



An overview of proposals towards the privacy-preserving publication of trajectory data

Àlex Miranda-Pascual^{1,2} · Patricia Guerra-Balboa¹ · Javier Parra-Arnau² · Jordi Forné² · Thorsten Strufe¹

Published online: 4 September 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

The privacy risks of processing human locations and their trajectories have been demonstrated by a large number of studies and real-world incidents. As a result, many efforts are aimed at making human location trajectories available for processing while protecting the privacy of individuals. A majority of these, however, are based on concepts and evaluation methodologies that do not always provide convincing results or obvious guarantees. The processing of locations and trajectories yields benefits in numerous domains, from municipal development over traffic engineering to personalized navigation and recommendations. It can also enable a variety of promising, entirely new applications, and is, therefore, the focus of many ongoing projects. With this article, we describe common trajectory types and representations and give a classification of meaningful utility measures, describe risks and attacks, and systematize previously published privacy notions. We then survey the field of protection mechanisms, classifying them into approaches of syntactic privacy, masking for differential privacy (DP), and generative approaches with DP for synthetic data. Key insights are that syntactic notions have serious drawbacks, especially in the field of trajectory data, but also that a large part of the literature that claims DP guarantees is considerably flawed. We also gather evidence that there may be hidden potential in the development of synthetic data generators, probably especially using deep learning with DP, since the utility of synthetic data has not been very satisfactory so far.

Keywords Privacy-preserving data publishing · Trajectory privacy · Syntactic notions · Differential privacy · Synthetic data · Utility metrics

Àlex Miranda-Pascual and Patricia Guerra-Balboa have contributed equally to this work.

Please note that preliminary results have been previously published in the 23rd Privacy Enhancing Technologies Symposium [90].

✉ Javier Parra-Arnau
javier.parra@upc.edu

Àlex Miranda-Pascual
alex.miranda.pascual@upc.edu

Patricia Guerra-Balboa
patricia.balboa@kit.edu

Jordi Forné
jordi.forne@upc.edu

Thorsten Strufe
thorsten.strufe@kit.edu

¹ Karlsruhe Institute of Technology, KASTEL Security Research Labs, Karlsruhe, Germany

² Department of Network Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain

1 Introduction

Trajectory data mining and analysis have emerged as a significant field of study owing to their extensive range of applications [76]. Beyond their potential to enhance everyday life through navigation and route recommendations, these processes have found diverse institutional data-analytic applications in both public and private sectors. The remarkable growth of trajectory analysis owes to the ability of personal devices (e.g., wearables, smartphones [7]) and navigation systems to collect, process, and analyze data with accuracy, coupled with their pervasive availability, all of which have been made possible by recent technological advancements. The domains benefiting from trajectory analyses span traffic management, urban planning, transportation system design, routing advice, and homeland security, among others [50].

While data analysis brings about economic and societal benefits, concerns related to privacy risks are on the rise [95, 112]. Thus, safeguarding data subjects and minimiz-

ing potential harm inflicted upon them assume paramount importance. Consequently, legal frameworks in the European Union and other regions explicitly restrict the collection, processing, and sharing of personal data. For instance, the European General Data Protection Regulation (GDPR) mandates the anonymization of personal data as a means to bypass processing restrictions [43]. Therefore, ensuring rigorous privacy preservation when analyzing location trajectories is not only a matter of best practice but also a legal obligation.

In essence, trajectories are sequences of timestamped locations, such as GPS coordinates. While seemingly innocuous with regard to user privacy, trajectories can inadvertently expose precise home locations and even reveal accurate behavioral patterns [105]. They readily disclose specific activities and their durations for individual users. By exploiting this information, one can infer circumstances and trends that impact sensitive aspects of an individual's life, including health status, religious beliefs, social relationships, and sexual preferences [23].

In this paper, we investigate the feasibility of publishing complete trajectory databases while ensuring privacy guarantees for this purpose. To achieve this, the field of *statistical disclosure control* (SDC) is employed, aiming to prevent the identification of confidential information with specific individuals when releasing data [66]. The objective is to transform a raw database into a sanitized version thereof that reduces the risk of disclosure while preserving the *utility* of the data, i.e., ensuring that statistical analyses yield similar results in both the original and sanitized databases. However, adapting SDC techniques to protect human mobility trajectories presents considerable challenges, as discussed in the following sections. Prominent privacy metrics in the field, such as k -anonymity [102] or ϵ -differential privacy (ϵ -DP) [38], are not immediately applicable to sequential and high-dimensional data.

The uniqueness of human traces implies that adversaries can easily attack seemingly protected data with minimal background knowledge about the individuals, such as their home or workplace locations [29, 130]. In this sense, de Montjoye et al. [31] show that knowing only four spatio-temporal points at low resolution is sufficient to uniquely identify 95% of individuals in a large-scale database. Moreover, through the use of auxiliary public information, such as road maps, speed limits, or simple spatio-temporal correlation models, a sanitized trajectory can be reconstructed within an obfuscated area [10, 125]. These factors contribute to inadequate privacy protection. Although numerous proposals exist in the literature, most suffer from evident deficiencies, including vulnerabilities to simple attacks or compromised utility due to the loss of information contained in trajectory data or the publication of impossible trajectories.

Last but not least, many applications involving trajectory data analysis require repeated computations as they

often monitor specific conditions, such as traffic. However, regularly publishing updated versions of a database while preserving privacy adds further complexity to the challenge. Each publication introduces some degree of information leakage about the individuals in the database, making it difficult to ensure that combinations of published sanitized data will not compromise privacy at any given moment.

The aforementioned issues raise significant concerns about the current state of the art in trajectory privacy. They cast doubt on the effectiveness of existing technologies in guaranteeing individuals' privacy and achieving an acceptable balance between privacy and data utility. Consequently, a comprehensive systematization of use cases, limitations, and misconceptions in the field is imperative, along with the establishment of a standardized classification that aids researchers in selecting appropriate privacy metrics, developing suitable mechanisms, and adequately measuring utility.

Contributions and Related Work: This paper presents a comprehensive and systematic analysis of the state of the art in *privacy-preserving trajectory publication*, which aims to publish trajectory databases that guarantee privacy while preserving utility. Our analysis explores two broad techniques for achieving privacy protection: *masking*, which involves modifying the original database [66], and *synthetic data generation*, which generates new data preserving certain statistical properties of the original database [66]. In this paper, we extend our previous work [90], in which we only covered DP masking techniques, by adding novel systematizations of two large families of trajectory protection mechanisms: *syntactic masking mechanisms*, based on k -anonymity and its extensions (syntactic privacy notions), and *DP synthetic data generation*. This version also includes the analysis of *DP masking mechanisms* from our original publication [90] for completeness. Overall, in this paper, we completely cover the private publication of *entire trajectory databases* under formal privacy notions by reviewing 38 mechanisms in the literature, doubling the number of mechanisms reviewed in our previous work [90]. Note that we do not cover orthogonal topics such as the publication of aggregated statistics in this systematization.

The review and analysis of privacy technologies for trajectory data in this paper include the following sections: *preliminary concepts*, which cover utility metrics and attacks (Sect. 2); *privacy notions* (Sect. 3); *syntactic masking mechanisms* (Sect. 4); *DP masking mechanisms* (Sect. 5); and *DP mechanisms for the generation of synthetic data* (Sect. 6). The contributions of this paper towards the systematization of knowledge in the field are as follows:

- A systematic analysis of how the utility of sanitized trajectory data can be measured, including a novel classification of utility metrics.

- A review of syntactic privacy notions and DP adaptations and granularity variations proposed for trajectory data. The paper also discusses and recompiles the challenges and limitations of DP as a privacy notion in the context of trajectories.
- Proposals of novel taxonomies of privacy-protecting technologies for trajectory data, along with systematic surveys of the state of the art and recent advances in the literature. The taxonomies cover:
 - **syntactic masking mechanisms**, along with an examination of the common structure and techniques they employ;
 - **DP masking mechanisms**, including mathematical proofs demonstrating that a notable percentage of the algorithms erroneously claim to satisfy DP; and
 - **synthetic trajectory generation with DP**, exploring of the current methods’ open problems.

In the sequel, we briefly describe the main differences between our work and previous surveys in this field. Primate et al [99] provide a deep analysis of location-privacy protection mechanisms, including a division of the protection mechanisms into online and offline methods. However, the authors do not cover trajectory privacy extensively since their main focus is on the more general field of location privacy. Note that trajectory data is inherently more complex than simple location data: trajectories are not only comprised of visited locations but also include correlations and connections between them. In consequence, attacks, privacy-protection mechanisms, and limitations are notably different, even though these data types share a close relationship. In addition, Portela et al. [97] focus on trajectory anonymization mechanisms under syntactic notions, although they do not explore those under DP. Fiore et al [44] offer a thorough overview and classification of attacks on trajectory databases and discuss privacy-preserving mechanisms. However, they do not study the various privacy and utility metrics available in the literature for trajectory protection, nor the limitations of DP for trajectory data. More recently, Jin et al. [69] conduct a survey with an analysis and empirical evaluation of trajectory-privacy models to quantify their privacy and utility, but do not consider DP mechanisms in depth.

In particular, our work also explores DP masking mechanisms for private database publication, which the aforementioned surveys do not fully cover. Other works focus on orthogonal topics, such as location privacy (without discussing trajectories) [68].

2 Preliminaries

2.1 Trajectories and their data sets

Trajectories correspond to a path or trace generated or drawn by a *moving object*, usually referred to as an *individual* or *user* (we will refer as such independently on what they are, e.g., a person walking, or a car carrying various people).

Different types of trajectories exist. *Raw trajectories* consist of an ordered sequence of spatio-temporal points $T = \langle p_1, \dots, p_m \rangle$, also written as $T = p_1 \rightarrow \dots \rightarrow p_m$, where $|T| := m$ denotes the *length* of T and $p_i = (x_i, y_i, t_i)$ corresponds to the location (x_i, y_i) at timestamp t_i . Trajectories respect the temporal order (i.e., t_{i+1} must happen strictly after t_i), which ensures there are no movements back in time, and no one is in two different locations at once. The term *subtrajectory* usually refers to a subset of a trajectory, including those formed by non-necessarily consecutive locations, while *n-grams* (also called *subsequences*) are subtrajectories formed by n consecutive spatio-temporal points. The *prefixes* of a trajectory $T = \langle p_1, \dots, p_m \rangle$ are the *n-grams* ($n \leq m$) starting at p_1 , i.e., $\langle p_1, \dots, p_n \rangle$.

Semantic trajectories are alternative representations where every spatio-temporal point contains additional *semantic meaning*, such as a name and description (e.g., “coffee shop” or “work”), possibly augmented with additional information such as the number of visitors or opening hours. In this latter case, locations are called *point of interest* (POI). More complex trajectories, called *multiple aspect trajectories* [89], additionally consider any possible type of recordable information, like weather variations, transportation mode, or the current heart rate or emotions of individuals. Simplified trajectories have been suggested, such as $T = \langle (x_1, y_1), \dots, (x_m, y_m) \rangle$, where time is omitted and only the order of locations is retained [18, 19, 57, 61].

We will refer to the *spatial* and *temporal* aspects as *dimensions* of a trajectory, which are both commonly represented as numerical data. Semantic locations additionally have a *categorical dimension*.

Trajectory databases consist of one or multiple trajectories from individuals, usually over a shared region. We can represent them as collections of rows, where each row contains the data of a single individual:

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

where T_i denotes a trajectory belonging to user i . The length of each trajectory is denoted here by m_i and depends on each user. In some contexts, the same user can contribute multiple

trajectories to the database. In this latter case, i is just a label of the trajectory and does not necessarily relate to a user.

Differences in structure between such databases exist. 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) [9, 40]. Further types include those with irregular recordings, with spatio-temporal points only included when the user is at a relevant location [13].

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 viewed as a sequence $D = \{S_1, \dots, S_t, \dots\}$, where each *update* S_i represents the information corresponding to time i :

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

The database at time t is denoted $D_t = \{S_1, \dots, S_t\}$ and called a *stream prefix*. Note that since some databases consist of non-periodically recorded trajectories, “gaps” in this representation are possible. Hence, 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 its protection exceptionally difficult. Long trajectories cause problems due to the *curse of dimensionality* [4, 35], and the sparseness and uniqueness of trajectories can aid in re-identification. Another risk factor is the semantic meaning of points since this information can be enough to expose individuals.

A notorious statistical property of trajectory databases is the presence of correlation. Two conceptually different correlations are present in trajectory data (see also Fig. 1): *Correlations between trajectories* refers to the case when multiple user’s records are correlated. In families’ trajectories, for instance, we are bound to observe high correlations between their corresponding records as they engage in shared activities. Furthermore, an extreme case is regular repetitions of trajectories contributed by the same individual. *Correlations between attributes* refers to the correlation in the data a single user contributes to the database. In the case of trajectories, it refers to the correlations within the spatio-temporal and semantic dimensions. A high-correlation level exists between close timestamps due to the laws of physics, route distribution, or social patterns. It is also termed *auto-correlation* for time series data. We present in Sect. 3.2.3 the implications of correlation in privacy.



Fig. 1 Two types of correlation in a trajectory database. In the green ellipse, we see correlations between users living in the same house so that their initial steps are the same. In the blue ellipse, we see the autocorrelation of the user trajectory

2.2 Utility metrics

Privacy mechanisms aim to balance two conflicting goals: strong privacy and high utility. Typically, these mechanisms introduce obfuscation and remove detail in the data, so improving privacy usually comes with a reduction in utility. Measuring the utility and privacy provided by an algorithm is laborious since trajectory data are very complex, and the presence of semantic values can complicate its study.

In this section, we explore utility metrics to assess protected trajectories and provide a new classification. We identify two major goals a sanitization mechanism can aim to preserve: data and statistics. Here, we understand *data preservation* as how much of the output data correspond to the original one (i.e., remains unaltered after sanitization); and *statistic preservation* as the preservation of specific properties of the database (e.g., numbers of visits to important locations), usually extracted from it with query functions. Our classification thus follows this idea, dividing into *data* and *statistics preservation*. Since assuring data realism is a significant utility condition, we also introduce the orthogonal category of *realism assurance*. The right-most columns of Tables 3, 4 and 5 show the types of metrics used for mechanisms explored in this work.

2.2.1 Data preservation

These metrics measure utility based on the number or proportion of data that is left unaltered after sanitization, or the extent to which it is changed. Technically speaking, not modifying the database would yield the highest possible utility, obviously at the cost of total privacy loss. We further distinguish two subclasses for data preservation: *total*, which looks at how much data remains exactly the same; and *close*, which instead measures some distance since, typically, per-

turbing trajectories slightly (e.g., moving locations by a few meters) does not strongly hinder the utility of the mechanism in question.

Location preservation [36] is a good example of total data preservation: one maintains high utility when the protected trajectories include many locations present in the original data and not fake ones. Similarly, some proposals evaluate utility as the number or percentage of discarded trajectories [36, 37] or locations [20, 36, 93, 117], or as the size of the restricted area of the map with perturbation [122].

A popular way [12, 22, 28, 36, 60, 64, 78, 82, 92, 104, 140] to quantify close data preservation is by using *similarity measures*, which output a value representing how different two trajectories are. For example, in mechanisms such that a one-to-one correspondence between the original and sanitized trajectories exists, we can use similarity measures to compute the average values between each pair. Similarity measures are also frequently used within the privacy mechanism to, for example, determine which trajectories should be clustered and merged, considering it preferable to cluster the most similar ones. We previously explored in depth existing similarity measures [90], and we provide a quick recompilation in Table 1. There are also other compilations of trajectory similarity functions [86, 107, 111, 114, 119]. Traffic management is one exemplary use case that can benefit from this group of metrics.

Another popular utility metric is to compare the size of anonymized regions or the final resolution of sanitized data with respect to the original [2, 3, 54, 91, 98, 117]. Other close metrics include *map inference metric* [122], which infers the geometry of the road maps drawn by trajectories between the original and sanitized databases. *Preservation range query* [28] represents the percentage of obfuscated locations in a data set that remain at a distance no greater than δ from their original counterpart.

2.2.2 Statistics preservation

In contrast with the previous categories, this one does not look at the preservation of the data comprising the database, but at specific extractable information. These statistics are extracted using query functions, and therefore the *relative error query function* [2, 3, 12, 18, 19, 33, 34, 36, 57–59, 124, 139] is frequently employed to study their preservation. Given the query q , it computes the difference between the outputs when using the original database D and the sanitized D' as

$$\text{error}(q) = \frac{|q(D) - q(D')|}{\max\{q(D), b\}},$$

where b is the sanity bound used for extremely selective queries (usually chosen to be equal to 0.1% or 1% of $|D|$).

Since these queries can be defined to extract any information from the database, we find multiple diverse examples in the literature. Some of the most common ones relate to visitor numbers and location popularity. For example, *frequent sequential pattern mining* looks at the k most common subtrajectories in the original and sanitized database, either by seeing if they match over the databases [3, 18, 19, 34, 57–59, 61, 96, 98, 138] or by comparing the counts of such [12, 19, 122]. Similarly, *count queries* [18, 19, 34, 57–59, 82, 139] can be utilized to check whether the number of visitors to locations is retained or not. Additionally, some metrics tackle the preservation of *number of trajectories* [82], *most visited locations* [34, 58, 84], *hotspots* [28], *location popularity* [122], *frequency over roads* [37], *flow density* [12, 93, 122], *being inside/outside a region* [2, 3, 36, 60, 93], the start and end points distribution (*trip error* [34, 57–59, 61, 122]) and home and work distributions [54].

Another popular metric type is *trajectory length preservation* [34, 54, 57–59, 61, 84, 122, 141]. Three variations have been suggested in the literature, varying in usefulness: preservation of the *total travel distance* (i.e., the sum of the physical length between locations), the trajectory *diameter* (i.e., the maximum physical length between any two of its points), and the total number of points in the trajectory.

These metrics are of special interest for commercial purposes, where specific information on trajectories is needed rather than whole trajectories. For example, vendors may be interested in placing their advertising banners on the busiest streets, and city hall may be interested in the distribution of start and end points to decide where to build parking lots. Note this information can be preserved and extracted from sanitized trajectories, without being similar to the original ones in all other respects.

2.2.3 Realism assurance

Finally, we introduce this category that measures the ability of an algorithm to output realistic values. It is motivated by the fact that some methods produce *geospatial inconsistencies* (i.e., with points in illogical places) or *unreachable points* (i.e., a consecutive pair of locations is unattainable in the given time [36]). Accordingly, reachability is a straightforward guarantee of realism, which can be checked by measuring the distance between consecutive points (x_i, y_i, t_i) , $(x_{i+1}, y_{i+1}, t_{i+1})$ to see if they are indeed reachable, i.e., if $d((x_i, y_i), (x_{i+1}, y_{i+1})) \leq v(t_{i+1} - t_i)$ where v is the maximum velocity of the user. Similarly, the

Table 1 Comparison between similarity measures [90] (based on Magdy et al.'s comparison [86])

Similarity measure	Checks similarity by comparing	Road network	Dimensions	Can compare different lengths	Allows local time shifting	Robust to noise	Is a metric	Computational cost
Euclidean distance (and L^p -norms)	Points		S		–		✓	$\mathcal{O}(n)$
Hausdorff distance	Shape	✓	S	✓	–		✓	$\mathcal{O}(n \log(n))$
Fréchet distance	Shape	✓	S	✓	–		✓	$\mathcal{O}(nm \log(nm))$
Dynamic time warping (DTW) [6, 72]	Points (time series)	✓	S-T	✓	✓			$\mathcal{O}(nm)$
Time warp edit distance (TWED) [88]	Points (time series)	✓	S-T	✓	✓		✓	$\mathcal{O}(nm)$
Enhanced weighted DTW (EWDTW) [5]	Points (time series)	✓	S-T	✓	✓			$\mathcal{O}(nm)$
Piecewise DTW (PDTW) [73]	Points (time series)	✓	S-T	✓	✓	✓		$\mathcal{O}(nm)$ N, M final lengths
Little and Gu [81]	Movement speed and path	✓	S-T	✓	✓			$\mathcal{O}(nm)$
Longest common subsequences (LCSS) [118]	Points (time series)	✓	S-T	✓	✓	✓		$\mathcal{O}(nm)$
Edit distance on real sequences (EDR) [17]	Points (time series)	✓	S-T	✓	✓	✓		$\mathcal{O}(nm)$
Linear spatio-temporal distance (LSTD) [3]	Points (time series)	✓	S-T	✓	✓	✓		$\mathcal{O}(n + m)$
Edit distance with real penalty (ERP) [16]	Points (time series)	✓	S-T	✓	✓		✓	$\mathcal{O}(nm)$
Hwang et al. [67]	Time at intersections	✓	S-T	✓				$\mathcal{O}(nm P)$ $P = \{\text{intersections}\}$
Longest overlapping road segment (LORS) [123]	Road segments	✓	S	✓	–	✓		$\mathcal{O}(nm)$
Longest common road segment (LCRS) [133]	Road segments	✓	S	✓	–	✓		$\mathcal{O}(nm)$
Spatio-temporo-categorical distance [28]	Points		S-T-C	✓	✓		✓	$\mathcal{O}(nm)$

For dimensions, “S”, “T”, and “C” stand for spatial, temporal, and categorical, respectively; and, for computational cost, n and m correspond to the length of the two compared trajectories

previously mentioned map inference metric [122] can be used to check for geospatially incoherent points.

Observe that there currently are only a few metrics in the literature that fall into this last category, but we believe that checking or ensuring realism is essential when providing privacy mechanisms. Hence, we introduce this category to demonstrate this notable gap.

2.2.4 Conclusions on utility metrics

To sum up, mechanisms should naturally achieve good utility, and one needs to be aware that some metrics are better suited for different use cases. Notably, there is no universal utility metric for all applications, and therefore a single proposal can use multiple ones in its evaluation to widen its scope.

Data-preservation metrics are excellent for scenarios where the whole trajectory is considered, such as traffic management. Total data preservation is usually a stronger statement than its close counterpart; however, it can sometimes provide disproportionately poor values for unsuitable mechanisms. For example, if looking at location preservation, a total-preservation metric will output “no utility” given a mechanism that perturbs the coordinates of all points (such as in some DP mechanisms). In such cases, it may be more suitable to use a close variant instead. Statistics-preservation metrics are convenient for publishing information like popular locations or sequences, but they do not reflect the preservation of the whole structure of trajectories.

Assuring that the database contains realistic values is essential. Beyond reflecting good utility, it may furthermore complicate attacks such as those that aim at reconstructing original trajectories.

2.3 Risks and attacks

Having explored how we can measure the utility of trajectory data, we now discuss possible privacy risks. The main goal of trajectory privacy is to protect against risks and threats when unintended actors get access to the data.

We illustrate the tangible risks associated with a lack of privacy protection in trajectory data in the following examples. The New York City taxi data set, which included around 173 million taxi trips and the corresponding tips [115], was published in 2013. Since then, plenty of attacks on this data, using *background knowledge*, quickly appeared: Tockar [113, 115] used paparazzi photos to link celebrities’ identities to the corresponding trip in the data discovering where they went, which establishments they visited, and how much they tipped. Deneau [45] figured out that one could link stops with daily praying time to identify Muslim cab drivers. These examples are excellent representatives of two important privacy risk classes [66], *identity* and *attribute disclosure*. We

review them in the context of trajectory data in the following subsections.

Furthermore, *sensitive location disclosure* represents a risk that does not refer to leaking private information relating to users, but rather to locations. Disclosure examples are the discovery of secret Israeli and US army bases through the publication of running trails recorded by Strava through soldiers’ mobile apps [55, 62].

To show the privacy risks in human traces, we expose the possible attacks and threats of the literature. The attacks correspond to the major classification of Fung et al. [48], adapted by Jin et al. [69] (adding *group linkage attacks*), with our extension of *reconstruction* and *prediction attacks*. We also provide examples, some of which have previously been extensively surveyed [44, 69].

2.3.1 Identity disclosure

Identity disclosure is the primary risk: It happens when an adversary is able to assign an individual to their corresponding record or records in a database. Such assignation may be possible from the database alone (if it directly contains identifying information) by combining the database with external knowledge or auxiliary data, or by probabilistic inference.

Record linkage attacks (or **identity linkage attacks**) attempt to infer individuals’ identities. *Re-identification attacks* [87] are the simplest form of this type. They utilize auxiliary information, i.e., information exposed through other means and thus available to the adversary. In particular, *personal context linking attacks* [56, 125] use known information about a victim (e.g., they have been to a coffee shop) to discover their trajectory in the database.

Some record linkage attacks aim to discover uniquely identifiable traits to determine the victim’s path. In the case of trajectories, little information suffices to do so. For example, knowing four locations of an individual is sufficient to uniquely identify 95% of trajectories [31], i.e., using background information to successfully reduce the possible trajectories corresponding to the victim to exactly one. Furthermore, if using highly accurate GPS data, two points are proven to be sufficient to uniquely identify all individuals in the database [101].

Attack models can be designed to use location probability distributions, mobility preferences and patterns, exposed locations, and physical encounters in order to detect the unique traits more successfully [44]. Along this line, De Mulder et al [32] show that human movement is characterized by strong regularities and can link 80% of users in real databases. Freudiger et al. [47] exploit the uniqueness of home and work locations to design an attack model that identifies trajectories of real databases. Rossi et al [101] show that one can uniquely identify up to 95% of users when using movement data such as traveled distance, speed, and direction. In addition, loca-

tion traces can reveal speed and acceleration patterns that can identify the type of vehicle that generated the trajectory (car, truck, or motorcycle) [135]; and knowing the physical dimension of the vehicle can also help identify trajectories in vehicular data [42]. It is also easier to single out individuals in trajectories drawn by location systems of mobile devices using minor information such as the size of the users' social network [136] or their writing style in posts attached to locations [52]. We refer the interested reader to Fiore et al.'s survey [44] for a comprehensive list of similar attacks.

Membership attacks (a generalization of **table attacks** [48, 69] for non-tabular data) aim to discover whether or not a specific individual is present in the database, regardless of whether their records can be directly identified. For example, if the database shows one trajectory leaving a home location, then an adversary can deduce an inhabitant participated in the database. Learning merely the presence or absence of an individual in a trajectory database can be a direct privacy threat (e.g., consider a database of trajectories with traffic violations). Well-known examples include adaptations of *membership linkage attacks* and *membership inference attacks* in trajectory data [100, 108].

2.3.2 Attribute disclosure

The second risk is attribute disclosure: An adversary learns additional information about the previously unknown individual without necessarily identifying their exact record in the database. In trajectory data, this includes the whereabouts and temporal information (e.g., when no one is at home). The disclosure of the *user's spatial* and *temporal information* [125] is sensitive itself but can also be indirectly damaging, since it may be associated with semantic knowledge and values. Presence at a hospital for extended amounts of time allows adversaries to infer a user's health status; while being at a place and time where a specific protest is happening may leak information about a user's political opinions.

In **attribute linkage attacks**, adversaries aim to learn attributes by relying on their ability to unambiguously assign the victim to a set of records that share the same revealing attribute [48], or an exceptional distribution of attributes. In the example of Muslim taxi drivers mentioned above, the attacker inferred an attribute: the victims' religion, even though they did not identify anyone's trajectory. Sui et al [109] observe that 40% of the records that cannot be immediately identified in their data and seem anonymous were instead homogeneous and directly disclose the shared attribute.

Users' most sensitive locations are another attribute that can be exposed, for example, point-clustering algorithms that can deterministically find them already exist [142]. Gambs et al [49] demonstrate how this violates the privacy of sensitive attributes.

Group linkage attacks [69] discover connections between individuals. Relationships are particular attribute cases, and both social links and kinship can be inferred from correlated movement [23]. Their disclosure may entail different threats. Predisposition to hereditary diseases, communication between dissidents, homophily in friendships sharing religious and political views, or homosexual partnerships in certain jurisdictions are just a few prominent examples.

Another attack type is **probabilistic attacks**, which aims to improve the probabilistic belief on the sensitive information of a victim after accessing the published data [48]. One typical example is the *Bayesian inference attack*, where the attacker adversary the difference between prior and posterior beliefs about sensitive information, succeeding in the attack when this difference is high (or the posterior exceeds a chosen threshold). We describe in more detail its implications to trajectory data in Sect. 3.2.3.

Reconstruction attacks aim at rebuilding trajectories in the database. For example, Buchholz et al [10] introduce a reconstruction algorithm that can construct trajectories closer to the original data than the perturbed one. Similarly, *filtering attacks* [120] also aim at reducing noise added. On the other hand, Xu et al. [128] develop an iterative attack that can exploit the uniqueness and regularity of human mobility to step-by-step recover individual's trajectories from mobility data without using any background knowledge.

Finally, we point out that 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. Additionally, adversaries can infer whether users will be home or not, and plan, e.g., a robbery. As an instance of this, Song et al. [105] demonstrate successful *movement pattern predictions* [49] with up to 93% average chance to correctly predict mobility behavior.

3 Privacy notions

There are two well-known families of privacy notions in SDC [66]: syntactic and semantic notions [24]. *Syntactic notions* specify conditions a sanitized database should exhibit; while *semantic notions*¹ describe guarantees that the mechanism chosen for releasing the data should satisfy [30].

Algorithms for trajectory anonymization have been proposed based on these two families. These are essentially represented by the formal privacy guarantees of k -anonymity (syntactic) and ϵ -DP (semantic). The former assures the pri-

¹ Do not confuse "semantic privacy notions" with "semantic meaning" of a location. The term "semantic privacy" comes from the related cryptographic notion of semantic security, while the term "semantic meaning" of a location relates to its real-world definition and aspects (i.e., the location is a restaurant).

vacy principle of *indistinguishability* [131], under which an attacker cannot distinguish an individual from a group, defending then against record linkage attacks. DP, on the other hand, assures the principle of *uninformativeness* [79], i.e., an attacker cannot considerably improve on their prior knowledge after accessing the database, and protects against probabilistic attacks. These notions have then been adapted to trajectory data, introducing new variations, such as (k, δ) -anonymity [2] and ℓ -trajectory privacy [13].

There are also other mechanisms defined exclusively for trajectory anonymization that do not achieve any formal notion of privacy. Fiore et al. [44] groups them under *mitigation*, which aims at reducing privacy risks without pursuing any well-known privacy principle. These include [44, 69, 122]: *obfuscation* (adding noise), *cloaking* or *path confusion* (increasing sample coarsening; or selectively removing points), *segmentation* (splitting trajectories), *swapping* (segmenting but reconnecting trajectories of different users), *mix-zones* (swapping but restricted to specific regions), *dummy* (creating synthetic trajectories), *time perturbation* (making the speed constant), and *heat-map modification*. We do not explore them in this paper because they do not verify any formal privacy notion.

In general, semantic notions can provide stronger privacy guarantees than syntactic notions because they do not require assumptions about the adversary's knowledge. Further benefits over syntactic notions are, for instance, that the sequential composition in DP holds: Specific subsequent publications of the same data yield well-defined leakage that can be controlled. We hence devote the greater part of the paper to semantic notions under DP, but still cover syntactic notions.

3.1 Syntactic notions under k -anonymity and its extensions

In this subsection, we give an overview of the syntactic notions proposed for trajectory data.

3.1.1 Base notions

The notion of k -anonymity [102, 103] and its extensions, such as l -diversity [85] and t -closeness [79], are traditional representatives in the field. We say that a database is *k-anonymous* if the information about any individual in the database cannot be distinguished from that of at least $k - 1$ others, or, more formally, if each *quasi-identifier* (QID) value $D(QID)$ appears in at least k records [103]. QIDs are the set of attributes in the database known to the attacker that can be linked to external information to re-identify individuals [102]. The privacy designer chooses which attributes are considered QIDs, and this decision may vary for each user and the knowledge of the attacker. Even though k -anonymity is secure against simple

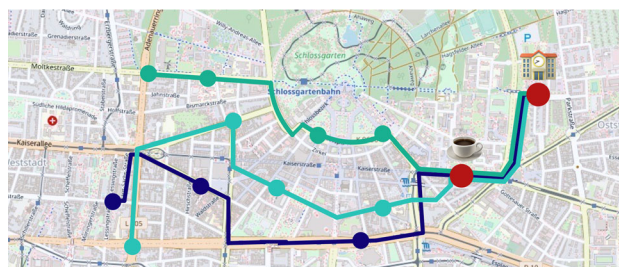


Fig. 2 Three trajectories from a database that visit the same coffee shop

re-identification attacks, it may still expose other information:

Example 1 Suppose we have a 3-anonymous database, such that each point is visited by at least 3 users (here, every spatio-temporal point is a QID), and suppose an attacker knows that their victim has visited a particular coffee shop at 15:00. Looking at the database, the attacker sees Fig. 2, where only three trajectories including this point. Thus, the attacker cannot learn which trajectory corresponds to the victim. However, the three trajectories end up at the same university, indicating that the victim works or studies there.

The notion of l -diversity is introduced to try to solve this last problem. A k -anonymous database is said to be (*distinct*) l -diverse if the number of distinct values for the sensitive attribute in the equivalence class (i.e., records containing $D(QID)$) is at least l [85]. That is, for all the individuals sharing the same quasi-identifier, there are always at least l different values for each sensitive attribute. This protects against some attribute linkage attacks, where the adversary cannot determine which of at least l values correspond to the victim. In the above example, l -diversity is satisfied if there are at least l ending locations between all trajectories that visit the coffee shop at 15:00.

Even though l -diversity ensures the variety of the sensitive attributes, it can still disclose information. For example, we could have scenarios where even though l -diversity is satisfied, 80% of the trajectories that visit the coffee shop end up at the university. This fact would allow the attacker to conclude that it is likely a student or university employee. Similarly, we could have diverse sensitive attributes, but semantically close, such as “university”, “library”, “university cafeteria”, etc. To better protect against these attacks, t -closeness is introduced, which ensures that the difference between the distribution of the sensitive attribute in the equivalence class and its distribution in the whole database does not exceed a threshold t [79].

3.1.2 Determining the QIDs in trajectory data

It is necessary to define what the QIDs are when introducing k -anonymity. Anonymization can be divided according to the type of QIDs assumed [8]: *QID-blind* anonymization defines constant QIDs for all users (e.g., each location is a QID), while *QID-aware* anonymization allows a user's QIDs to be different from others. Using the latter can lead to complications in the implementation, since QIDs and sensitive information can vary drastically between users, or depend on external characteristics. Next, we review some of the proposed QID-blind definitions.

For starters, every point can be defined as a QID (such as in Example 1). This means that if a point (or a set thereof) appears in one trajectory, then it must appear in at least $k - 1$ different trajectories. This is the simplest type of protection, and while it is straightforward, it does not ensure that sequences are k -anonymous. Note that the sequentiality of data is central to trajectories and it remains unprotected under this definition.

Domingo-Ferrer and Trujillo-Rasua [36] go even further and introduce a variation protecting any proper subset of locations. Formally, their definition of trajectory k -anonymity is satisfied if the probability of correctly linking the anonymized trajectory T' with the original T , given a strict subset S of locations of T , is no greater than $\frac{1}{k}$. In other words, the probability of correct identification is worse than choosing a trajectory at random from a set of k trajectories. The authors also introduce an adaptation of l -diversity, called *location k -diversity*, that checks whether the probability of correctly identifying a location that belongs to the original trajectory T is not greater than $\frac{1}{k}$. Being related to l -diversity, this notion aims at protecting attribute linkage of locations.

To take advantage of data sequentiality, we can consider subtrajectories as QID. For example, Poulis et al [98] say that a data set is k^m -anonymous if every subtrajectory of length at most m is contained in at least k distinct trajectories. This model assumes that the adversary knows some background information and is useful in situations where the attacker does not have access to many of a user's locations. Although its utility remains high, it does not protect against all background knowledge attackers, such as adversaries who have background knowledge about more than one trajectory in the database [98].

On the contrary, Gramaglia et al. [54] argue that the only way to ensure indistinguishability against an adversary who owns a generic subset of the victim's trajectory is to ensure full-length trajectory k -anonymity (i.e., each user's trajectory is equal to at least k others' after anonymization). The recurring problem of attribute leakage is particularly apparent in this scenario, where knowledge of a single location can reveal all of a user's unprotected information.

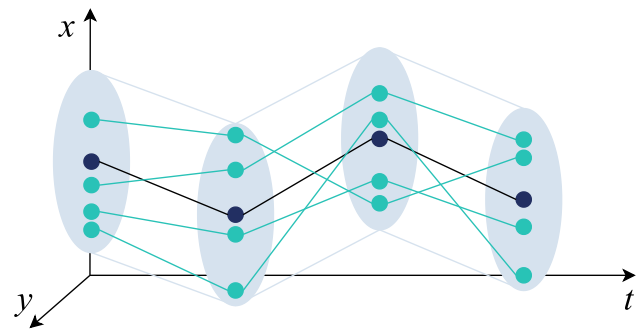


Fig. 3 Cylinder with radius δ centered on a trajectory. If, for all trajectories, $k - 1$ other trajectories are enclosed in this region, then (k, δ) -anonymity is satisfied [2]

3.1.3 Variations for trajectory data

Several attempts have been made to translate or adapt these notions specifically for trajectory data. Many variations of k -anonymity play with the definition of QIDs, as we mentioned earlier. Some adapt k -anonymity to better represent the sequential nature of trajectories. For example, we say that a database is (k, δ) -anonymous [2] for each trajectory, there exist at least $k - 1$ other trajectories such that at any given timestamp, the corresponding locations are no more than a distance of $\delta/2$ apart. This allows these k trajectories to be placed in a cylinder of radius δ , as shown in Fig. 3. However, (k, δ) -anonymity does not satisfy trajectory k -anonymity (Domingo-Ferrer and Trujillo-Rasua's definition [36]) for $\delta > 0$ [116]. Thus, it does not hide an original trajectory in a set of k indistinguishable anonymized trajectories.

Gramaglia et al. [53] identify the need for concepts that protect against probabilistic attacks. They introduce $k^{\tau, \epsilon}$ -anonymity, an uninformative extension of k^m -anonymity, which limits the maximum additional knowledge an attacker is allowed to learn. Also limiting the attacker's knowledge, $(K, C)_L$ -privacy [20] ensures that the adversary (who knows at most L locations) cannot distinguish the victim's trajectory from $K - 1$ other records, and their confidence in the sensitive-value inference is bounded by C . This notion protects against both record linkage and attribute linkage attacks. Furthermore, this notion is equivalent to l -diversity if $C = \frac{1}{l}$ and L equals the maximum length of the trajectories. Therefore, it is a more general notion. There are other notions with the same goal as l -diversity, such as c -safety [92], which bounds the probability of inferring whether a user has visited a sensitive location. More formally, a database is said to be c -safe with respect to a set of locations Q if, for every quasi-identifying sequence S_Q , the probability of the attacker inferring any user's set of sensitive locations (for any given set) is bounded by $c \in [0, 1]$.

3.1.4 Conclusions on syntactic privacy notions

Although syntactic notions can generally provide higher utility when compared to other privacy notions (e.g., DP), they present major privacy problems. Syntactic notions must assume what the attacker may or may not know, i.e., which attributes of the database are known and which are not, in order to define what is considered a QID and what is a sensitive attribute. In these cases, the attacker's background knowledge is assumed, and if the attacker has more information than assumed, the protection may not hold. They are also vulnerable to various well-known attacks (e.g., k -anonymity falls victim to attribute linkage attacks). These two shortcomings, along with the fact that the privacy guarantees of the model are not preserved after repeated independent application of the model (i.e., *composability* [106]), limit the use of syntactic technology to continuously protect trajectory data. Furthermore, data sets with sparse or short trajectories pose a great challenge for these anonymization methods. In these cases, the data must be heavily modified or even deleted, which inevitably leads to a significant loss of utility.

3.2 Differentially private notions

Differential privacy (DP) [38] is the best-known semantic notion. It aims to hide the presence or absence of any user in the database such that an analyst can extract statistics about the whole population, while an adversary cannot learn more than a limited amount about any user. The difference between the output probability of a DP mechanism, given a database that contains a user's data and one that does not, is bounded. Thus, the publication of the anonymized output reveals only bounded information about individuals, since the inference capability of any attacker is restricted.

Formally, a randomized algorithm \mathcal{M} is said to be ε -*differentially private* (ε -DP) [38] if for all *neighboring* databases $D, D' \in \mathcal{D}$ (i.e., differing in exactly one entry) and all measurable $S \subseteq \text{Range}(\mathcal{M})$,

$$\mathbb{P}\{\mathcal{M}(D) \in S\} \leq e^\varepsilon \mathbb{P}\{\mathcal{M}(D') \in S\}, \quad (3.1)$$

where \mathcal{D} is the fixed universe of all possible input databases of \mathcal{M} .

The *privacy budget* $\varepsilon > 0$ represents a measure of the privacy loss after seeing the output. From Equation (3.1), we obtain $\ln(\mathbb{P}\{\mathcal{M}(D) \in S\}) - \ln(\mathbb{P}\{\mathcal{M}(D') \in S\}) \leq \varepsilon$ for all measurable $S \subseteq \text{Range}(\mathcal{M})$, establishing thus a bound ε over the difference in distributions of outputs between two neighboring databases. Intuitively, the smaller ε , the stronger the provided privacy, i.e., if ε is small enough, then the difference between the two mentioned distributions is negligible. Thus, the attacker has no reasonable criteria to choose between the two possible input databases, limiting the

amount of information that can be learned about any given individual.

One strong point of this notion is that it does not make any assumptions about the background knowledge of the attacker. DP is a *worst-case* guarantee [15], which means it protects the privacy of any database (including outliers) against the strongest attackers.

A popular variation, called *approximate DP* or (ε, δ) -DP [39] requires instead that

$$\mathbb{P}\{\mathcal{M}(D) \in S\} \leq e^\varepsilon \mathbb{P}\{\mathcal{M}(D') \in S\} + \delta,$$

relaxing the definition to ensure bounds for rare events. In this case, for every neighboring databases $D, D', \mathbb{P}\{\mathcal{M}(D) \in S\} \leq e^\varepsilon \mathbb{P}\{\mathcal{M}(D') \in S\}$ holds for all measurable subsets S with probability at least $1 - \delta$.

Additionally, both original and approximate DP offer two beneficial properties: *composability* ensures that the combination of multiple DP algorithms is still DP, and *post-processing* implies that subsequent processing does not affect the privacy of data published with DP guarantees. These are given by the *sequential* and *parallel composition* theorems, and the *post-processing* property [39].

3.2.1 Central versus Local DP

The original, *central* DP notion assumes the presence of a trusted party (data curator) who executes the mechanisms protecting the sensitive data. If no party with shared trust exists, it is necessary to distribute the curation to all participants. The corresponding *local ε -differential privacy* (ε -LDP) [39, 71], assumes every individual holds a database containing their records and shares them only after local obfuscation. They hence contribute partial answers to queries on the whole data, enforcing DP locally.

Formally, a randomized algorithm \mathcal{M} that takes as input a user's record is said to be ε -LDP [127] if for all possible pairs of user's records x, x' and all measurable $S \subseteq \text{Range}(\mathcal{M})$, $\mathbb{P}\{\mathcal{M}(x) \in S\} \leq e^\varepsilon \mathbb{P}\{\mathcal{M}(x') \in S\}$.

This notion is stronger than central DP since there is no need to assume a trusted party. However, it is usually harder to achieve with the same utility constraints since each user needs to perturb their own record, which does not happen in the central case. Hence, the total amount of noise may be higher in the local scenario. The differences between these notions demand new hypotheses and conditions to satisfy them, as well as adapted mechanisms. A well-known example to achieve local DP in questionnaires is the randomized response.

3.2.2 Level of granularity

DP is a mathematical guarantee, so it is crucial to specify exactly what information is protected by it. The specification hinges on how the concept of neighboring databases is instantiated. Hence, various adaptations of the concept of *neighborhood* (i.e., what is considered a single entry in the database) have been suggested in the literature. We refer to the neighborhood definition as the *level of granularity* [39] of a DP notion. For example, the original DP notion aims to protect the entire existence of an individual's records or entries in a database, thus assuming a one-to-one correspondence between record and user.

In trajectory data, where several points form each user's record, the concept of granularity has special relevance. The neighborhood definition directly impacts the privacy guarantee offered. We explore the details of the most common granularity notions in our first paper [90] and provide a quick summary of these in Table 2.

In terms of privacy protection, *user-level privacy* is the strongest, followed by *element-level*, *ℓ -trajectory*, and *w-event privacy*. Finally, we fear *event-level* to be unreliable regarding trajectory privacy.

Although choosing user-level privacy may result in excessive loss of utility in the complex field of trajectory publication, none of the other granularity notions in Table 2 can provide convincing privacy guarantees. All of them allow identity disclosure, and none provide effective protection against attribute disclosure. Even if participation in a database was not sensitive information, leaking user attributes seems unacceptable in terms of privacy. Element-level privacy could be a promising attempt to protect against attribute disclosure. However, it has not been adapted to trajectory data yet. Therefore, it is difficult to assess the impact of this notion on the utility of anonymized trajectories.

3.2.3 Challenges and limitations of DP in trajectory privacy protection

DP has become the formal and de facto mathematical standard for privacy-preserving data release. Yet, recent works [21, 25, 26, 46, 70, 74, 83, 129] have demonstrated various challenges and shortcomings that this notion encounters when applied to trajectories. First, we discuss some challenges and difficulties of the application of DP to trajectories that are yet to be overcome in the literature.

Infinite streaming context. Trajectory data analysis usually requires users to continuously share spatio-temporal updates. One of the advantages of DP is its composition property. It allows publishing subsequent database updates with linearly increasing privacy loss: with r updates, the release consumes $r\epsilon$ privacy budget. The main obstacle to protecting subsequent releases of dynamic data is that the overall privacy

budget is consumed completely at some time [77]. The situation worsens when aiming to publish sanitized databases rather than global statistics since the corresponding sensitivity is usually much higher. The possibility of protection is finite in time, and parametrization gets complicated: the larger the number of releases, the smaller the ϵ assigned to each of them, and thus, the more noise added. This problem affects various use cases of trajectory data release. Traffic-jam prediction and avoidance are examples where users need to update their locations and trajectories in real time. Standard DP hence cannot be used sensibly in the streaming context, while granularity adaptations to this context, such as event-level and w -event privacy, still show serious privacy deficiencies [90]. Therefore, the DP adaptation to dynamic trajectory sharing is still an open challenge in the scientific community.

Outlier protection [58] is related to the significant utility loss incurred by the amount of noise that *outlying* sequences or trajectories (i.e., that they differ significantly from mostly any others) require to be protected. As we mentioned in Sect. 2.1, trajectories are high-dimensional and unique [31], increasing the chances of singling out or identifying records in comparison with simpler databases. In particular, the sensitivity of this type of data remains high. However, DP is a worst-case metric and it must therefore add larger amounts of noise to hide these outlying records. This is because, in most DP mechanisms, the noise added is directly proportional to the sensitivity and inversely to ϵ .

Therefore, if we assume the sensitivity is fixed, the only way of reducing noise is by increasing ϵ . This problem leads to two undesirable opposites: choosing a smaller ϵ to protect the outliers, which itself leads to lower utility in the whole of the database, or choosing a larger ϵ , leaving the outliers especially unprotected. Observe that this choice feels excessive since, with larger ϵ , non-outliers likely remain protected; but it is only the privacy concerns of possible outliers that impede this scenario because they can be outliers even after sanitization.

The challenge of finding a good trade-off between obfuscation and ϵ remains open in the literature. Some works [58] already proposed additional outlier-control mechanisms to ensure that the outliers plausibly blend into a crowd of users' trajectories. Such techniques could help attain better ϵ while avoiding the associated immense protection lost.

On the other hand, we also have intrinsic limitations of the DP notion, especially notorious in the trajectory context, that require modifications of the metric itself.

First, we encounter the **Bayesian inference threat**, which implies prior knowledge of an attacker. Taking the example by Gursoy et al. [58]: Suppose that 10% of a population lives in a district. The prior expected percentage of patients from this neighborhood in the only hospital is around 10%. Imagine now that the released data shows that 70% of trajectories

Table 2 Granularity notions and their concept of neighborhood

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

stopping in the hospital are from this district. Since the difference in values between the prior and posterior beliefs is notable, we assert that there is a privacy leakage (i.e., a health-related problem in the district). However, data should not disclose health-related information when the goal is to predict traffic jams, and there is no need to learn about health situations. DP by itself does not provide any guarantee against this phenomenon. We cannot measure how much we modify the distance between prior and posterior beliefs or if it is enough to hide sensitive information. Protection against this attack must ensure that the difference between prior and posterior about a sensitive attribute or information from data participants is sufficiently small.

This attack should not be confused with the *inference privacy fallacy* [75]. Bounding all the posterior vs. prior beliefs would end in zero utility and no possible inference process. However, we aim to protect the people participating in the database from sensitive inferences that are unnecessary for the data-analysis purpose.

Finally, problems regarding **correlation in trajectory data** in databases have recently been observed in several works [14, 21, 25, 74, 83, 129]. DP inherently assumes the database is a simple, independent random sample. This assumption implies that the database records are identically distributed (i.e., follow the same probability distribution) and independent (in particular, non-correlated). As we explained in Sect. 2.1, this is not the case for trajectory data.

One problem for DP caused by correlation relates to the difference between theoretic and real-world sensitivity:

Example 2 Suppose that Alice and Bob are married and an adversary who wants to infer the origin of Alice’s trajectory. The corresponding inference attack determines how probable the output database is, conditioned to Alice starting at a selected point or not, and chooses the answer that maximizes the probability. Now, given their relationship, Alice’s and Bob’s trajectories share points in their daily life. These could relate to their home or their favorite supermarket. The origins of Alice’s and Bob’s trajectories hence are highly correlated.

Suppose we select a location and query the database for the number of trajectories starting at this point. If we assume independence, the sensitivity of such a query is 1 (user-level), as two neighboring databases can differ in a single user’s trajectory, and each trajectory has only



Fig. 4 The green location is naturally no sensible alternative for the original blue point. Jumping from one location to another far away in seconds is not possible in real life, which is easily modeled with correlations. Changing that location also would imply changing the nearby points

one origin. Therefore, ϵ -DP is satisfied by adding Laplace noise drawn from $\text{Lap}(\frac{1}{\epsilon})$. However, in reality, Alice’s and Bob’s answers are positively correlated. Therefore, with very high probability, the difference in counts between a database where Alice started in the selected location and another where Alice did not is 2, since Bob’s answer also changes.

The correlation model, considered background knowledge, helps an attacker to infer Alice’s record as the probability distributions will be further apart than the expected ϵ bound.

Cao et al. [14] demonstrate how this problem greatly affects protection under event-level privacy due to the autocorrelation between nearby spatial points. As we see in Fig. 4, each spatio-temporal point affects other nearby points, simply due to the laws of physics and external limitations, such as road networks. As we mentioned, event-level privacy aims to protect the existence of each spatio-temporal point in the database. However, if the attacker uses autocorrelation knowledge, then the difference between the output distributions of Equation (3.1), conditioned to whether the target spatio-temporal point is in the database or not, will not be bounded by ϵ anymore. This helps the attacker to guess whether the point was originally in the database by just looking at the output.

Attribute correlations allow an adversary to invert simple perturbations: Applying time-series filters, such as the Kalman or Wiener filters, effectively removes the noise added by sanitization mechanisms, as shown by Wang et al. [120]. The post-processing property of DP should intuitively pre-

vent such attacks. However, it relies on the independence of records and breaks due to correlation.

Some notions of DP attempt to take correlations into account to overcome this issue, such as *Bayesian DP* [129] or *dependent DP* [83]. Unfortunately, they have not been analyzed in the context of trajectory privacy yet, and their adaptation is all but straightforward.

4 Syntactic masking mechanisms

Having seen the privacy notions available for trajectory privacy, we now examine the various protection mechanisms for trajectories. In this section, we look at masking mechanisms that provide syntactic privacy. First, we consider the many techniques used to construct such mechanisms.

4.1 Fundamental techniques for syntactic notions

There are several anonymization techniques to enforce syntactic privacy in trajectory data. Here we refer to *techniques* as the technical concepts upon which various specific mechanisms are designed. Classifications of these techniques exist in the literature (e.g., [97]), and although they generally include the same classes, there is no standardized classification. We provide here a list of techniques grouped into *perturbative* and *non-perturbative masking*, following Willenborg and deWaal's division [66, 126] of masking models. The difference between them is that non-perturbative techniques do not distort data, unlike perturbative ones, but rather suppress or reduce details in the original data set [66]. In this way, they preserve the truthfulness of data without distorting it, albeit losing information.

The following is an overview of the most important techniques in this area. Note that the techniques mainly ensure k -anonymity and not all of them can provide l -diversity or t -closeness.

4.1.1 Perturbative masking

We introduce some of the many perturbative anonymization techniques.

Clustering, or *microaggregation* (as it is known in the SDC field), refers to the technique of replacing a group of trajectories with a single element. More often, the locations of each trajectory, rather than the trajectory itself, are clustered at each timestamp. Clustering is usually done in two steps [3]. First, the universe of locations is partitioned into clusters such that each shares common attributes (usually given by some kind of similarity measure, e.g., they are nearby locations or share semantic information) and contains at least k elements. Second, each record in the cluster is replaced by the prototype of the cluster (usually a random location of the

cluster or its centroid is chosen). Unlike in generalization² (see Sect. 4.1.2), the data is not simply reduced in resolution, but rather replaced by different yet similar records.

Condensation starts with a partitioning of the universe of locations into exactly k locations, and then, for each group, these k locations are regenerated following the distribution and covariance of the originals. This technique does not preserve the original data, but its chosen statistics.

Other techniques modify the points within each trajectory. *Permutation* splits all trajectories into points or subsequences and then reconstructs the trajectories by sampling the split trajectories. In this way, the locations are permuted between the trajectories.

Finally, we also find *data addition*, such as the creation of new trajectories, or the *repetition* of trajectories in the database. This technique can be used to increase the number of entries in the database, sometimes even to ensure that the original and sanitized databases have the same number of trajectories. Since it adds new trajectories that were not previously in the database (although they could be copies of existing ones), we classify this technique as perturbative.

4.1.2 Non-perturbative masking

Only two techniques represent non-perturbative masking: generalization and suppression.

Generalization equalizes different records by reducing the precision of the trajectories or by grouping samples into larger areas. For example, spatio-temporal points can be transformed into regions consisting of an area of the map and a time window containing the original point. There are slight variations, e.g., in *grouping*, points are replaced by a set containing them, i.e., if p_1, \dots, p_n are to be grouped, then each p_i is changed to $\{p_1, \dots, p_n\}$. Closely related, *merging* consists of joining similar locations or trajectories (usually pair by pair) until the syntactic notion is satisfied. Generalization can be seen as the first (non-perturbative) step of clustering².

Suppression removes location samples or entire trajectories from the database. It is particularly effective in combination with other techniques, where it helps to remove locations or trajectories that are challenging to anonymize [2], such as those that are isolated or visited by a single user.

² We use the term “generalization” to refer to its original definition [102, 110], but we find instances in the literature where “clustering” and “generalization” are used interchangeably. For example, “generalization” has also been used to define the technique that additionally returns the generalized data to its original domain [91], which we refer to as “clustering”. In this paper, we use these terms differently to avoid confusion and also to make clear whether the approach is perturbative or not.

4.2 Syntactic mechanisms

In recent years, a large number of trajectory-anonymization mechanisms have been introduced. In this subsection, we review just some of the most relevant proposals. We note that the vast majority of the anonymization algorithms combine several of the aforementioned techniques. Therefore, we list them according to their core technique, but they can (and usually do) use others as well. We also provide a quick guide to them in Table 3.

4.2.1 Clustering based

The first mechanism to use k -anonymity to tackle trajectory anonymization is *Never Walk Alone* (NWA) [2]. It consists of a two-step greedy algorithm that first groups trajectories into clusters and then performs a specific clustering technique called *minimum space translation* to achieve (k, δ) -anonymity. It also suppresses outlier trajectories. Because of its historical relevance, NWA is the baseline against which most other mechanisms are compared and has been thoroughly analyzed. For example, trajectories must be defined over the same timestamps and have the same number of locations to ensure (k, δ) -anonymity. These are hard constraints to meet [36]. Subsequent variations were later introduced such as W4M [3], where the Euclidean distance function is replaced by the EDR similarity measure [17] to avoid this problem. This change allows time adjustments and avoids the preprocessing step that cuts trajectories to equal lengths, which harms the utility. They mention other improvements, such as defining trajectories directly over road networks.

The rest of the proposals first generalize the data and then return to the original domain (i.e., clustering according to our classification, even though the papers refer to it as “generalization”). One of the earliest proposals is TGA [93, 94], which is defined as a condensation-like approach. First, a random trajectory is chosen and grouped with its $k - 1$ closest trajectories. This process is then repeated until less than k trajectories remain ungrouped, which are suppressed. The points in each group of size k are either clustered into the minimum spatio-temporal bounding boxes or removed. Finally, the trajectories are returned to their original domain by randomly reconstructing representations from the original data set.

Monreale et al. [91] introduce another algorithm. In their mechanism, important locations are extracted from trajectories to create clusters, the center of which produces a tessellation of the rest of the map. For each trajectory, all consecutive points in the same region are replaced by its centroid. The authors also mention the possibility of adapting the notion to provide k -anonymity in the following ways: ensuring that each location appears in at least k different trajectories, that if someone goes from one region to another,

at least $k - 1$ other trajectories also do so, and that there is a dispersion of locations between each region. The choice of important locations can vary from user to user and must be defined before anonymization.

Finally, TOPF [37] uses frequent paths to preserve data. Defined over road networks, the method studies the frequency of roads (i.e., connections between points) and removes the infrequent ones, grouping the rest to ensure k -anonymity. Then, one representative of each group is chosen as the most similar trajectory to all others. These choices ensure that the mechanism preserves frequent patterns and has been empirically proven to outperform NWA.

4.2.2 Permutation based

Other perturbative algorithms include *SwapLocations* and *ReachLocations* [36]. Ensuring trajectory k -anonymity, the former first clusters trajectories using microaggregation and then permutes the locations inside each cluster, as well as allowing for local suppression. The latter consists instead of only a permutation step, where locations are swapped by others that are reachable (or removed if not possible). It is also defined over road networks, which leads to better utility, but achieves location k -diversity instead of trajectory k -anonymity. Both mechanisms guarantee the publication of real locations, but only *ReachLocations* actually ensures the reachability constraint is satisfied.

4.2.3 Generalization based

One of the most well-known generalization approaches is CAST [92], which anonymizes semantic trajectories to satisfy c -safety. The three-step algorithm suppresses, for each user, their quasi-identifying locations; then groups them into sets of equal size with minimal effect on the categorical dimension; finally, it generalizes QIDs and other sensitive attributes until the database is c -safe. The ultimate goal of this mechanism is to protect sequential pattern mining results. However, they do not use the spatio-temporal dimensions in their computation, which limits their use cases. In addition, generalization can only be applied to trajectories of equal length, which can cause problems with certain databases.

Poulis et al. [98] also propose methods based on location generalization. In the basic framework, SeqAnon, locations are generalized to nearby ones (starting with the shortest and least frequent trajectories) until the number of each trajectory exceeds k and k^m -anonymity can be satisfied. They also define two variants: SD-SeqAnon, which considers the categorical dimension of trajectories and tries to generalize according to their distance; and U-SeqAnon, which further satisfies predefinable utility constraints and uses suppression when generalization is not possible. Here, the utility constraints limit how much generalization each location receives

Table 3 Summary of the explored mechanisms with syntactic privacy guarantees according to our classification and the privacy notion they satisfy with the choice of QIDs

Privacy notion	QIDs	Clustering Techniques	Permutation	Repetition	Generalization	Traj. suppression	Point suppression		
(k, δ) -anonymity	Trajectories	●	○	○	○	●	○		
k -anonymity	Traj. with the same points	●	○	○	○	●	○		
k -anonymity	Points & Subtraj. of length 2	●	○	○	○	●	●		
k -anonymity	Traj. with the same points	●	○	○	○	○	○		
k -anonymity	Any proper subset	●	●	○	○	○	●		
l -diversity	Any proper subset	○	●	○	○	○	●		
c -safety	Predefined points set	○	○	○	●	○	●		
k^m -anonymity	Subtraj. of length m	○	○	○	●	○	○		
k -anonymity	Trajectories	○	○	○	○	○	○		
k -anonymity	Trajectories	○	○	○	○	○	○		
l -diversity	Semantic values	○	○	○	○	○	○		
t -closeness	Semantic values	○	○	○	○	○	○		
k -anonymity	Trajectories	○	○	●	○	●	○		
$(K, C)_L$ -privacy	Subtraj. of length L	○	○	○	○	○	●		
Name	Dimensions	<i>Traj. preservation</i>	Other	Similarity measures	Size of anon. region	Other	Frequent sequences	Inside/outside region	Other
	Data type	Total	Close	Utility metrics			Statistics		
NWA [2]	S	○	○	○	●	○	○	○	○
W4M [3]	S-T	○	○	○	●	○	●	●	○

Table 3 continued

Name	Dimensions Data type	Road networks	Traj. preservation		Similarity measures		Frequent sequences		Other
			Total	Utility metrics	Close	Other	Statistics	Inside/outside region	
TGA [93, 94]	S-T		o	o	o	o	o	o	o
Monreale et al. [91]	S		o	o	o	o	o	o	o
TOPF [37]	S	✓	•	o	o	o	o	o	•
SwapLocations [36]	S-T		•	•	•	o	o	o	o
ReachLocations [36]	S-T	✓	•	•	•	o	o	o	o
CAST [92]	C		o	o	•	o	o	o	o
SeqAnon [98]	S-T		o	o	•	o	o	o	o
SD-SeqAnon [98]	S-T-C		o	o	o	•	•	o	o
U-SeqAnon [98]	S-T-C		o	o	•	o	•	o	o
GLOVE [54]	S-T		o	o	o	•	o	o	•
Tu et al. [117]	S-T-C		o	o	o	•	o	o	o
BF-P2kA [96]	S		o	o	o	o	•	o	o
Chen et al. [20]	S-T		o	o	•	o	o	o	o

We also show in the table the dimensions of the data (“S”, “T”, and “C” stand for spatial, temporal, and categorical, respectively) and whether they are defined over road networks. The table also specifies which classes of utility metrics are used in the evaluation of the mechanisms (cf. Sect. 2.2). We chose the most representative utility metrics according to the selected mechanisms

by ensuring that all elements in the generalized locations span a single utility constraint.

Furthermore, Gramaglia et al. [54] argue that it is not necessary to provide a uniform generalization because many records require little generalization. To this end, they introduce an anonymization mechanism, GLOVE, that preserves the truthfulness of mobile data at record level. The algorithm first removes all outlying locations (not in the basic algorithm, but in a variation of it). Then it applies a non-uniform generalization by iteratively merging the two trajectories with the weakest effect on accuracy. This is done so that each sample undergoes an independent minimal accuracy reduction to ensure k -anonymity, where accuracy is defined in terms of the size of the spatial and temporal regions: the larger the size at merging, the lower the accuracy. Compared to other mechanisms in the spatio-temporal dimensions, namely W4M and TGA, GLOVE provides better utility results in anonymizing mobile trajectory data. This is partly due to the fact that the algorithm does not use perturbative techniques, unlike the others.

Finally, Tu et al. [117] propose a mechanism similar to GLOVE that tries to solve a more complex scenario. They work with semantic trajectories and address protection against attacks that reveal semantic information. As shown before, k -anonymity is not enough to protect against these attribute disclosures, and therefore it is the first trajectory-anonymization mechanism that satisfies k -anonymity, l -diversity, and t -closeness. The algorithm first merges locations to ensure l -diversity and t -closeness, and then merges trajectories to ensure k -anonymity. It also suppresses points that lead to a huge reduction in granularity when merged. Its experimental evaluation leads to a stronger notion of privacy with little reduction in utility.

4.2.4 Suppression based

Although suppression is usually used as a subordinate technique (as we have seen), some proposals revolve around it. This is the case of BF-P2kA [96]. This algorithm first constructs a prefix tree, prunes the subsequences that do not reach the k threshold in counts, and then adds the counts of the pruned trajectories to the closest trajectory under the LCSS similarity measure [118]. However, any trajectory published in BF-P2kA corresponds to a real trajectory without any other protection, and even though k -anonymity is satisfied, the indistinguishable set may contain the same trajectory k times, thus revealing it (i.e., a *homogeneity attack*).

Chen et al. [20], on the other hand, define a method based purely on suppression. It first identifies the minimal subsequences that can be removed to ensure $(K, C)_L$ -privacy, and then deletes them following either local or global suppression (i.e., suppression of a subsequence or all of its instances

in the database). It preserves instances of spatio-temporal points and frequent sequences in the trajectory data.

4.2.5 Conclusions on syntactic mechanisms

Some of the general problems of syntactic mechanisms arise directly from the techniques used.

Gramaglia et al. [53] state, regarding perturbative techniques, that to preserve data veracity, one cannot rely on randomized, perturbed, permuted, or synthetic data since the addition of fictitious data introduces unpredictable record-level biases in the final sanitized data sets. In addition, such mechanisms can lead to the creation of impossible trajectories with unreachable locations or geospatial inconsistencies. For example, the clustering-based methods (e.g., [91, 93]) can produce new points that may be illogical, such as coordinates on buildings or rivers.

The privacy problems of non-perturbative approaches can lead to a severe loss of utility. For example, generalization can be inadequate if applied inappropriately. Models that generalize a single trajectory dimension (e.g., [37, 92]) are vulnerable to attacks on the other dimensions, which may still contain sensitive information. Also, the manner in which the generalization “regions” are defined is crucial, as inappropriate choices can lead to information loss due to unnecessary generalization [11], or result in data that is susceptible to background knowledge attacks.

To make matters worse, generalization methods that generalize over all points may be ineffective for databases with long trajectories. This is known as the *curse of dimensionality* [4, 35], which refers to the events that occur only when working with high-dimensional data (here referring to trajectory length) that do not exist in lower dimensions. In other words, if we generalize point by point so that each trajectory is indistinguishable from $k - 1$ others, the generalization regions of each point will be much larger for long trajectory lengths than in the smaller cases.

Generalization also works poorly on its own, as two experiments on real databases illustrate: just to guarantee 2-anonymity in Gramaglia et al.’s setting [54], the precision must be coarsened from 100 m and 1 min granularity to 20 km and 8 h regions; while Abul et al. [2] show that 99.9% of information is completely lost when generalizing improperly. Similarly, suppression can drastically change the size of the sanitized database, removing important information when used alone.

The other major problem relates to the basic notion of privacy, k -anonymity, which does not protect against attribute linkage attacks, as mentioned above. Although we have some mechanisms that achieve a higher notion of privacy, the vast majority still only guarantee k -anonymity. Furthermore, some proposals [54, 96, 117], consider the QIDs as the whole

trajectory, where the problem of attribute linkage is even more apparent.

Even though we conclude that syntactic mechanisms suffer from many shortcomings, we can draw some insights from the proposals presented. First, many syntactic mechanisms take into account the temporal (e.g., [3, 20, 36, 54, 93, 94, 98, 117]) and even the categorical dimensions (e.g., [92, 98, 117]) of trajectory data and use this information in their anonymization processes. Additionally, we find proposals [36, 37] that consider data over road networks. We believe that such mechanisms can be used to inspire DP algorithms that consider more complex data, since the literature, as we will see in the following sections, has very few proposals for doing so.

Second, the use of suppression has been shown to significantly aid in anonymization at little cost to utility. The structure of trajectory data often contains sparse and far-away trajectories and locations, so suppressing them can allow for better utility results in the rest of the database while maintaining the same level of privacy. Since suppression can be implemented to any protection algorithm, evaluations and comparisons can be defined as to whether this improvement is achievable.

Finally, regarding their utility metrics, we see in Table 3 that both data and statistic preservation metrics are often used. Note, however, that since many of the mechanisms use suppression, many of the authors use the percentage of removed points or trajectories in their evaluation. In addition, calculating the size of the anonymized region is also a popular metric, since the anonymized databases satisfy k -anonymity.

5 DP masking mechanisms

Next, we examine masking algorithms that adapt trajectory databases for publication with DP guarantees. The corresponding state of the art we review in this work covers the static-context publication in which the sanitized database is released just once in its entirety, without subsequent updates. We classify them according to their fundamental concept. We provide an overview of our classification in Table 4, including information on the privacy notion they satisfy, their properties, and the utility metrics used for their evaluation. Observe that most of the reviewed proposals aim to achieve user-level DP.

Note that DP algorithms require a randomized approach since deterministic algorithms cannot achieve DP guarantees [39]. The two classical mechanisms to provide DP are the *Laplace* and *exponential mechanisms* [39]. Nearly all the algorithms presented in this section leverage these mechanisms in some way.

5.1 Noisy counts

We include in this class the anonymization approaches that add Laplace noise to the count of trajectories or their subsequences.

5.1.1 Exploration tree

Chen et al. [18] first construct an exploration prefix tree from the trajectory database. Each node is labeled with a possible location, which can only be an element of a predefined finite set of locations (the *universe of locations*). Every possible prefix trajectory is represented uniquely as a walk from the root node to another (i.e., we represent prefix $\langle p_1, \dots, p_m \rangle$ by the node obtained after walking through the tree following the labels: $root \rightarrow p_1 \rightarrow \dots \rightarrow p_m$). This node stores the number of times (i.e., the *counts*) the prefix appeared in the database. The tree includes all the possible trajectories drawn from the universe of locations, including those not present in the database (i.e., with a count of 0).

This way the count of each prefix of length n is stored at the n th level of the tree. To guarantee DP, Laplace noise is added to the count of each node (including the 0 ones, potentially creating sequences not contained in the original data). Since each trajectory has only one prefix of length n , the sensitivity of the mechanism is 1. A node with a noisy count of 0 becomes a leaf; otherwise expands until a maximum allowed length.

Then, to release the trajectory database, we only need to explore the resulting tree. Based on the noisy prefix tree, we can draw the sanitized database by traversing it once, calculating the number r of trajectories terminated at each node, and appending r copies of the prefix saved in that node to the output. Since creating and exploring the tree are inverse operations, there is a one-to-one correspondence between the database and the prefix tree. Note we need a post-processing module to maintain the tree consistency (i.e., the sum of counts of descendant nodes cannot be higher than that of their ancestors).

In subsequent work, the same authors improved this approach by introducing an n -gram exploration tree [19] that looks at n -grams instead of prefixes, which leads to higher counts in each node and higher sensitivity. In this case, each trajectory could add its total length to a node count. Therefore, the sensitivity is the maximum trajectory length, l_{max} , allowed in the database (any trajectory longer than l_{max} is cut before introducing it in the data). The authors also add Laplace noise on the n -grams counts. Once again, by exploring the tree, we recover the perturbed version of the original trajectories, obtaining a sanitized trajectory database. The proposed solution [19] additionally offers the possibility of creating trajectories using a Markov process, where they compute the probabilities using the noisy counts. This option

Table 4 Summary of explored DP-based mechanisms according to our classification and exact privacy notion they

Privacy notion	Classification	Ref.	Correct DP notion		Laplace Mech. (Δ)		Exponential Properties		Unb. loc. univ.	Realism
User-level ϵ -DP*	Noisy counts	Exploration tree	[122]		l_{max}	o			✓	
			[18]		1	o				
			[19]		l_{max}	o				
			[33]		l_{max}	o				
		Sequence tree	[141]	x	•	o				
			[139]	x	•	o				
			[134]	x	•	o				
		Trajectory count	[138]	x	•	o			✓	✓
		Tree + Markov	[12]	x	•	o			✓	
		Clustering			o	o				
		Random centroid			o	o				
		Exp.: k -means	[22, 64]	x	•	o			✓	
			[78]	x	•	o			✓	
		Exp.: Hilbert curves	[60]	x	•	o			✓	
	Universal clustering	[140]	x	•	o			✓		
Event-level $(0, \delta)$ -DP	Sampling + interpolation	[104]		o	o			✓		
User-level (ϵ, δ) -DP		[82]		ΔX	o			✓		
ϵ -LDP	Perturbation	[28]		o	Δd_{tp}			✓	✓	
Privacy notion	Classification	Ref.	SM: <i>Euclidean</i>	SM: <i>Hausdorff</i>	SM: <i>Other</i>	SM: <i>Other</i>	SM: <i>Other</i>	SM: <i>Other</i>	SM: <i>Other</i>	SM: <i>Spatial density</i>
			Total data preserv.	Close data preserv.	Statistics preserv.	Statistics preserv.	Statistics preserv.	Statistics preserv.	Statistics preserv.	Realism assurance
User-level ϵ -DP*	Noisy counts	Exploration tree	[122]	•	o	o	o	•	•	•
			[18]	o	o	o	o	•	•	o
			[19]	o	o	o	o	•	•	o
			[33]	o	o	o	o	•	•	o

Table 4 continued

Privacy notion	Classification	Ref.	Utility metrics				Realism assurance						
			SM: Euclidean	SM: Hausdorff	SM: Other	Other	Statistics preserv.	Seq. freq.	Spatial density	Other			
	Sequence tree	[141]	o	o	o	o	o	o	o	o	o	o	o
		[139]	o	o	o	o	o	o	o	o	o	o	o
	Trajectory count	[134]	o	o	o	o	o	o	o	o	o	o	o
	Tree + Markov	[138]	o	o	o	o	o	o	o	o	o	o	o
		[12]	o	o	o	o	o	o	o	o	o	o	o
	Clustering Random centroid		o	o	o	o	o	o	o	o	o	o	o
	Exp.: k -means	[22, 64]	o	o	o	o	o	o	o	o	o	o	o
		[78]	o	o	o	o	o	o	o	o	o	o	o
	Exp.: Hilbert curves	[60]	o	o	o	o	o	o	o	o	o	o	o
	Universal clustering	[140]	o	o	o	o	o	o	o	o	o	o	o
	Event-level $(0, \delta)$ -DP	Sampling + interpolation	o	o	o	o	o	o	o	o	o	o	o
		[104]	o	o	o	o	o	o	o	o	o	o	o
	User-level (ϵ, δ) -DP	[82]	o	o	o	o	o	o	o	o	o	o	o
	ϵ -LDP	Perturbation	o	o	o	o	o	o	o	o	o	o	o
		[28]	o	o	o	o	o	o	o	o	o	o	o

satisfy “Correct DP notion” labels mechanisms that incorrectly claim DP. We show if the algorithm uses the Laplace or exponential mechanism, and the corresponding sensitivity (Δ) of correct proposals (sensitivity is not well-defined for the incorrect algorithms). Next, we cover basic properties: whether they consider time, allow for an unbounded location universe, and assure realism. We then specify which classes of utility metrics are used to evaluate the mechanism (cf. Sect. 2.2). We highlight the most representative metrics according to the selected mechanisms. “Close data preservation” includes two specific similarity measure (SM) types: Euclidean and Hausdorff distances. “Statistics preservation” includes “location visit counts” (including location popularity metrics), “frequent sequences” and “spatial density”. For noisy counts and clustering, colored cells indicate the original proposals from which the others in each family stem
 *It provides ϵ -DP only when restricted to certain spatial areas

does not create a modified database from the original (masking) but instead generates synthetic data (see Sect. 6).

Other proposals modify these algorithms in various ways. Firstly, Wang and Kankanhalli [122] define sensitive zones and apply Chen et al's method [19] only to these zones, which provides better utility. However, their privacy notion is weaker since they do not provide DP for the whole database but only for sensitive zones.

DPLG [33] constructs the same noisy n -gram tree (therefore, the sensitivity of each node count is l_{max}) but provides a non-uniformly distributed privacy level by regulating the amount of noise added, so the location will be more or less protected depending on the area of the map it is.

All the exploration-tree-based methods have some common problems: For instance, it is necessary to assume a fixed and discrete universe of possible locations and set the maximum length of trajectories. We need these strong assumptions to bound their sensitivity. Also, the size of the trees increases exponentially with the number of locations and allowed length of trajectories. Note that limiting length would considerably reduce utility. Hence, a small location universe is required to perform these methods, which is not usually the case in real-world applications. Additionally, the mechanisms only retain spatial information and counts, with the loss of temporal information further reducing utility.

The spatio-temporal correlations of human trajectories, their regularity, and self-similarity can be easily represented by autocorrelation models (see Sect. 3.2.3). Some of the new sequences generated by the processes do not follow realistic patterns and hence can easily be identified and removed from the data by the adversary. The accuracy of this attack depends on the quality of the adversary's correlation model. The Laplace mechanism, however, does not consider correlations and is bound to choose impossible or highly unlikely sequences when adding noise to the original zero counts of these hypothetical trajectories. A simple stochastic model aggregating road-map information and physical movement laws will suffice to eliminate these cases.

5.1.2 Sequence tree

More recent approaches try to build trees storing the counts of subsequences in each node instead of only one location (i.e., *sequence trees*). This is the case of NTPT [141]. This mechanism first tries to overcome data sparseness by simplifying the trajectories. By performing an optimal segmentation process, the trajectories are divided into sequences, and then, it constructs a prefix tree where each node stores a sequence. Afterward, it adds Laplace noise to the counts of each node.

Related approaches are presented in [134, 139], with the difference that they rely on a similarity factor. More specifically, they save sequences of spatio-temporal points in a tree structure according to the number of location points they have

in common. As usual, they add Laplace noise to the count of each sequence node.

5.1.3 Trajectory count

Finally, one work considers the correlation between individuals in the database [138]. Here, the authors measure the correlation coefficients between the different trajectories in the database, which translate into privacy risk: the more correlated trajectories are, the more risk they pose. Therefore, they allocate different privacy budgets adding more Laplace noise to the counts of the risky ones.

5.1.4 Correctness of DP in noisy-counts mechanisms

We would like to note that the above suggestions [134, 138, 139, 141] suffer from a common formal mistake and do not provide DP. They output perturbed counts of only those segments, subsequences, or trajectories present in the original database, but do not change the output of hypothetical sequences with zero counts, as in the exploration-tree-based methods we discussed. These conditions contradict the definition of DP, and thus cannot provide DP. We show in the following proof that a meaningful DP mechanism cannot simply change the counts of the elements in the database:

Proposition 1 *Let \mathcal{M} be a randomized algorithm with domain \mathcal{D} .³ Suppose \mathcal{M} changes the counts of the rows of $D \in \mathcal{D}$ (where it is possible to change a positive count into 0, but not the other way around). If \mathcal{M} is ε -DP, then \mathcal{M} is the void algorithm (i.e., it outputs the empty set independently of the input).*

Proof Let \mathcal{M} be an ε -DP algorithm, as described in the statement. By definition, the output domain of \mathcal{M} is a subset $\mathcal{S} \subseteq \mathcal{D}$.

Fix $D \in \mathcal{D}$. For every $x \in D$, denote $k_x < \infty$ as the number of times x appears in D and D_x as the database obtained after removing all elements x from D . For every $x \in D$, there exists a sequence of neighboring databases of \mathcal{D} :

$$D = D_0 \rightarrow D_1 \rightarrow \dots \rightarrow D_{k_x-1} \rightarrow D_{k_x} = D_x,$$

i.e., D_{i-1} and D_i are neighboring for all $i \in \{1, \dots, k_x\}$. Then, since \mathcal{M} is ε -DP, we obtain for all measurable $S \subseteq \mathcal{S}$ and $x \in D$ that

$$\mathbb{P}\{\mathcal{M}(D) \in S\} \leq e^\varepsilon \mathbb{P}\{\mathcal{M}(D_1) \in S\}$$

³ We will use Dwork and Roth's definition of database [39], defined as a multiset drawn from \mathcal{X} , the universe of database rows (represented too by their histograms from $\mathbb{N}^{|\mathcal{X}|}$). To simplify notation, we use \mathcal{D} to denote a set of finite databases.

$$\begin{aligned} &\leq e^{2\varepsilon} \mathbb{P}\{\mathcal{M}(D_2) \in S\} \leq \dots \\ &\leq e^{(k_x-1)\varepsilon} \mathbb{P}\{\mathcal{M}(D_{k_x-1}) \in S\} \\ &\leq e^{k_x\varepsilon} \mathbb{P}\{\mathcal{M}(D_x) \in S\} = 0. \end{aligned}$$

Let $S_D \subseteq \mathcal{S}$ be the set of all possible outputs of $\mathcal{M}(D)$. It is clear that $\mathbb{P}\{\mathcal{M}(D) \in S_D\} = 1$. Furthermore, S_D is contained in the discrete set $\{S \text{ multiset} \mid \text{for all } x \in S, x \in D\}$, and therefore S_D is discrete, and

$$\mathbb{P}\{\mathcal{M}(D) \in S_D\} = \sum_{s \in S_D} \mathbb{P}\{\mathcal{M}(D) = s\}.$$

For every non-empty $s \in S_D$, we select an element $x \in s$. By the previous inequalities, we obtain that

$$\mathbb{P}\{\mathcal{M}(D) = s\} \leq e^{k_x\varepsilon} \mathbb{P}\{\mathcal{M}(D_x) = s\} = 0,$$

since $x \notin D_x$ and $x \in s$. Therefore,

$$\begin{aligned} 1 = \mathbb{P}\{\mathcal{M}(D) \in S_D\} &= \sum_{s \in S_D} \mathbb{P}\{\mathcal{M}(D) = s\} \\ &= \mathbb{P}\{\mathcal{M}(D) = \emptyset\}. \end{aligned}$$

Since $\mathcal{M}(D)$ is a discrete random variable, it proves that it can only output the empty set. Then, we repeat the proof for every possible database $D \in \mathcal{D}$, proving that \mathcal{M} is the void algorithm. \square

This fact is not reflected in the privacy analysis of these papers [134, 138, 139, 141], as the authors provide proof of the DP tools they incorporated, such as the Laplace mechanism, but do not of the privacy met by their global algorithm. Consequently, if the count of the victim’s trajectory is positive after perturbation, and this trajectory contains a quasi-identifier known by the attacker, such as their home or work, the victim and the rest of its path can still be identified.

In general, a DP mechanism needs to be able to output any possible output independently of the database. We formalize this statement with the precise hypotheses in Propositions 2 and 3, which cover the bounded and unbounded scenarios of DP. Recall that in *unbounded* DP, two databases are neighboring if we obtain one from the other by adding or removing one element; and that in *bounded* DP, these are neighboring if we obtain them instead by replacing one element with another [74, 80].

Proposition 2 *Let \mathcal{M} be a randomized algorithm that satisfies unbounded ε -DP, \mathcal{D} its domain, and $\text{Range}(\mathcal{M})$ the set of all possible outputs of \mathcal{M} . Then, given any measurable $S \subseteq \text{Range}(\mathcal{M})$, if there exist $D \in \mathcal{D}$ such that $\mathbb{P}\{\mathcal{M}(D) \in S\} > 0$, it is also true for all other $D' \in \mathcal{D}$.*

Proof Consider a measurable $S \subseteq \text{Range}(\mathcal{M})$ such that there exist $D \in \mathcal{D}$ in a way that $\mathbb{P}\{\mathcal{M}(D) \in S\} > 0$. We then

proceed by *reductio ad absurdum*: that is, we assume that there exists $D' \in \mathcal{D}$ such that $\mathbb{P}\{\mathcal{M}(D') \in S\} = 0$ and we will end in a contradiction.

Since we assume all databases are finite, there exists a finite sequence of neighboring databases from D to D' of length k . As in the proof of Proposition 1, we obtain

$$\mathbb{P}\{\mathcal{M}(D) \in S\} \leq e^{k\varepsilon} \mathbb{P}\{\mathcal{M}(D') \in S\} = 0.$$

This contradicts that $\mathbb{P}\{\mathcal{M}(D) \in S\} > 0$. \square

Proposition 3 *Let \mathcal{M} be a randomized algorithm that satisfies bounded ε -DP, \mathcal{D} its domain, and $\text{Range}(\mathcal{M})$ the set of all possible outputs of \mathcal{M} . Then, given any measurable $S \subseteq \text{Range}(\mathcal{M})$, if there exist $D \in \mathcal{D}$ such that $\mathbb{P}\{\mathcal{M}(D) \in S\} > 0$, it is also true for all other $D' \in \mathcal{D}$ such that $|D'| = |D|$.*

Proof This proof is the same as that of Proposition 2, but we must impose that $|D| = |D'|$ to ensure that there is a sequence of neighboring databases between D and D' . \square

5.1.5 Conclusions on noisy counts

We conclude that the only noisy-count mechanisms that achieve acceptable privacy guarantees are the original exploration-tree approaches [18, 19, 33, 122]. However, due to their high computational cost for large databases, we only see these methods used for cases with reduced universes, such as the analysis of public-transport lines of a city.

These algorithms excel at preserving statistics (e.g., location counts). This result is reflected in Table 4, where we see that many of the algorithms evaluate their utility using statistic-preservation metrics. On the other hand, we find fewer evaluations using data-preservation metrics and, in particular, no similarity measures.

5.2 Clustering

The next category contains mechanisms [12, 22, 60, 64, 78, 140] 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 M_1 , which generalizes the set of locations by grouping them into clusters, and a releasing mechanism M_2 , which outputs resulting trajectories drawn from the generalized sets. To achieve DP publication, both M_1 and M_2 have to be DP.

5.2.1 Exponential clustering

Hua et al. [64] is the first proposal using clustering. Their idea for M_1 is to cluster and merge concurrent locations

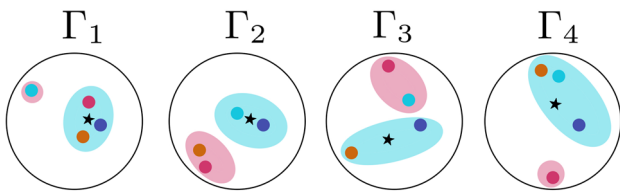


Fig. 5 Example of how trajectory data are anonymized through clustering techniques. Different trajectories are represented in different colors, with points corresponding to the physical location over each timestamp. The colored areas represent the clusters defined by the selected partition, and the stars denote the centroids of each subset. In this case, trajectories are of length $|T| = 4$, and the selected partition contains $m = 2$ subsets

from different trajectories, following a probabilistic partitioning based on the exponential mechanism. Then, using the Laplace mechanism, M_2 connects the merged locations and forms the final generalized trajectories.

Specifically, the authors suggest a score function to measure distances between trajectories crossing the corresponding locations at each timestamp. Using the exponential mechanism and this score function, they choose one of the candidate partitions (into m groups) of Γ_i , the set of locations of the database at time i . Finally, the locations of each subset are clustered together and replaced by their corresponding centroid (see Fig. 5).

After selecting a partition and replacing the locations with centroids, the location set Γ_i is replaced by a smaller one, $\tilde{\Gamma}_i$, which contains perturbed information. They build the new trajectories from this reduced set $\tilde{\Gamma}_i$ using the mechanism M_2 , which draws sequences from $\tilde{\Gamma}_i$ at random. The counts are attributed following the Laplace mechanism until obtaining a sanitized database of the same size as the original.

Works based on this proposal also exist. For example, Chen et al [22] import Hua et al’s proposal [64] as the final part of their recurrent neural network to ensure privacy. Later, Li et al [78] design an M_2 algorithm with bounded Laplace noise. In addition, Han et al. [60] propose a new private cluster mechanism M_1 based on Hilbert curves, where it is not necessary to fix the number of clusters in advance.

5.2.2 Correctness of DP in exponential-clustering mechanisms

After studying these approaches [22, 60, 64, 78], we observe an issue when applying the exponential mechanism. The exponential mechanism [39] selects the best element of a certain given set \mathcal{R} , the range of this mechanism. The best assignments for each database are chosen using a *score function* u , which associates scores to each element in the database: the higher the score, the higher its chances to be chosen. More formally, given a database $D \in \mathcal{D}$, the exponential mechanism outputs $r \in \mathcal{R}$ with probability pro-

portional to $\exp\left(\frac{\varepsilon u(D,r)}{2\Delta u}\right)$, where $u: \mathcal{D} \times \mathcal{R} \rightarrow \mathbb{R}$ is the aforementioned score function and

$$\Delta u := \max_{r \in \mathcal{R}} \max_{\substack{\text{neighb.} \\ D, D'}} |u(D, r) - u(D', r)|$$

is its sensitivity. Note, in particular, that \mathcal{R} needs to be data independent.

In the original framework [64] (from which the others stem out), the exponential mechanism is used to output the centroids of the partitions of the location set at every timestamp i . In this work, the score function is defined as $u: \mathcal{D} \times \tau \rightarrow \mathbb{R}$, with τ being the set of partitions of the locations set at time i of a specific database D . The previous expression is not well-defined, since τ depends on the chosen element $D \in \mathcal{D}$, and varies when changing to another D , as mentioned in the paper. As a direct consequence, Δu is not theoretically computable (even if fixing D , since the definition compares two different databases), and an exponential mechanism cannot be defined. Hence, one cannot claim the algorithm ensures DP via the exponential mechanism.

This error leads to some anomalies in the suggested proposal. First, the cluster size does not affect the privacy guaranteed: i.e., we can choose to partition into sets of size 1, which would simply be a mechanism outputting the original unmodified database, providing no privacy. Secondly, $u(D, r) \leq 1$ for all possible combinations, would imply that the absolute difference between any possible score function is at most 1. If the exponential mechanism were correctly applied, it would mean that changing the whole database has the same effect as changing one record, which is highly improbable.

Having explained why the mechanism is not the exponential mechanism, we discuss why it is not DP. We know that given two different sets, S and S' , their sets of partitions into m groups, \mathcal{P}_S^m and $\mathcal{P}_{S'}^m$, are disjoint. For example, consider $S = \{1, 2, 3\}$ and $S' = \{1, 2\}$. The only partition of S' into two clusters is $\mathcal{P}_{S'} = \{\{1\}, \{2\}\}$, while for S we have $\mathcal{P}_S^{(1)} = \{\{1, 2\}, \{3\}\}$, $\mathcal{P}_S^{(2)} = \{\{1, 3\}, \{2\}\}$ or $\mathcal{P}_S^{(3)} = \{\{2, 3\}, \{1\}\}$. It is then easy to see that $\mathcal{P}_S^2 \cap \mathcal{P}_{S'}^2 = \emptyset$.

More formally, consider two neighboring databases $D, D' \in \mathcal{D}$ and their respective location set at time i , Γ_i and Γ'_i . Let $\mathcal{P}, \mathcal{P}' \subseteq \text{Range}(\mathcal{M})$ be the set of all possible partitions of Γ_i and Γ'_i , respectively, into m groups. As mentioned, $\mathcal{P} \cap \mathcal{P}' = \emptyset$, so

$$1 = \mathbb{P}\{\mathcal{M}(D) \in \mathcal{P}\} \leq e^\varepsilon \mathbb{P}\{\mathcal{M}(D') \in \mathcal{P}\} = 0,$$

resulting in a contradiction with the definition of DP. As we already proved in Proposition 2, if an output is possible for a database, it needs to be possible for all the remaining ones, which simply cannot happen if the range of outputs is data dependent.

In summary, the protection mechanisms (M_1 , in particular) of the proposed approaches [22, 60, 64, 78] do not provide DP because they do not use correct exponential mechanisms, since their abstract range of outputs is completely dependent on the input database. Furthermore, we can construct an example attack that shows how DP breaks in this case [90].

The only way to avoid this issue would be to define a data-independent universe of locations, for instance, based on the city map, and output a partition of this universe. This way, the mechanism could achieve DP. Being independent of the actual patterns in the data could incur a significant utility loss in some scenarios.

5.2.3 Universal clustering

Recently, Zhao et al. [140] introduce a protection proposal independent of the specific clustering algorithm. It allows one to choose any preferred clustering and run it on the database without modification. They add Laplace noise to location coordinates (using the polar form) and to the counts of these data in the cluster. Finally, the authors calculate the noisy centroid according to the noisy counts and locations and release these centroids. The noisy count algorithm they use is the same as in the works [134, 139, 141] that we have shown to lack DP guarantees in Sect. 5.1. Furthermore, following this scheme, we cannot release more than the corresponding centroids since there is no private way of establishing connections between centroids and thus forming trajectories without using the original data. The authors do not propose any mechanism for trajectory release (M_2).

5.2.4 Random centroid

Finally, we highlight DPTD [12], which introduces a generalization module that clusters the locations without consuming privacy budget (the proposed solution chooses a random location instead of the centroid). For the release method M_2 , the authors adapt the noisy prefix tree structure by Chen et al. [19] to reduce the consumption of the privacy budget and provide higher utility. Instead of adding Laplace noise to the odd layers of the tree, they predict the new count with a Markov process. This Markov process uses the frequencies of the original database, apparently without protection (i.e., no noise or perturbation added to the frequencies). Although the authors attempt to reduce the privacy budget consumed, the generalization step indirectly uses the database in its election of the centroid, thus breaking DP. The publications also contain neither analyses nor proofs of privacy, so the actual protection achieved remains unclear.

5.2.5 General problems

Apart from the privacy issues we have explained in each proposal, we find general problems. First, the generation of impossible trajectories challenges the utility of the resulting output. Specifically, the presented methods can create trajectories in which two consecutive locations are unreachable in the given time and unrealistic centroids placed at impossible locations, such as in the middle of a river or on top of a building.

Another limitation is that the used score function of the exponential mechanism only depends on physical distance and therefore does not consider time. These proposals are thus inapplicable for non-periodically recorded and variable-length trajectories, which represent a majority of real-world databases.

Similarly, a problem arises related to stationary sequences when disregarding time. When a driver stops, the spatial location remains constant during each timestamp until the car starts to move again (e.g., see Fig. 5, where the dark blue point is constantly in the same location at each timestamp because it represents a stop position in the trajectory). The constant spatial points will be substituted by the corresponding centroids at each timestamp. However, since merging locations is only based on distances, the sanitized data will likely not reflect this stop. In Fig. 5, we can see that the locations of the dark blue stationary trajectory change into different locations at each timestamp. This produces an apparent random movement that hides the stop.

5.2.6 Conclusions on clustering

This category of approaches overcomes the applicability problem of those using trees (see Sect. 5.1), as they do not need to assume a small universe of locations. However, we can still identify several deficiencies: merging without considering time and using naïve mechanisms for releasing data (M_2) can yield poor utility and facilitate correlation attacks. Also, as mentioned above, all of these proposals contain erroneous DP analyses or proofs. It hence remains unclear which protection they provide.

Unlike in the noisy-counts algorithms, these clustering mechanisms evaluate utility mainly using similarity measures rather than statistic-preservation metrics (see Table 4), even though these last ones could be used in the utility evaluation.

In combination with the development of better release mechanisms and rigorous privacy analyses, these approaches promise to be a fruitful path for future potential research.

5.3 Sampling and interpolation

Another type of mechanism is based on point sampling and interpolation [82, 104]. The sampling technique consists of selecting a subset of the database (in this case, trajectory points), while interpolation is used to counteract the size reduction due to sampling by reconstructing intermediate points of the trajectories. The sampling techniques used do not satisfy ϵ -DP, but rather (ϵ, δ) -DP, and interpolation is conducted without affecting the privacy guarantees.

Shao et al. [104] present two mechanisms, SFI and IFS, for ship-trajectory privacy based on these techniques. SFI first randomly samples points over each trajectory and then redraws trajectories using a cubic Bézier interpolation (the “a priori” mechanism). IFS first interpolates and then samples (the “a posteriori” mechanism). The mechanisms are proven to achieve event-level $(0, \delta)$ -DP. In their experimentation, the authors conclude that SFI works better than IFS for small values of δ and not-so-smooth trajectories.

Similar to the mechanisms discussed in the previous subsections, this algorithm ignores the temporal dimension, and impossible trajectories can thus occur. Furthermore, even though SFI and IFS guarantee high utility for smooth ship trajectories, we believe this result might not extrapolate well for other trajectory types, like people or road vehicles, which can contain sharper turns and need to fit into a road network.

Another proposal is VTDP [82], which consists of a three-phased sampling with a final interpolation step and satisfies (ϵ, δ) -DP. Each of the sampling phases constructs from the previous following a well-known distribution. The first phase considers position and counts, the second additionally considers moving speed, and the third adds the temporal component. Interpolation is computed simply using the basic formulas between speed, acceleration, and time. The algorithm also uses the Laplace mechanism during the first phase to find how many elements points are to be sampled. The sensitivity of this mechanism is $\Delta X = \max_{D, D'} \|x_i - x'_i\|$, where x_i and x'_i are the optimal counts of points P_i returned by an optimization process depending on D and D' , respectively. However, there is no bound or further analysis of this sensitivity. Without a bound, it is not possible to apply this mechanism to satisfy DP properly.

The mechanism aims at preserving the original distributions and maintaining high utility throughout. With this privacy guarantee, the probability of protection against attacks such as record linkage is only $1 - \delta$. However, the authors evaluate their proposal over a database consisting only of a section of an arterial road, which asks whether the mechanism will maintain the same utility results over other trajectory databases.

5.4 Local perturbation

While LDP proposals for location privacy start to appear [121], we only find one protection mechanism [28] that perturbs semantic trajectories to satisfy ϵ -LDP. Recall that these trajectories are a time-ordered sequence of POIs visited by a user. The authors integrate public knowledge to improve the utility without affecting the privacy budget ϵ . The proposed mechanism utilizes this public knowledge to partition the set of all POIs into spatio-tempo-categorical regions, such that each contains some number of POIs.

The mechanism is divided into four parts: first, it generalizes every POI into the corresponding region; it partitions these new trajectories into n -grams, which are then individually perturbed following the exponential mechanism to ensure ϵ -LDP, where the score function is a distance function d_w defined over the spatial, temporal and categorical dimensions; then trajectories are reconstructed by minimizing the distance function; and finally, the mechanism returns to the initial domain by randomly picking a POI for each section, making sure that consecutive locations in a trajectory are reachable in the corresponding time.

This mechanism demonstrates several advantages over those described above. First of all, ϵ -LDP is a stronger privacy guarantee than ϵ -DP since there is no need for a trusted curator. Furthermore, it does consider the temporal dimension (and the categorical dimension of the trajectories). It also takes into consideration publicly available information to improve the overall utility of the mechanism, without any effects on the privacy budget, and ensures that the published data is realistic.

However, it also faces some challenges: First, to adapt the mechanism to a multiple-release setting (i.e., the same user contributing more than one trajectory), the user needs to know in advance how many trajectories they want to share, to divide the overall privacy budget by this number [28]. Adapting this approach to a streaming scenario will encounter the same challenge.

Second, the sensitivity of the exponential mechanism, Δd_w , depends on the fixed data universe. This means that it can be reasonable in small spatial areas, short time intervals, and reduced semantics, but if we consider huge spatio-temporo-categorical domains, the amount of noise needed will spoil the utility results. The authors also point out in their utility analysis that the error increases with trajectory length. The mechanism hence lends itself to small regions, for instance, the mobility within a city, rather than databases covering large areas.

It is also worth mentioning that this approach has been presented as a solution for societal-contact-tracing applications. In other use cases (e.g., traffic management), driving patterns and traffic flow are more important than semantic

values. Adapting the approach to fields such as these seems interesting, but has not yet been investigated.

6 Generating synthetic data with DP

Instead of masking techniques, i.e., creating a modified version of the original data [66], we can design *synthetic data generation* methods that generate new (artificial) data with similar statistical properties to the real data. The utility of the synthetic mechanism is considered good if the results of an analysis performed on a synthetic data set (artificial data generated by the mechanism) are close to what we would get with real data [66].

A synthetic data generative mechanism can be based on physics models using public and scientific knowledge and run simulations based on them (synthesis without real data), or it can use real databases to capture and structure the distributions and then build the generator. These original real data are called *training data* [41]. The goal of this section is to analyze the privacy concerns and state-of-the-art privacy techniques in the publication of *synthetic data* in the latter case, where these have been generated from a real training database. Accordingly, in the rest of this section, we will refer to the synthetic data generation mechanisms using training data simply as generative models.

A generative model performs two separate operations:

- *Information extraction*: Given one or more training databases, the module extracts or learns the information needed for the generative process.
- *Generation*: The training data are no longer accessible. Now the generative module creates new databases using the information learned by the extraction module.

Generators can be obtained using different techniques. Particularly for private trajectory generation, we have classified the approaches of the literature into two categories:

- i) *Frequentist inference-based approaches*: approaches that explicitly define the aggregated statistics to be extracted and use *frequentist inference* [27] techniques to do so (see Table 5). Here, each extracted statistic corresponds to a parameter of the final model used to sample a synthetic database.
- ii) *End-to-end machine learning (ML) approaches* [51], which use deep learning optimization to extract directly the global distribution of the trajectory dataset, i.e., the entire inference process is performed by a single model, often a neural network, without manual parameter extraction or separate inference stages as in the previous category.

In frequentist inference-based approaches, it is necessary to define manually which *model parameters*, the aggregated statistics from the database, we want to learn. For example, one parameter may be the distribution of the most visited locations in the database, so that the fake databases sampled from the model preserve the location distribution. These approaches have the advantage of being easy to interpret since one knows at all times what information is being memorized. However, the realism of the generated samples is limited to the learned properties [66], which usually involve only linear correlation or simple frequentist inference, while unlikely to anticipate all possible statistics an analyst will be interested in.

End-to-end ML approaches emerge in search of more realistic results, able to copy more than just the statistical properties. They aim to learn directly the distribution of the whole database, ideally capturing non-linear correlations and more complex properties that frequentist inference-based approaches cannot achieve. Several proposals for ML generative models in trajectory data have been presented. The most relevant ones can be reviewed in Luca et al's survey [84].

Being able to generate realistic synthetic data has many use cases [41]. Synthetic data is easier to obtain in large quantities and is flexible and scalable. The capacity to generate large databases makes learning algorithms more effective and accurate. In addition, the ability to sample outliers is interesting for stress-testing models.

Despite its advantages, synthetic data (whether from manual frequentist inference or end-to-end ML generators) poses some challenges. Among these, we are particularly interested in the privacy issues: even if synthetic data is fake and there is no one-to-one correspondence between records and users (so record linkage attacks are limited), it can still compromise the privacy of the individuals in the training data, since it has learned and is trying to mimic that database. If the generated samples are very similar to the training databases or exploit ML phenomena such as *overfitting* [132], new attacks such as *membership inference attacks* [108] can threaten the privacy of users whose records were in the training data.

One of the solutions proposed in the literature is to ensure that the generative mechanism satisfies DP. In essence, we want an attacker with access to a generator to be unable to determine whether a trajectory was in the training database or not. However, this simple broad interpretation leads to different scenarios depending on the type of access the attacker has. For instance, we could assume that the attacker only has access to the synthetic output database of trajectories (black-box setting), or we could assume that the attacker also has access to the trained generative mechanism with all the learned parameters (white-box attack) [63]. Obviously, if we can protect the training database against a white-box setting, we can also ensure protection against a black-box setting.

To guarantee privacy in the latter case, we need to ensure that the information extraction mechanism satisfies DP, i.e., Equation (3.1) holds for $\text{Range}(\mathcal{M}_L)$, the set of possible output features (including probability distributions) learned by the training process \mathcal{M}_L , and for all pairs of neighboring training databases.

Therefore, we review here the recent proposals for synthetic trajectory data generation with DP guarantees.

6.1 Frequentist inference-based approaches

Next, we analyze the traditional generative models that are enhanced with DP to protect their training data (in this domain more properly called sample data). All these models base their generation mechanism on sampling data from learned distributions. However, they differ in the methods used to extract the information. The explored proposals belong to one of two main categories (see Table 5): those that rely on a tree structure to infer mobility patterns, and those that perform multiple feature extraction directly from the database. The taxonomy of this classification is made explicit in Fig. 6.

6.1.1 Tree based

These proposals base their mechanism on learning the probabilities of sampling the next location conditioned on a previous n -gram, a Markov process of order $n - 1$:

$$\begin{aligned} & \mathbb{P}\{p_{i+1} \mid p_1 \rightarrow \dots \rightarrow p_i\} \\ & \approx \mathbb{P}\{p_{i+1} \mid p_{i-n+2} \rightarrow \dots \rightarrow p_i\}. \end{aligned}$$

The probabilities of this equation are called *transition probabilities*. They are computed from the counts of the training database. Given a training database D , an n -gram $s = p_{i-n+2} \rightarrow \dots \rightarrow p_i$, and p a possible location in the universe, we compute the transition probabilities in a frequentist manner:

$$\mathbb{P}\{p_{i+1} \mid p_{i-n+2} \rightarrow \dots \rightarrow p_i\} = \frac{c(D, s \rightarrow p)}{c(D, s)},$$

where $c(D, s)$ is the total number of occurrences of s in the database and $c(D, s \rightarrow p)$ is the number of occurrences of the sequence $p_{i-n+2} \rightarrow \dots \rightarrow p_i \rightarrow p$. The generative module uses the learned transition probabilities to sample data sequences.

One of the first ideas was based on the noisy-count method by Chen et al. [19] (see Sect. 5.1.1). After constructing an exploration tree that adds Laplace noise to the counts of the n -grams, we can use these noisy counts $\tilde{c}(D, s) = c(D, s) + z$ with $z \sim \text{Lap}(\frac{\Delta c}{\epsilon})$ to compute the transition probabilities.

Thanks to the post-processing property of DP [39], the generation process that uses only the noisy transition probabilities to sample sequences still guarantees DP.

In fact, this generative mechanism satisfies DP. However, it suffers from all the problems of the implemented DP mechanisms mentioned in Sect. 5.2.2, such as its assumption of a discretized and small universe with highly frequent n -grams.

The computational cost of the mechanism increases exponentially with respect to the height k of the tree, quickly becoming unfeasible for larger location universes Γ and long trajectories. Therefore, this method is only applicable in small data domains, such as the public transportation lines of a city. We refer to the problem of infeasible computational cost on large data domains or universes of possible locations as the *scalability problem*. We show the computational cost of all the data-generation proposals with respect to Γ , which are included in the last column of Table 5.

DPT [61] aims at overcoming the scalability problem. It follows n -grams [18, 19] by proposing a tree construction with noisy counts using Laplace noise and consequently sampling from the noisily computed transition probabilities. However, to mitigate the scalability problem, they incorporate a *hierarchical reference system* (HRS), which attempts to simplify the universe of locations into a grid system. While the universe of possible locations is a continuous domain, they associate each location with a cell of the grid, obtaining a discrete domain with a finite number of possible cells. They use grids of different resolutions to capture motion at different speeds: Slow motions are stored in high-resolution grids, while faster motions are stored in lower-resolution grids. Each resolution grid is connected to a tree, creating a forest. This is a data-dependent process, so it is necessary to add calibrated noise to the HRS construction. First, a reference system is constructed based on public knowledge. Then, some of the trees are dropped from the model by an optimization process using the noisy counts (after adding Laplace noise) of the transitions occurring in each tree.

Despite the progress of DPT compared to the previous work, certain shortcomings can still be identified. For example, as shown in their evaluation, for small ϵ and large tree heights, the optimization process produces a small subset of reference systems that suffer utility loss. If one tries to avoid this by keeping a smaller tree height, only the low-order Markov model holds. As we explained in Sect. 2.2, an important aspect of the utility of a mechanism is ensuring the realism of the output trajectories. However, in DPT, location points connected to the same cell of a grid are replaced by their centroids, which can produce impossible locations (e.g. the middle of a river). In terms of privacy, a good point of this proposal is that it protects all trajectories contributed by the same user, i.e., the user-level granularity with multiple trajectories shared by each user. This can help to deal with correlations between trajectories attacks (see Sect. 3.2.3).

Table 5 Summary of the explored mechanisms for synthetic trajectory generation with DP guarantees according to our classification

Privacy notion	Classification	Name	Number of model parameters	Traj. correlation Protection against	Autocorrelation	Bayesian inference	Outlier attack	Linkage attacks	
User-level ϵ -DP	Tree based	n -grams [19]	1					✓	
		DPT [61]	2	✓				✓	
	Multiple-distribution extraction	DP-STAR [57]	4						✓
		AdaTrace [58]	4			✓		✓	✓
		OptaTrace [59]	4			✓		✓	✓
		DP-MODR [34]	3						✓
Privacy notion	Diameter error Statistics preservation Utility	Count query	Most popular loc.	Freq. seq.	Length dist.	Trip dist.	Time Complexity		
User-level ϵ -DP		•		•				$\mathcal{O}(\Gamma ^{k+1})$	
		•		•		•		$\mathcal{O}(9^k \Gamma)$	
		•		•		•		$\mathcal{O}(\Gamma ^2)$	
		•		•		•		$\mathcal{O}(\Gamma ^2)$	
		•		•		•		$\mathcal{O}(\Gamma ^2)$	
		•		•		•		$\mathcal{O}(\Gamma ^3)$	

We show the number of model parameters extracted from each method and the privacy risks they protect against. The table also specifies which classes of utility metrics are used in the evaluation of the mechanisms (cf. Sect. 2.2). We chose the most representative metrics according to the selected mechanisms, appearing at least in two different proposals. We also show the time complexity of the mechanism with respect to the size of the data universe $|\Gamma|$ and, in the case of tree-based approaches, the height k of the tree

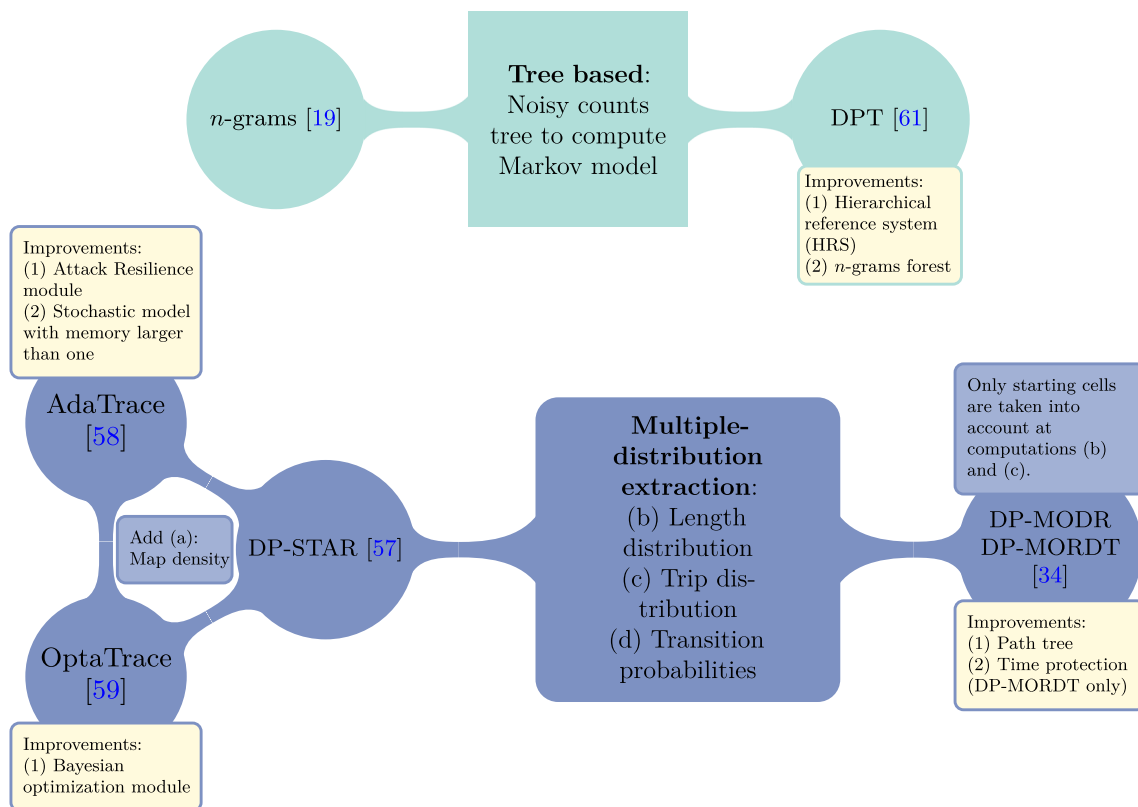


Fig. 6 Taxonomy of frequentist inference-based synthetic DP generators for trajectory data. We represent the tree-based approaches in the upper green figure and the multiple-distribution-extraction approaches in the lower blue figure. The core features common to all the mechanisms in each class are summarized in the large rectangles. We highlight

the main improvements of a mechanism over its predecessors in the yellow boxes. The initial proposals, n -gram, and DP-STAR, do not have any boxes, since they are not based on any previous proposals. Note that the three mechanisms connected in a cycle additionally use the map density distribution as one of their parameters

In general, the tree-based proposals compute only Markov probabilities, which leads to a loss of information and realism, since trajectories are more complex than just transitions, e.g. they also save information about start and end points, trip lengths, or the influence of the destination in the driving pattern.

6.1.2 Multiple-distribution extraction

Recent proposals have sought to avoid the loss of information and realism caused by relying solely on the computation of Markov probabilities. The proposed methods extract the transition probabilities as well as other feature distributions, such as the distributions of start and end locations. The combination of distributions is used to sample values and the results are combined into trajectory data.

The first mechanism of this type is DP-STAR [57]. DP-STAR starts by using a preprocessing model to reduce the maximum length of each trajectory, or in other words, to simplify it to just a few relevant points (reducing the maximum length). To solve the continuous domain problem, DP-STAR

uses an adaptive grid as a reference system (RS). This RS is not a set as in DPT, but a unique grid with a non-uniform adaptive structure. This adaptive grid preserves the density information of the database (or the actual number of reported locations in each cell), as popular areas are covered by smaller cells with higher quantities, and empty areas are covered by larger cells with lower quantities. In addition, the addition of Laplace noise to the density counts is DP-STAR's means of ensuring DP. Despite all these components, DP-STAR's use of a single adaptive grid makes for a simpler model than DPT.

DP-STAR overcomes tree-based limitations primarily through its extraction model. During extraction, DP-STAR decomposes these four statistical distributions into the following.

- (a) The distribution of location density by area (as retained by the adaptive grid RS).
- (b) The distribution of the start and end points (tree-based proposals suffer from the loss of correlation between the

start and end points, which are pruned by the tree structure due to their sparseness).

- (c) The transition probabilities from one cell to another.
- (d) The length distribution of the original trajectories.

DP-STAR's route-length estimation (step (d)) uses the exponential mechanism to preserve the probability distribution of lengths for a trip within fixed start and end cells. While tree-based methods suffer from correlation loss between sparse start and end points pruned by the tree structure, DP-STAR learns the distribution of trajectory start and end points (step (b)). DP-STAR adds Laplace noise to the probability counts to compute the empirical distribution of trip start and endpoints. DP-STAR uses a first-order Markov chain model to compute the transition probabilities from one cell to another (step (c)), which are stored in a transition matrix with values perturbed by Laplace noise.

The generator of DP-STAR works in four steps. First, start and end cells are generated according to the noisy trip distribution. Second, the corresponding route lengths of the trip are determined according to the noisy length distribution. Third, the intermediate cells are sampled according to the noisy matrix. Fourth, the locations in each cell of the final trajectory path are randomly sampled.

Later, the same authors present an improvement of DP-STAR called AdaTrace [58]. The basis of this procedure is the same as in DP-STAR, but they introduce a Markov model of a higher order than one, thus storing more information about the autocorrelation of the trajectories, and a specific defense module to overcome the deficiencies of DP against Bayesian inference and outlier-leakage attacks (see Sect. 3.2.3).

Gursoy et al. [59] subsequently introduced OptaTrace. Its improvement over DP-STAR and AdaTrace is the addition of an optimization module that attempts to minimize the error between real and synthetic databases using Bayesian optimization.

We raise several concerns about the accuracy of DP-STAR and its derivatives AdaTrace and OptaTrace. First, we ask whether the random sampling of locations within the cell by the generator may produce incoherent locations. Further, we consider it likely that during decomposition D , the influence of shorter trajectories over longer trajectories will cause a bias in the learning process. DP-STAR uses the length of the trajectory to normalize the number of transitions from cell C_i to cell C_j in each trajectory. The result is that the sensitivity of the total number of transitions from C_i to C_j in the database is set to 1, a convenient value for the usefulness of the Laplace mechanism. However, this sensitivity limit also means that a transition that has occurred n times will be retained as more likely to occur if the n occurrences were recorded in a short rather than a long trajectory. The resulting bias in the learning process is not considered in DP-STAR.

Two additional approaches, DP-MODR and DP-MODRT [34], aim at approximating regional mobility only, instead of generating exact trajectories. Thus, they generate synthetic mobility traces between coarse-grained cells mapped to cities instead of exact location trajectories. In contrast to DP-STAR, DP-MODR only preserves the distribution of starting cells and the mean length given a starting cell but does not directly store end-cell information (although this information should be stored indirectly through the mobility transition probabilities). In addition, the discretization step is performed uniformly without consuming ϵ or keeping any real density information. Otherwise, this method follows the distribution extraction and generation by sampling from learned distributions.

The tree-extracted features are the starting-cell histogram, the mean length starting in each cell, and the transition cost matrix, which instead of storing the noisy transition probabilities (Markov order 1) stores the logarithm of them. These tree features are extracted using the same DP techniques of previous work on noisy extraction [57–59]. As a novelty, the transition cost matrix is integrated into cost-sensitive path trees, i.e., instead of saving probabilities, they save the negative logarithm of the probabilities, so that a higher number means more cost and less probability. Each cell is the root of a path tree in which the transition costs along the possible paths starting from that cell are stored, allowing us to compute the total cost of each path.

Then, trajectories are generated recursively using the learned features. First, the noisy histogram is used to select how many trajectories start in each cell, then the sampling lengths are selected, and finally, the trajectory sequence is generated using the costs aggregated in the cell trees until a maximum length is reached.

Essentially, the proposed mechanism is similar to the previous one of the same group. The main difference is only the storage of the initial properties instead of the start–end distributions and a more deterministic sampling process.

The same work proposes an adaptation, called DP-MODRT [34], that considers temporal information of real trajectories. The procedure is the same, but the discretization step is done in both spatial and temporal domains. The temporal domain is discretized in intervals and the result is a set of time-dependent cells (e.g., $(C_1, 1)$). This increases the number of possible states (cell and time pairs) and thus the cost matrix. However, this matrix will have a large number of entries with infinite costs, since the probability of many transitions is 0 (which happens, e.g., because transitions back in time are impossible).

In both cases, the final output is a sequence of cells, not a trace of points as in DP-STAR, AdaTrace, and OptaTrace. This makes comparing them difficult. For example, it is challenging to compare the position distribution error when some mechanisms output real trajectories while others output cells.

Also, all evaluation metrics implemented by DP-MODR are at the cell level. They report better results than AdaTrace. However, the grid size of AdaTrace cannot be fixed, nor do they report this data. If the grid size is different, the comparison would be meaningless.

6.1.3 Conclusions on frequentist inference-based approaches

Most of these proposals completely ignore the temporal dimension by considering the trajectories as an ordered sequence of locations. With such a model, a lot of information is lost. For example, if we want to predict traffic jams, it is really important to know at what time they will occur, not just that they might occur at some time.

None of these proposals take correlations between records into account, as they all make the common assumption of independent data, which does not hold in trajectory databases (as we already mentioned in Sect. 2.1).

To avoid scalability problems, the new mechanisms use a lattice structure. This can lead to information loss when the number of cells is small, and also to impossible locations when we return to raw trajectories from cell trajectories.

Furthermore, we want to highlight AdaTrace as the only method presented in this section that provides protection against probabilistic attacks and reports good results in terms of utility. DP-MODR also reports good results compared to AdaTrace. However, due to the lack of grid information, we question the reliability of these results.

6.2 End-to-end machine learning generators with dp

Note that to the best of our knowledge, private learning has not yet been applied to trajectory generation. However, it could be an interesting place to explore, so we want to shed some light on it here.

There are generative machine learning models that can generate output with similar properties based on training data. While this idea has been applied to trajectory data as well [84], the corresponding approaches do not take privacy into account. Private learning can indeed be achieved by DP. The use of the DP stochastic gradient descent (SGD) [1] is a notable approach towards this end. This technique consists of introducing noise (usually Gaussian or Laplace) into the SGD-optimization iteration process of the learning module. Since the noise is added in the optimization process, it is possible to generalize this method to many ML models, including those that attempt to generate trajectories.

However, some doubts have been raised about the actual privacy achieved by DP-SGD. Humphries et al [65] show that it does not provide meaningful protection against membership inference attacks. Zhang et al [137] also show that decreasing ϵ does not reduce the threat of an adversary suc-

cessfully breaking privacy at the same rate as utility degrades. In conclusion, the application of privacy-enhancing technologies to synthetic trajectory ML generation is an open field that has not yet been explored.

7 Conclusions

Human location traces reveal a wide range of highly sensitive information. However, processing this information in a privacy-preserving way could be very helpful in many areas and enable completely new applications. In many efforts, various mechanisms have been developed and published to achieve privacy in trajectory data.

In this article, we have systematized the knowledge on the private publication of trajectory data. We have classified the protection mechanisms, broadly following their common distinction into syntactic privacy, semantic privacy of the publication process, and DP generation of synthetic trajectories. We also classified and described the corresponding threats, potential attacks, privacy notions, and metrics to measure the utility preserved in the protected trajectories.

In this effort, we carried out a comprehensive and systematic analysis of the published protection mechanisms. The first part of this systematization covered syntactic protection in trajectory data by reviewing some of the most relevant mechanisms in the literature. We emphasized their drawbacks regarding privacy: They are vulnerable to known attacks and require assumptions regarding the attacker's background knowledge. The syntactic mechanisms for trajectory data are varied and use several techniques in their computation. However, these techniques can be inefficient when used independently and can lead to false trajectories and results. Despite these drawbacks, we believe that syntactic mechanisms can be a starting point for improving semantic mechanisms, as several examples consider the temporal and semantic dimensions of trajectories, or that use road networks to improve utility. Suppression may also be a technique whose applicability to other mechanisms can be explored.

We also classified the masking mechanisms for trajectories that claim differential privacy into four categories and, most importantly, proved formal errors in their analyses. We showed that many of the protection mechanisms in DP masking presented obvious flaws. In this context, we reiterate the importance of the temporal dimension in trajectory privacy and the need to consider it. In addition, given that a significant fraction of the reviewed proposals do not actually provide DP as they claim to do, we would like to emphasize the importance of carefully proving whether a mechanism does so. We note that relying on pre-existing incomplete proofs can be dangerous, and it is crucial that practitioners independently verify the privacy protection provided by their mechanisms.

Furthermore, many of these proposals rely on a well-known DP mechanism but fail to define or adapt it correctly. This raises concerns that the hypotheses of core DP mechanisms, such as the exponential and Laplace mechanisms, are not correctly understood in the literature. Among the proposals on DP masking that we reviewed, we distinguish Cunningham et al.'s mechanism [28] as the most promising solution: It addresses the existing limitations and uses public knowledge to enhance the mechanism's utility.

As a third class, we investigated the privacy of generators for synthetic trajectories. While we find several approaches based on frequentist inference (traditional approaches) with DP, we point out the lack of DP-enhancing technologies in deep learning approaches. Traditional approaches meet the privacy requirements of DP, but there is still room for improvement concerning utility, both in terms of realism and temporal information.

We have also highlighted the wide range of utility metrics that can be used to evaluate a mechanism and its output. In particular, there is no universal metric, and the suitability depends heavily on the scenario. A notable takeaway is the importance of publishing realistic data (or using realistic metrics), as unrealistic data generally hinders utility and can be easily detected and attacked by adversaries.

One important challenge for future work is to address the different correlations that are implicit in trajectories. We would like to highlight that there is a need for more robust metrics that are adapted to the aforementioned trajectory properties.

In addition to overcoming these shortcomings, much remains to be done in real-world scenarios. Much of the research focuses on limited use cases, such as social contact tracing, or is limited to small location universes. Other areas (e.g., driving pattern detection) remain unexplored.

In conclusion, we are confident that research into privacy in the public release of human mobility data is still an open and quite fruitful area with many research challenges that lie ahead.

Author Contributions All authors contributed to the study conception and design. The first draft of the manuscript was written by Àlex Miranda-Pascual and Patricia Guerra-Balboa and all authors commented on previous versions of the manuscript and critically revised the work. All authors read and approved the final manuscript.

Funding Javier Parra-Arnau is the recipient of a “Ramón y Cajal” fellowship (ref. RYC2021-034256-I) funded by the Spanish Ministry of Science and Innovation and the European Union—“NextGenerationEU”/PRTR (Plan de Recuperación, Transformación y Resiliencia). This work also received support from “la Caixa” Foundation (fellowship code LCF/BQ/PR20/11770009), the European Union’s H2020 program (Marie Skłodowska-Curie grant agreement No 847648), from the Government of Spain under the project “COMPROMISE” (PID2020-113795RB-C31/AEI/10.13039/501100011033), and from the BMBF project “PROPOLIS” (16KIS1393K). This work has also received support from the project “MOBILYTICS” (TED2021-129782B-I00),

funded by MCIN/AEI/10.13039/501100011033 and the European Union “NextGenerationEU”/PRTR, and was also funded by the Generalitat de Catalunya, under AGAUR grant “2021 SGR 01413”. The authors at KIT are supported by KASTEL Security Research Labs (Topic 46.23 of the Helmholtz Association) and Germany’s Excellence Strategy (EXC 2050/1 ‘CeTI’; ID 390696704).

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose. The authors have no Conflict of interest to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Research data policy and data availability Data sharing is not applicable to this article since no datasets were generated or analyzed during the current work.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 308–318 (2016)
2. Abul, O., Bonchi, F., Nanni, M.: Never walk alone: Uncertainty for anonymity in moving objects databases. In: Proceedings of the IEEE International Conference on Data Engineering (ICDE), pp. 376–385 (2008)
3. Abul, O., Bonchi, F., Nanni, M.: Anonymization of moving objects databases by clustering and perturbation. *Inform. Syst.* **35**(8), 884–910 (2010)
4. Aggarwal, C.C.: On k -anonymity and the curse of dimensionality. In: Proceedings of the International Conference on Very Large Databases (VLDB), pp. 901–909 (2005)
5. Anantasech, P., Ratanamahatana, C.A.: Enhanced weighted dynamic time warping for time series classification. In: Proceedings of the International Conference on Information and Communication Technologies (ICICT), pp. 655–664 (2019)
6. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Proceedings of the ACM SIGKDD Conference International Data Science, Data Mining (KDD), AAAIWS’94, pp. 359–370 (1994)
7. Blumberg, A.J., Eckersley, P.: On locational privacy, and how to avoid losing it forever. *Electron Front Found (EFF)* (2009)
8. Bonchi, F., Lakshmanan, L.V., Wang, H.W.: Trajectory anonymity in publishing personal mobility data. *ACM SIGKDD Explor. Newsl.* **13**(1), 30–42 (2011). <https://doi.org/10.1145/2031331.2031336>
9. Brinkhoff, T.: A framework for generating network-based moving objects. *GeoInformatica* **6**(2), 153–180 (2002)
10. Buchholz, E., Abuadba, A., Wang, S., Nepal, S., Kanhere, S.S.: Reconstruction attack on differential private trajectory protection mechanisms (2022)
11. Byun, J.W., Kamra, A., Bertino, E., Li, N.: Efficient k -anonymization using clustering techniques. In: Proceedings of

- the Advanced Databases: Concept, Systems, Applications (DAS-FAA), pp. 188–200 (2007)
12. Cai, S., Lyu, X., Li, X., Ban, D., Zeng, T.: A trajectory released scheme for the internet of vehicles based on differential privacy. *IEEE Trans. Intell. Transp. Syst.* (2021)
 13. Cao, Y., Yoshikawa, M.: Differentially private real-time data release over infinite trajectory streams. In: *Proceedings of the IEEE International Conference on Mobile Data Management (MDM)*, vol. 2, pp. 68–73 (2015)
 14. Cao, Y., Yoshikawa, M., Xiao, Y., Xiong, L.: Quantifying differential privacy under temporal correlations. In: *Proceedings Under International Conference on Data Engineering (ICDE)*, pp. 821–832 (2017)
 15. Chen, B.C., Kifer, D., LeFevre, K., Machanavajjhala, A., et al.: Privacy-preserving data publishing. *Found Trends Database* **2**(1–2), 1–167 (2009)
 16. Chen, L., Ng, R.: On the marriage of Lp-norms and edit distance. In: *Proceedings of the International Conference on Very Large Databases (VLDB)*, pp. 792–803 (2004)
 17. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: *Proceedings of the ACM SIGMOD International Conference on Management Data (MOD)*, pp. 491–502 (2005). <https://doi.org/10.1145/1066157.1066213>
 18. Chen, R., Fung, B., Desai, B.C.: Differentially private trajectory data publication (2011)
 19. Chen, R., Acs, G., Castelluccia, C.: Differentially private sequential data publication via variable-length n -grams. In: *Proceedings ACM Conference Computing, Communications security (CCS)*, pp. 638–649 (2012)
 20. Chen, R., Fung, B.C., Mohammed, N., Desai, B.C., Wang, K.: Privacy-preserving trajectory data publishing by local suppression. *Inf. Sci.* **231**, 83–97 (2013)
 21. Chen, R., Fung, B., Yu, P.S., Desai, B.C.: Correlated network data publication via differential privacy. *VLDB J.* **23**(4), 653–676 (2014)
 22. Chen, S., Fu, A., Shen, J., Yu, S., Wang, H., Sun, H.: RNN-DP: a new differential privacy scheme base on recurrent neural network for dynamic trajectory privacy protection. *J. Netw. Comput. Appl.* **168**(102), 736 (2020)
 23. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery, Data Mining (KDD)*, pp. 1082–1090 (2011). <https://doi.org/10.1145/2020408.2020579>
 24. Clifton, C., Tassa, T.: On syntactic anonymity and differential privacy. *Trans. Data Privacy* **6**(2), 161–183 (2013)
 25. Clifton, C., Tassa, T.: On syntactic anonymity and differential privacy. In: *Proceedings of the IEEE International Conference on Data Engineering Workshop (ICDEW)*, pp. 88–93 (2013)
 26. Cormode, G.: Personal privacy vs population privacy: learning to attack anonymization. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery, Data Mining (KDD)*, pp. 1253–1261 (2011)
 27. Cox, D.R.: *Principles of Statistical Inference*. Camb. Univ. Press (2006). <https://doi.org/10.1017/CBO9780511813559>
 28. Cunningham, T., Cormode, G., Ferhatosmanoglu, H., Srivastava, D.: Real-world trajectory sharing with local differential privacy (2021)
 29. Dai, C., Pi, D., Becker, S.I., Wu, J., Cui, L., Johnson, B.: CenEEGs: valid EEG selection for classification. *ACM Trans. Knowl. Discov. Data* **14**(2), 1–25 (2020)
 30. De Capitani di Vimercati, S., Foresti, S., Livraga, G., Samarati, P.: Data privacy: Definitions and techniques. *Int. J. Uncertain. Fuzz. Knowl. Based Syst.* **20**, 793–818 (2012)
 31. de Montjoye, Y.A., Hidalgo, C.A., Verleysen, M., Blondel, V.D.: Unique in the crowd: the privacy bounds of human mobility. *Sci. Rep.* **3**(1), 1–5 (2013)
 32. De Mulder, Y., Danezis, G., Batina, L., Preneel, B.: Identification via location-profiling in GSM networks. In: *Proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES)*, pp. 23–32 (2008)
 33. Deldar, F., Abadi, M.: A differentially private location generalization approach to guarantee non-uniform privacy in moving objects databases. *Knowl. Based Syst.* **225**(107), 084 (2021)
 34. Deldar, F., Abadi, M.: Enhancing spatial and temporal utilities in differentially private moving objects database release. *Int. J. Inf. Secur.* **20**(4), 511–533 (2021)
 35. Deußner, C., Passmann, S., Strufe, T.: Browsing unicity: on the limits of anonymizing web tracking data. In: *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pp. 777–790 (2020) <https://doi.org/10.1109/SP40000.2020.00018>
 36. Domingo-Ferrer, J., Trujillo-Rasua, R.: Microaggregation- and permutation-based anonymization of movement data. *Inf. Sci.* **208**, 55–80 (2012). <https://doi.org/10.1016/j.ins.2012.04.015>
 37. Dong, Y., Pi, D.: Novel privacy-preserving algorithm based on frequent path for trajectory data publishing. *Knowl. Based Syst.* **148**, 55–65 (2018)
 38. Dwork, C.: Differential privacy. In: *Proceedings of the International Colloquium on Automata, Languages (ICALP)*, pp. 1–12 (2006)
 39. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. *Found Trends Theor. Comput. Sci.* **9**(3–4), 211–407 (2014)
 40. ECML/PKDD (2015) Taxi trajectory prediction (I). <https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data>
 41. El Emam, K., Mosquera, L., Hoptroff, R.: Practical synthetic data generation: balancing privacy and the broad availability of data. O'Reilly Media (2020)
 42. Escher, S., Sontowski, M., Berling, K., Köpsell, S., Strufe, T.: How well can your car be tracked: analysis of the European C-ITS pseudonym scheme. In: *Proceedings of the IEEE Vehicular Technology Conference (VTC-Spring)*, pp. 1–6 (2021)
 43. European Commission, Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). OJ L 119 (2016)
 44. Fiore, M., Katsikouli, P., Zavou, E., Cunche, M., Fessant, F., Le Hello, D., Aïvodji, U., Olivier, B., Quertier, T., Stanica, R.: Privacy in trajectory micro-data publishing: a survey. *Trans. Data Priv.* **3** (2020)
 45. Franceschi-Bicchierai, L.: Redditor cracks anonymous data trove to pinpoint Muslim cab drivers. *Mashable* (2015). <https://mashable.com/archive/redditor-muslim-cab-drivers>
 46. Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., Ristenpart, T.: Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In: *Proceedings of the Conference on USENIX Security Symposium*, pp. 17–32 (2014)
 47. Freudiger, J., Shokri, R., Hubaux, J.P.: Evaluating the privacy risk of location-based services. In: *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*, vol. 7035 (2011)
 48. Fung, B., Wang, K., Chen, R., Yu, P.: Privacy-preserving data publishing: a survey of recent developments. *ACM Comput. Surv.* **42**, 213 (2010). <https://doi.org/10.1145/1749603.1749605>
 49. Gams, S., Killijian, M.O., del Prado Cortez, M.N.: Show me how you move and I will tell you who you are. In: *Proceedings of the ACM SIGSPATIAL International Workshop Security Privacy*

- GIS & LBS (SPRINGL), vol. 4, pp. 34–41 (2010). <https://doi.org/10.1145/1868470.1868479>
50. Gangadharan, S.P.: How can big data be used for social good? The Guardian (2013) <https://www.theguardian.com/sustainable-business/how-can-big-data-social-good>. Accessed on 2021-01-18
 51. Glasmachers, T.: Limits of end-to-end learning. In: Proceedings of the Conference on Machine Learning (ACML), PMLR, pp. 17–32 (2017). <https://proceedings.mlr.press/v77/glasmlachers17a.html>
 52. Goga, O., Lei, H., Parthasarathi, S.H.K., Friedland, G., Sommer, R.: Teixeira R Exploiting innocuous activity for correlating users across sites. In: Proceedings of the ACM International WWW Conference, pp. 447–458 (2013)
 53. Gramaglia, M., Fiore, M., Tarable, A., Banchs, A.: Preserving mobile subscriber privacy in open datasets of spatiotemporal trajectories. In: Proceedings of the Joint Conference of the IEEE Computer Society (INFOCOM), pp. 1–9 (2017)
 54. Gramaglia, M., Fiore, M., Furno, A.: Stanica R GLOVE: towards privacy-preserving publishing of record-level-truthful mobile phone trajectories. ACM/IMS Trans. Data Sci. **2**, 3 (2021). <https://doi.org/10.1145/3451178>
 55. Gritten, D.: Strava app flaw revealed runs of Israeli officials at secret bases. BBC (2022). <https://www.bbc.com/news/world-middle-east-61879383>
 56. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), pp. 31–42 (2003). <https://doi.org/10.1145/1066116.1189037>
 57. Gursoy, M.E., Liu, L., Truex, S., Yu, L.: Differentially private and utility preserving publication of trajectory data. IEEE Trans. Mob. Comput. **18**(10), 2315–2329 (2018)
 58. Gursoy, M.E., Liu, L., Truex, S., Yu, L., Wei, W.: Utility-aware synthesis of differentially private and attack-resilient location traces. In: Proceedings of the ACM SIGSAC Conference on Computing Communication Security (CCS), pp. 196–211 (2018)
 59. Gursoy, M.E., Rajasekar, V., Liu, L.: Utility-optimized synthesis of differentially private location traces. In: Proceedings of the IEEE International Conference on Trust, Security and Privacy Intelligent Systems with Applications (TPS-ISA), pp. 30–39 (2020)
 60. Han, Q., Xiong, Z., Zhang, K.: Research on trajectory data releasing method via differential privacy based on spatial partition. In: Security, Communication and Network (2018)
 61. He, X., Cormode, G., Machanavajjhala, A., Procopiuc, C.M., Srivastava, D.: DPT: differentially private trajectory synthesis using hierarchical reference systems. VLDB J. **8**(11), 1154–1165 (2015)
 62. Hern, A.: Fitness tracking app Strava gives away location of secret US army bases. The Guardian (2018). <https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases>
 63. Hu, H., Salicic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X.: Membership inference attacks on machine learning: a survey. ACM Comput. Surv. **54**(11s), 1–37 (2022)
 64. Hua, J., Gao, Y., Zhong, S.: Differentially private publication of general time-serial trajectory data. In: Proceedings of the Joint Conference on IEEE Computing, Communication and Society (INFOCOM), pp. 549–557 (2015)
 65. Humphries, T., Rafuse, M., Tulloch, L., Oya, S., Goldberg, I., Hengartner, U., Kerschbaum, F.: Differentially private learning does not bound membership inference (2020)
 66. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E.S., Spicer, K., de Wolf, P.P.: Statistical Disclosure Control. Wiley, New York (2012)
 67. Hwang, J.R., Kang, H.Y., Li, K.J.: Spatio-temporal similarity analysis between trajectories on road networks. In: Proceedings of the Perspectives on Conceptual Modeling (ER), Berlin, Heidelberg, pp. 280–289 (2005)
 68. Jiang, H., Li, J., Zhao, P., Zeng, F., Xiao, Z., Iyengar, A.: Location privacy-preserving mechanisms in location-based services: a comprehensive survey. ACM Comput. Surv. **54**(1) (2021)
 69. Jin, F., Hua, W., Francia, M., Chao, P., Orlowska, M.: Zhou X A survey and experimental study on privacy-preserving trajectory data publishing (2021). <https://doi.org/10.36227/techrxiv.13655597.v1>
 70. Kasiviswanathan, S.P., Smith, A.: On the ‘semantics’ of differential privacy: a Bayesian formulation. J. Priv. Confid. **6**(1) (2014)
 71. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? SIAM J. Comput. **40**(3), 793–826 (2011)
 72. Keogh, E., Ratanamahatana, C.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. **7**, 358–386 (2005). <https://doi.org/10.1007/s10115-004-0154-9>
 73. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery, Data Mining (KDD) (2000)
 74. Kifer, D., Machanavajjhala, A.: No free lunch in data privacy. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (MOD), pp. 193–204 (2011)
 75. Kifer, D., Abowd, J.M., Ashmead, R., Cumings-Menon, R., Leclerc, P., Machanavajjhala, A., Sexton, W., Zhuravlev, P.: Bayesian and frequentist semantics for common variations of differential privacy: applications to the 2020 census (2022)
 76. Kong, X., Li, M., Ma, K., Tian, K., Wang, M., Ning, Z., Xia, F.: Big trajectory data: a survey of applications and services. IEEE Access **6**, 58295–58306 (2018)
 77. Leal, B.C., Vidal, I.C., Brito, F.T., Nobre, J.S., Machado, J.C.: δ -DOCA: achieving privacy in data streams. Proc. Int. Workshop Data Priv. Manag. (DPM) **11025**, 279–295 (2018)
 78. Li, M., Zhu, L., Zhang, Z., Xu, R.: Achieving differential privacy of trajectory data publishing in participatory sensing. Inf. Sci. **400**, 1–13 (2017)
 79. Li, N., Li, T., Venkatasubramanian, S.: t -Closeness: privacy beyond k -anonymity and l -diversity. In: Proceedings of the IEEE International Conference on Data Engineering (ICDE), pp. 106–115 (2007)
 80. Li, N., Lyu, M., Su, D., Yang, W.: Differential Privacy: From Theory to Practice. Morgan & Claypool (2016)
 81. Little, J.J., Gu, Z.: Video retrieval by spatial and temporal structure of trajectories. In: Proceedings of the SPIE, Storage, Retrieval Media Databases, vol. 4315, pp. 545 – 552. <https://doi.org/10.1117/12.410966> (2001)
 82. Liu, B., Xie, S., Wang, H., Hong, Y., Ban, X., Mohammady, M.: VTDP: privately sanitizing fine-grained vehicle trajectory data with boosted utility. IEEE Trans. Depend. Secur. Comput. **18**(6), 2643–2657 (2021). <https://doi.org/10.1109/TDSC.2019.2960336>
 83. Liu, C., Chakraborty, S., Mittal, P.: Dependence makes you vulnerable: differential privacy under dependent tuples. Proc. Symp. Netw. Distrib. Syst. Secur. (NDSS) **16**, 21–24 (2016)
 84. Luca, M., Barlacchi, G., Lepri, B., Pappalardo, L.: A survey on deep learning for human mobility (2020)
 85. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkatasubramanian, M.: l -diversity: Privacy beyond k -anonymity. ACM Trans. Knowl. Discov. Data **1**(1), 3-es (2007)
 86. Magdy, N., Sakr, M., Abdelkader, T., Elbahnasy, K.: Review on trajectory similarity measures. In: Proceedings of the IEEE International Conference on Intelligent Computing and Information Systems (ICICIS) (2015). <https://doi.org/10.1109/IntelCIS.2015.7397286>
 87. Maouche, M., Mokhtar, S.B., Bouchenak, S.: AP-attack: A novel user re-identification attack on mobility datasets. In: Proceedings

- of the EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous), Association for Computing Machinery, New York, pp. 48–57 (2017). <https://doi.org/10.1145/3144457.3144494>
88. Marteau, P.F.: Time warp edit distance with stiffness adjustment for time series matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(2), 306–318 (2009). <https://doi.org/10.1109/TPAMI.2008.76>
 89. Mello, R., Bogorny, V., Alvares, L., Santana, L., Ferrero, C., Frozza, A.A., Schreiner, G., Renso, C.: MASTER: a multiple aspect view on trajectories. *Trans. GIS* (2019). <https://doi.org/10.1111/tgis.12526>
 90. Miranda-Pascual, À., Guerra-Balboa, P., Parra-Arnau, J., Forné, J., Strufe, T.: SoK: differentially private publication of trajectory data. In: *Proceedings of the International Symposium on Privacy Enhancing Technology (PoPETs)*, vol. 2023, pp. 496–516 (2023). <https://doi.org/10.56553/popets-2023-0065>
 91. Monreale, A., Andrienko, G., Andrienko, N., Giannotti, F., Pedreschi, D., Rinzivillo, S., Wrobel, S.: Movement data anonymity through generalization. *Trans. Data Priv.* **3**, 91–121 (2010)
 92. Monreale, A., Trasarti, R., Pedreschi, D., Renso, C., Bogorny, V.: C-safety: a framework for the anonymization of semantic trajectories. *Trans. Data Priv.* **4**, 73–101 (2011)
 93. Nergiz, M., Atzori, M., Saygin, Y., Güç, B.: Towards trajectory anonymization: a generalization-based approach. *Trans. Data Priv.* **2**, 47–75 (2009)
 94. Nergiz, M.E., Atzori, M., Saygin, Y.: Perturbation-driven anonymization of trajectories. *Tech. Rep. 2007-TR-017*, ISTI-CNR, Pisa (2007)
 95. Ovide, S.: Just collect less data, period. *New York Times* (2020). <https://www.nytimes.com/2020/07/15/technology/just-collect-less-data-period.html>. Accessed on 2021-01-18
 96. Pensa, R., Monreale, A., Pinelli, F., Pedreschi, D.: Pattern-preserving k -anonymization of sequences and its application to mobility data mining. In: *Proceedings of the International Workshop Privacy Locat.-Based Application (PiLBA)*, vol. 397 (2008)
 97. Portela, T.T., Vicenzi, F., Bogorny, V.: Trajectory data privacy: research challenges and opportunities. In: *Proceedings of the X Brazilian Symposium on Geoinformatics (GEOINFO)* (2019)
 98. Poulis, G., Skiadopoulos, S., Loukides, G., Gkoulalas-Divanis, A.: Apriori-based algorithms for k^m -anonymizing trajectory data. *Trans. Data Priv.* **7**, 165–194 (2014)
 99. Primault, V., Boutet, A., Mokhtar, S.B., Brunie, L.: The long road to computational location privacy: a survey. *Comput. Res. Repos.* (arXiv CoRR) (2018)
 100. Pyrgelis, A., Troncoso, C., Cristofaro, E.D.: Knock knock, who's there? Membership inference on aggregate location data. *Comput. Res. Repos.* (arXiv CoRR) (2017). [arXiv:1708.06145](https://arxiv.org/abs/1708.06145)
 101. Rossi, L., Walker, J., Musolesi, M.: Spatio-temporal techniques for user identification by means of GPS mobility data. *EPJ Data. Sci.* **4**(1), 1–16 (2015). <https://doi.org/10.1140/epjds/s13688-015-0049-x>
 102. Samarati, P.: Protecting respondents identities in microdata release. *IEEE Trans. Knowl. Data Eng.* **13**(6), 1010–1027 (2001). <https://doi.org/10.1109/69.971193>
 103. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k -Anonymity and its enforcement through generalization and suppression. *Tech. rep, SRI Int* (1998)
 104. Shao, D., Jiang, K., Kister, T., Bressan, S., Tan, K.L.: Publishing trajectory with differential privacy: a priori vs. a posteriori sampling mechanisms. In: *Proceedings of the International Conference Database, Expert System Application (DEXA), Lecture Notes Computer Science (LNCS)*, vol. 8055, pp. 357–365 (2013)
 105. Song, C., Qu, Z., Blumm, N., Barabási, A.L.: Limits of predictability in human mobility. *Science* **327**, 1018–21 (2010). <https://doi.org/10.1126/science.1177170>
 106. Soria-Comas, J., Domingo-Ferrer, J.: Big data privacy: challenges to privacy principles and models. *Data Sci. Eng.* **1**(1), 21–28 (2016)
 107. Sousa, R.S.D., Boukerche, A., Loureiro, A.A.F.: Vehicle trajectory similarity: models, methods, and applications. *ACM Comput. Surv.* (2020). <https://doi.org/10.1145/3406096>
 108. Stadler, T., Oprisanu, B., Troncoso, C.: Synthetic data— anonymisation groundhog day. *arXiv preprint* (2021)
 109. Sui, K., Zhao, Y., Liu, D., Ma, M., Xu, L., Zimu, L., Pei, D.: Your trajectory privacy can be breached even if you walk in groups. In: *Proceedings of the IEEE/ACM International Symposium on Quality Service (IWQoS)*, pp. 1–6 (2016)
 110. Sweeney, L.: Achieving k -anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzz. Knowl. Based Syst.* **10**(5), 571–588 (2002). <https://doi.org/10.1142/S021848850200165X>
 111. Tao, Y., Both, A., Silveira, R.I., Buchin, K., Sijben, S., Purves, R.S., Laube, P., Peng, D., Toohey, K., Duckham, M.: A comparative analysis of trajectory similarity measures. *GIScience Remote Sens.* **58**(5), 643–669 (2021). <https://doi.org/10.1080/15481603.2021.1908927>
 112. Tarnoff, B.: Big data for the people: it's time to take it back from our tech overlords. *The Guardian* (2018). <https://www.theguardian.com/technology/2018/mar/14/tech-big-data-capitalism-give-wealth-back-to-people>. Accessed on 2021-01-18
 113. Tockar, A.: Riding with the stars: Passenger privacy in the NYC taxicab dataset (2014). <https://agkn.wordpress.com/author/atockar/>
 114. Toohey, K., Duckham, M.: Trajectory similarity measures. *ACM Spec. Interest Group Spatial Inform. (SIGSPATIAL Special)* **7**, 43–50 (2015). <https://doi.org/10.1145/2782759.2782767>
 115. Trotter, J.: Public NYC taxicab database lets you see how celebrities tip. *Gawker* (2014). <https://www.gawker.com/the-public-nyc-taxicab-database-that-accidentally-track-1646724546>
 116. Trujillo-Rasua, R., Domingo-Ferrer, J.: On the privacy offered by (k, δ) -anonymity. *Inf. Syst.* **38**, 491–494 (2013). <https://doi.org/10.1016/j.is.2012.12.003>
 117. Tu, Z., Zhao, K., Xu, F., Li, Y., Su, L., Jin, D.: Protecting trajectory from semantic attack considering k -anonymity, l -diversity, and t -closeness. *IEEE Trans. Netw. Serv. Manag.* **16**(1), 264–278 (2019). <https://doi.org/10.1109/TNSM.2018.2877790>
 118. Vlachos, M., Gunopulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pp. 673–684 (2002)
 119. Wang, H., Su, H., Zheng, K., Sadiq, S., Zhou, X.: An effectiveness study on trajectory similarity measures. In: *Proceedings of the Australasian Database Conference (ADC)*, pp. 13–22 (2013)
 120. Wang, H., Xu, Z., Jia, S., Xia, Y., Zhang, X.: Why current differential privacy schemes are inapplicable for correlated data publishing? *World Wide Web* **24**, 1–23 (2021)
 121. Wang, H., Hong, H., Xiong, L., Qin, Z., Hong, Y.: L-SRR: Local differential privacy for location-based services with staircase randomized response. In: *Proceedings of the ACM SIGSAC Conference on Computing Communication Security (CCS)*, pp. 2809–2823 (2022). <https://doi.org/10.1145/3548606.3560636>
 122. Wang, N., Kankanhalli, M.S.: Protecting sensitive place visits in privacy-preserving trajectory publishing. *Comput. Secur.* **97**(101), 949 (2020)
 123. Wang, S., Bao, Z., Culpepper, J., Zizhe, X., Liu, Q., Qin, X.: Torch: a search engine for trajectory data. In: *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 535–544 (2018). <https://doi.org/10.1145/3209978.3209989>

124. Wang, W., Yang, G., Bao, L., Ma, K., Zhou, H., Bai, Y.: Travel Trajectory Frequent Pattern Mining Based on Differential Privacy Protection. *Wirel Commun, Mob Comput* (2021)
125. Wernke, M., Skvortsov, P., Dürr, F., Rothermel, K.: A classification of location privacy attacks and approaches. *Pers. Ubiquitous Comput.* **18**, 163–175 (2012)
126. Willenborg, L., de Waal, T.: *Elements of Statistical Disclosure Control*. Springer, Berlin (2001)
127. Xiong, X., Liu, S., Li, D., Cai, Z., Niu, X.: A comprehensive survey on local differential privacy. *Secur. Commun. Netw.* (2020). <https://doi.org/10.1155/2020/8829523>
128. Xu, F., Tu, Z., Li, Y., Zhang, P., Fu, X., Jin, D.: Trajectory recovery from ash: user privacy is NOT preserved in aggregated mobility data. In: *Proceedings of the ACM International WWW Conference*, pp. 1241–1250 (2017). <https://doi.org/10.1145/3038912.3052620>
129. Yang, B., Sato, I., Nakagawa, H.: Bayesian differential privacy on correlated data. In: *Proceedings of the ACM SIGMOD International Conference on Management Data (MOD)*, pp. 747–762 (2015)
130. Yang, Y., Cai, J., Yang, H., Zhang, J., Zhao, X.: TAD: a trajectory clustering algorithm based on spatial-temporal density analysis. *Expert Syst. Appl.* **139**(112), 846 (2020)
131. Yao, C., Wang, L., Wang, S.X., Jajodia, S.: Indistinguishability: the other aspect of privacy. In: *Proceedings of the VLDB Workshop Secure Data Management (SDM)*, pp. 1–17 (2006)
132. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: analyzing the connection to overfitting. In: *Proceedings of the IEEE Computing Security Foundation Symposium (CSF)*, pp. 268–282. IEEE (2018)
133. Yuan, H., Li, G.: Distributed in-memory trajectory similarity search and join on road network. In: *Proceedings International Conference on Data Engineering (ICDE)*, pp. 1262–1273 (2019). <https://doi.org/10.1109/ICDE.2019.00115>
134. Yuan, S., Pi, D., Zhao, X., Xu, M.: Differential privacy trajectory data protection scheme based on R-tree. *Expert. Syst. Appl.* **182**(115), 215 (2021)
135. Zan, B., Sun, Z., Gruteser, M., Ban, X.: Linking anonymous location traces through driving characteristics. In: *Proceedings of the ACM Conference on Data and Application Security and Privacy (CODASPY)*, pp. 293–300 (2013)
136. Zang, H., Bolot, J.: Anonymization of location data does not work: a large-scale measurement study. In: *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 145–156 (2011)
137. Zhang, Z., Liu, Q., Huang, Z., Wang, H., Lu, C., Liu, C., Chen, E.: GraphMI: extracting private graph data from graph neural networks (2021)
138. Zhao, J., Mei, J., Matwin, S., Su, Y., Yang, Y.: Risk-aware individual trajectory data publishing with differential privacy. *IEEE Access* **9**, 7421–7438 (2020)
139. Zhao, X., Dong, Y., Pi, D.: Novel trajectory data publishing method under differential privacy. *Expert Syst. Appl.* **138**(112), 791 (2019)
140. Zhao, X., Pi, D., Chen, J.: Novel trajectory privacy-preserving method based on clustering using differential privacy. *Expert Syst. Appl.* **149**(113), 241 (2020)
141. Zhao, X., Pi, D., Chen, J.: Novel trajectory privacy-preserving method based on prefix tree using differential privacy. *Knowl. Based Syst.* **198**(105), 940 (2020)
142. Zhou, C., Frankowski, D., Ludford Finnerty, P., Shekhar, S., Terveen, L.: Discovering personal gazetteers: An interactive clustering approach. In: *Proceedings of the ACM International Conference on Information and Knowledge Management (GIS)*, pp. 266–273 (2004). <https://doi.org/10.1145/1032222.1032261>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.