least k elements.

The main strength of the Mondrian algorithm lies in its ability to operate in spaces of more than two dimensions, creating multidimensional partitions rather than being limited to purely spatial data. However, its main drawback hide in the nature of the rectangular regions it produces: these can vary significantly in size, sometimes becoming excessively large and thus too coarse to preserve utility, or excessively small, leading to over-partitioning and potential privacy risks. Moreover, since the partitions are axis-aligned and not inherently hierarchical, the resulting structure may fail to represent geographic areas in a consistent or meaningful way.

Interesting location mining algorithm

This algorithm is discussed in [36] and it wants to ensure that an individual's participation in a statistical database does not substantially increase the risk to his privacy. This process must account for the possibility of combining internal attributes with external data to uniquely identify individuals. However, subject to this constraint, it is important that the released data remain as "useful" as possible. More in particular, figure 2.4 shows that, if differential privacy is not preserved, when an individual decides to opt out from the location history database, an interesting location can change from Region A to Region A'. This change can give to the adversary more specific information related on the location that the individual visits regularly. This algorithm promise to protect against this kind of attack proposing a new differentially private solution.

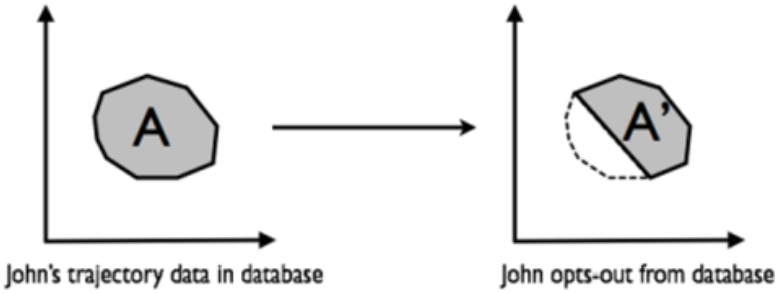

Figure 2.4: An interesting location changes from Region A to Region A' when John opts out from the location history database [36].

First, some definitions are necessary. A stay point is the center (x, y) of a circle

region with a δ radius in which a trajectory stays for at least a time period of T. An individual interesting location is a region containing more that r stay points for the individual. Another possible definition of an interesting location is a region that satisfies the condition that it is an individual interesting location for at least m individuals if each individual has more than r stay points in the region.

In the conventional differential privacy solution, the privacy mechanism is applied to the output results. This solution consists of applying the privacy mechanism to both the pre-processing step and the algorithm outputs. The first step splits the main problem into smaller subproblems and enables one to obtain reasonable "local" sensitivity to the subproblems. Then, the Laplace noise perturbation privacy mechanism is applied to the DBSCAN outputs, namely: the interesting regions and their corresponding stay point counts, in the interesting location mining algorithm.

The interesting location mining algorithm consists of two steps:

- algorithm 1: differentially private region quadtree-based spatial decomposition, and
- algorithm 2: differentially private interesting location extraction based on DBSCAN clustering algorithm.

The first algorithm takes as input the set of stay points S, the spatial region R, and a threshold parameter T. To guarantee privacy, Laplace noise is first applied to the cardinality of S, generating a perturbed count S'. This perturbed value is then used to decide whether the region should be further partitioned using a quadtree structure:

- if the noisy count is greater than $3T$, the region is further divided;
- otherwise, the process stops, and the region is defined as a partition P, with S_P denoting the subset of stay points contained within it.

The output of this step is therefore the set of partitions together with their associated subsets of stay points.

The second algorithm takes as input the subsets of stay points associated with the partitions from Algorithm 1, together with a refined threshold parameter r' and the parameters of the DBSCAN clustering algorithm. For each subset, DBSCAN is applied in order to detect dense groups of points. For each resulting cluster:

- Laplace noise is added to the count of stay points in the cluster;
- if the noisy count exceeds r', the cluster is considered an interesting location with its corresponding noisy count;
- the centroid of the cluster is computed as the mean of the points, and Laplace noise is added to its coordinates, which are then used to identify the location.

The output of this step consists of the set of clusters identified as interesting locations and their noisy counts.

2.5.2 Algorithms protecting OD-matrices

OIGH

The OIGH (Optimal Identical Generalization Hierarchy) algorithm [46] was developed specifically for datasets in which all quasi-identifiers share the same generalization hierarchy, a category known as Identical Generalization Hierarchy (IGH) data. This situation occurs when sensitive attributes belong to the same domain and can therefore be generalized using a single hierarchical structure.

The key idea behind OIGH is to exploit four properties that characterize IGH data and allow a more efficient search for k-anonymity:

- if a node is k-anonymous, then all its direct generalizations at higher levels are also k-anonymous;
- if a node is not anonymous, all its direct specializations at lower levels will also fail to satisfy k-anonymity;
- nodes located at the same level of the lattice share the same precision;
- precision always increases when moving upward in the lattice.

These properties ensure that the optimal k-anonymous solution will always be found at the lowest level where k-anonymous nodes appear in the lattice.

To exploit this, OIGH adopts a depth-first search strategy, exploring the lattice from the root node downwards. Once a k-anonymous node is detected, all its ancestors at higher levels can automatically be considered k-anonymous, without further checks. Conversely, if a node is identified as non-anonymous, all its descendants at lower levels are automatically discarded.

The strength of this approach is that it avoids examining large portions of the lattice: the search focuses only on levels below the first k-anonymous level encountered, which results in a significant speedup compared to traditional uniform generalization algorithms that must scan the full solution space.

Nonetheless, OIGH has two important limitations. First, it can only be applied to IGH datasets, which represent a rather specific class of data. Second, like other uniform generalization methods, it applies the same level of generalization to all values of each quasi-identifier. This can lead to unnecessary information loss when the data distribution is uneven or heterogeneous.

ATG

Adaptive Tree Generalization (ATG) methods [20] represent a family of algorithms designed to achieve k-anonymity in OD-matrices by exploiting a hierarchical representation of space. The general idea behind these methods is to view the anonymization process as an optimization problem structured over a spatial tree, where each node represents a geographic area at a specific level of aggregation. The goal of ATG approaches is to minimize information loss while ensuring that each generalized region includes at least k individuals or trips. To do this, the algorithms iteratively generalize origins and destinations according to a cost function that balances two factors:

- the spatial generalization cost, which increases as areas become larger; and
- the suppression cost, associated with removing OD pairs that cannot be safely anonymized.

Two main variants of the algorithm exist: ATG-Dual and ATG-Soft. While both share the same conceptual foundation, they differ in computational complexity and flexibility. ATG-Dual seeks an optimal balance between generalization and suppression through dual optimization, whereas ATG-Soft simplifies this process by fixing the balancing parameter in advance. The following section focuses on ATG-Soft, which is the version adopted for the experiments in this work.

ATG-Soft The ATG-Soft algorithm [20] is a simplified version of the ATG-Dual method, designed to anonymize large-scale origin–destination matrices more efficiently. Its goal is to achieve k-anonymity while obtaining a balance between spatial generalization and information loss.

The idea behind ATG-Soft is to treat generalization and suppression as an optimization problem defined over a tree-structured hierarchy of values. Unlike ATG-Dual, which requires solving a dual problem to find the optimal value of the parameter λ, ATG-Soft fixes this parameter in advance, usually at 10% of the total map size. This simplification makes the process much lighter computationally.

The algorithm works in two phases.

- Phase 1: Origin Generalization. ATG-Soft starts by partitioning the set of origins. It uses an objective function that minimizes the gap between the total number of trips leaving each origin and a target volume v_{target}. This ensures that the corresponding destination maps are generalized to a comparable degree, preventing imbalances across origins.
- Phase 2: Destination Generalization. Once origins are partitioned, the algorithm moves on to destinations. Here, it solves a simplified pruning problem

that balances two costs: the aggregation penalty (α_d) and the suppression penalty (σ_d), with λ acting as a regularization parameter.

The optimization problem can be expressed as:

$$\min_x \sum_{d \in T} (x_p(d) - x_d)(\alpha_d - \lambda \sigma_d) - \lambda C \tag{2.5}$$

where α_d is the aggregation penalty, σ_d is the suppression penalty, and C is the suppression constraint.

A key property of ATG-Soft is that it enforces a strict constraint on the generalization level: no flow $o \to d$ can be generalized beyond $|o| + |d| > \lambda$, even if this means suppressing the record altogether. This guarantees that spatial resolution never drops below a minimum acceptable threshold.

The strengths of ATG-Soft are related on its efficiency, which permits it to avoids the heavy dual optimization step making it faster than ATG-Dual, and scalability because yt can process large OD matrices within reasonable time. On the other hand, its main drawbacks are the parameter tuning and the risk on imbalance. Choosing a good value for v_{target} requires either expert knowledge or empirical testing. Moreover, since origins and destinations are handled separately, poor parameter calibration can result in uneven generalizations.

Overall, ATG-Soft can be seen as a compromise: it sacrifices some precision for a large gain in speed. This makes it well-suited for scenarios where massive volumes of mobility data need to be anonymized quickly, and where moderate privacy guarantees are considered acceptable.

K^m-anonymization: a priori anonymization

In this section is presented the problem of publishing set-valued data, while trying to preserve the privacy of individuals associated to them. This is done introducing a new extended concept for the k-anonymity: the k^m-anonymity.

Consider a database D, which stores information about items purchased at a supermarket by various customers. If the adversary has some partial knowledge about a subset of items purchased by a person, observing the direct publication of D may result in unveiling the identity of that person associated with that particular transaction. For example, assume that Bob went to the supermarket on a particular day and purchased a set of items including coffee, bread, brie cheese, diapers, milk, tea, scissors, light bulb. Assume that some of the items purchased by Bob were on top of his shopping bag (e.g., brie cheese, scissors, light bulb) and were spotted by his neighbor Jim, while both were on the same bus. Bob would not like Jim to find out other items that he bought. However, if the supermarket decides to publish its transactions and there is only one transaction containing brie cheese, scissors, and

light bulb, Jim can immediately infer that this transaction corresponds to Bob and he can find out his complete shopping bag contents [37].

In [37], authors propose such a k^m-anonymization model, for transactional databases: assuming that the maximum knowledge of an adversary is at most m items in a specific transaction, the model wants to prevent him from distinguishing the transaction from a set of k published transactions in the database. Equivalently, for any set of m or less items, there should be at least k transactions, which contain this set, in the published database *D'*.

This concept can be extended to trajectories. Imagine a dataset containing the columns `latitude_start`, `latitude_end`, `longitude_start` and `longitude_end`. If the dataset is 5^2-anonymous, this means that there are at least 5 sample of the dataset with the same 2 subset of the four core column. This case is a restricted case because, theoretically m can go from 1 to 4, but in the reality the attacker will know the latitude and/or the longitude. For this reason m can strictly be 2 or 4.

Unlike the k-anonymity problem in relational databases, there is no fixed, well-defined set of quasi-identifier attributes and sensitive data. A subset of items in a transaction could play the role of the quasi-identifier for the remaining (sensitive) ones and vice-versa. This means that, assuming that the dataset is 5^2-anonymous, both the case in which the attacker know the latitude and the longitude is covered.

If D is not k^m-anonymous, it can be transformed to a k^m-anonymous database *D'* by using generalization. There are different anonymization techniques that can be used: the count-tree, the optimal anonymization, the direct anonymization and the apriori-based anonymization [47]. The one that is explained more in detail in this study is the apriori-based anonymization.

The algorithm is inspired by the apriori principle, which states that if an itemset J of size i leads to a privacy breach, then every superset of J will also cause a breach. This principle allows us to progressively enforce the necessary generalizations. Specifically, the algorithm evaluates potential privacy risks in increasing order of adversarial knowledge: starting from the case where the adversary knows only one item per trajectory, then two, and so forth, up to the case where the adversary may know m items.

At each iteration i, the database is scanned and a count-tree is populated with all i-itemsets. Itemsets containing elements already generalized are ignored. Each transaction t is first expanded to include all possible generalizations, and then reduced by removing items already generalized. For example, after the first iteration ($i = 1$), suppose the generalization map contains $< \{a_1, a_2\} \rightarrow A >$. In the second iteration ($i = 1$), the transaction $t_4 = \{a_1, a_2, b_2\}$ is first expanded to $t_4 = \{a_1, a_2, b_2, A, B\}$, and then reduced to $t_4 = \{b_2, A, B\}$ since a_1 and a_2 are already generalized. This process considerably reduces the number of candidate itemsets to be inserted into the count-tree. The algorithm proceeds as follows.

- An empty generalization map and a set of reduced transactions are initialized.

- The reduced transactions are filled with items that are not yet present in the generalization map.
- For each iteration $i = 1, ..., m$:
  - all reduced transactions are explored to generate combinations of i elements (i-itemsets);
  - the frequency of each combination is counted;
  - if an itemset is found to have a support smaller than k, it is generalized.

2.5.3 Algorithms protecting trajectories

Grid-based anonymization

The basic idea of grid-based generalization is to partition the data space into grids such that all points falling into the same grid are uniformly represented by the grid.

In the example in Figure 2.5: a trajectory $(P_1, P_2, ..., P_8)$ with eight points is fit in a 2D space that is partitioned into six grids denoted as $G_1, G_2, ..., G_6$. Then the trajectory is transformed into a new format (G_4, G_5, G_2) with respect to time intervals $P_1.t$-$P_3.t$, $P_4.t$–$P_6.t$ and $P_7.t$–$P_8.t$.

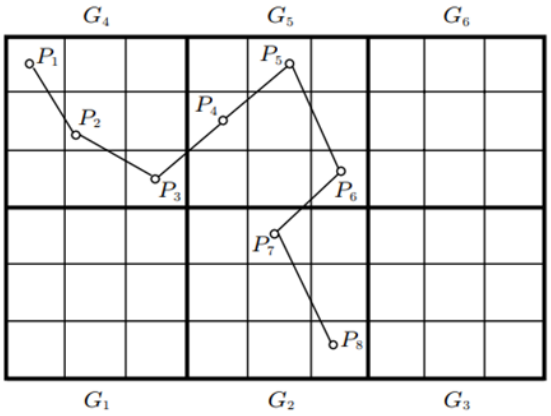

Figure 2.5: Example of grid-based generation [48]

There actually exists different approaches for partitioning the space [49]:

- Common Regular Partitioning (CRP): the simplest method is to define a single, regular partitioning that is used by all the objects.
- Individual Regular Partitioning (IRP): not all objects require the same level of location privacy. Objects requiring higher levels of privacy construct and

use a regular partitioning with larger partitions, while objects requiring lower levels of privacy define and use a regular partitioning with smaller partitions.

- Individual Irregular Partitioning (IIP): objects may have different location privacy requirements in different regions of space. For example, most objects (users) desire a higher level of location privacy when being at home or the work place than when being in transition or when being in other general areas of the city. Objects can be allowed to individually define privacy levels for regions in space that reflect their needs. The definition of these regions can be either manual, or can be aided by discovering frequent (presumably sensitive) locations of individual objects.

Mix zones

The concept of "mix" has been applied to anonymous communication in a network. A mix-network consists of normal message routers and mix-routers. The basic idea is that a mix-router collects k equal-length packets as input and reorders them randomly before forwarding them, thus ensuring unlinkability between incoming and outgoing messages [50]. This concept has been extended to LBS, namely, mix-zones [51].

When users enter a mix-zone, they change to a new, unused pseudonym. In addition, they do not send their location information to any location-based application when they are in the mix-zone. When an adversary that sees a user u exits from the mix-zone cannot distinguish u from any other user who was in the mix-zone with u at the same time. The adversary is also unable to link people entering the mix-zone with those coming out of it.

A set of users S is said to be k-anonymized in a mix-zone Z if all following conditions are met [50]:

- The user set S contains at least k users, i.e., $|S| \geq k$.
- All users in S are in Z at a point in time, i.e., all users in S must enter Z before any user in S exits.
- Each user in S spends a completely random duration of time inside Z.
- The probability of every user in S entering through an entry point is equally likely to exit in any of the exit points.

Mix-zones impose limits on the services available to mobile users inside a mix-zone because they cannot update their locations until exiting the mixzone. To minimize disruptions caused to users, the placement of mix-zones in the system should be optimized to limit the total number of mix-zones required to achieve a certain degree of anonymity.

In a road network, vehicle movements are constrained by many spatial and temporal factors, such as physical roads, directions, speed limits, traffic conditions, and road conditions. For this reason mix-zones must be designed in a proper way to protect trajectory privacy in road networks. This is because an adversary can gain more background information from physical road constraints and delay characteristics to link entering events and exiting events of a mix-zone with high certainty. For example imagine two cars entering a mix-zone: the first with a speed of 10km/h; the second with a speed of 100km/h. An attacker can easily understand that the first one exiting the mix-zone will be the fastest one. Or if turn is not allowed in the intersection, an adversary knows that a vehicle entering the mix-zone from a specific road can only go in one side.

An effective solution for vehicular mix-zones is to construct non-rectangular, adaptive mix-zones that start from the center of an road segment intersection on its outgoing road segments, as depicted in Figure 2.6. The length of each mixzone on an outgoing segment is determined based on the average speed of the road segment, the time window, and the minimum pairwise entropy threshold.

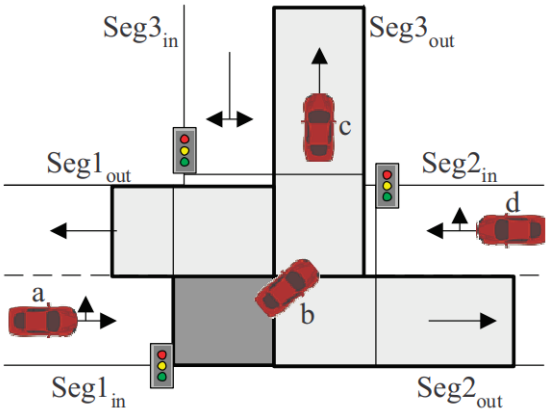

Figure 2.6: Non-rectangular, adaptive vehicular mix-zones [50]

Dummy trajectories

Without relying on a trusted third party to perform anonymization, a mobile user can generate fake location trajectories, called dummies, to protect trajectory privacy [50]. Given a real user location trajectory T_r and a set of user-generated dummies T_d, the degree of privacy protection for the real trajectory is measured by the following metrics [52]:

- Short-term Disclosure (SD). This parameter specifies requirement for protecting the current user location. Given a set of current locations (including both

true and dummy locations), SD is the probability of successfully identifying the true user location

$$SD = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{|D_i|} \tag{2.6}$$

where m is the number of time slots in a trajectory, D_i is the set of true and dummy locations at the i-th time slot, and $|D_i|$ is the size of D_i.

- Long-term Disclosure (LD). This parameter specifies requirement for protecting the user trajectory. Given n trajectories, among which k trajectories have intersected with other trajectories and $(n-k)$ trajectories do not have any intersection. Thus, for those $(n-k)$ trajectories, there are exactly $(n-k)$ possible trajectories. For those k trajectories, all possible trajectories have to be considered by exhaustively traversing intersections from the start point of each trajectory to the end point. The number of possible trajectories among k trajectories is denoted as T_k. Consequently, LD is defined as

$$LD = \frac{1}{T_k + (n-k)} \tag{2.7}$$

- Distance deviation (dst). It is the average of distance difference among trajectories of dummies and the user. It is formulated as

$$dst = \frac{1}{m} \frac{1}{n} \sum_{k=1}^{n} \sum_{j=1}^{m} dist(PL_i^j, L_{dk}^i) \tag{2.8}$$

where *dist* is distance between the true user location and dummy locations in unit of cell size.

Once defined those metrics, it is possible to define the concept of privacy profile, which is the combination of the metrics that the user wants to satisfy. For this reason, given a privacy profile, dummy trajectories are generated to satisfy the user privacy profile. The most known schemes are the random and rotation pattern schemes

Random Pattern Scheme In this scheme, the starting point and the destination of a dummy are first selected. Then, a dummy will move randomly from the starting point to the destination, dividing the space into cells created taking into account the speed of the dummy and its movement in the space. This naive scheme demonstrates that even after a long term observation, it's difficult for adversaries to identify true user since dummies also exhibit long-term, consistent movement patterns [52].

Rotation Pattern Scheme The main idea behind this scheme is to create some intersections between trajectories of dummies and the user. Given a user trajectory, its dummy is generated rotating the known user trajectory. The rotation point of the user trajectory is an intersection point. Since there are three requirements in privacy profile, first it is necessary to understand which is the solution space for the given privacy profile. Then, within the solution space, compute the metrics to understand which of the different solutions is the best to use (the one with the smallest metrics). With proper selection of dummy trajectories, it is possible to minimize the number of dummies so as to satisfy the user privacy requirements.

In order to derive the solution space it is crucial to consider both of the rotation angle and the rotation point within a true user trajectory. This means that, given the user trajectory, the different metrics will be computed changing the rotation point and its angle, so as to create different dummies intersecting the real trajectory in different point and with different angles. The Figure 2.7 is describing this step.

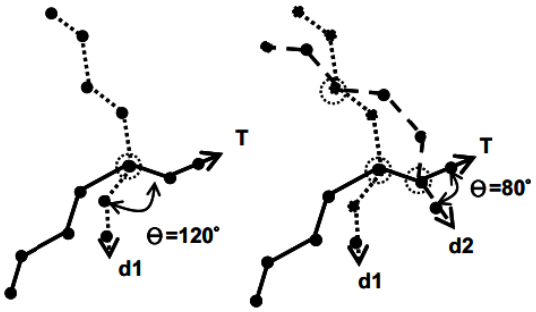

Figure 2.7: Example of rotating step [52]

If the disclosures are still larger than the required disclosures, one should repeat the procedure to add one additional dummy until the all requirements in privacy profile are satisfied.

NWA

The next algorithm that is resumed in this work belongs to the category of algorithms that cluster locations and subsequently release trajectories through these clusters with some perturbation to guarantee privacy. They follow a common structure that consists of two privacy mechanisms: a generalization mechanism M1, which generalizes the set of locations by grouping them into clusters, and a releasing mechanism M2, which outputs resulting trajectories drawn from the generalized sets. The algorithm described in this section is called Never Walk Alone (NWA) [53]. NWA is developed along three main phases:

- Pre-processing: aimed at enforcing larger equivalence classes of trajectories

w.r.t. same time span;

- Clustering: based on greedy clustering method and enhanced with techniques to keep low the radius of produced clusters;
- Space Translation: transforming each cluster found into a (k, δ)-anonymity set.

The input of the algorithm are a database of trajectories D, an anonymity threshold k, an uncertainty threshold δ, and the time granularity π used in the pre-processing step to create equivalence classes of trajectories, as explained in the next section. The output of the algorithm is a (k, δ)- anonymized database D'.

Pre-processing The fist task of NWA is the partitioning of the input database into equivalence classes according to the time span. This means creating groups containing all the trajectories that have the same starting time and the same ending time. If this procedure is performed on raw data this often brings to a large number of very small equivalence classes. In order to overcome this problem, the pre-processing procedure is driven by the integer number π: only one timestamp every 7 can be the starting or ending point of a trajectory.

Clustering This phase clusters trajectories based on a greedy clustering scheme. For each equivalence class, a set of appropriate pivot trajectories are selected as cluster centers. They are chosen as the farthest trajectory from the previous pivot (excepted the first one, chosen as the farthest trajectory from the dataset center). For each cluster center, its nearest $k - 1$ trajectories are assigned to the cluster, such that the radius of the bounding trajectory volume of the cluster is not larger than a certain threshold.

When a cluster cannot be created around a new pivot, the latter is simply deactivated, i.e., it will not be used as pivot but, in case, it can be used in the future as member of some other cluster and the process goes on with the next pivot. When a remaining object cannot be added to any cluster without violating the radius threshold, it is simply trashed. Notice that this process can lead to solutions with a too large trash.

Space Translation Finally, each cluster is transformed into a k-anonymized aggregate trajectory by aggregating the trajectories point per point with the median.

In NWA, information distortion occurs in three different ways:

- First, in the pre-processing step, some initial and final points of a trajectory are possibly cut with the aim of building larger equivalence classes of trajectories having the same time span.

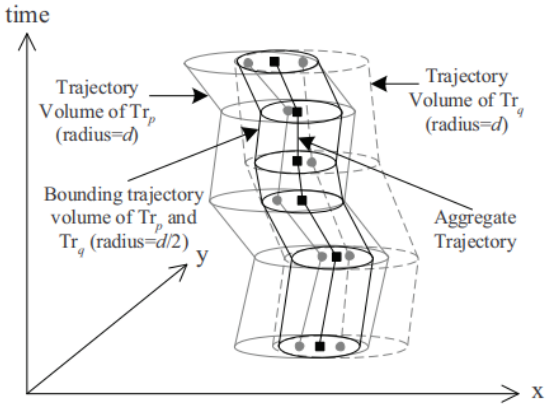

Figure 2.8: An example of the clustering approach [50]

- Second, trajectories ending in the trash bin are completely removed and will not appear in the released dataset *D'*.
- Third, trajectories not ending in the trash bin are space-translated to achieve (k, δ)-anonymity.

For this reason the work propose a unique measure able to capture these three different kinds of information distortion. For each trajectory $\tau \in D$, let τ' be its correspondent in the (k-δ)-anonymized dataset *D'*. For each time t in which τ is defined:

$$ID(\tau[t], \tau'[t]) = \begin{cases} \text{Dist}(\tau[t], \tau'[t]) & \text{if } \tau'[t] \text{ is defined;} \\ \Omega & \text{otherwise.} \end{cases}$$

where Ω is a constant value used to penalize removed points and corresponding to the maximal point translation recorded in the experiment, and *Dist* is the Euclidean distance.

DPTD

The last algorithm analysed is the DPTD [41]. The purpose of [41] is to design an ϵ-differentially private trajectory releasing algorithm, synthesizing a database S_D corresponds to raw database D.

The proposed method can be divided into three main phases: generalization, Markov probability prediction modeling, and the construction of a noisy prefix tree.

Generalization In the first phase, all locations within the spatial universe are clustered using the DBSCAN algorithm. For each cluster, the centroid is

then computed. Subsequently, the spatial domain is partitioned using a quadtree structure. Each region is recursively divided into four equal sub-regions if the number of locations it contains exceeds a predefined threshold. This process continues until every sub-region contains fewer locations than the threshold. For privacy purposes, the centroids obtained are further masked by substituting them randomly with one of the real positions inside the corresponding sub-region.

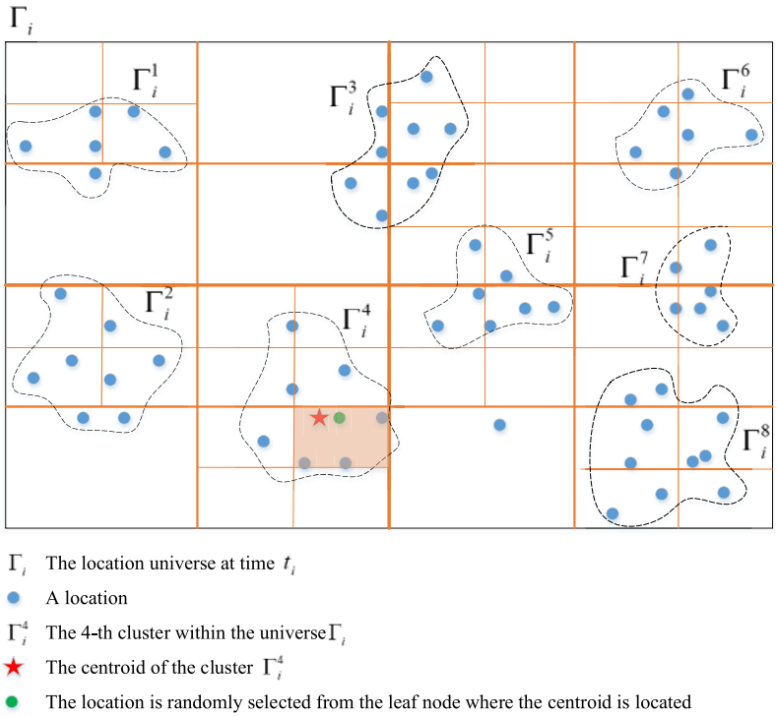

Figure 2.9: DPTD generalization process [41]

The work also states that since the centroids have selected randomly, the privacy budget that should have been used to protect them has been saved.

At this point the generalized dataset is synthesized in suck a way that the new dataset contains the IDs of the trajectories and the different clusters for each time interval. Basically the trajectory as a sequence of points has been transformed as a sequence of clusters.

Markov Probability Prediction Model The second phase is based on a first-order Markov model, which assumes that the probability of a future transition depends only on the current state. In this context, the transition probability from one cluster to another is determined solely by the current cluster.

Noisy Prefix Tree Construction In the final phase, a prefix tree of traversed clusters is generated. Each level of the tree represents a different timestamp: after the root, at height 1 there are the clusters traversed in the first time interval with the respective count. At height 2 there are clusters traversed in the second time interval related to their father with their counts. And so on.

To guarantee differential privacy, Laplace noise is added to the counts of the tree. Specifically, when the height of a layer l is odd, Laplace noise with a privacy budget of $\epsilon/\lceil h/2 \rceil$ is applied to the nodes of layer l, where h is the total height of the noisy prefix tree. Nodes in layer $l+1$ are then obtained by multiplying the noisy count of their parent node by the Markov transition probability of moving from the parent cluster to the child cluster.

The main drawback that this work is not considering is the fact that when accessing probability transition table for even nodes, it is doing it without paying any budget. The strong assumption that the work is doing is that the Markov transition probability is calculated using the generalized dataset, but this will not reveal the users' privacy, based on the assumption that an attacker cannot accurately get the transition probability among all the clusters in the generalized dataset. This means that an attacker cannot neither access to this table nor is able to compute those probabilities by his own.

2.6 Geo-indexing systems

Geo-indexing systems are alternative ways of representing locations on a (spherical or flat) land surface in recursively smaller areas, creating a hierarchy. Each area has to satisfy the following properties [54]:

- each area belongs to a single level in the area hierarchy,
- the aggregation of areas of the same level in the hierarchy results in the total surface of the Earth,
- each area is uniquely identified by an alphanumeric code, where the number of characters of the code identifies the level in the hierarchy. Thus, larger areas need fewer characters to identify themselves than smaller areas.

The main advantage of these systems is that they allow the identification of very small areas of the earth's surface with a text string of a small number of characters. Indeed they achieve a great deal of precision. However, their main drawback is that they were not designed to cluster locations, but rather to store coordinates more easily in databases or encode them in URLs. Nevertheless, they do represent a way of having the space divided recurrently in advance, which is a way of speeding up

all the algorithms that need the division. Examples of these libraries are GeoHash[8], Google S2[9] or Uber H3[10].

2.6.1 Introduction to H3

Uber is a technology platform that provides on-demand transportation and delivery services by connecting passengers and drivers (or couriers) through a mobile or web application. The company acts as an intermediary, managing the booking, payment, and rating processes, without directly owning the vehicles used. Every day, millions of events occur in the Uber marketplace. Every minute, riders request rides, driver-partners start trips, and users request food, among other actions on the platform.

These events empower Uber to better understand and optimize the marketplace for users across their services. For instance, these events might tell them that there is more demand than supply in a certain part of a city and adjust pricing in response[11].

Deriving information and insights from data in the Uber marketplace requires analyzing data across an entire city. Because cities are geographically diverse, this analysis needs to happen at a fine granularity. Analysis at the finest granularity, the exact location where an event happens, is very difficult and expensive. Analysis on areas, such as neighborhoods within a city, is much more practical.

Figure 2.10: The maps depict the process of bucketing points with H3: cars in a city; cars in hexagons; and hexagons shaded by number of cars. Picture from note 11.

Data points are bucketed into hexagons and can be written using the hexagonally

[8]Source: `https://www.ibm.com/docs/en/streams/4.3.0?topic=334-geohashes`, accessed on October 11, 2025

[9]Source: `http://s2geometry.io/`, accessed on October 11, 2025

[10]Source: `https://h3geo.org/`, accessed on October 11, 2025

[11]Source: `https://www.uber.com/en-IT/blog/h3/`, accessed on October 11, 2025

bucketed data. For example, in the Uber interests, the rising of the price is computed by measuring supply and demand in hexagons in each city that are served. These hexagons form the basis for the analysis of the Uber marketplace. For those reasons Uber decided to create H3 combining the benefits of a hexagonal global grid system with a hierarchical indexing system.

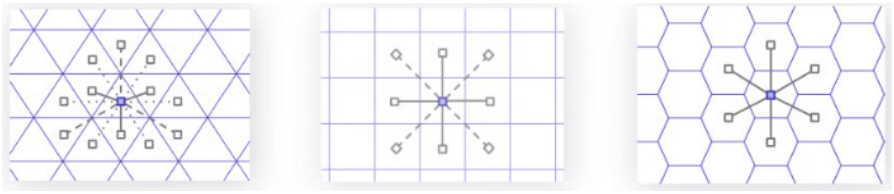

Figure 2.11: Distances from a triangle to its neighbors (left), a square to its neighbors (center), and a hexagon to its neighbors (right). Picture from note 11.

Using a hexagon as the cell shape is critical for H3. As depicted in Figure 2.11, hexagons have only one distance between a hexagon centerpoint and its neighbors' ones, compared to two distances for squares or three distances for triangles. This property greatly simplifies performing analysis and smoothing over gradients.

H3 supports sixteen resolutions. Each finer resolution has cells with one seventh the area of the coarser resolution. Hexagons cannot be perfectly subdivided into seven hexagons, so the finer cells are only approximately contained within a parent cell.

H3 index	No. of cells	Average Hex. Area (km²)
0	122	4,357,449.416078381
1	842	609,788.441794133
2	5,882	86,801.780398997
3	41,162	12,393.434655088
4	288,122	1,770.347654491
5	2,016,842	252.903858182
6	14,117,882	36.129062164
7	98,825,162	5.161293360
8	691,776,122	0.737327598
9	4,842,432,842	0.105332513
10	33,897,029,882	0.015047502
11	237,279,209,162	0.002149643
12	1,660,954,464,122	0.000307092
13	11,626,681,248,842	0.000043870
14	81,386,768,741,882	0.000006267
15	569,707,381,193,162	0.000000895

Table 2.6: Total number of cells and the corresponding area in km^2 for each level of the Uber H3 hierarchy

Chapter 3

ODkAnon

This chapter describes the ODkAnon algorithm for the anonymisation of OD-matrices, leveraging the Uber H3 geo-indexing library. The version used is the H3 version 4.2.2.

Before diving into the technical details, it is useful to provide a short overview of the proposed algorithm. At a high level, ODkAnon leverages the hierarchical structure of the H3 hexagonal spatial indexing system to protect mobility data while retaining as much spatial detail as possible. The algorithm iteratively generalizes origins and destinations until all OD pairs satisfy the anonymity threshold. Unlike traditional approaches, which typically apply the same level of aggregation to both dimensions, ODkAnon adopts an adaptive strategy: at each step, it dynamically decides whether to generalize origins or destinations, based on the structure and density of the data.

3.1 Suppression algorithm

Since hexagons may be very sparse or associated with very low counts (even when their siblings contain large volumes), they may need to be progressively generalized into larger and larger parent hexagons.

The function `fast_pre_generalization_filter` implements a pre-filtering step that identifies and removes OD pairs (origin–destination) that cannot achieve k-anonymity even after multiple levels of spatial generalization. The goal is to suppress those records that may lead to pointless huge generalizations.

It defines a suppression budget, defined as the the maximum percentage of rows that can be removed, and a maximum number of generalization levels (`max_generalization_levels`) to explore to understand if within these levels the node can be k-anonymized. The algorithm works as follows.

- **Generalization Mapping**. For each OD pair, the algorithm computes parent

hexagons at progressively coarser resolutions from original hexagon resolution up to `max_generalization_levels` (Algorithm 1, for cycle at row 3).

- **Aggregated Counts**. At each generalization level, the OD pairs are grouped by their generalized hexagons, and the aggregated counts are computed. If a group reaches or exceeds the threshold k, the corresponding rows are marked as valid (Algorithm 1, for cycle at row 10).

- **Detection of Problematic Rows**. Rows that do not reach the threshold at any generalization level are identified as problematic, meaning they cannot be made k-anonymous (Algorithm 1, row 15).

- **Suppression Strategy**. If the number of problematic rows is within the suppression budget, all of them are removed. If not, the algorithm suppresses only the rows with the lowest counts, ensuring minimal information loss (Algorithm 1, if condition at row 16).

The function returns a filtered OD-matrix where only valid rows remain, along with the number of suppressed rows. The "valid" rows are those rows that, within a fixed number of generalizations, can become k-anonymous.

Algorithm 1 Suppression algorithm

Require: OD-matrix, threshold k, max generalization levels L, suppression budget β

Ensure: Filtered OD-matrix

1: $n \leftarrow |OD|$, max_supp $\leftarrow \lfloor n \cdot \beta \rfloor$
2: Initialize H3 mapping cache
3: **for** $\ell = 0$ **to** L **do**
4: **for** each row in OD **do**
5: start_gen$_\ell \leftarrow$ generalize(start_h3, ℓ)
6: end_gen$_\ell \leftarrow$ generalize(end_h3, ℓ)
7: **end for**
8: **end for**
9: valid_pairs $\leftarrow \emptyset$
10: **for** $\ell = 0$ **to** L **do**
11: Group by (start_gen$_\ell$, end_gen$_\ell$)
12: Compute agg_count for each group
13: valid_pairs $\leftarrow$ valid_pairs $\cup$ {rows with agg_count $\geq k$}
14: **end for**
15: problematic $\leftarrow$ all_rows $\setminus$ valid_pairs
16: **if** |problematic| $\leq$ max_supp **then**
17: Suppress all problematic rows
18: **else**

19: Sort problematic rows by increasing count
20: Suppress the first max_supp rows
21: **end if**
return OD-matrix with valid rows only

3.2 Tree structure creation

To handle the spatial generalization of OD-matrices, the algorithm is based on two hierarchical H3 trees: `tree_start` and `tree_end`. The main idea is to represent space not as a fixed grid, but as a tree structure, where each node corresponds to an H3 hexagon at a given resolution, and its children represent finer subdivisions.

The inputs given to this part of the code are the OD-matrix, the target resolution, which is the smallest resolution considered (in this case is 10) and the column in the OD-matrix to consider: first the column containing the starting hexagons then the one considering the ending ones.

At this point the algorithm follows these steps:

- **Hexagons extraction**. Extract hexagons from the OD-matrix (Algorithm 2, row 1).
- **Root identification**. Find the minimal optimal resolution (the biggest hexagon with count equal to 1): this hexagon will be the root of the tree (Algorithm 2, for cycle at row 3).
- **Hierarchy construction**. Build the hierarchical paths from the minimum resolution up to the target resolution (Algorithm 2, for cycle at row 13).
- **Node creation**. Create the nodes and establish the parent–child relationships between them (Algorithm 2, row 17).
- **Count population**. Populate the counts based on the OD-matrix data: trip counts associated with each hexagon are inserted into the leaf nodes and then propagated upwards, so that each node represents the total number of trips across that area (Algorithm 2, for cycle at row 23).

Algorithm 2 H3 Hierarchical Tree Construction

Require: OD-matrix, target resolution R_{target}, hex column C
Ensure: Optimized H3 hierarchical tree
1: $H \leftarrow$ extract unique hexagons from column C
2: $H_{coverage} \leftarrow$ obtain full coverage at resolution R_{target}
3: **for** $r = 0$ **to** R_{target} **do**

```
    ancestors ← ∅
    for each h ∈ H do
        ancestor ← parent of h at resolution r
        ancestors ← ancestors ∪ {ancestor}
    end for
    stats[r] ← |ancestors|
end for
R_min ← max{r : stats[r] = 1}
nodes ← ∅
for each h ∈ H_coverage do
    path ← path from h to resolution R_min
    for each p ∈ path do
        if p ∉ nodes then
            nodes[p] ← new H3 node
        end if
    end for
    Establish parent-child relationships along path
end for
counts ← group OD by column C and sum counts
for each (h, c) ∈ counts do
    h_target ← map h to resolution R_target
    Propagate count c from h_target up to the root
end for
    return Hierarchical tree with aggregated counts
```

3.3 Generalization algorithm for *k*-anonymity

The class `OptimizedH3GeneralizedODMatrix` aims to efficiently anonymize very large OD-matrices. It integrates sparse data structures, hierarchical H3 trees, and dynamic balancing strategies.

The inputs to this core function are the OD-matrix, the two trees created with the Algorithm 2 and the parameter k to satisfy k-anonymity. The workflow of the algorithm is structured as follows:

- **Initialization**. A sparse matrix representation (CSR/CSC format) is built, where rows correspond to destination hexagons and columns to origin hexagons. This drastically reduces memory usage compared to dense matrices.
- **Precomputation of Sibling Groups**. Using the tree structures, the algorithm precomputes sibling groups, i.e., sets of hexagons sharing the same

parent. These groups represent the potential candidates for aggregation during the generalization process. Even single-child groups are included, ensuring that the algorithm can continue generalizing also when only one descendant is available.

- **Generalization Strategy**. At each iteration, the algorithm identifies the cell with the minimum count in the sparse OD-matrix. If all cells are above the anonymity threshold k, the process stops. Otherwise, a new generalization step is applied. To decide where to generalize, the algorithm uses a dynamic balancing strategy:
  - It tracks the ratio between the number of origins and destinations.
  - If this ratio deviates significantly (beyond $\pm 3\%$) from its initial value, the algorithm forces generalization on the "dominant" axis (origins if too many columns, destinations if too many rows).
  - Otherwise, it alternates between the two axes to maintain balance.

  Within the chosen axis, the algorithm selects the best sibling group to aggregate, using a cost function based on the number of trips.
- **Application of Sparse Generalization**. The selected sibling group is merged into its parent node in the sparse matrix. The corresponding row(s) or column(s) are summed, and the matrix is updated while preserving efficiency.
- **Termination**. The process continues iteratively until every OD pair has at least k trips, or no further aggregation is possible.

Algorithm 3 Optimized OD Generalization

function RUN_OPTIMIZED_GENERALIZATION($OD, tree_{start}, tree_{end}, k$)
    Initialize sparse matrix with INITIALIZE_OPTIMIZED_MATRIX
    $step \leftarrow 0$
    while minimum cell value $< k$ **do**
        Select axis based on ratio balance (*columns/rows alternation*)
        $(group, parent, cost) \leftarrow$ GET_BEST_GENERALIZATION_FAST(axis)
        if no valid generalization found **then**
            Try alternative axis
            if still none **then**
                break
            end if
        end if
        APPLY_SPARSE_GENERALIZATION(group, parent, axis)
        $step \leftarrow step + 1$

```
    end while
    return Final generalized sparse matrix
end function
function INITIALIZE_OPTIMIZED_MATRIX
    Remove zero-count cells from OD
    Extract used start and end hexagons
    Map hexagons to target resolution using tree_start, tree_end
    Build sparse OD matrix (rows, cols, counts)
    Precompute sibling groups for both axes
    return sparse OD matrix
end function
function GET_BEST_GENERALIZATION_FAST(axis)
    best ← ∞
    for each parent node in hierarchy (start or end) do
        siblings ← children of parent
        present ← siblings currently in matrix
        if present ≠ ∅ and consistent with siblings then
            cost ← aggregated count of present
            if cost < best then
                update best group, parent, and cost
            end if
        end if
    end for
    return (group, parent, cost) if found, else None
end function
function APPLY_SPARSE_GENERALIZATION(group, parent, axis)
    if axis = columns then
        Merge columns of group into new column parent
        Update sparse matrix and start mappings
    else
        Merge rows of group into new row parent
        Update sparse matrix and end mappings
    end if
    Remove old sibling groups involving group
    Add new sibling group including parent (if applicable)
end function
```

Chapter 4

Experiments

In this chapter, the framework adopted for the experimental evaluation is introduced. First, the datasets used in the experiments are presented, highlighting their main characteristics. Then, the benchmark algorithms against which ODkAnon is compared are described. Finally, the performance indicators employed in the evaluation are defined, covering both utility- and privacy-oriented metrics.

4.1 Datasets

The proposed approaches have been evaluated on three different datasets, each with distinct characteristics.

1. The first dataset comes from the NetMob25 challenge[1] and is a merge of a dataset containing GPS data and a dataset containing demographic data. Particularly merging the two dataframe on the `person_id` it is possible to obtain a structure that contains both the GPS data and the demographic data. The dataset collets data obtained between October 2022 and May 2023 and focused on residents aged 16 to 80 in Ile-de-France (3,337 participants took part to the collection). The most relevant attribute in the demographic data is the `WEIGHT_INDIV`. Each participant is assigned a weight representing how many individuals in the Ile-de-France region share the same socio-demographic profile. This profile is defined by the cross-tabulation of several variables: department of residence (8 departments), age group (16–25,26–45,46–65,66–80), sex (male, female), socio-professional category (craftsmen, executives, intermediate professions, employees and workers, retirees and other inactives), number of cars in the household (0, 1, or 2+), household size (1, 2, 3, or 4+ people),

[1] `https://netmob.org/www25/`, accessed on October 11, 2025

and highest diploma obtained (lower or upper secondary, Bac+2, Bac+5 or doctorate). The final dataset will contain a person identifier, the GPS data (with the time) and the weight. This dataset is denoted as **paris** and initially has 81,289 rows and 8 columns.

2. The second dataset contains trip data from a car-sharing service operating in Turin, Italy. It is denoted as **turin** and initally has 873,240 rows and 6 columns. This data was originally not anonymized and has been collected in 2017 for research purposes. Each row represents a single trip and includes the starting and the ending point grouped into a single object called `coordinates`, as well as the initial and the final time of the trip, the plate of the car and its VIN (Vehicle Identification Number), a unique code of 17 alphanumeric characters associated to each single motor vehicle. The sensitive attribute that is protected is the plate of the car. Since it is a car-sharing dataset, the purpose is not to protect the user, but the car.

3. The third dataset describes a complete year, from July 2013 to June 2014, of the trajectories for all the 442 taxis running in the city of Porto, Portugal[2]. It is denoted as **porto** and it initially has 1,710,670 rows and 9 columns. Each entry corresponds to one completed trip. Since these taxis operate through a taxi dispatch central, using mobile data terminals installed in the vehicles, the dataset contains both information related on the phone call and a column called `POLYLINE` which contains a list of GPS coordinates mapped as a string. Each pair of coordinates is identified as [LONGITUDE, LATITUDE]. This list contains one pair of coordinates for each 15 seconds of trip. The last list item corresponds to the trip's destination while the first one represents its start: those two couples are the one useful for the purpose of this study. Also in this case the purpose is to protect the vehicle rather than the user.

In order to be properly processed by the algorithm, the three datasets need to undergo a pre-processing phase to ensure that data is structured in the same format for all the three datasets and removing sampling errors. The `paris` dataset is the only one that is already in the required format and therefore does not require additional pre-processing.

First, the datasets need to contain the same four core columns: `start_lon`, `start_lat`, `end_lon` and `end_lon`, which respectively identifies the longitude and the latitude of the starting point, and the longitude and the latitude of the ending point.

[2] `https://www.kaggle.com/datasets/crailtap/taxi-trajectory?resource=download`, accessed on October 11, 2025

- The `turin` dataset is first filtered to filter out the possible sampling errors present in the coordinates: particularly there are points in (0,0) and in Ethiopia. This process brings the number of the rows of the dataset from 873,240 to 873,234. Then the four core columns are extracted by the column `coordinates`.

- The `porto` dataset needs first a filter to remove those rows with no coordinates or with one single coordinate. This step reduces the number of rows of the dataset from 1,710,670 to 1,674,160. At this point the latitude and the longitude of the starting and ending points are extracted by the `POLYLINE` column considering only the first couple as the starting point and the last one as the ending one.

At this stage of the processing, the three datasets contain only valid geographic data collected in the four core columns `start_lon`, `start_lat`, `end_lon` and `end_lon`. The minimum spatial resolution is set to 10, corresponding to the resolution of the H3 hexagonal grid used to partition the space. A finer resolution would have produced an excessive number of cells, generating too sparse OD-matrices and obtaining an inefficient generalization. Two new columns are then added: `start_h3` and `end_h3` which identify the ID of the H3 hexagons at resolution 10 which contains respectively the starting and the ending point of the trip. The original processed dataset is now aggregated considering only the columns `start_h3` and `end_h3` and the count related to the couple: the number of couple with the same starting and ending hexagons in the dataset. This is the way the OD matrix is computed (and stored in the variable `od_matrix)`.

- Within the 81,240 rows of the Paris dataset, there are 72,569 unique couples of hexagons (71 of them have a count bigger or equal to 10, the remaining 72,498 have a count smaller than 10). The area of the trips is partitioned into 29,350 starting hexagons and 32,217 ending hexagons (both at resolution 10).

- Within the 873,243 rows of the Turin dataset, there are 507,514 unique couples of hexagons (3,863 of them have a count bigger or equal to 10, the remaining 503,651 have a count smaller then 10). The area of the trips is partitioned into 3,606 starting hexagons and 3,608 ending hexagons (both at resolution 10).

- Within the 1,674,160 rows of the Porto dataset, there are 501,472 unique couples of hexagons (28,963 of them have a count bigger or equal to 10, the remaining 472,509 have a count smaller then 10). The area of the trips is partitioned into 8,444 starting hexagons and 18,359 ending hexagons (both at resolution 10).

Finally, a further filter is applied to consider only the rows in the center of the city both for Turin, Porto and Paris. After this filtering step, the OD matrices contain 458,918 rows for Turin, 501,472 rows for Porto, and 23,264 rows for Paris.

Considering that for the `paris` dataset other analysis will be done regarding the population-protection view, also the weights must be considered. For this reason the aggregated dataset `od_matrix` also contains the column `total_weight` which identifies the sum of the individual weights related to that couple of starting and ending hexagons.

4.2 Benchmark

In this work, the proposed approach is evaluated against different anonymization algorithms, adapted from existing sources:

- Mondrian [35]: partitions the data by recursively cutting along the coordinates of the flows, represented as 4D points. Details of the algorithm can be found at 2.5.1. Original Python implementation available at: `https://github.com/Nuclearstar/K-Anonymity/blob/master/k-Anonymity.ipynb`.

- OIGH algorithm [46], which leverages horizontal cuts within the hierarchies of origins and destinations. Details of the algorithm can be found at 2.5.2. The implementation used is the one proposed by the authors of [20].

- ATG [20] based on optimization over tree-structured hierarchies. Details of the algorithm can be found at 2.5.2. The implementation used is the one proposed by the authors of [20].

All those algorithms have needed a phase for choosing the different hyperparameters to set before the actual running of the different codes (as also described in Section 2.5). First of all the anonymity parameter k for k-anonymity has been set to 10 for the three algorithms. Moreover, the algorithms need as input the OD-matrix and the two hexagon hierarchical trees.

The Mondrian algorithm requires the specification of two variables:

- `featureColumns` is the set of features used for recursive partitioning of the dataset. In this case, this vector is defines as `featureColumns = ['start_lon', 'start_lat', 'end_lon', 'end_lat']`, which corresponds to a 4D representation of the OD flows.

- `sensitiveColumn` is the attribute to protect, set to `sensitiveColumn = 'person_id'`. This means that the algorithm ensures that every partition contains at least k distinct `person_id`.

The ATG-Soft algorithm requires the specification of two key hyperparameters:

- `suppr_thres_frac` defines the maximum fraction of records that can be suppressed instead of generalized when they fail to meet the anonymity threshold. This parameter is set to 10% of the total volume to balance utility and privacy.
- `target_vol` represents the target volume parameter used in the origin generalization heuristic. The author of [20] suggests to find a value for this paramter in a way that the number of origins and destination hexagons is similar to obtain a balanced representation of the data through the hexagons. For this reason in Paris and Porto the value is set to 1000, while in Turin is 2500.

Finally, the OIGH algorithm needs nothing more but the three core components: k, the OD-matrix and the two hierarchical trees.

4.3 Performance indicators

The anonymization alters the values of the OD-matrix, as well as the zone sizes, depending on the aggregation of the geographical areas in the hierarchy. It is crucial to assess data utility and privacy preservation with respect to the original data.

When applying anonymization techniques to protect data, it is essential to ensure that the process introduces only the minimum amount of generalization or perturbation required to satisfy the k-anonymity constraint. While hiding sensitive data is a priority, the resulting dataset must remain useful for further analysis. Excessive perturbation may strengthen privacy protection, but it can also compromise data utility, making it difficult to extract meaningful insights. For this reason, it is crucial to compute some metrics that evaluate the quality and usability of data.

Data utility metrics measure how well the anonymized OD-matrix preserves the original data characteristics, comparing the pre- and post-anonymization versions. The following are the metrics used in this work.

Discernability Metric

The first metric is one that attempts to capture in a straightforward way the desire to maintain discernibility between tuples as much as is allowed by a given setting of k [55]. This discernibility metric (C_{DM}) assigns a penalty to each tuple based on how many tuples in the transformed dataset are indistinguishable from it. If an unsuppressed tuple is part of an equivalence class Eq of size $|Eq|$, then that tuple is assigned a penalty of $|Eq|$. If a tuple is suppressed, then it is assigned

a penalty of $|D|$, the size of the input dataset: in this way, a suppressed tuple cannot be distinguished from any other tuple in the dataset, hence it needs a penalization larger than any non-suppressed tuple ($|D| \geq |Eq|$). The metric is defined mathematically as:

$$C_{DM} = \sum_{Eqs.t.|Eq| \geq k} |Eq|^2 + \sum_{Eqs.t.|Eq|<k} |D||Eq| \tag{4.1}$$

In this expression, the sets Eq refer to the equivalence classes of tuples in D induced by the anonymization. The first sum computes penalties for each non-suppressed tuple, the second for suppressed tuples. Small values of this metric indicate small equivalence classes and no suppression, with resulting OD matrices retaining more utility and information. Notice that this metric is not normalized, and can assume very large values.

Normalized Average Equivalence Class Size

The C_{AVG} metric [35] evaluates the average size of equivalence classes in a dataset after anonymization, normalized with respect to the anonymity parameter k. Formally, it is defined as:

$$C_{AVG} = \frac{\left(\frac{|D^+|}{total\_equiv\_classes}\right)}{k} \tag{4.2}$$

D^+ is the set of non-suppressed records, leading to $|D^+| \leq |D|$, and $total\_equiv\_classes$ is the number of equivalence classes that respects k-anonymity. This metric measures how much the average class size exceeds the minimum anonymity requirement. A value of $C_{AVG} = 1$ indicates that, on average, equivalence classes contain exactly k records, meaning the dataset is minimally compliant with the k-anonymity constraint. Higher values of C_{AVG} suggest that equivalence classes are significantly larger than the threshold, which implies stronger anonymity but may also lead to greater information loss due to excessive generalization.

Notice that this metric does not penalize suppression and does not compare the obtained generalization with the original data.

Generalization Distance Metric

The Generalization Distance Metric (GDM) measures the spatial distortion introduced when original geographic points (origins and destinations) are generalized into an anonymized location. Particularly, in case of Mondrian and H3 hierarchy, the anonymized point is defined as the center of the rectangle and the hexagon respectively.

Let each trip in the dataset be represented by its origin and destination $(x_s^{(i)}, y_s^{(i)}), (x_d^{(i)}, y_d^{(i)})$, where $(x_s^{(i)}, y_s^{(i)})$ and $(x_d^{(i)}, y_d^{(i)})$ are the latitude–longitude coordinates of the origin and destination, respectively. During the anonymization process, each point is mapped into a new anonymized point. In the case of ODkAnon the anonymzed point is the center of the hexagon, while in the case of Mondrian is the center of the rectangle. This point is defined as $(x_{s,g}^{(i)}, y_{s,g}^{(i)}), (x_{d,g}^{(i)}, y_{d,g}^{(i)})$.

The distance between the original point and its generalized representation is then computed using the geodesic distance. When calculating the geodesic distance between two points, the curvature of the Earth is taken into account. In other words, it corresponds to the shortest path along the Earth's surface between the two locations. This differs from the Euclidean distance, which instead computes the straight-line distance between two points, ignoring the Earth's curvature. For short distances, the difference between the two measures is negligible, since the curved path can be approximated by a straight line. However, for points that are far apart, the difference can become substantial. In any case, the geodesic distance is considered in this work, as it provides a more accurate measurement. The `geodesic` function is provided by the `GeoPy` library. The distances computed are defined as:

$$d_j^{(i)} = dist\Big((x_j^{(i)}, y_j^{(i)}), (x_{j,g}^{(i)}, y_{j,g}^{(i)})\Big), \quad j \in \{s, d\} \tag{4.3}$$

The overall generalization error is then evaluated by aggregating these distances across all trips. For example, the metric considers descriptive statistics such as mean and median distance and standard deviation.

This metric therefore provides a quantitative evaluation of how far generalized data points deviate from their original spatial position: smaller values indicate that the generalization process preserves spatial accuracy, while larger values suggest higher distortion and, consequently, lower data utility.

Mean Generalization Error

The mean generalization error is defined by [20] as follows:

$$\bar{G} = \frac{1}{|D^+|} \sum_{Eq_o \to d \, s.t. |Eq_o \to d| \geq k} (|o| + |d|) \, |Eq_{o \to d}| \tag{4.4}$$

where D^+ is the set of non-suppressed records and and $|D^+|$ is their total volume. All non-suppressed equivalence classes $Eq_{o \to d}$ that respect k-anonymity ($Eq_{o \to d} \geq k$) are considered, which means only the ones belonging to D^+. $|o|$ and $|d|$ represent the number of original areas aggregated over the hierarchy, for origins and destinations, respectively. This metric provides an estimation of the average information loss introduced during the generalization process. A larger value of $\bar{G}$ indicates that

origins and/or destinations have been generalized into coarser spatial units, reducing the precision of the mobility representation. On the other hand, lower values of $\bar{G}$ reflect finer partitions, which preserve more spatial detail but may offer weaker privacy guarantees. Notice that this metric cannot be computed for algorithms that do not use a hierarchy.

Reconstruction Loss

This metric quantifies how much the generalized data deviates from the original data. The reconstruction loss E is computed in this way [20]:

$$E = \frac{1}{|D|} \sum_{o,d \in leaves(T)} ||\tilde{D}_{o \to d}| - |D_{o \to d}|| \tag{4.5}$$

where T is the hierarchy tree, and the finer-grained tiles in this hierarchy for the origins and destinations, i.e., the leaves of the tree are considered. $D_{o \to d}$ is the number of records in these original finer-grained tiles. Considering the leaves of the tree T ensures that the loss is computed with respect to the original maximum granularity (level 10 of the H3 hierarchy), regardless of how much aggregation has been performed. The anonymized coarser-grained equivalence classes $Eq_{o \to d}$ that respect k-anonymity ($Eq_{o \to d} \geq k$) should be mapped to the finer-grained ones. For every pair of origin cells (o, d) at the maximum granularity (the leaves of the hierarchical structure T), it is found which generalised pair (o', d') they belong to after anonymisation. However, such information is now aggregated over multiple tiles, and it is not possible to infer the exact original values from the anonymized version. Then, it is crucial to uniformly assign the anonymized records to the finer-grained tiles, proportionally to their size. This is defined as the reconstructed finer-grained volume of trips $\tilde{D}_{o \to d}$. The metric is normalised by the total volume of the flows $/D/$ for readability. Notice that D also includes the flows that have been suppressed during the anonymisation, and the finer-grained tiles without any trips that generalize to tiles with at least a trip. In a scenario where, during the anonymization process, new trips are not created, i.e., they do not increase but are possibly suppressed, the worst possible obtained value is 2.

Notice that this metric can also not be computed for algorithms that do not use a hierarchy.

Chapter 5

Results

This chapter contains the different experiments done, which consider the following dimensions:

- 3 different datasets: Paris, Turin and Porto;
- 4 algorithms: ODkAnon, ATG-Soft, OIGH, and Mondrian;
- 5 data utility metrics: Discernability Metric C_{DM}, Normalized Average Equivalence Class Size C_{AVG}, Generalization Mistance Metric (GDM), Mean Generalization Error $\bar{G}$, and Reconstruction Loss E;

Moreover, a more precise analysis is done on the Paris dataset considering, in addition to the ones above, these dimensions:

- 4 different ways of segmenting the dataset: whole dataset, sex, age, and socio-professional category;
- 2 protection targets: participants, and population;
- 2 metrics evaluation computation: based on participants (trips), and based on population (trips multiplied by representativeness).

This additional analysis is possible thanks to the specific characteristics of the Paris dataset, which includes the `WEIGHT_INDIV` attribute representing how many individuals in the population each participant represents. This allows for the evaluation of anonymization from both the perspective of survey participants and the broader population they represent.

5.1 Results over different datasets

In this section, the performance the different anonymization algorithms is compared across the three datasets introduced in 4.1. The evaluation considers some standard

Table 5.1: Summary of the three evaluated datasets.

Dataset	Trips	Unique OD pairs	Origin hexagons	Destination hexagons
`paris`	81,289	72,569	29,350	32,217
`turin`	873,234	507,514	3,606	3,608
`porto`	1,674,160	501,472	8,444	18,359

utility and privacy metrics C_{DM}, C_{AVG}, $\bar{G}$, E, together with the ad-hoc created metric GDM. All the metrics are defined in 4.3. Tables 5.2, 5.3 and 5.4 report the obtained results.

Table 5.2: Result comparison on the Paris dataset. $\bar{G}$ and E are not defined for Mondrian.

	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)	Time (s)
ODkAnon	5.4×10^7	13.2	601.8	1.91	933.9	26.4
ATG-Soft	1.1×10^8	46.5	6,807.6	1.98	7,275.0	29.8
OIGH	3.6×10^7	80.5	6,869.0	1.99	7,345.8	14.3
Mondrian	4.0×10^5	1.3	-	-	351.8	7.2

Regarding the cost metrics (C_{DM} and C_{AVG}), Mondrian outperforms the other algorithms with values that are considerably low. However, it plays a different game compared to the other methods: since it does not rely on a predefined hierarchy, it can freely generate optimal rectangles whose size is tuned to include just above k points. ODkAnon shows intermediate results in cost metrics. While, taking into account C_{DM}, ATG-Soft and OIGH always are an order of magnitude larger than ODkAnon. On the other hand, C_{AVG} has not a clear winner. OIGH is for sure the one with highest value, but ODkAnon and ATG-Soft exhibit a fluctuating behavior. This is caused the characteristics of the different datasets.

The metrics $\bar{G}$ and E, unavailable for Mondrian, provide important insights into the geometric quality of the generated partitions. ODkAnon consistently presents the lowest values for $\bar{G}$, suggesting more compact partitions compared to ATG-Soft and OIGH, which show values in the thousands. The reason of this behavior hides in the fact that ODkAnon is aggregating origins and destinations simultaneously, creating a much finer resolution for the hexagons. ATG-Soft, instead, is aggregating the origin hexagons first to find a map where each sone has a departing volume around a given parameter. Then, for each origin, an aggregation of the destination hexagons is selected to minimizes the error. This operation brings to fine origin hexagons, but it may bring to really large destination hexagons. Moreover, unlike ATG, ODkAnon automatically balances generalisation between origins and destinations without parameter tuning. In these experiments,

Table 5.3: Result comparison on the Turin dataset. $\bar{G}$ and E are not defined for Mondrian.

	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)	Time (s)
ODkAnon	4.1×10^9	77.1	280.1	1.80	691.3	30.6
ATG-Soft	5.7×10^{10}	71.9	3,073.8	1.88	1,105.4	4.9
OIGH	1.2×10^{10}	886.7	3,328.7	1.97	1,721.3	2.5
Mondrian	1.2×10^7	1.2	-	-	118.6	2,446.1

Table 5.4: Result comparison on the Porto dataset. $\bar{G}$ and E are not defined for Mondrian.

	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)	Time (s)
ODkAnon	6.2×10^9	208.1	270.8	1.71	730.9	14.8
ATG-Soft	9.7×10^{10}	41.6	2,277.9	1.82	806.9	2.7
OIGH	4.2×10^{10}	1,765.1	2,240.3	1.97	3,177.3	1.5
Mondrian	1.5×10^7	1.5	-	-	81.1	3094.0

as the authors suggest in [20], the parameter has been choosen trying to create an OD-matrix with same number of origins and destinations. On the other hand, OIGH applies a single cut to the hierarchy trees so that all cells satisfy k-anonymity. This behavior inevitably enlarges some hexagons more than necessary, since even already compliant cells are forced to generalize further.

The reconstruction loss metric E is particularly significant as it measures the deviation from original data when reconstructing fine-grained flows from generalized equivalence classes. Lower values indicate better preservation of the original trip distribution patterns. ODkAnon achieves the best results across all datasets. This superior performance suggests that ODkAnon's generalization strategy better preserves the underlying spatial flow patterns when data is reconstructed at maximum granularity. ATG-Soft shows moderate reconstruction loss, while OIGH consistently exhibits the highest deviation from original patterns. These higher values indicate that when fine-grained trips are reconstructed from the generalized equivalence classes, there is greater distortion in the spatial distribution compared to the original data.

The GDM metric reinforces these findings. Since Mondrian has not a hierarchy to respect, it is systematically achieving the lowest distance, being able to create small rectangular partitions with a number of points close to k. For the same reasons just described above, ODkAnon is maintaining moderate distances with respect to the ATG-Soft and OIGH. As an example, the Turin dataset is generating

the generalization hexagons shown in Figure 5.1. ODkAnon is the only algorithm able to maintain fine-resolution hexagons thanks to its behavior that balance the generalization of origins and destinations, producing a 32×32 OD matrix. In contrast, OIGH yields a 1×89 matrix, while ATG-Soft results in 357×328.

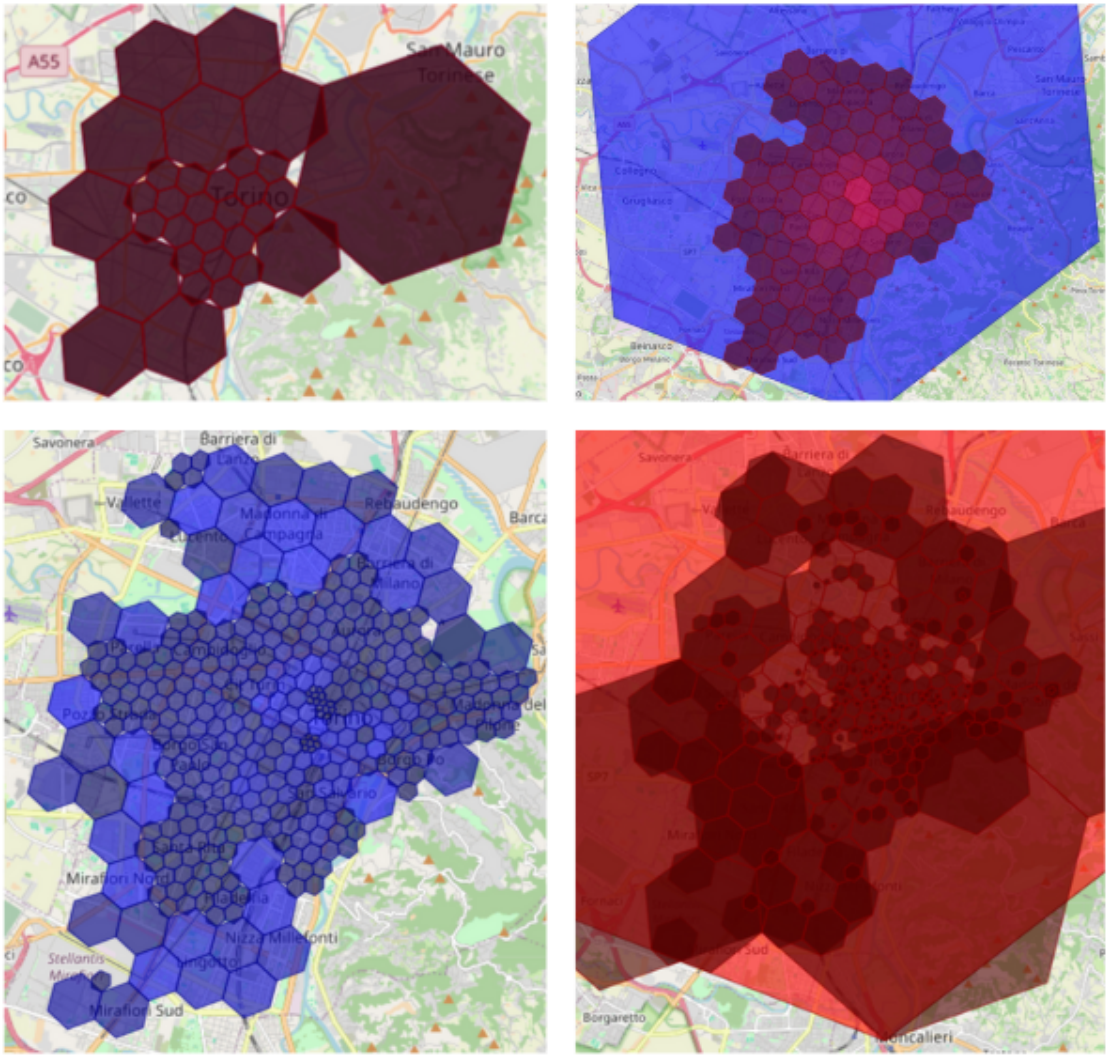

Figure 5.1: On the top left the generalization hexagons created by the ODkAnon, on the top right the ones from OIGH, on the bottom left the origins of ATG-Soft and on the bottom right the destination od ATG-Soft

The analysis of execution times reveals heterogeneous performance characteristics. ATG-Soft and OIGH are the most efficient, being able to complete the whole process in very short time. ODkAnon requires slightly longer but still moderate times. Mondrian shows variable behavior depending on the size of the dataset: it is

excellent on Paris but considerably slower on Turin and Porto.

These results highlight fundamental trade-off in spatial anonymization:

- ODkAnon offers a balanced compromise across all the metrics, maintaining acceptable costs, good geometric performance and reasonable execution time, making it suitable for applications requiring a balance between privacy and utility.
- ATG-Soft prioritize computational efficiency but at the cost of inferior quality and geographical localization metrics, suggesting orientation toward application scenarios where processing speed is prioritized over spatial precision.
- OIGH prioritize computational efficiency as well, proving the worst results over all the presented algorithms.
- Mondrian excels in minimizing cost metrics and geographical localization but sacrifices computational efficiency on larger datasets.

For these reasons, the choice of optimal algorithm depends on specific application requirements. ODkAnon configures as a versatile solution for most standard application scenarios, offering balanced performance across all evaluated parameters. For real-time applications or those with strict computational constraints, ATG-Soft and OIGH represent viable alternatives. For scenarios in which there is no need for a hierarchical structure but the priority is on local spatial structure preservation, Mondrian emerges as an ideal candidate despite elevated processing times.

5.2 Results over the Paris datasets

This part of the work will focus on the Paris dataset only. In particular, thanks to its characteristics, the work will deepen the difference between participant-protection and population-protection. Moreover, the dataset will be segmented to study the behavior of the algorithm to different segments of population. The dataset has been filtered out to contain only on the trips within the Île-de-France (starting and ending within the region). For the algorithms that allow suppression, a maximum threshold of 10% of trip suppression has been fixed. For each run, the maximum computation time defined is up to two hours. The runs that did not provide a result in time are reported as N/A in the tables.

5.2.1 Results over the whole population

The ODkAnon algorithm is first applied on the whole dataset. The process begins by protecting the participants in the survey, setting $k = 10$ for obtaining a k-anonymous OD-matrix. For both the origins and the destination, 29 zones are

obtained, merging the original thousands of resolution-10 hexagons according to the hierarchy.

When protecting the population, k should be adapted, accounting for the representativeness of each participant. Given that a participant on average accounts for 2,674 people, the threshold is set to $k = 10 \times 2{,}674$ in order to keep a fair comparison of the two approaches. In this case the same 29 destination zones are obtained, but more fine-grained 35 origin zones.

The comparison in the origin hexagons for the two approaches is reported in Figure 5.2, with a zoom over the Paris region with observed differences. Protecting the population produces a different anonymization: in particular, 7 smaller hexagons (red tiles) are generalized to their parent node (blue tiles). Likely, these zones contain fewer trips from the participants—hence, they must be aggregated to satisfy 10-anonymity—but said participants represent a sufficient amount of people, allowing to maintain a higher resolution when protecting the population.

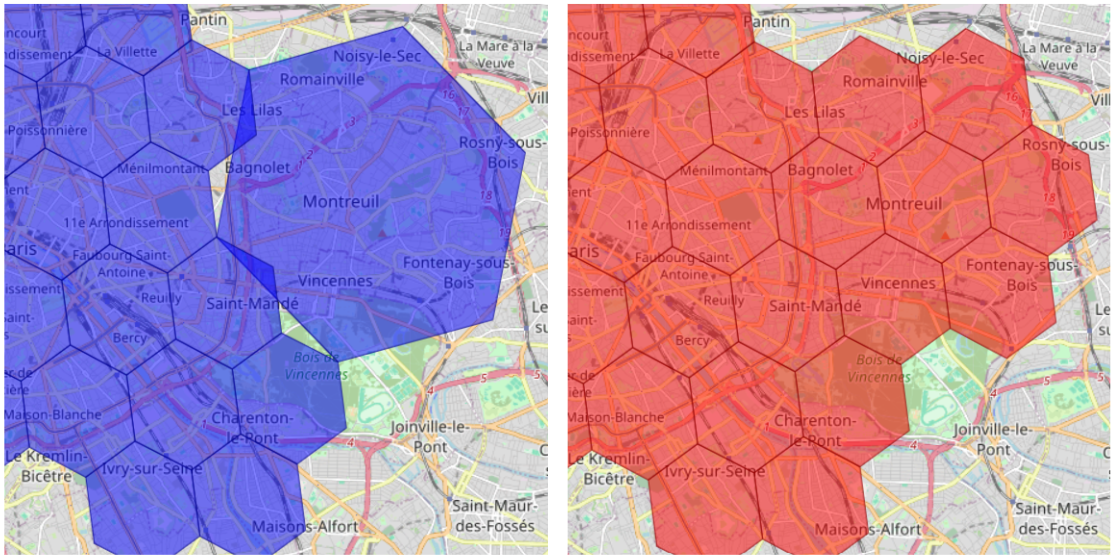

Figure 5.2: Detail over Paris of the origin generalization hexagons for the participant-protecting (left, blue hexagons) and the population-protecting (right, red hexagons) definitions.

Privacy metrics validate the strength of the anonymization. When generating the anonymous OD-matrix for the participants, the impact on the population OD-matrix is evaluated, and vice versa. The minimum k-anonymity obtained in such cases is measured, with the results reported in Table 5.5.

The results show that protecting the participants leads to a population OD-matrix that is no longer k-anonymous: 21 cells fall below the anonymity threshold, with a minimum value of 10,274 compared to the required 26,742. On the other

hand, when the protection is applied to the population, the participants' OD matrix does not reach the same level of anonymization $k = 10$, as 13 cells fall below the threshold, and the minimal value is 4.

Table 5.5: k-anonymity property computed in different scenarios.

Participant-protecting		**Population-protecting**	
$k_{dataset}$	$k_{population}$	$k_{dataset}$	$k_{population}$
10	10,274	4	26,742

The utility of data is compared by applying different algorithms to the whole dataset. Table 5.6 shows the utility metrics in the survey participant-protecting scenario. In general, ATG-Soft and OIGH obtain relatively lower utility than ODkAnon. Given that OIGH does not allow suppression, it over-generalizes sparse hexagons and thus loses more information. While ATG-Soft allows for suppression, its performance highly depends on the pre-definition of zones. As the authors of [20] acknowledged, ATG-Soft needs proper tuning to achieve better performance. Mondrian does not consider H3 hexagons for hierarchy definition: it aggregates coordinates into rectangles. The more flexible generalization allows Mondrian to get the best performance on C_{DM} and C_{AVG}, since it does not aggregate over hierarchy, $\bar{G}$ and E cannot be derived. While evaluating the results, it is important to keep in mind that non-homogeneity gives an advantage to ATG-Soft and Mondrian in terms of metrics, as it may happen that origin or destination overlap: while this offers greater flexibility to the algorithms, it may prevent real-world analysis of the results. For instance, when analyzing the number of trips arriving at a specific zone, hexagons at different resolutions covering that zone might need to be considered.

Utility is also evaluated from the population's point of view. In general, the results are similar to calculating metrics on the participants, see Table 5.7 for more details. Notice that when computing C_{DM} on population, the weights make it get much larger values, while the other metrics are normalized.

Results for the population-protecting scenario are reported in Table 5.8 and Table 5.9. In this case, ATG-Soft and Mondrian did not return results within two hours of computation. This is because now the number of trips is much larger (because the original trips are now multiplied by the representativeness), and the two algorithms scale poorly. In general, the utility metrics computed both on participants and population are on par with the participant-protecting scenario. In short, protecting participants or the population has little impact on data utility, but neither approach can guarantee the same level of privacy from the other perspective.

Table 5.6: Result comparison on the whole dataset, protecting the participants, calculating metrics on the participants. $\bar{G}$ and E are not defined for Mondrian.

	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)	Time (s)
ODkAnon	5.4×10^7	13.2	601.8	1.91	933.9	26.4
ATG-Soft	1.1×10^8	46.5	6,807.6	1.98	7,272.0	29.8
OIGH	3.6×10^7	80.5	6,869.0	1.99	7,345.8	14.3
Mondrian	4.0×10^5	1.3	-	-	351.8	7.2

Table 5.7: Result comparison on the whole dataset, protecting the participants, calculating metrics on the population. $\bar{G}$ and E are not defined for Mondrian.

	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)	Time (s)
ODkAnon	1.3×10^{14}	12.8	592.1	1.92	933.9	26.4
ATG-Soft	2.2×10^{14}	44.2	6,824.5	1.88	7,272.0	29.8
OIGH	2.4×10^{14}	77.6	6,867.3	1.99	7,345.8	14.3
Mondrian	3.3×10^{14}	13.5	-	-	351.8	7.2

5.2.2 Segmenting the population over sex

Moreover, the population was segmented taking into account three different attributes available in the NetMob dataset: sex, age, and profession. In particular, the division was carried out by sex (men and women), by age groups (from 10 to 19 years old, from 20 to 29, from 30 to 39, from 40 to 49, from 50 to 59, from 60 to 99, and above 70), and by profession into eight categories.

Results for the sex segmentation are reported in Tables 5.10, 5.11, 5.12, and 5.13. The following section presents the results for age and socio-professional categories. Again, a two-hour deadline was imposed for the computation of every k-anonymized dataset. When protecting the population, ATG-Soft and Mondrian were not able to meet the time limit–hence, the metrics on these two algorithms were not evaluated.

When protecting the participants, anonymizing the male dataset produces a 2×5 matrix, whereas anonymizing the female dataset results in a 29×29 matrix. This indicates that protecting men is more challenging, as it requires very coarse hexagons, while for women the resulting hexagons remain much finer. Furthermore, when applying protection to the population, the difference becomes even more pronounced: the anonymized male dataset reduces to a 2×2 matrix.

These differences are mainly observed when using ODkAnon (see in particular Tables 5.10 and 5.11). Indeed, ODkAnon is the only algorithm able to reach very high utility metrics for women. This observation raises the question of whether the difference lies in the data distribution or in the algorithms' choices, and will be

Table 5.8: Result comparison on the whole dataset, protecting the population, calculating metrics on the participants. ATG–Soft and Mondrian did not provide a result within 2 hours of computation time.

	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)	Time (s)
ODkAnon	5.3×10^7	13.1	539.0	1.91	876.8	26.2
ATG-Soft			N/A			>7,200.0
OIGH	3.6×10^7	80.5	6,869.0	1.99	7,345.8	13.1
Mondrian	N/A		-	-	N/A	>7,200.0

Table 5.9: Result comparison on the whole dataset, protecting the population, calculating metrics on the population. ATG–Soft and Mondrian did not provide a result within 2 hours of computation time.

	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)	Time (s)
ODkAnon	1.1×10^{14}	12.7	530.9	1.92	876.8	26.2
ATG-Soft			N/A			>7,200.0
OIGH	2.3×10^{14}	77.6	6,867.3	1.99	7,345.8	13.1
Mondrian	N/A		-	-	N/A	>7,200.0

evaluated in depth in future work.

5.2.3 Other results

In the following, results are reported for the age and the socio-professional category. For each segment, $k = 10$ is kept for the k-anonymous OD matrix. Keeping $k = 10$ on much smaller datasets greatly reduces the data utility, and overgeneralizes the geographic area, leaving fewer than 5 origins/destinations.

Tables 5.14, 5.15, 5.16, 5.17 present the results for the different combinations of protecting either the participants or the population, and calculating the metrics over either the participants or the population, segmenting the dataset according to participants' age. The same results, this time segmenting according to socio-professional categories, is shown in Tables 5.18, 5.19, 5.20, and 5.21.

Again, when protecting the population, ATG-Soft and Mondrian computation exceeded our two-hour computation limit.

Table 5.10: Result comparison segmenting on sex, protecting the participants, calculating metrics on the participants. $\bar{G}$ and E are not defined for Mondrian.

	Sex	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	M	4.6×10^7	151.6	4,388.3	1.43	4,077.4
	F	1.8×10^7	9.5	446.7	1.51	931.2
ATG-Soft	M	3.3×10^7	49.0	5,140.5	1.88	7,556.8
	F	4.1×10^7	67.4	5,300.2	1.90	7,671.5
OIGH	M	8.0×10^6	37.9	4,843.3	1.99	7,334.2
	F	1.0×10^7	42.6	5,033.7	1.99	7,345.6
Mondrian	M	1.7×10^5	1.3	-	-	441.0
	F	2.2×10^5	1.4	-	-	391.2

Table 5.11: Result comparison segmenting on sex, protecting the participants, calculating metrics on the population. $\bar{G}$ and E are not defined for Mondrian.

	Sex	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	M	2.1×10^{14}	141.4	4,377.5	1.87	4,077.4
	F	4.1×10^{13}	9.4	433.5	1.88	931.2
ATG-Soft	M	1.0×10^{14}	45.3	5,180.0	1.88	7,556.8
	F	1.6×10^{14}	66.2	5,340.5	1.89	7,671.5
OIGH	M	4.9×10^{13}	35.2	4,842.4	1.99	7,334.2
	F	7.2×10^{13}	42.3	5,031.6	1.99	7,345.6
Mondrian	M	1.0×10^{14}	10.9	-	-	441.0
	F	1.7×10^{14}	14.0	-	-	391.2

Table 5.12: Result comparison segmenting on sex, protecting the population, calculating metrics on the participants. ATG–Soft and Mondrian did not provide a result within 2 hours of computation time.

	Sex	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	M	1.3×10^8	302.8	6,726.8	1.86	5,373.4
	F	1.8×10^7	9.9	450.3	1.87	932.7
ATG-Soft	M			N/A		
	F					
OIGH	M	8.0×10^6	37.9	4,843.3	1.99	7,334.2
	F	1.0×10^7	42.6	5,033.7	1.99	7,345.6
Mondrian	M	N/A		-	-	N/A
	F			-	-	

Table 5.13: Result comparison segmenting on sex, protecting the population, calculating metrics on the population. ATG–Soft and Mondrian did not provide a result within 2 hours of computation time.

	Sex	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	M	7.5×10^{14}	292.5	6,724.4	1.92	5,373.4
	F	4.1×10^{13}	9.9	437.2	1.89	932.7
ATG-Soft	M			N/A		
	F					
OIGH	M	4.9×10^{13}	35.2	4,842.4	1.99	7,334.2
	F	7.2×10^{13}	42.3	5,031.6	1.99	7,345.6
Mondrian	M	N/A		-	-	N/A
	F			-	-	

Table 5.14: Result comparison segmenting on age, protecting the participants, calculating metrics on the participants. $\bar{G}$ and E are not defined for Mondrian.

	Age	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	[10-20]	5.9×10^5	21.3	801.2	1.81	5,136.2
	[20-30]	3.1×10^7	156.9	4,154.3	1.85	5,316.9
	[30-40]	2.2×10^7	128.5	3,917.3	1.84	5,236.4
	[40-50]	8.6×10^6	66.8	2,481.9	1.84	4,143.4
	[50-60]	5.5×10^6	52.9	2,057.5	1.84	4,012.2
	[60-70]	6.7×10^6	69.3	2,583.5	1.83	5,295.4
	>70	9.5×10^5	26.2	1,082.6	1.83	5,065.9
ATG-Soft	[10-20]	8.8×10^5	939.0	1,176.0	2.00	13,384.7
	[20-30]	1.1×10^7	137.6	3,703.9	1.99	7,921.1
	[30-40]	7.6×10^6	113.0	3,400.2	1.99	8,043.6
	[40-50]	1.0×10^7	65.4	3,427.5	1.89	7,922.3
	[50-60]	5.0×10^6	93.0	2,823.5	1.99	7,982.7
	[60-70]	2.9×10^6	46.3	2,213.8	1.54	7,971.1
	>70	1.3×10^6	115.1	1,467.0	1.99	13,796.6
OIGH	[10-20]	2.2×10^5	18.7	743.6	1.99	7,870.1
	[20-30]	2.3×10^6	19.6	3,146.5	1.99	7,272.9
	[30-40]	1.4×10^6	16.1	2,808.3	1.99	7,387.1
	[40-50]	1.5×10^6	16.7	2,938.1	1.99	7,371.6
	[50-60]	1.0×10^6	13.2	2,349.6	1.99	7,393.8
	[60-70]	2.3×10^6	61.1	2,177.1	1.99	8,012.8
	>70	3.7×10^5	23.0	923.3	1.99	7,967.1
Mondrian	[10-20]	1.3×10^1	1.4	-	-	722.0
	[20-30]	9.5×10^1	1.3	-	-	504.0
	[30-40]	6.7×10^1	1.1	-	-	502.9
	[40-50]	7.0×10^1	1.1	-	-	503.1
	[50-60]	7.9×10^1	1.6	-	-	540.9
	[60-70]	3.7×10^1	1.2	-	-	587.0
	>70	1.9×10^1	1.6	-	-	755.7

Table 5.15: Result comparison segmenting on age, protecting the participants, calculating metrics on the population. $\bar{G}$ and E are not defined for Mondrian.

	Age	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	[10-20]	1.3×10^{13}	38.4	8,160.6	1.81	5,136.2
	[20-30]	7.6×10^{13}	96.9	4,106.5	1.84	5,316.9
	[30-40]	1.4×10^{14}	124.1	3,950.2	1.82	5,236.4
	[40-50]	4.6×10^{13}	68.5	2,469.1	1.86	4,143.4
	[50-60]	3.1×10^{13}	54.6	2,069.9	1.84	4,012.2
	[60-70]	6.3×10^{13}	83.6	2,591.8	1.83	5,295.4
	>70	6.4×10^{12}	27.8	1,054.1	1.78	5,065.9
ATG-Soft	[10-20]	2.0×10^{13}	168.4	1,176.0	2.00	13,384.7
	[20-30]	3.0×10^{13}	85.6	3,704.0	1.99	7,921.1
	[30-40]	5.0×10^{13}	110.2	3,385.1	1.99	8,043.6
	[40-50]	5.4×10^{13}	65.7	3,451.3	1.88	7,922.3
	[50-60]	4.0×10^{13}	95.8	2,829.9	1.99	7,982.7
	[60-70]	2.3×10^{13}	55.7	2,221.9	1.57	7,971.1
	>70	1.1×10^{13}	125.5	1,467.0	1.99	13,796.6
OIGH	[10-20]	5.3×10^{12}	33.8	744.7	1.99	7,870.1
	[20-30]	6.4×10^{12}	12.2	3,146.5	1.99	7,272.9
	[30-40]	9.6×10^{12}	15.7	2,804.5	1.99	7,387.1
	[40-50]	1.1×10^{13}	17.0	2,937.8	1.99	7,371.6
	[50-60]	8.1×10^{12}	13.6	2,349.5	1.99	7,393.8
	[60-70]	2.5×10^{13}	73.6	2,175.1	1.99	8,012.8
	>70	3.1×10^{12}	25.1	921.5	1.99	7,967.1
Mondrian	[10-20]	2.0×10^{13}	19.3	-	-	722.0
	[20-30]	2.5×10^{13}	7.5	-	-	504.0
	[30-40]	2.7×10^{13}	7.7	-	-	502.9
	[40-50]	3.0×10^{13}	8.2	-	-	503.1
	[50-60]	6.4×10^{13}	15.6	-	-	540.9
	[60-70]	2.2×10^{13}	9.6	-	-	587.0
	>70	1.4×10^{13}	14.2	-	-	755.7

Table 5.16: Result comparison segmenting on age, protecting the population, calculating metrics on the participants. ATG–Soft and Mondrian did not provide a result within 2 hours of computation time.

	Age	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	[10-20]	6.4×10^5	42.6	942.8	1.82	9,172.9
	[20-30]	3.2×10^7	156.7	4,234.6	1.85	5,298.5
	[30-40]	2.3×10^7	128.4	4,025.4	1.84	5,168.2
	[40-50]	2.3×10^7	133.5	3,879.9	1.84	5,352.8
	[50-60]	1.5×10^7	105.7	3,187.4	1.84	5,164.6
	[60-70]	6.9×10^6	69.2	2,643.6	1.88	5,233.8
	>70	9.6×10^5	26.1	1,105.8	1.83	4,966.5
ATG-Soft	[10-20]					
	[20-30]					
	[30-40]					
	[40-50]			N/A		
	[50-60]					
	[60-70]					
	>70					
OIGH	[10-20]	2.2×10^5	18.7	743.6	1.99	7,870.1
	[20-30]	1.1×10^7	137.6	3,714.1	1.99	7,940.0
	[30-40]	1.4×10^6	16.1	2,808.3	1.99	7,387.1
	[40-50]	1.5×10^6	16.7	2,938.1	1.99	7,371.6
	[50-60]	5.1×10^6	93.0	2,784.4	1.99	8,063.7
	[60-70]	2.3×10^6	61.1	2,177.1	1.99	8,012.8
	>70	3.7×10^5	23.0	923.3	1.99	7,967.1
Mondrian	[10-20]			-	-	
	[20-30]			-	-	
	[30-40]			-	-	
	[40-50]	N/A		-	-	N/A
	[50-60]			-	-	
	[60-70]			-	-	
	>70			-	-	

Table 5.17: Result comparison segmenting on age, protecting the population, calculating metrics on the population. ATG–Soft and Mondrian did not provide a result within 2 hours of computation time.

	Age	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	[10-20]	1.5×10^{13}	80.7	950.1	1.91	9,172.9
	[20-30]	8.6×10^{13}	102.6	4,184.3	1.93	5,298.5
	[30-40]	1.6×10^{14}	132.9	4,054.2	1.94	5,168.2
	[40-50]	1.6×10^{14}	143.0	3,850.1	1.93	5,352.8
	[50-60]	1.1×10^{14}	115.9	3,206.1	1.94	5,164.6
	[60-70]	7.5×10^{13}	89.8	2,641.5	1.95	5,233.8
	>70	7.9×10^{12}	30.4	1,083.1	1.94	4,966.5
ATG-Soft	[10-20]					
	[20-30]					
	[30-40]					
	[40-50]			N/A		
	[50-60]					
	[60-70]					
	>70					
OIGH	[10-20]	5.3×10^{12}	33.8	744.7	1.99	7,870.1
	[20-30]	3.1×10^{13}	85.6	3,714.0	1.99	7,940.0
	[30-40]	9.6×10^{12}	15.7	2,804.5	1.99	7,387.1
	[40-50]	1.1×10^{12}	17.0	2,937.8	1.99	7,371.6
	[50-60]	4.1×10^{13}	95.8	2,791.7	1.99	8,063.7
	[60-70]	2.5×10^{13}	73.6	2,175.1	1.99	8,012.8
	>70	3.1×10^{12}	25.1	921.5	1.99	7,967.1
Mondrian	[10-20]			-	-	
	[20-30]			-	-	
	[30-40]			-	-	
	[40-50]	N/A		-	-	N/A
	[50-60]			-	-	
	[60-70]			-	-	
	>70			-	-	

Table 5.18: Result comparison segmenting on socio-professional category, protecting the participants, calculating metrics on the participants. $\bar{G}$ and E are not defined for Mondrian.

	Cat.	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	Cat. 1	9.1×10^5	24.6	1,131.1	1.82	5,425.5
	Cat. 2	2.3×10^7	110.0	3,666.6	1.86	4,087.0
	Cat. 3	1.2×10^7	95.8	3,034.3	1.84	4,965.7
	Cat. 4	8.8×10^6	82.7	2,486.3	1.84	5,305.5
	Cat. 5	5.2×10^5	38.1	940.5	1.81	9,503.4
	Cat. 6	6.3×10^6	68.5	2,451.1	1.83	5,431.7
	Cat. 7	4.7×10^6	48.1	1,842.0	1.84	5,267.7
	Cat. 8	5.9×10^5	17.5	789.1	1.81	4,514.8
ATG-Soft	Cat. 1	1.1×10^6	108.5	1,428.0	2.00	14,454.8
	Cat. 2	2.2×10^7	62.0	4,545.4	1.97	7,782.7
	Cat. 3	4.1×10^6	84.2	2,655.1	1.82	8,107.7
	Cat. 4	4.3×10^6	45.3	2,230.0	1.88	8,003.0
	Cat. 5	7.0×10^5	84.0	1,142.0	2.00	14,037.7
	Cat. 6	2.4×10^6	60.3	2,175.9	1.99	7,929.4
	Cat. 7	4.3×10^6	84.4	2,535.3	1.99	7,816.8
	Cat. 8	2.3×10^6	154.4	2,001.0	2.00	13,765.6
OIGH	Cat. 1	3.4×10^5	21.7	909.4	1.99	8,377.9
	Cat. 2	4.3×10^6	27.5	4,087.0	1.99	7,301.6
	Cat. 3	8.7×10^5	12.0	2,222.5	1.99	7,452.7
	Cat. 4	6.5×10^5	10.3	1,896.8	1.99	7,369.9
	Cat. 5	1.6×10^5	16.8	712.5	1.99	8,106.6
	Cat. 6	2.4×10^6	60.3	2,132.0	1.99	7,932.6
	Cat. 7	4.3×10^6	84.4	2,566.6	1.99	7,836.6
	Cat. 8	5.4×10^5	30.8	1,215.0	1.99	8,015.8
Mondrian	Cat. 1	1.9×10^1	1.6	-	-	836.4
	Cat. 2	1.6×10^2	1.4	-	-	503.6
	Cat. 3	7.1×10^1	1.6	-	-	601.1
	Cat. 4	5.3×10^1	1.4	-	-	529.0
	Cat. 5	1.1×10^1	1.3	-	-	898.4
	Cat. 6	3.7×10^1	1.2	-	-	550.5
	Cat. 7	7.0×10^1	1.6	-	-	582.2
	Cat. 8	1.9×10^1	1.2	-	-	645.8

Table 5.19: Result comparison segmenting on socio-professional category, protecting the participants, calculating metrics on the population. $\bar{G}$ and E are not defined for Mondrian.

	Cat.	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	Cat. 1	3.8×10^{12}	19.9	1,138.3	1.80	5,425.5
	Cat. 2	6.5×10^{13}	77.6	3,666.1	1.87	4,087.0
	Cat. 3	1.0×10^{14}	108.8	3,018.6	1.83	4,965.7
	Cat. 4	1.2×10^{14}	119.0	2,549.0	1.85	5,305.5
	Cat. 5	6.6×10^{12}	53.2	940.3	1.84	9,503.4
	Cat. 6	5.2×10^{13}	78.7	2,410.3	1.81	5,431.7
	Cat. 7	1.8×10^{13}	41.9	1,833.4	1.85	5,267.7
	Cat. 8	2.0×10^{12}	13.2	829.6	1.79	4,514.8
ATG-Soft	Cat. 1	5.0×10^{12}	88.7	1,428.0	1.99	14,454.8
	Cat. 2	5.0×10^{13}	43.3	4,591.9	1.97	7,782.7
	Cat. 3	3.6×10^{13}	96.0	2,643.8	1.83	8,107.7
	Cat. 4	4.4×10^{13}	66.8	2,236.7	1.90	8,003.0
	Cat. 5	9.5×10^{12}	115.6	1,142.0	1.99	14,037.7
	Cat. 6	2.3×10^{13}	70.2	2,165.1	1.99	7,929.4
	Cat. 7	2.2×10^{13}	73.1	2,351.1	1.99	7,816.8
	Cat. 8	1.0×10^{13}	118.5	2,001.0	2.00	13,765.6
OIGH	Cat. 1	1.5×10^{12}	17.7	898.4	1.99	8,377.9
	Cat. 2	1.5×10^{13}	19.3	4,089.8	1.99	7,301.6
	Cat. 3	8.0×10^{12}	13.7	2,220.0	1.99	7,452.7
	Cat. 4	1.0×10^{13}	14.8	1,895.7	1.99	7,369.9
	Cat. 5	2.4×10^{12}	23.1	715.6	1.99	8,106.6
	Cat. 6	2.3×10^{13}	70.2	2,121.4	1.99	7,932.6
	Cat. 7	2.2×10^{13}	73.1	2,561.2	1.99	7,836.6
	Cat. 8	2.7×10^{12}	23.7	1,223.9	1.99	8,015.8
Mondrian	Cat. 1	7.3×10^{12}	10.7	-	-	836.4
	Cat. 2	7.9×10^{13}	10.9	-	-	503.6
	Cat. 3	5.7×10^{13}	16.2	-	-	601.1
	Cat. 4	5.7×10^{13}	15.3	-	-	529.0
	Cat. 5	7.9×10^{12}	11.2	-	-	898.4
	Cat. 6	2.1×10^{13}	9.4	-	-	550.5
	Cat. 7	4.4×10^{13}	13.7	-	-	582.2
	Cat. 8	3.3×10^{12}	4.9	-	-	645.8

Table 5.20: Result comparison segmenting on socio-professional category, protecting the population, calculating metrics on the participants.ATG–Soft and Mondrian did not provide a result within 2 hours of computation time.

	Cat.	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	Cat. 1	9.4×10^5	24.7	1,149.9	1.82	5,339.1
	Cat. 2	6.6×10^7	219.9	5,689.6	1.85	5,347.6
	Cat. 3	1.2×10^7	95.7	3,072.9	1.84	4,916.7
	Cat. 4	9.1×10^6	82.6	2,548.2	1.84	5,239.0
	Cat. 5	5.2×10^5	19.0	868.9	1.82	4,826.7
	Cat. 6	2.6×10^6	34.2	1,648.6	1.83	4,028.3
	Cat. 7	1.2×10^7	96.1	2,861.5	1.84	5,236.0
	Cat. 8	1.4×10^6	35.0	1,234.7	1.81	5,641.5
ATG-Soft	Cat. 1	N/A				
	Cat. 2					
	Cat. 3					
	Cat. 4					
	Cat. 5					
	Cat. 6					
	Cat. 7					
	Cat. 8					
OIGH	Cat. 1	3.4×10^5	21.7	909.4	1.99	8,377.9
	Cat. 2	4.3×10^6	27.5	4,087.0	1.99	7,301.6
	Cat. 3	8.7×10^5	12.0	2,222.5	1.99	7,452.7
	Cat. 4	3.1×10^6	75.5	2,234.4	1.99	8,058.0
	Cat. 5	1.6×10^5	16.8	712.5	1.99	8,106.6
	Cat. 6	2.4×10^6	60.3	2,132.0	1.99	7,932.6
	Cat. 7	4.3×10^6	84.4	2,566.6	1.99	7,836.6
	Cat. 8	5.4×10^5	30.8	1,215.0	1.99	8,015.8
Mondrian	Cat. 1	N/A		-	-	N/A
	Cat. 2			-	-	
	Cat. 3			-	-	
	Cat. 4			-	-	
	Cat. 5			-	-	
	Cat. 6			-	-	
	Cat. 7			-	-	
	Cat. 8			-	-	

Table 5.21: Result comparison segmenting on socio-professional category, protecting the population, calculating metrics on the population. ATG–Soft and Mondrian did not provide a result within 2 hours of computation time.

	Cat.	C_{DM}	C_{AVG}	$\bar{G}$	E	GDM (m)
ODkAnon	Cat. 1	4.5×10^{12}	21.5	1,146.9	1.94	5,339.1
	Cat. 2	2.2×10^{14}	161.1	5,647.8	1.92	5,347.6
	Cat. 3	1.1×10^{14}	115.6	3,055.4	1.93	4,916.7
	Cat. 4	1.4×10^{14}	126.1	2,599.8	1.94	5,239.0
	Cat. 5	7.2×10^{12}	28.3	858.2	1.95	4,826.7
	Cat. 6	2.1×10^{13}	42.6	1,618.6	1.95	4,028.3
	Cat. 7	6.6×10^{13}	88.9	2,848.1	1.95	5,236.0
	Cat. 8	6.7×10^{12}	28.9	1,286.4	1.94	5,641.5
ATG-Soft	Cat. 1	N/A				
	Cat. 2					
	Cat. 3					
	Cat. 4					
	Cat. 5					
	Cat. 6					
	Cat. 7					
	Cat. 8					
OIGH	Cat. 1	1.5×10^{12}	17.7	898.4	1.99	8,377.9
	Cat. 2	1.5×10^{13}	19.3	4,089.8	1.99	7,301.6
	Cat. 3	8.0×10^{12}	13.7	2,220.0	1.99	7,452.7
	Cat. 4	4.7×10^{13}	104.1	2,230.3	1.99	8,058.0
	Cat. 5	2.4×10^{12}	23.1	715.6	1.99	8,106.6
	Cat. 6	2.3×10^{13}	70.2	2,121.4	1.99	7,932.6
	Cat. 7	2.2×10^{13}	73.1	2,561.2	1.99	7,836.6
	Cat. 8	2.7×10^{12}	23.7	1,223.9	1.99	8,015.8
Mondrian	Cat. 1	N/A		-	-	N/A
	Cat. 2			-	-	
	Cat. 3			-	-	
	Cat. 4			-	-	
	Cat. 5			-	-	
	Cat. 6			-	-	
	Cat. 7			-	-	
	Cat. 8			-	-	

Chapter 6

Conclusion and perspectives

This work has presented ODkAnon, a novel k-anonymization algorithm specifically designed for OD-matrices that leverages the H3 hexagonal spatial indexing system. The analysis shows that ODkAnon can indeed be used as a practical tool for anonymizing mobility data, but its applicability depends on the context and on the requirements of the use case.

In particular, ODkAnon proves especially valuable in scenarios where there is a need to balance privacy protection with data utility. Unlike approaches that impose uniform levels of generalization, ODkAnon adapts its strategy dynamically, choosing whether to generalize origins or destinations depending on the structure of the matrix. This makes it particularly effective for datasets that are sparse and highly unbalanced, where uniform strategies would result in excessive information loss. It is not the fastest algorithm available, but it achieves consistent and competitive results across all metrics. ODkAnon is particularly recommended in situations when hierarchical consistency and spatial homogeneity are required, since ODkAnon guarantees non-overlapping partitions and preserves the hexagonal structure of H3.

6.1 Answers to the research questions

The research has successfully answered the three research questions posed at the beginning of this work:

- **RQ1: How can the H3 hexagonal spatial indexing system be used to partition geographic areas in a different way than the traditional rectangular approaches, such as the Mondrian algorithm?** The H3 hexagonal spatial indexing system has proven to be an effective alternative to traditional rectangular approaches like the Mondrian algorithm. Unlike rectangular partitions that can create irregular geographic areas of different sizes, H3 provides more consistent spatial aggregations through its hierarchical

hexagonal structure. The hexagonal grid better shapes geographic areas with uniform distance properties to neighbors, providing more accuracy in mobility data representation than traditional rectangular approaches.

- **RQ2: Can OD-matrices be generalised adaptively by applying different levels of spatial aggregation to origins and destinations, in order to achieve *k*-anonymity while minimising information loss?** The adaptive generalization strategy implemented in ODkAnon demonstrates that OD-matrices can be generalized by applying different levels of spatial aggregation to origins and destinations independently. The algorithm's dynamic balancing mechanism, which alternates between generalizing origins and destinations based on the matrix dimensions ratio, successfully minimizes information loss while achieving k-anonymity. This approach goes in a different direction with respect to other methods (like OIGH) that apply uniform aggregation levels to both dimensions, often resulting in unnecessary spatial detail loss. Moreover, ODkAnon creates homogeneous, non-overlapping hexagons of varying sizes, ensuring more reliable mobility data representation compared to algorithms like ATG that generate overlapping areas.
- **RQ3: How does the proposed approach perform in terms of privacy protection when evaluated both at the individual level and at the population level using weighted mobility data?** The comprehensive evaluation using the Paris dataset reveals significant differences between individual-level and population-level privacy protection approaches. When protecting survey participants, the resulting anonymization may not adequately protect the broader population they represent, and vice versa. This finding is particularly relevant for weighted mobility surveys where participants represent larger population segments. The analysis across demographic segments (sex, age, socio-professional categories) further demonstrates that privacy challenges vary significantly across different population groups, with some requiring much coarser generalizations than others.

6.2 Limitations and future work

This work presents several limitations opening different directions for future research.

- Optimal Generalization Heuristics. The proposed algorithms rely on heuristic strategies to determine generalization levels, which may not always guarantee an optimal solution. Future research could explore optimization frameworks or approximation algorithms to achieve closer-to-optimal generalizations while maintaining efficiency.

- Partitioning Dataset Challenges. Results also suggest that achieving anonymity may depend strongly on the dataset's intrinsic distribution rather than solely on the applied algorithm. For example, as already discussed, anonymizing the male dataset tends to require much coarser aggregations, while the female dataset can often be preserved at finer levels of detail. This suggests that protecting men is inherently more complex, as it forces stronger reductions in spatial resolution. At the same time, when applying ODkAnon, the anonymized female dataset consistently shows higher utility than the male one. This observation points to a qualitative difference that goes beyond simple matrix sizes: it raises the question of whether the challenge lies in the intrinsic structure of the data or in the way algorithms make their generalization choices. Exploring these dynamics in depth represents an important direction for future work.

- Scalability. Although the proposed approach shows reasonable performance on the evaluated datasets, its scalability to very large metropolitan areas or even national-scale mobility datasets remains to be tested. The hierarchical tree construction and sparse matrix operations could become computationally demanding when applied to extremely large spatial extents.

- Temporal Dimension. The current experiments are limited to static OD-matrices. However, many mobility applications require the analysis of time-series data. Extending the approach to handle temporal dimensions, while maintaining privacy protection, represents a natural and important research opportunity.

- Privacy Models. The methodology is grounded exclusively in k-anonymity. While widely adopted, this model is known to suffer from vulnerabilities such as homogeneity and background knowledge attacks. Future work should investigate the integration of complementary models like l-diversity, t-closeness, or differential privacy, in order to provide stronger theoretical guarantees.

6.3 Practical implications

From a practical perspective, the work has important implications for both practitioners and policymakers. For transportation authorities, ODkAnon provides a concrete solution to publish mobility data that comply with privacy regulations such as GDPR, while maintaining sufficient spatial detail for meaningful analysis. For statistical agencies, the distinction between participant-protecting and population-protecting approaches offers valuable insights when conducting mobility surveys, underlining the need to safeguard both individual respondents and the populations they represent. Finally, the observed differences in anonymization

requirements across demographic groups highlight the importance of fairness-aware privacy protection mechanisms, ensuring that mobility data publishing does not systematically disadvantage specific segments of the population.

Bibliography

[1] Shunjiang Ni and Wenguo Weng. «Impact of travel patterns on epidemic dynamics in heterogeneous spatial metapopulation networks». In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 79.1 (2009), p. 016111 (cit. on p. 1).

[2] Vitaly Belik, Theo Geisel, and Dirk Brockmann. «Natural human mobility patterns and spatial spread of infectious diseases». In: *Physical Review X* 1.1 (2011), p. 011001 (cit. on p. 1).

[3] Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie. «Urban computing with taxicabs». In: *Proceedings of the 13th international conference on Ubiquitous computing*. 2011, pp. 89–98 (cit. on p. 1).

[4] Jing Yuan, Yu Zheng, and Xing Xie. «Discovering regions of different functions in a city using human mobility and POIs». In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012, pp. 186–194 (cit. on p. 1).

[5] Guande Qi, Xiaolong Li, Shijian Li, Gang Pan, Zonghui Wang, and Daqing Zhang. «Measuring social functions of city regions from large-scale taxi behaviors». In: *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE. 2011, pp. 384–388 (cit. on p. 1).

[6] Woo-Sung Jung, Fengzhong Wang, and H Eugene Stanley. «Gravity model in the Korean highway». In: *Europhysics Letters* 81.4 (2008), p. 48005 (cit. on p. 1).

[7] Segun Goh, Keumsook Lee, Jong Soo Park, and MY Choi. «Modification of the gravity model and application to the metropolitan Seoul subway system». In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 86.2 (2012), p. 026102 (cit. on p. 1).

[8] Bei Pan, Yu Zheng, David Wilkie, and Cyrus Shahabi. «Crowd sensing of traffic anomalies based on human mobility and social media». In: *Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems*. 2013, pp. 344–353 (cit. on p. 1).

[9] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. «U-air: When urban air quality inference meets big data». In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 1436–1444 (cit. on p. 1).

[10] Kai Zhao, Mohan Prasath Chinnasamy, and Sasu Tarkoma. «Automatic city region analysis for urban routing». In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE. 2015, pp. 1136–1142 (cit. on p. 1).

[11] Weixiong Rao, Kai Zhao, Eemil Lagerspetz, Pan Hui, and Sasu Tarkoma. «Energy-aware keyword search on mobile phones». In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012, pp. 59–64 (cit. on p. 1).

[12] Xiangjie Kong, Menglin Li, Kai Ma, Kaiqi Tian, Mengyuan Wang, Zhaolong Ning, and Feng Xia. «Big trajectory data: A survey of applications and services». In: *IEEE access* 6 (2018), pp. 58295–58306 (cit. on p. 1).

[13] Àlex Miranda-Pascual, Patricia Guerra-Balboa, Javier Parra-Arnau, Jordi Forné, and Thorsten Strufe. «SoK: Differentially private publication of trajectory data». In: *Proceedings on Privacy Enhancing Technologies* (2023) (cit. on pp. 1, 2, 10, 12, 13, 15, 19, 21, 22).

[14] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. «Limits of predictability in human mobility». In: *Science* 327.5968 (2010), pp. 1018–1021 (cit. on p. 1).

[15] Chenglong Dai, Dechang Pi, Stefanie I Becker, Jia Wu, Lin Cui, and Blake Johnson. «CenEEGs: Valid EEG selection for classification». In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14.2 (2020), pp. 1–25 (cit. on p. 2).

[16] Yuqing Yang, Jianghui Cai, Haifeng Yang, Jifu Zhang, and Xujun Zhao. «TAD: A trajectory clustering algorithm based on spatial-temporal density analysis». In: *Expert Systems with Applications* 139 (2020), p. 112846 (cit. on p. 2).

[17] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. «Unique in the crowd: The privacy bounds of human mobility». In: *Scientific reports* 3.1 (2013), p. 1376 (cit. on p. 2).

[18] Erik Buchholz, Alsharif Abuadbba, Shuo Wang, Surya Nepal, and Salil Subhash Kanhere. «Reconstruction attack on differential private trajectory protection mechanisms». In: *Proceedings of the 38th annual computer security applications conference*. 2022, pp. 279–292 (cit. on pp. 2, 11).

[19] Marius Wernke, Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. «A classification of location privacy attacks and approaches». In: *Personal and ubiquitous computing* 18.1 (2014), pp. 163–175 (cit. on p. 2).

[20] Benoit Matet, Angelo Furno, Marco Fiore, Etienne Côme, and Latifa Oukhellou. «Adaptative generalisation over a value hierarchy for the k-anonymisation of Origin–Destination matrices». In: *Transportation Research Part C: Emerging Technologies* 154 (2023), p. 104236 (cit. on pp. 2, 13, 27, 50, 51, 53, 54, 57, 61).

[21] Latanya Sweeney. «Achieving k-anonymity privacy protection using generalization and suppression». In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 571–588 (cit. on pp. 3, 16).

[22] Julián Salas and Josep Domingo-Ferrer. «Some basics on privacy techniques, anonymization and their big data challenges». In: *Mathematics in Computer Science* 12.3 (2018), pp. 263–274 (cit. on pp. 6, 16).

[23] Welfare. Secretary's Advisory Committee on Automated Personal Data Systems. *Records, Computers, and the Rights of Citizens: Report*. US Department of Health, Education & Welfare, 1973 (cit. on p. 6).

[24] Jaap-Henk Hoepman. «Privacy design strategies». In: *IFIP International Information Security Conference*. Springer. 2014, pp. 446–459 (cit. on p. 7).

[25] Kamyar Hasanzadeh, Anna Kajosaari, Dan Häggman, and Marketta Kyttä. «A context sensitive approach to anonymizing public participation GIS data: From development to the assessment of anonymization effects on data quality». In: *Computers, Environment and Urban Systems* 83 (2020), p. 101513 (cit. on p. 8).

[26] Fengmei Jin, Wen Hua, Matteo Francia, Pingfu Chao, Maria E Orlowska, and Xiaofang Zhou. «A survey and experimental study on privacy-preserving trajectory data publishing». In: *IEEE Transactions on Knowledge and Data Engineering* 35.6 (2022), pp. 5577–5596 (cit. on pp. 10, 15).

[27] Mohamed Maouche, Sonia Ben Mokhtar, and Sara Bouchenak. «Ap-attack: a novel user re-identification attack on mobility datasets». In: *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 2017, pp. 48–57 (cit. on p. 10).

[28] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. «Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data». In: *Proceedings of the 26th international conference on world wide web*. 2017, pp. 1241–1250 (cit. on p. 10).

[29] Yuan Zhong, Nicholas Jing Yuan, Wen Zhong, Fuzheng Zhang, and Xing Xie. «You are where you go: Inferring demographic attributes from location check-ins». In: *Proceedings of the eighth ACM international conference on web search and data mining*. 2015, pp. 295–304 (cit. on p. 11).

[30] Hao Wang, Zhengquan Xu, Shan Jia, Ying Xia, and Xu Zhang. «Why current differential privacy schemes are inapplicable for correlated data publishing?» In: *World Wide Web* 24.1 (2021), pp. 1–23 (cit. on p. 11).

[31] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. «Knock knock, who's there? Membership inference on aggregate location data». In: *arXiv preprint arXiv:1708.06145* (2017) (cit. on p. 11).

[32] Eunjoon Cho, Seth A Myers, and Jure Leskovec. «Friendship and mobility: user movement in location-based social networks». In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011, pp. 1082–1090 (cit. on p. 11).

[33] Benjamin CM Fung, Ke Wang, Rui Chen, and Philip S Yu. «Privacy-preserving data publishing: A survey of recent developments». In: *ACM Computing Surveys (Csur)* 42.4 (2010), pp. 1–53 (cit. on p. 12).

[34] Marco Fiore et al. «Privacy in trajectory micro-data publishing: a survey». In: *arXiv preprint arXiv:1903.12211* (2019) (cit. on p. 12).

[35] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. «Mondrian multidimensional k-anonymity». In: *22nd International conference on data engineering (ICDE'06)*. IEEE. 2006, pp. 25–25 (cit. on pp. 12, 22, 23, 50, 52).

[36] Shen-Shyang Ho and Shuhua Ruan. «Differential privacy for location pattern mining». In: *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. 2011, pp. 17–24 (cit. on pp. 12, 24).

[37] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. «Privacy-preserving anonymization of set-valued data». In: *Proceedings of the VLDB Endowment* 1.1 (2008), pp. 115–125 (cit. on pp. 13, 29).

[38] Ronaldo dos Santos Mello, Vania Bogorny, Luis Otavio Alvares, Luiz Henrique Zambom Santana, Carlos Andres Ferrero, Angelo Augusto Frozza, Geomar Andre Schreiner, and Chiara Renso. «MASTER: A multiple aspect view on trajectories». In: *Transactions in GIS* 23.4 (2019), pp. 805–822 (cit. on p. 14).

[39] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. «l-diversity: Privacy beyond k-anonymity». In: *Acm transactions on knowledge discovery from data (tkdd)* 1.1 (2007), 3–es (cit. on p. 16).

[40] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. «t-closeness: Privacy beyond k-anonymity and l-diversity». In: *2007 IEEE 23rd international conference on data engineering*. IEEE. 2006, pp. 106–115 (cit. on p. 17).

[41] Sujin Cai, Xin Lyu, Xin Li, Duohan Ban, and Tao Zeng. «A trajectory released scheme for the internet of vehicles based on differential privacy». In: *IEEE Transactions on Intelligent Transportation Systems* 23.9 (2021), pp. 16534–16547 (cit. on pp. 18, 36, 37).

[42] Cynthia Dwork, Aaron Roth, et al. «The algorithmic foundations of differential privacy». In: *Foundations and trends® in theoretical computer science* 9.3–4 (2014), pp. 211–407 (cit. on p. 20).

[43] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. «Differentially private event sequences over infinite streams». In: (2014) (cit. on p. 20).

[44] Yang Cao and Masatoshi Yoshikawa. «Differentially private real-time data release over infinite trajectory streams». In: *2015 16th IEEE international conference on mobile data management*. Vol. 2. IEEE. 2015, pp. 68–73 (cit. on p. 21).

[45] Hilal Asi, John Duchi, and Omid Javidbakht. «Element level differential privacy: The right granularity of privacy». In: *arXiv preprint arXiv:1912.04042* (2019) (cit. on p. 21).

[46] Waranya Mahanan, W Art Chaovalitwongse, and Juggapong Natwichai. «Data privacy preservation algorithm with k-anonymity». In: *World Wide Web* 24.5 (2021), pp. 1551–1561 (cit. on pp. 26, 50).

[47] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. «Local and global recoding methods for anonymizing set-valued data». In: *The VLDB Journal* 20.1 (2011), pp. 83–106 (cit. on p. 29).

[48] She Sun, Shuai Ma, Jing-He Song, Wen-Hai Yue, Xue-Lian Lin, and Tiejun Ma. «Experiments and analyses of anonymization mechanisms for trajectory data publishing». In: *Journal of Computer Science and Technology* 37.5 (2022), pp. 1026–1048 (cit. on p. 30).

[49] Gyozo Gidofalvi, Xuegang Huang, and Torben Bach Pedersen. «Privacy-preserving data mining on moving object trajectories». In: *2007 International Conference on Mobile Data Management*. IEEE. 2007, pp. 60–68 (cit. on p. 30).

[50] Chi-Yin Chow and Mohamed F Mokbel. «Trajectory privacy in location-based services and data publication». In: *ACM Sigkdd Explorations Newsletter* 13.1 (2011), pp. 19–29 (cit. on pp. 31, 32, 36).

[51] Alastair R Beresford and Frank Stajano. «Location privacy in pervasive computing». In: *IEEE Pervasive computing* 2.1 (2004), pp. 46–55 (cit. on p. 31).

[52] Tun-Hao You, Wen-Chih Peng, and Wang-Chien Lee. «Protecting moving trajectories with dummies». In: *2007 International conference on mobile data management*. IEEE. 2007, pp. 278–282 (cit. on pp. 32–34).

[53] Osman Abul, Francesco Bonchi, and Mirco Nanni. «Never walk alone: Uncertainty for anonymity in moving objects databases». In: *2008 IEEE 24th international conference on data engineering*. Ieee. 2008, pp. 376–385 (cit. on p. 34).

[54] Javier Rodriguez-Viñas, Ines Ortega-Fernandez, and Eva Sotos Martínez. «Hexanonymity: a scalable geo-positioned data clustering algorithm for anonymi‡sation purposes». In: *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2023, pp. 396–404 (cit. on p. 38).

[55] Roberto J Bayardo and Rakesh Agrawal. «Data privacy through optimal k-anonymization». In: *21st International conference on data engineering (ICDE'05)*. IEEE. 2005, pp. 217–228 (cit. on p. 51).