JOURNAL OF SPATIAL INFORMATION SCIENCE
Number 19 (2019), pp. 57–103
doi:10.5311/JOSIS.2019.19.493

RESEARCH ARTICLE

Privacy, space, and time: a survey on privacy-preserving continuous data publishing

Manos Katsomallos[1], Katerina Tzompanaki[1], and Dimitris Kotzinos[1]

[1]ETIS UMR 8051, University of Paris-Seine, University of Cergy-Pontoise, ENSEA, CNRS , Cergy-Pontoise, France

Received: January 31, 2019; returned: May 20, 2019; revised: August 5, 2019; accepted: November 22, 2019.

Abstract: Sensors, portable devices, and location-based services, generate massive amounts of geo-tagged, and/or location- and user-related data on a daily basis. The manipulation of such data is useful in numerous application domains, e.g., healthcare, intelligent buildings, and traffic monitoring, to name a few. A high percentage of this data carries information of users' activities and other personal details, and thus its manipulation and sharing gives rise to concerns about the privacy of the individuals involved. To enable the secure—from the users' privacy perspective—data sharing, researchers have already proposed various seminal techniques for the protection of users' privacy. However, the continuous fashion in which data is generated nowadays and the high availability of external sources of information pose more threats and add extra challenges to the problem. In this survey, we visit the works done on data privacy for continuous data publishing and report on the proposed solutions, with a special focus on solutions concerning location or geo-referenced data.

Keywords: privacy-preserving algorithms, continuous data publishing, location privacy, microdata privacy, statistical data privacy

1 Introduction

Data privacy is becoming an increasingly important issue both at a technical and at a societal level, and introduces various challenges ranging from the way we share and publish data sets to the way we use online and mobile services. Personal information, also

described as *microdata*, has acquired increasing value and is, in many cases, used as the 'currency' [11] to pay for access to various services, i.e., users are asked to exchange their personal information with the service provided. This is particularly true for many *Location-Based Services* (LBSs), e.g., Google Maps [5], Waze [9], etc.; these services exchange their 'free' service with collecting and using user-generated data, such as timestamped geolocalized information. Besides navigation and location-based services, social media applications (e.g., Facebook [3], Twitter [8], Foursquare [4], etc.) take advantage of user-generated and user-related data, to make relevant recommendations and show personalized advertisements. In this case, the location is also part of the important required personal data to be shared. Last but not least, *data brokers* (e.g., Experian [2], TransUnion [7], Acxiom [1], etc.) collect data from public and private resources, e.g., censuses, bank card transaction records, voter registration lists, etc. Most of this data is georeferenced and contain, directly or indirectly, location information; protecting the location of the user has become one of the most important privacy goals so far.

These different sources and types of data, on the one hand give useful feedback to the involved users and/or services, and, on the other hand, when combined together, provide valuable information to various internal/external analytical services. While these activities happen within the boundaries of the law [108], it is important to be able to protect the privacy (by anonymizing, perturbing, encrypting, etc.) of the corresponding data before sharing, and to take into account the possibility of correlating, linking, and crossing diverse independent data sets. Especially the latter is becoming quite important in the era of Big Data, where the existence of diverse linked data sets is one of the promises; as an example, one can refer to the discussion on Entity Resolution problems using Linked Open Data in [42]. In some cases, personal data might be so representative that even if de-identified, when integrated with a small amount of external data, one can trace back to their original source. An example case is shown in [36], where it was discovered that four mobility traces are enough to identify 95% of the individuals in a data set. The case of location is actually one of great interest in this context, since space brings its own particular constraints. The ability to combine and correlate additional information impacts the ways we protect sensitive data and affects the privacy guarantees we can provide. Besides the explosion of online and mobile services, another important aspect is that a lot of these services actually rely on data provided by the users (*crowdsourced* data) to function, with prominent example efforts being Wikipedia [10] and OpenStreetMap [6]. Data from crowdsourced based applications, if not protected correctly, can be easily used to identify personal information, such as location or activity, and thus lead indirectly to cases of user surveillance [83].

Privacy-preserving processes usually introduce noise in the original or the aggregated data set in order to hide the sensitive information. In the case of *microdata*, a privacy-protected version, containing some synthetic data as well, is generated with the intrinsic goal to make the users indistinguishable. In the case of *statistical* data (i.e., the results of statistical queries over the original data sets), a privacy-protected version is generated by adding noise on the actual statistical values. In both cases, we end up affecting the quality of the published data set. The privacy and the utility of the 'noisy' output are two contrasting desiderata, which need to be measured and balanced. Furthermore, if we want to account for external additional information (e.g., linked or correlated data) and at the same time to ensure the same level of protection, we need to add additional noise, inevitably deteriorating the quality of the output. This problem becomes particularly pertinent in the Big Data era, as the quality or *Veracity* is one of the five dimensions (known as the five

'V's') that define Big Data, and where there is an abundance of external information that cannot be ignored. Since this needs to be taken into account *prior* to the publishing of the data set or the aggregated statistics there of, introducing external information into privacy-preserving techniques becomes part of the traditional processing flow while keeping an acceptable quality to privacy ratio.

As we can observe in the examples mentioned above, there are many cases where data is not protected at source (what is also described as *local* data privacy protection) for various reasons, e.g., the users do not want to pay extra, it is impossible due to technical complexity, because the quality of the expected service will be deteriorated, etc. Thus, the burden of the privacy-preserving process falls on the various aggregators of personal/private data, who should also provide the necessary technical solutions to ensure data privacy for every user (what is also described as *global* data privacy protection).

The discussion so far explains and justifies the current situation in the privacy-preserving scientific area. As a matter of fact, a wealth of algorithms have been proposed for privacy-preserving data publishing, either for microdata or statistical data. Moreover, privacy-preserving algorithms are designed specifically for data published at one point in time (used in what we call *snapshot* data publishing) or data released over or concerning a period of time (used in what we call *continuous data publishing*). In that respect, we need to be able to correctly choose the proper privacy algorithm(s), which would allow users to share protected copies of their data with some guarantees. The selection process is far from trivial, since it is essential to:

1. select an appropriate privacy-preserving technique, relevant to the data set intended for public release;
2. understand the different requirements imposed by the selected technique and tune the different parameters according to the circumstances of the use case based on, e.g., assumptions, level of distortion, etc. [72];
3. get the necessary balance between privacy and data utility, which is a significant task for any privacy algorithm as well as any privacy expert.

Selecting the wrong privacy algorithm or configuring it poorly may put at risk the privacy of the involved individuals and/or end up deteriorating the quality and therefore the utility of the data set.

In data privacy research, privacy in continuous data publishing scenarios is the area that is concerned by studying the privacy problems created when sensitive data is published continuously, either infinitely (e.g., streaming data) or by multiple continuous publications over a known period of time (e.g., finite time series data). This specific subfield of data privacy becomes increasingly important since it:

(i) includes the most prominent cases, e.g., location (trajectory) privacy problems, and
(ii) provides the most challenging and yet not well charted part of the privacy algorithms since it is rather new and increasingly complex.

In this context, this survey seeks to offer a guide that would allow its users to choose the proper algorithm(s) for their specific use case accordingly. Additionally, data in continuous data publishing use cases require a timely processing because their value usually decreases over time depending on the use case as demonstrated in Figure 1. For this reason, we provide an insight into time-related properties of the algorithms, e.g., if they work on infinite,

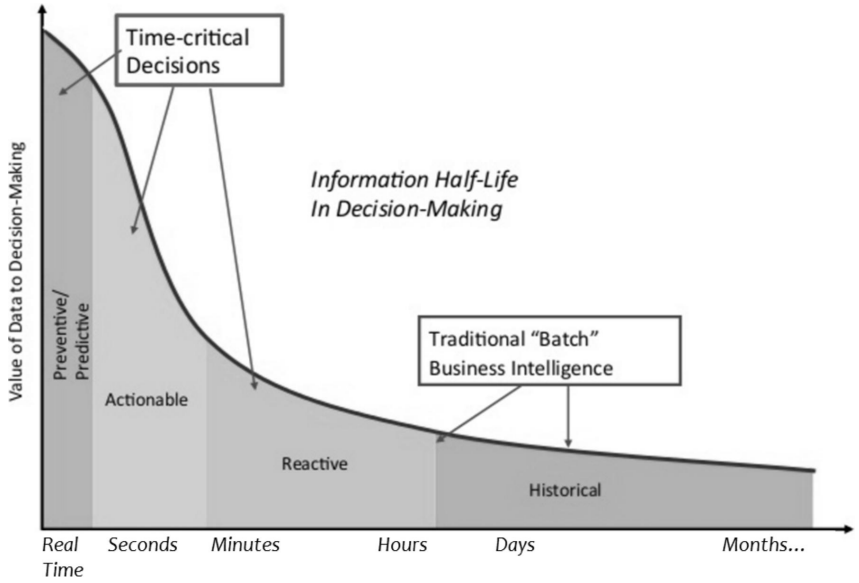


Figure 1: Value of data for decision-making over time from less than seconds to more than months [56].

real-time data, or if they take into consideration existing data dependencies. The importance of continuous data publishing is stressed by the fact that, commonly, many types of data have such properties, with geospatial data being a prominent case. A few examples include—but are not limited to—data being produced while tracking the movement of individuals for various purposes (where data might also need to be privacy-protected in real-time and in a continuous fashion); crowdsourced data that are used to report measurements, such as noise or pollution (where again we have a continuous timestamped and usually georeferenced stream of data); and even isolated data items that might include location information, such as photographs or social media posts. Typically, in such cases, we have a collection of data referring to the same individual or set of individuals over a period of time, which can also be infinite. Additionally, in many cases, the privacy-preserving processes should take into account implicit correlations and restrictions that exist, e.g., space-imposed collocation or movement restrictions. Since this data is related to most of the important applications and services that enjoy high utilization rates, privacy-preserving continuous data publishing becomes one of the emblematic problems of our time.

Since the domain of data privacy is vast, several surveys have already been published with different scopes. A group of surveys focuses on specific different families of privacy-preserving algorithms and techniques. For instance, Simi et al. [101] provide an extensive study of works on k-anonymity and Dwork [37] focuses on differential privacy. Another group of surveys focuses on techniques that allow the execution of data mining or machine learning tasks with some privacy guarantees, e.g., Wang et al. [111] and Ji et al. [63]. In a more general scope, Wang et al. [48] analyze the challenges of privacy-preserving data publishing and offer a summary and evaluation of relevant techniques. Additional surveys look into issues around Big Data and user privacy. Indicatively, Jain et al. [62], and

Soria-Comas and Domingo-Ferrer [104] examine how Big Data conflict with pre-existing concepts of privacy-preserving data management, and how efficiently k-anonymity and ε-differential privacy deal with the characteristics of Big Data. Others narrow down their research to location privacy issues. To name a few, Chow and Mokbel [34] investigate privacy protection in continuous LBSs and trajectory data publishing, Chatzikokolakis et al. [28] review privacy issues around the usage of LBSs and relevant protection mechanisms and metrics, Primault et al. [92] summarize location privacy threats and privacy-preserving mechanisms, and Fiore et al. [47] focus only on privacy-preserving publishing of trajectory microdata. Finally, there are some surveys on application-specific privacy challenges. For example, Zhou et al. [123] have a focus on social networks, and Christin et al. [35] give an outline of how privacy aspects are addressed in crowdsensing applications. Nevertheless, to the best of our knowledge, there is no up-to-date survey that deals with privacy under continuous data publishing covering diverse use cases. Such a survey becomes very useful nowadays, due to the abundance of continuously user-generated data sets that could be analyzed and/or published in a privacy-preserving way, and the quick progress made in this research field.

This survey is organized as follows. We begin by providing a general description of the field of data privacy, and the most prominent anonymization and obfuscation/noise-inducing algorithms in the literature (Section 2). The main content of the survey (Section 3) spans works related to the continuous publishing of data points or to the re-publishing of (or parts of) a data set along time, with regard to the privacy of the individuals involved. More particularly, we divide the works in two categories, based on the type of data to be published: microdata—the data in their original format—or statistical data—statistical query results over microdata. In all cases, we use the same set of properties to characterize the algorithms, which facilitates their comparison. Finally (Section 4), we put these works into perspective and discuss various future research lines in this area.

2 Background

In this section, we introduce some relevant terminology and background knowledge around the problem of continuous publishing of sensitive data sets. First, we categorize data as we view them in the context of continuous data publishing. Second, we define data privacy, we list the kinds of attacks that have been identified in the literature, as well as the desired privacy levels that can be achieved, and the basic privacy operations that are applied to achieve data privacy. Third, we provide a brief overview of the seminal works on privacy-preserving data publishing, used also in continuous data publishing, fundamental in the domain and important for the understanding of the rest of the survey.

To accompany and facilitate the descriptions in this section, we provide the following running example.

Example 2.1. Users interact with an LBS by making queries in order to retrieve some useful location-based information or just reporting user-state at various locations. This user–LBS interaction generates user-related data, organized in a schema with the following attributes: *Name* (the unique identifier of the table), *Age*, *Location*, and *Status* (Table 1a). The 'Status' attribute includes information that characterizes the user's state or the query itself, and its value varies according to the service functionality. Subsequently, the generated data

is aggregated (by issuing count queries over them) in order to derive useful information about the popularity of the venues during the day (Table 1b).

Name	Age	Location	Status
Donald	27	Le Marais	at work
Daisy	25	Belleville	driving
Huey	12	Montmartre	running
Dewey	11	Montmartre	at home
Louie	10	Latin Quarter	walking
Quackmore	62	Opera	dining

(a) Microdata

Location	Count
Belleville	1
Latin Quarter	1
Le Marais	1
Montmartre	2
Opera	1

(b) Statistical data

Table 1: Example of raw user-generated (a) microdata, and related (b) statistical data for a specific timestamp.

2.1 Data

2.1.1 Categories

As this survey is about privacy, the data that we are interested in, contain information about individuals and their actions. We firstly classify the data based on their content:

- *Microdata*—the data items in their raw, usually tabular, form pertaining to individuals or objects.
- *Statistical data*—the outcome of statistical processes on microdata.

An example of microdata is displayed in Table 1a, while an example of statistical data in Table 1b. Data, in either of these two forms, may have a special property called *continuity*, i.e., their values change and can be observed through time. Depending on the span of observation, we distinguish the following categories:

- *Finite data*—data is observed during a predefined time interval.
- *Infinite data*—data is observed in an uninterrupted fashion.

Example 2.2. Extending Example 2.1, Table 2 shows an example of continuous data observation, by introducing one data table for each consecutive timestamp. The two data tables, over the time-span $[t_1, t_2]$ are an example of finite data. Infinite data is the whole series of data obtained over the period $[t_1, \infty)$ (infinity is denoted by '...').

We further define two sub-categories applicable to both finite and infinite data: *sequential* and *incremental* data; these two subcategories are not exhaustive, i.e., not all data sets belong to the one or the other category. In sequential data, the value of the observed variable changes, depending on its previous value. For example, trajectories are finite sequences of location stamps, as naturally the position at each timestamp is connected to the position at the previous timestamp. In incremental data, an original data set is augmented in each subsequent timestamp with supplementary information. For example, trajectories can be considered as incremental data, when at each timestamp we consider all the previously visited locations by an individual, incremented by his current position.

Name	Age	Location	Status
Donald	27	Le Marais	at work
Daisy	25	Belleville	driving
Huey	12	Montmartre	running
Dewey	11	Montmartre	at home
Louie	10	Latin Quarter	walking
Quackmore	62	Opera	dining

t_1

Name	Age	Location	Status
Donald	27	Montmartre	driving
Daisy	25	Montmartre	at the mall
Huey	12	Latin Quarter	sightseeing
Dewey	11	Opera	walking
Louie	10	Latin Quarter	at home
Quackmore	62	Montmartre	biking

t_2 ...

(a) Microdata

Location	Count		
	t_1	t_2	...
Belleville	1	0	...
Latin Quarter	1	2	...
Le Marais	1	0	...
Montmartre	2	3	...
Opera	1	1	...

(b) Statistical data

Table 2: Continuous data observation of (a) microdata, and corresponding (b) statistics at multiple timestamps.

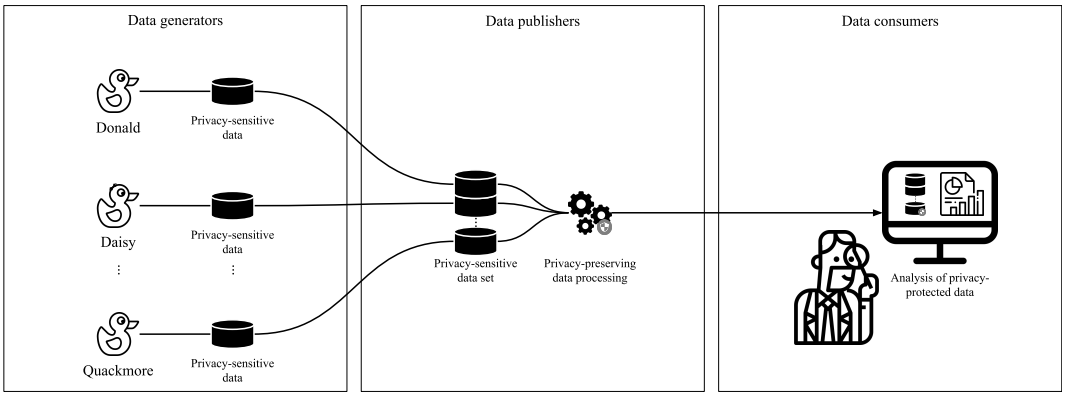
2.1.2 Processing and publishing

We categorize data processing and publishing based on the implemented scheme, as:

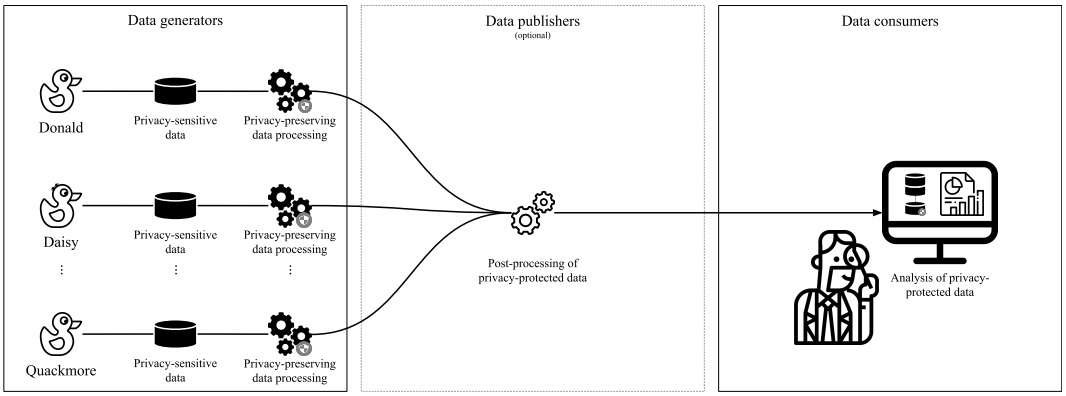
- *Global*—data is collected, processed, and privacy-protected, and then published by a central (trusted) entity, e.g., [20, 65, 87].
- *Local*—data is stored, processed, and privacy-protected on the side of data generators before sending it to any intermediate or final entity, e.g., [14, 44, 68].

In the case of location data privacy, the existing literature is divided in *service-* and *data*-centric methods [34]. The service-centric methods correspond to scenarios where individuals share their privacy-protected location with a service to get some relevant information (local publishing scheme). The data-centric methods relate to the publishing of user-generated data to data consumers (global publishing scheme).

There is a long-standing debate whether the local or the global architectural scheme is more efficient with respect to not only privacy, but also organizational, economic, and security factors [74]. On the one hand, in the global privacy scheme (Figure 2a), the dependence on third-party entities poses the risk of arbitrary privacy leakage from a compromised data publisher. Nonetheless, the expertise of these entities is usually superior to that of the majority of (non-technical) data generators' in terms of understanding privacy permissions/policies and setting-up relevant preferences. Moreover, in the global architecture, less distortion is necessary before publicly releasing the aggregated data set, naturally because the data sets are larger and users can be 'hidden' more easily. On the other hand, the local privacy scheme (Figure 2b) facilitates fine-grained data management, offering to every individual better control over their data [53]. Nonetheless, data distortion at an early



(a) Global scheme



(b) Local scheme

Figure 2: The usual flow of user-generated data, optionally harvested by data publishers, privacy-protected, and released to data consumers, according to the (a) global and (b) local privacy schemes.

stage might prove detrimental to the overall utility of the aggregated data set. The so far consensus is that there is no overall optimal solution among the two designs. Most service-providing companies prefer the global scheme, mainly for reasons of better management and control over the data, while several privacy advocates support the local privacy scheme that offers users full control over what and how data is published. Although there have been attempts to bridge the gap between them, e.g., [19], the global scheme is considerably better explored and implemented [97]. For this reason, most of the works in this survey span this context.

We distinguish between two publishing modes for private data: *snapshot* and *continuous*. In snapshot publishing (also appearing as *one-shot* or *one-off* publishing), the system processes and releases a data set at a specific point in time and thereafter is not concerned anymore with the specific data set. For example, in Figure 3a (ignore the privacy-preserving step for the moment) individuals send their data to an LBS provider, consider-

ing a specific time point. In continuous data publishing the system computes, and publishes augmented or updated versions of one data set in different time points, and without a predefined duration. In the context of privacy-preserving data publishing, privacy preservation is tightly coupled with the data processing and publishing stages.

As already discussed in Section 1, in this survey we are studying the continuous data publishing mode, and thus we do not include works considering the snapshot paradigm. We make this deliberate choice as privacy-preserving continuous data publishing is a more complex problem, receiving more and more attention from the scientific community in the recent years, as shown by the increasing number of publications in this area. Moreover, the use cases of continuous data publishing abound, with the proliferation of the Internet, sensors, and connected devices, which produce and send to servers huge amounts of continuous personal data in astounding speed.

We identify two main data processing and publishing modes:

- *Batch*—data is considered in groups in specific time intervals.
- *Streaming*—data is considered per timestamp, infinitely.

Batch data processing and publishing (Figure 3b) is performed (usually offline) over both finite and infinite data, while streaming processing and publishing (Figure 3c) is by definition connected to infinite data (usually in real-time).

2.2 Privacy

When personal data is publicly released, either as microdata or statistical data, individuals' privacy can be compromised, i.e, an adversary becomes certain about an individual's personal information with a probability higher than a desired threshold. In the literature, this compromise is know as *information disclosure* and is usually categorized as [48, 81, 90]:

- *Presence disclosure*—the participation (or absence) of an individual in a data set is revealed.
- *Identity disclosure*—an individual is linked to a particular record.
- *Attribute disclosure*—new information (attribute value) about an individual is revealed.

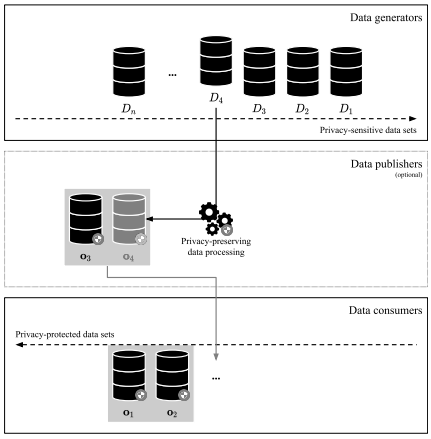
In the literature, identity disclosure is also referred to as *record linkage*, and presence disclosure as *table linkage*. Notice that identity disclosure can result in attribute disclosure, and vice versa.

To better illustrate these definitions, we provide some examples based on Table 1. Presence disclosure appears when by looking at the (privacy-protected) counts of Table 1b, we can guess if Quackmore has participated in Table 1a. Identity disclosure appears when we can guess that the sixth record of (a privacy-protected version of) the microdata of Table 1a belongs to Quackmore. Attribute disclosure appears when it is revealed from (a privacy-protected version of) the microdata of Table 1a that Quackmore is 62 years old.

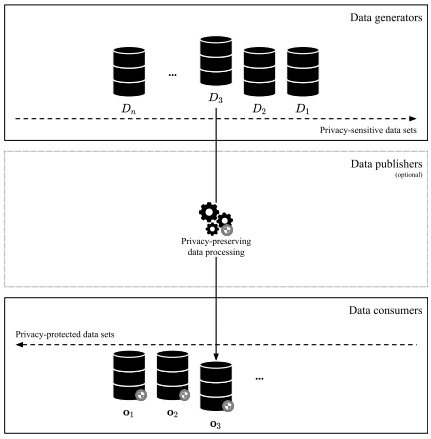
2.2.1 Levels

The information disclosure that a data release may entail is often linked to the protection level that a privacy-preserving algorithm is trying to achieve. More specifically, in continuous data publishing the privacy protection level is considered with respect to not only the

(a) Snapshot mode



(b) Batch mode



(c) Streaming mode

Figure 3: The different data processing and publishing modes of continuously generated data sets. (a) Snapshot publishing, (b) continuous publishing—batch mode, and (c) continuous publishing—streaming mode. $\boldsymbol{o}_x$ denotes the privacy-protected version of the data set D_x or statistics thereof, while '…' denote the continuous data generation and/or publishing, where applicable. Depending on the data observation span, n can either be finite or tend to infinity.

users but also to the *events* occurring in the data. An event is considered as a pair of an identifying attribute of an individual and the sensitive data (including contextual information), and can be seen as a correspondence to a record in a database, where each individual may participate once. Data publishers typically release events in the form of data points' sequences usually indexed in time order (time series), and geotagged, e.g., ('Dewey', 'at home at Montmartre at t_1'), …, ('Quackmore', 'dining at Opera at t_1'). The term 'users' is used to refer to the *individuals*, also known as *participants*, who are the source of the processed and published data. Therefore, they should not be confused with the consumers of

the released data sets. Users are subject to privacy attacks, and thus are the main point of interest of privacy protection mechanisms. In more detail, the privacy protection levels are:

- *Event* [39,40]—*any single event* of any individual is protected.
- *User* [39,40]—*all the events* of any individual, spanning the observed event sequence, are protected.
- *w-event* [70]—*any sequence of w events*, within the released event series, of any individual is protected.

Figure 4 demonstrates the application of the possible protection levels on the statistical data of Example 2.2. For instance, in event-level (Figure 4a) it is hard to determine whether Quackmore was dining at Opera at t_1. Moreover, in user-level (Figure 4b) it is hard to determine whether Quackmore was ever included in the released event series at all. Finally, in 2-event-level (Figure 4c) it is hard to determine whether Quackmore was ever included in the released event series between the timestamps t_1 and t_2, t_2 and t_3, etc. (i.e., for a window $w = 2$).

Location	Count		
	t_1	t_2	…
Belleville	1	0	…
Latin Quarter	1	2	…
Le Marais	1	0	…
Montmartre	2	3	…
Opera	1	1	…

(a) Event-level

Location	Count		
	t_1	t_2	…
Belleville	1	0	…
Latin Quarter	1	2	…
Le Marais	1	0	…
Montmartre	2	3	…
Opera	1	1	…

(b) User-level

Location	Count		
	t_1	t_2	…
Belleville	1	0	…
Latin Quarter	1	2	…
Le Marais	1	0	…
Montmartre	2	3	…
Opera	1	1	…

(c) 2-event-level

Figure 4: Protecting the data of Table 2b on (a) event-, (b) user-, and (c) 2-event-level. A suitable distortion method can be applied accordingly.

Contrary to event-level that provides privacy guarantees for a single event, user- and w-event-level offer stronger privacy protection by protecting a series of events. In use-cases that involve infinite data, event- and w-event-level attain an adequate balance between data utility and user privacy, whereas user-level is more appropriate when the span of data observation is predefined. w-event- is narrower than user-level protection due to its sliding window processing methodology. In the extreme cases where w is set to either 1 or to the size of the entire length of the event series, w-event- matches event- or user-level protection, respectively. Although the described levels have been coined in the context of *differential privacy* [38], a seminal privacy method that we will discuss in more detail in Section 2.2.4, it is possible to apply their definitions to other privacy protection techniques as well.

2.2.2 Attacks

Information disclosure is typically achieved by combining supplementary (background) knowledge with the released data or by setting unrealistic assumptions while designing the privacy-preserving algorithms. In its general form, this is known as *adversarial* or *linkage* attack. Even though many works directly refer to the general category of linkage attacks, we distinguish also the following sub-categories, addressed in the literature:

- *Sensitive attribute domain* knowledge. Here we can identify *homogeneity and skewness* attacks [81,85], when statistics of the sensitive attribute values are available, and *similarity attack*, when semantics of the sensitive attribute values are available.
- *Complementary release* attacks [107] with regard to previous releases of different versions of the same and/or related data sets. In this category, we also identify the *unsorted matching* attack [107], which is achieved when two privacy-protected versions of an original data set are published in the same tuple ordering. Other instances include: (i) the *join* attack [112], when tuples can be identified by joining (on the (quasi-)identifiers) several releases, (ii) the *tuple correspondence* attack [49], when in case of incremental data certain tuples correspond to certain tuples in other releases, in an injective way, (iii) the *tuple equivalence* attack [60], when tuples among different releases are found to be equivalent with respect to the sensitive attribute, and (iv) the *unknown releases* attack [99], when the privacy preservation is performed without knowing the previously privacy-protected data sets.
- *Data dependence*
  - within one data set. Data tuples and data values within a data set may be correlated, or linked in such a way that information about one person can be inferred even if the person is absent from the database. Consequently, in this category we put assumptions made on the data generation model based on randomness, like the random world model, the independent and identically distributed data (i.i.d.) model, or the independent-tuples model, which may be unrealistic for many real-world scenarios. This attack is also known as the *deFinetti's attack* [71].
  - among one data set and previous data releases, and/or other external sources [32,72,82,121]. The strength of the dependence between a pair of variables can be quantified with the utilization of *correlations* [105]. Correlation implies dependence but not vice versa, however, the two terms are often used as synonyms. The correlation among nearby observations, i.e., the elements in a series of data points, are referenced as *autocorrelation* or *serial correlation* [91]. Depending on the evaluation technique, e.g., *Pearson's correlation coefficient* [105], a correlation can be characterized as *negative*, *zero*, or *positive*. A negative value shows that the behavior of one variable is the *opposite* of that of the other, e.g., when the one increases the other decreases. Zero means that the variables are not linked and are *independent* of each other. A positive correlation indicates that the variables behave in a *similar* manner, e.g., when the one decreases the other decreases as well.

    The most prominent types of correlations might be:
    - *Temporal* [115]—appearing in observations (i.e., values) of the same object over time.
    - *Spatial* [15,77]—denoted by the degree of similarity of nearby data points in space, and indicating if and how phenomena relate to the (broader) area where they take place.
    - *Spatiotemporal*—a combination of the previous categories, appearing when processing time series or sequences of human activities with geolocation characteristics, e.g., [52].

    Contrary to one-dimensional correlations, spatial correlation is multi-dimensional and multi-directional, and can be measured by indicators (e.g., *Moran's I* [88]) that reflect the *spatial association* of the concerned data. Spa-

tial autocorrelation has its foundations in the *First Law of Geography* stating that "everything is related to everything else, but near things are more related than distant things" [109]. A positive spatial autocorrelation indicates that similar data is *clustered*, a negative that data is dispersed and is close to dissimilar ones, and when close to zero, that data is *randomly arranged* in space.

A common practice for extracting data dependencies from continuous data, is by expressing the data as a *stochastic* or *random process*. A random process is a collection of *random variables* or *bivariate data*, indexed by some set, e.g., a series of timestamps, a Cartesian plane $\mathbb{R}^2$, an n-dimensional Euclidean space, etc. [102]. The values a random variable can take are outcomes of an unpredictable process, while bivariate data is pairs of data values with a possible association between them. Expressing data as stochastic processes allows their modeling depending on their properties, and thereafter the discovery of relevant data dependencies. Some common stochastic processes modeling techniques include:

- *Conditional probabilities* [58]—probabilities of events in the presence of other events.
- *Conditional Random Fields* (CRFs) [75]—undirected graphs encoding conditional probability distributions.
- *Markov processes* [96]—stochastic processes for which the conditional probability of their future states depends only on the present state and it is independent of its previous states (*Markov assumption*).
  - *Markov chains* [50]—sequences of possible events whose probability depends on the state attained in the previous event.
  - *Hidden Markov Models* (HMMs) [16]—statistical Markov models of Markov processes with unobserved states.

The first sub-category of attacks has been mainly addressed in works on snapshot microdata publishing, and is still present in continuous publishing; however, algorithms for continuous publishing typically accept the proposed solutions for the snapshot publishing scheme (see discussion over k-anonymity and l-diversity in Section 2.2.4). This kind of attacks is tightly coupled with publishing the (privacy-protected) sensitive attribute value. An example is the lack of diversity in the sensitive attribute domain, e.g., if all users in the data set of Table 1a shared the same *running* Status (the sensitive attribute). The second and third subcategory are attacks emerging (mostly) in continuous publishing scenarios. Consider again the data set in Table 1a. The complementary release attack means that an adversary can learn more things about the individuals (e.g., that there are high chances that Donald was at work) if he/she combines the information of two privacy-protected versions of this data set. By the data dependence attack, the status of Donald could be more certainly inferred, by taking into account the status of Dewey at the same moment and the dependencies between Donald's and Dewey's status, e.g., when Dewey is at home, then most probably Donald is at work. In order to better protect the privacy of Donald in case of attacks, the data should be privacy-protected in a more adequate way (than without the attacks).

2.2.3 Operations

Protecting private information, which is known by many names (obfuscation, cloaking, anonymization, etc.), is achieved by using a specific basic privacy protection operation. Depending on the intervention that we choose to perform on the original data, we identify the following operations:

- *Aggregation*—group together multiple rows of a data set to form a single value.
- *Generalization*—replace an attribute value with a parent value in the attribute taxonomy. Notice that a step of generalization, may be followed by a step of *specialization*, to improve the quality of the resulting data set.
- *Suppression*—delete completely certain sensitive values or entire records.
- *Perturbation*—disturb the initial attribute value in a deterministic or probabilistic way. The probabilistic data distortion is referred to as *randomization*.

For example, consider the table schema *User(Name, Age, Location, Status)*. If we want to protect the *Age* of the user by aggregation, we may replace it by the average age in her Location; by generalization, we may replace the Age by age intervals; by suppression we may delete the entire table column corresponding to *Age*; by perturbation, we may augment each age by a predefined percentage of the age; by randomization we may randomly replace each age by a value taken from the probability density function of the attribute.

It is worth mentioning that there is a series of algorithms (e.g., [18,23,67]) based on the *cryptography* operation. However, the majority of these methods, among other assumptions that they make, have minimum or even no trust to the entities that handle the personal information. Furthermore, the amount and the way of data processing of these techniques usually burden the overall procedure, deteriorate the utility of the resulting data sets, and restrict their applicability. Our focus is limited to techniques that achieve a satisfying balance between both participants' privacy and data utility. For these reasons, there will be no further discussion around this family of techniques in this article.

2.2.4 Seminal works

For completeness, in this section we present the seminal works for privacy-preserving data publishing, which, even though originally designed for the snapshot publishing scenario, have paved the way, since many of the works in privacy-preserving continuous publishing are based on or extend them.

Microdata Sweeney coined *k-anonymity* [107], one of the first established works on data privacy. A released data set features k-anonymity protection when the sequence of values for a set of identifying attributes, called the *quasi-identifiers*, is the same for at least k records in the data set. Computing the quasi-identifiers in a set of attributes is still a hard problem on its own [89]. k-anonymity constitutes an individual indistinguishable from at least $k-1$ other individuals in the same data set. In a follow-up work [106], the author describes a way to achieve k-anonymity for a data set by the suppression or generalization of certain values of the quasi-identifiers. Machanavajjhala et al. [85] pointed out that k-anonymity is vulnerable to homogeneity and background knowledge attacks. Thereby, they proposed *l-diversity*, which demands that the values of the sensitive attributes are 'well-represented' by l sensitive values in each group. Principally, a data set can be l-diverse by featuring

at least l distinct values for the sensitive field in each group (*distinct* l-diversity). Other instantiations demand that the entropy of the whole data set is greater than or equal to $\log(l)$ (*entropy* l-diversity) or that the number of appearances of the most common sensitive value is less than the sum of the counts of the rest of the values multiplied by a user defined constant c (*recursive (c, l)*-diversity). Later on, Li et al. [81] indicated that l-diversity can be void by skewness and similarity attacks due to sensitive attributes with a small value range. In such cases, θ-*closeness* guarantees that the distribution of a sensitive attribute in a group and the distribution of the same attribute in the whole data set is 'similar'. This similarity is bounded by a threshold θ. A data set features θ-closeness when all of its groups feature θ-closeness.

The main drawback of k-anonymity (and its derivatives) is that it is not tolerant to external attacks of re-identification on the released data set. The problems identified in [107] appear when attempting to apply k-anonymity on continuous data publishing (as we will also see next in Section 3.1). These attacks include multiple k-anonymous data set releases with the same record order, subsequent releases of a data set without taking into account previous k-anonymous releases, and tuple updates. Proposed solutions include rearranging the attributes, setting the whole attribute set of previously released data sets as quasi-identifiers or releasing data based on previous k-anonymous releases.

Statistical data While methods based on k-anonymity have been mainly employed for releasing microdata, *differential privacy* [38] has been proposed for releasing privacy-protected, high utility aggregates over microdata. Differential privacy ensures that any adversary observing a privacy-protected output, no matter his/her computational power or auxiliary information, cannot conclude with absolute certainty if an individual is included in the input data set. Moreover, it quantifies and bounds the impact that the addition/removal of the data of an individual to/from an input data set has on the derived privacy-protected aggregates.

In its formal definition, a *privacy mechanism* $\mathcal{M}$, which outputs a query answer with some injected randomness, satisfies ε-differential privacy for a user-defined privacy budget ε [87] if for all pairs of *neighboring* (i.e., differing by the data of an individual) data sets D and D', it holds that:

$$\Pr[\mathcal{M}(D) \in O] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in O],$$

where $\Pr[\cdot]$ denotes the probability of an event, and O is the world of possible outputs of a mechanism $\mathcal{M}$. As the definition implies, for low values of ε, $\mathcal{M}$ achieves stronger privacy protection since the probabilities of D and D' being true worlds are similar, but the utility of the mechanism's output is reduced since more randomness is introduced. The privacy budget ε has a non-zero and positive value, and is usually set to 0.01, 0.1, or, in some cases, $\ln 2$ or $\ln 3$ [76].

A typical mechanism example is the *Laplace mechanism* [41], which draws randomly a value from the probability distribution of Laplace(μ, b), where μ stands for the location parameter and $b > 0$ the scale parameter. Here, μ is equal to the original output value of a query function, and b is the sensitivity of the query function divided by ε. The Laplace mechanism works for any function with range the set of real numbers. A specialization of this mechanism for location data is the *Planar Laplace mechanism* [14], which is based on a multivariate Laplace distribution. For query functions that do not return a real number, e.g., 'What is the most visited country this year?' or in cases where perturbing the value of the output will completely destroy its utility, e.g., 'What is the optimal price for this

auction?', most works in the literature use the *Exponential mechanism* [41]. This mechanism utilizes a utility function u that maps (input data set D, output value r) pairs to utility scores, and selects an output value r from the input pairs, with probability proportional to $\exp(\frac{\varepsilon u(D,r)}{2\Delta u})$, where Δu is the sensitivity of the utility function. Another technique for differential privacy mechanisms is the *randomized response* [114]. It is a privacy-preserving survey method that introduces probabilistic noise to the statistics of a research by randomly instructing respondents to answer truthfully or 'Yes' to a sensitive, binary question. The technique achieves this randomization by including a random event, e.g., the flip of a fair coin. The respondents reveal to the interviewers only their answer to the question, and keep as a secret the result of the random event (i.e., if the coin was tails or heads). Thereafter, the interviewers can calculate the probability distribution of the random event, e.g., $\frac{1}{2}$ heads and $\frac{1}{2}$ tails, and thus they can roughly eliminate the false responses and estimate the final result of the research.

Differential privacy mechanisms satisfy two composability properties: *sequential* and *parallel* [87, 104]. Due to the sequential composability property, the total privacy level of two independent mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ over the same data set that satisfy ε_1 and ε_2, respectively, equals to $\varepsilon_1 + \varepsilon_2$. The parallel composability property dictates that, when the mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ are applied over disjoint subsets of the same data set, then the overall privacy level is $\max_{i\in\{1,2\}} \varepsilon_i$. Every time a data publisher interacts with (any part of) the original data set, it is mandatory to consume some of the available privacy budget according to the composability properties. This is a necessity, so as to ensure that there will be no further arbitrary privacy loss, when the released data sets will be acquired by adversaries (or simple users). However, *post-processing* the output of a differential privacy mechanism can be done without using any additional privacy budget. Naturally, using the same (or different) privacy mechanism(s) multiple times to interact with raw data in combination with already perturbed data, implies that the privacy guarantee of the final output will be calculated according to sequential composition.

Differential privacy methods are best for low sensitivity queries such as counts, because the presence/absence of a single record can only change the result slightly. However, sum and max queries can be problematic, since a single but very different value could change the output noticeably, making it necessary to add a lot of noise to the query's answer. Furthermore, asking a series of queries may allow the disambiguation between possible data sets, making it necessary to add even more noise to the outputs. For this reason, after a series of queries exhausts the available privacy budget, the data set has to be discarded. Keeping the original guarantee across multiple queries that require different/new answers, one must inject noise proportional to the number of the executed queries, and thus destroying the utility of the output.

A special category of differential privacy-preserving algorithms is that of *pan-private* algorithms [40]. Pan-private algorithms hold their privacy guarantees even when snapshots of their internal state (memory) are accessed during their execution by an external entity, e.g., subpena, security breach, etc. There are two intrusion types that a data publisher has to deal with when designing a pan-private mechanism: *single unannounced*, and *continual announced* intrusion. In the first, the data publisher assumes that the mechanism's state is observed by the external entity one unique time, without the data publisher ever being notified about it. In the latter, the external entity gains access to the mechanism's state multiple times, and the publisher is notified after each time. The simplest approach to deal with both cases is to make sure that the data in the memory of the mechanism have

constantly the same distribution, i.e., they are differentially private. Notice that this must hold throughout the mechanism's lifetime, even before/after it processes any sensitive data point(s).

The notion of differential privacy has highly influenced the research community, resulting in many follow-up publications ([72, 86, 120] to mention a few). We distinguish here *Pufferfish* [73] and *geo-indistinguishability* [14, 29]. *Pufferfish* is a framework that allows experts in an application domain, without necessarily having any particular expertise in privacy, to develop privacy definitions for their data sharing needs. To define a privacy mechanism using *Pufferfish*, one has to define a set of potential secrets $\mathcal{X}$, a set of distinct pairs $\mathcal{X}_{pairs}$, and auxiliary information about data evolution scenarios $\mathcal{B}$. $\mathcal{X}$ serves as an explicit specification of what we would like to protect, e.g., 'the record of an individual x is (not) in the data'. $\mathcal{X}_{pairs}$ is a subset of $\mathcal{X} \times \mathcal{X}$ that instructs how to protect the potential secrets $\mathcal{X}$, e.g., ('x is in the table', 'x is not in the table'). Finally, $\mathcal{B}$ is a set of conservative assumptions about how the data evolved (or were generated) that reflects the adversary's belief about the data, e.g., probability distributions, variable correlations, etc. When there is independence between all the records in the original data set, then ε-differential privacy and the privacy definition of ε-*Pufferfish*$(\mathcal{X}, \mathcal{X}_{pairs}, \mathcal{B})$ are equivalent. *Geo-indistinguishability* is an adaptation of differential privacy for location data in snapshot publishing. It is based on l-privacy, which offers to individuals within an area with radius r, a privacy level of l. More specifically, l is equal to εr if any two locations within distance r provide data with similar distributions. This similarity depends on r because the closer two locations are, the more likely they are to share the same features. Intuitively, the definition implies that if an adversary learns the published location for an individual, the adversary cannot infer the individual's true location, out of all the points in a radius r, with a certainty higher than a factor depending on l. The technique adds random noise drawn from a multivariate Laplace distribution to individuals' locations, while taking into account spatial boundaries and features.

Example 2.3. To illustrate the usage of the microdata and statistical data techniques for privacy-preserving data publishing, we revisit Example 2.2. In this example, users continuously interact with an LBS by reporting their status at various locations. Then, the reported data is collected by the central service, in order to be protected and then published, either as a whole, or as statistics thereof. Notice that in order to showcase the straightforward application of k-anonymity and differential privacy, we apply the two methods on each timestamp independently from the previous one, and do not take into account any additional threats imposed by continuity.

Name	Age	Location	Status
*	> 20	Paris	at work
*	> 20	Paris	driving
*	> 20	Paris	dining
*	≤ 20	Paris	running
*	≤ 20	Paris	at home
*	≤ 20	Paris	walking

t_1

Name	Age	Location	Status
*	> 20	Paris	driving
*	> 20	Paris	at the mall
*	> 20	Paris	biking
*	≤ 20	Paris	sightseeing
*	≤ 20	Paris	walking
*	≤ 20	Paris	at home

t_2 ...

Table 3: 3-anonymous event-level protected versions of the microdata in Table 2a.

First, we anonymize the data set of Table 2a using k-anonymity, with $k = 3$. This means that any user should not be distinguished from at least 2 others. Status is the sensitive attribute, thus the attribute that we wish to protect. We start by suppressing the values of the Name attribute, which is the identifier. The Age and Location attributes are the quasi-identifiers, so we proceed to adequately generalize them. We turn age values to ranges (≤ 20, and > 20), and generalize location to city level (Paris). Finally, we achieve 3-anonymity by putting the entries in groups of three, according to the quasi-identifiers. Table 3 depicts the results at each timestamp.

Location	Count
Belleville	1
Latin Quarter	1
Le Marais	1
Montmartre	2
Opera	1

(a) True counts

Noise →

Location	Count
Belleville	1
Latin Quarter	0
Le Marais	2
Montmartre	3
Opera	1

(b) Perturbed counts

Table 4: (a) The original version of the data of Table 2b, and (b) their 1-differentially event-level private version.

Next, we demonstrate differential privacy. We apply an ε-differentially private Laplace mechanism, with $\varepsilon = 1$, taking into account the count query that generated the true counts of Table 2b. The sensitivity of a count query is 1 since the addition/removal of a tuple from the data set can change the final result of the query by maximum 1 (tuple). Figure 5 shows how the Laplace distribution for the true count in Montmartre at t_1 looks like. Table 4b shows all the perturbed counts that are going to be released.

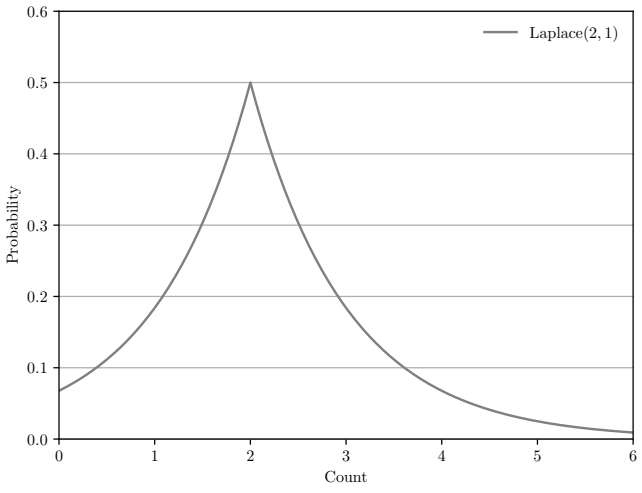


Figure 5: A Laplace distribution for location $\mu = 2$ and scale $b = 1$.

3 Privacy-preserving continuous data publishing

In this section we review the algorithms proposed in the literature for privacy-preserving continuous publishing. We organize the presentation following the data categories identified in Section 2: works that deal with privacy-preserving microdata are presented in Section 3.1, and works for privacy-preserving statistical data are presented in Section 3.2. Furthermore, we divide the works in each of the two sections based on the data type: finite and infinite data. This categorization will help the interested reader navigate with ease the provided reviews, depending on the nature of the data that he/she wants to manipulate and protect.

The accompanying Table 5 and Table 6 summarize all the works reviewed in this survey and put a more detailed index in the disposal of the reader. There are two main columns, concerning the:

- **Data**. In the first column of the Data part, we identify the considered data *Category* from the set of categories defined in Section 2.1.1, i.e., finite or infinite. We may also encounter here the subcategories sequential or incremental, when the corresponding algorithm is designed for these specific kinds of data. We outline (in bold) the cases where spatial data is explicitly considered; nevertheless, all other algorithms could be equally applied on location data as well. The second column of the Data part concerns the adopted *Publishing Mode*, i.e., batch or streaming, and the *Publishing Scheme*, i.e., global or local, as they are defined in Section 2.1.2.
- **Protection**. The second part of the table contains four columns. First, we mention the *Attack* scenario, i.e., complementary release or data dependence (see Section 2.2.2), together with more specific sub-categories, when available. Notice here, that several works based on differential privacy refer directly to the most general category that we have defined, namely the linkage attack. Second, we mention the base protection *Method*, which is mostly k-anonymity, differential privacy, or an extension thereof (see Section 2.2.4). Third, we mention the protection *Level*, i.e., event, user, or w-event (see Section 2.2.1). User level is the most popular because, inevitably, a user is more exposed when their data is included in many (continuous) releases. Notice that we have applied the privacy levels to algorithms other than differential privacy by looking whether the quantity of added distortion depends on the fact that a user may exist in multiple releases or not (user- vs event-level privacy). Finally, we list the privacy *Operation* that is utilized in the algorithm (see Section 2.2.3). Naturally, the most popular operation for microdata privacy protection is generalization, because in this way we can easily group tuples and hide identifiers in the groups. For statistics privacy protection, the most popular operation is perturbation, which introduces statistical noise to the computed statistics. Certain algorithms combine different operations in order to achieve better user privacy and/or data quality.

3.1 Microdata

As observed in Table 5, privacy-preserving algorithms for microdata rely mostly on k-anonymity or derivatives of it. Ganta et al. [51] revealed that k-anonymity methods are vulnerable to complementary release attacks (or *composition attacks* in the original publication). Consequently, the research community proposed solutions based on k-anonymity,

Microdata							
Article	Data			Protection			
	Category	Publishing		Level	Attack	Operation	Method
		Mode	Scheme				
(*k*, *δ*)-*anonymity* [12]	finite (sequential)	batch	global	user	complementary release	generalization, randomization	k-anonymity
Li et al. [79]	finite	batch	global	user	compl. release (unknown releases)	generalization, randomization	l-diversity
Erdogdu and Fawaz [43]	finite	batch/ streaming	local	user	dependence (temporal)	randomization	-
Jiang et al. [64]	finite (sequential)	batch	global	event	linkage	perturbation (Laplace)	differential privacy
Chen et al. [31]	finite (sequential)	batch	global	user	linkage	perturbation (Laplace)	differential privacy
Xiao et al. [117]	finite (sequential)	batch	local	user	dependence (temporal)	perturbation (multi-variate Laplace)	differential privacy
Promesse [93]	finite (sequential)	batch	local	event	linkage	perturbation	-
DP-Star [57]	finite (sequential)	batch	global	user	linkage	perturbation (Laplace)	differential privacy
(X, Y)-privacy [112]	infinite (sequential)	batch	global	user	compl. release (join)	generalization, specialization	k-anonymity
BCF-anonymity [49]	infinite (incremental)	batch	global	user	compl. release (tuple correspondence)	generalization, specialization	k-anonymity
m-invariance [116]	infinite	batch	global	user	compl. release	generalization, synthetic data	l-diversity
e-equivalence [60]	infinite	batch	global	user	compl. release (tuple equivalence)	generalization synthetic data	l-diversity
Shmueli and Tassa [99]	infinite (sequential)	batch	global	user	compl. release (unknown releases)	generalization, permutation	l-diversity
Zhou et al. [122]	infinite	streaming	global	event	same with k-anonymity [107]	generalization, randomization	k-anonymity
MaskIt [54]	infinite	streaming	local	event	dependence (temporal)	suppression	-
PLP [84]	infinite	streaming	local	event	dependence (spatiotemporal)	suppression	-
Al-Dhubhani and Cazalas [13]	infinite (sequential)	streaming	local	event	dependence (temporal)	perturbation (multi-variate Laplace)	geo-indistin-guishability
Ghinita et al. [52]	infinite (sequential)	streaming	local/ global	event	dependence (spatiotemporal)	generalization, perturbation	-
Ye et al. [119]	infinite (sequential)	streaming	global	event	linkage	generalization	l-diversity
Cao et al. [25,26]	finite/ infinite	streaming	global	user/ (w-)event	dependence (temporal)	perturbation (Laplace)	differential privacy

Table 5: Summary table of reviewed privacy-preserving algorithms for continuous micro-data publishing. Location-specific techniques are listed in bold.

focusing on different threats linked to continuous publication, as we review later on. However, notice that only a couple [79, 99] of the following works assume that data sets are privacy-protected *independently* of one another, meaning that the publisher is oblivious of the rest of the publications. On the other side, algorithms that are based on differential privacy are not concerned with so specific attacks as, by definition, differential privacy considers that the adversary may possess any kind of background knowledge. Later on, data dependencies were also considered for differential privacy algorithms, to account for the extra privacy loss entailed by them.

Statistical data							
Article	**Data**			**Protection**			
	Category	**Publishing**		**Level**	**Attack**	**Operation**	**Method**
		Mode	**Scheme**				
Kellaris et al. [69]	finite	batch	global	event	linkage	perturbation (Laplace)	differential privacy
Chen et al. [30]	finite (sequential)	batch	global	user	linkage	perturbation (Laplace)	differential privacy
Hua et al. [61]	finite (sequential)	batch	global	user	linkage	perturbation (exponential, Laplace)	differential privacy
Li et al. [80]	finite (sequential)	batch	global	user	linkage	perturbation (Laplace)	differential privacy
DPT [59]	finite (sequential)	batch	global	user	dependence (spatial)	perturbation (Laplace)	differential privacy
Song et al. [103]	finite	batch	global	event	dependence	perturbation (Laplace)	pufferfish
Fan et al. [46]	finite (sequential)	streaming	global	user	dependence (spatiotemporal)	perturbation (Laplace)	differential privacy
FAST [45]	finite	streaming	global	user	linkage	perturbation (Laplace)	differential privacy
CTS-DP [110]	finite	streaming	global	event	dependence (serial)	perturbation (Laplace)	differential privacy
Chan et al. [27]	finite/ infinite	streaming	global	event	linkage	perturbation (Laplace)	differential privacy
l-trajectory [24]	infinite (sequential)	streaming	global	w-event	linkage	perturbation (Laplace)	differential privacy
Bolot et al. [22]	infinite	streaming	global	w-event	linkage	perturbation (Laplace)	differential privacy
Kellaris et al. [70]	infinite	streaming	global	w-event	linkage	perturbation (Laplace)	differential privacy
RescueDP [113]	infinite	streaming	global	w-event	dependence (serial)	perturbation (Laplace)	differential privacy
RAPPOR [44]	infinite	streaming	local	user	linkage	randomization (randomized response)	differential privacy
PrivApprox [95]	infinite	streaming	global	event	linkage	randomization (randomized response)	differential privacy
Li et al. [78]	infinite	streaming	global	event	dependence (serial)	randomization	-
PeGaSus [33]	infinite	streaming	global	event	linkage	perturbation (Laplace)	differential privacy

Table 6: Summary table of reviewed privacy-preserving algorithms for continuous statistical data publishing. Location-specific techniques are listed in bold.

3.1.1 Finite observation

Wang and Fung [112] address the problem of anonymously releasing different projections (i.e., subsets of the attributes) of the same data set in subsequent timestamps. More precisely, the authors want to protect individual information that could be revealed from joining various releases of the same data set. To do so, instead of locating the quasi-identifiers in a single release, the authors suggest that the identifiers may span the current and all previous releases of the (projections of the) data set. Then, the proposed method uses the join of the different releases on the common identifying attributes. The goal is to generalize the identifying attributes of the current release, given that previous releases are immutable. The generalization is performed in a top down manner, meaning that the attributes are ini-

tially over-generalized, and step by step are specialized until they reach the point when predefined quality and privacy requirements are met. The privacy requirement is the so-called *(X, Y)-privacy* for a threshold k, meaning that the identifying attributes in X are linked with at most k sensitive values in Y, in the join of the previously released and current data sets. The quality requirement can be tuned into the framework. Namely, the authors propose three alternatives: the reduction of the class entropy [94, 98], the notion of distortion, and the discernibility [17]. The anonymization algorithm for releasing a data set in the existence of a previously released data set takes into account the scalability and performance problems that a join among those two may entail. Still, when many previous releases exist, the complexity would remain high.

Fung et al. [49] introduce the problem of privately releasing continuous incremental data sets. As a reminder, the invariant of this kind of releases is that at every timestamp t_i, the records previously released at t_j ($j < i$) are released again together with a set of new records. The authors first focus in two consecutive releases and describe three classes of possible attacks, which fall under the general category of complementary attacks. They name these attacks *correspondence attacks* because they rely on the principle that all tuples from an original data set D_1, from timestamp t_1, correspond to a tuple in the data set D_2, from timestamp t_2. Naturally, the opposite does not hold, as tuples added at t_2 do not exist in D_1. Assuming that the attacker knows the quasi-identifiers and the timestamp of the record of a person, they define the *backward*, *cross*, and *forward* (*BCF*) attacks. They show that combining two individually k-anonymized subsequent releases using one of the aforementioned attacks can lead to 'cracking' some of the records in the set of k candidate tuples rendering the privacy level lower than k. Except for the detection of cases of compromising BCF anonymity between two releases, the authors also provide an anonymization algorithm for a release $\boldsymbol{o}_2$ in the presence of a private release $\boldsymbol{o}_1$. The algorithm starts from the most possible generalized state for the quasi-identifiers of the records in D_2. Step by step, it checks which combinations of specializations on the attributes do not violate the BCF anonymity and outputs the most possible specialized version of the data set. The authors discuss how the framework extends to multiple releases and to different kinds of privacy methods (other than k-anonymity). It is worth noting that to maintain a certain quality for a release, it is essential that the delta among subsequent releases is large enough; otherwise the needed generalization level may destroy the utility of the data set.

Abul et al. [12] defined *(k, δ)-anonymity* for enabling high-quality moving-objects data sets publishing. The authors claim that the classical k-anonymity framework cannot be directly applied to such kind of data from a data-centric perspective. The traditional distortion techniques in k-anonymity, i.e., generalization or suppression, yield great loss of information. On the one hand, suppression diminishes the size of the database. On the other hand, generalization demands the existence of quasi-identifiers, the values of which are going to be generalized. In trajectories, however, all points can be equally considered as quasi-identifiers. Obviously, a generalization of all the trajectories points would yield great levels of distortion. For this reason, a new, spatial-based distortion method is proposed. After clustering the trajectories in groups of at least k elements, each trajectory is translated into a new one, in a vicinity of a predefined threshold δ. Of course, the newly generated trajectories should still form a k-anonymous set. The authors validate their theory by experimentally showing that the resulting distance of count queries executed over a data set and its (k, δ) version, remains low. However, a comparative evaluation to existing clustering

techniques, e.g., k-means would have been interesting, to better support the contributions on this part of the solution as well.

Erdogdu and Fawaz [43] consider the scenario where privacy-conscious individuals separate the data that they generate into sensitive and non-sensitive. The individuals keep the former unreleased and publish samples of the latter to a service provider. Privacy mapping, implemented as a stochastic process, distorts the non-sensitive data samples locally, and a separable distortion metric (e.g., Hamming distance) calculates the discrepancy of the distorted data from the original. The goal of the privacy mapping is to find a balance between the distortion and privacy metric, i.e., achieve maximum released data utility, while offering sufficient privacy guarantees. The authors assume that there is a data dependence (modeled with an HMM) between the two data sets, and thus the release of the distorted data set can reveal information about the sensitive one. They investigate both a simple attack setting and a complex one. In the simple attack, the adversary can make static assumptions, based only on the so far made observations that cannot be later altered. In the complex attack, past and future data releases affect dynamically the assumptions that an adversarial entity makes. In both cases, the framework quantifies the information leakage at any time point using a privacy metric that measures the improvement of the adversarial inference of the sensitive data set, which the individual kept secret, after observing the data released at that particular point. Throughout the process, the authors consider both the batch and the streaming processing schemes. However, the assumption that individuals are privacy-conscious can drastically limit the applicability of the framework. Furthermore, the metrics that the framework utilizes for the evaluation of the privacy guarantees that it provides are not intuitive.

Xiao et al. [116] consider the case when a data set is (re)published in different timestamps in an update (insert/delete tuple) manner. More precisely, they address data anonymization in continuous publishing by implementing m-*invariance*. In a simple k-anonymity (or l-diverse) scenario the privacy of an individual existing in two updates can be compromised by the intersection of the set of sensitive values. In contrast, an individual who exists in a series of m-invariant releases is always associated with the same set of m different sensitive values. To enable the publishing of m-invariant data sets, artificial tuples (*counterfeits*) may be added in a release. To minimize the noise added to the data sets, the authors provide an algorithm with two extra desiderata: limit the counterfeits and minimize the quasi-identifiers' generalization level. Still, the choice of adding tuples with specific sensitive values disturbs the value distribution with a direct effect on any relevant statistics analysis.

In the same update setting (insert/delete tuple), He et al. [60] introduce another kind of attack, namely the *equivalence* attack, not taken into account by the aforementioned m-invariance technique. The equivalence attack allows for sets of individuals to be considered equivalent as far as the sensitive attribute is concerned, in different timestamps. In this way, all the members of the equivalence class will be harmed, if the sensitive value is learned even for only one member. For a number of releases to be private, they have to be both m-invariant and e-equivalent ($e < m$). The authors propose an algorithm incorporating m-invariance, based on the graph optimization *min cut* problem, for publishing e-equivalent data sets. The proposed method can achieve better levels of privacy, in comparable times and quality as m-invariance.

Shmueli and Tassa [99] identified the computational inefficiency of anonymously releasing a data set, taking into account previous ones, in scenarios of continuous data pub-

lishing. The released data sets contain subsets of attributes of an original data set, while the authors propose an extension for attribute addition. Their algorithm can compute l-diverse anonymized releases (over different subsets of attributes) in parallel by generating $l-1$ so-called *fake* worlds. A fake world is generated from the base data set by randomly permutating non-identifier and sensitive values among the tuples, in such a way that minimal information loss (quality desideratum) is incurred. This is partially accomplished by verifying that the permutation is done among quasi-identifiers that are similar. Then, the algorithm creates buckets of tuples with at least l different sensitive values, in which the quasi-identifiers will then be generalized in order to achieve l-diversity (privacy protection desideratum). The generalization step is also conducted in an information-loss efficient way. All different releases will be l-diverse because they are created assuming the same possible worlds, with which they are consistent. Tuples/attributes deletion is briefly discussed and left as an open question. The article is contrasted with a previous work [100] of the same authors, claiming that the new approach considers a stronger adversary (the adversary knows all individuals with their quasi-identifiers in the data set and not only one), and that the computation is much more efficient, as it does not have an exponential complexity with respect to the number of previous publications.

Li et al. [79] identified a common characteristic in most of the privacy techniques: when anonymizing a data set all previous releases are known to the data publisher. However, it is probable that the releases are independent from each other, and that the data publisher is unaware of these releases when anonymizing the data set. In such a setting, the previous techniques would suffer from composition attacks. The authors define this kind of adversary and propose a hybrid model for data anonymization. More precisely, the publisher/adversary knows that an individual exists in two different anonymized versions of the same data set, he has a hold of the anonymized versions, but the anonymization is done independently (i.e., without considering the previously anonymized data sets) for each data set. The key idea in fighting a composition attack is to enforce the probability that the matches among tuples from two data sets are random, linking different rather than the same individual. To do so, the proposed privacy protection method exploits three preprocessing steps before applying a traditional k-anonymity or l-diversity algorithm. First, the data set is sampled so as to blur the knowledge of the existence of individuals. Then, especially in small data sets, quasi-identifiers are distorted by noise addition before the classical generalization step. The noise is taken from a normal distribution with the mean and standard deviation values calculated on the corresponding quasi-identifier values. In the case of sparse data, the sensitive values are generalized along with the quasi-identifiers. The danger of composition attacks is less prominent when using this method on top of k-anonymity rather than without, while having comparable quality results. The authors also provide a comparison to data set release using ε-differential privacy, demonstrating that their techniques are superior with respect to quality because in the opponent algorithm the noise is added up for each of the sensitive attribute to be protected. Even though the authors use in the experiments two different values for ε, a better experiment would have been to compare the quality/privacy ratio between the two methods. This is a good attempt to independently anonymize multiple times the same data set; nevertheless, the scenario is restricted to releases over the same database schema, using the same perturbation and generalization functions.

Jiang et al. [64] focus on ship trajectories with known starting and terminal points. More specifically, they study different noise addition mechanisms for publishing trajectories with

differential privacy guarantees. These mechanisms include adding global noise to the trajectory and local noise to either each location point or the coordinates of each point of the trajectory. The first two mechanisms sample noisy radius from an exponential distribution, while the latter adds noise drawn from a Laplace distribution to each coordinate of every location. By comparing these different techniques, they conclude that the latter offers better privacy guarantee and smaller error bound. Nonetheless, the resulting trajectory is noticeably distorted due to the addition of Laplace noise to the original coordinates. To tackle this issue, they design the *Sampling Distance and Direction* (SDD) mechanism. This mechanism allows the publishing of optimal next possible trajectory point by sampling, from the probability distribution of the exponential mechanism, a suitable distance and direction at the current position, while taking into account the ship's maximum speed constraint. Due to the fact that SDD utilizes the exponential mechanism, it outperforms the other three mechanisms and maintains a good utility-privacy balance.

Chen et al. [31] propose a non-interactive data-dependent privacy-preserving algorithm to generate a differentially private release of trajectory data. The algorithm relies on a noisy prefix tree, i.e., an ordered search tree data structure used to store an associative array. Each node represents a location, from a set of possible locations that any user can be present at, of a trajectory and contains a perturbed count, which represents the number of individuals at the current location, with noise drawn from a Laplace distribution. The privacy budget is equally allocated to each level of the tree representing a timestamp. At each level and for every node, the algorithm seeks for the children nodes with non-zero number of trajectories (non-empty nodes) to continue expanding them. An empty node has a noisy count lower than a threshold that is dependent on the available privacy budget and the height of the tree. All children nodes associate with disjoint data subsets, and thus the algorithm can utilize for every node all of the available budget at every tree level, according to the parallel composition theorem of differential privacy. To generate the anonymized database, it is necessary to traverse the prefix tree once in post-order, paying attention to terminating (empty) nodes. During this process, taking into account some consistency constraints helps to avoid erroneous trajectories due to the noise injection. Namely, each node of a path should have a count that is greater than or equal to the counts of its children, and each node of a path should have a count that is greater than the sum of the counts of all of its children. Increasing the privacy budget results in less average relative error because less noise is added at each level, and thus improves quality. By increasing the height of the tree, the relative error initially decreases as more information is retained from the database. However, after a certain threshold, the increase of height can result in less available privacy budget at each level, and thus more relative error due to the increased perturbation.

Xiao et al. [117] propose another privacy definition based on differential privacy that accounts for temporal correlations in geo-tagged data. Location transitions between two consecutive timestamps are determined by temporal correlations modeled through a Markov chain. A *δ-location* set includes all the probable locations a user might appear at, excluding locations of low probability. Therefore, the true location is hidden in the resulting set, in which any pair of locations are indistinguishable. The lower the value of δ, the more locations are included and hence, the higher the level of privacy that is achieved. The authors use the *Planar Isotropic Mechanism* (PIM) as perturbation mechanism, which they designed upon their proof that l_1-norm sensitivity fails to capture the exact sensitivity in a multidimensional space. For this reason, PIM utilizes instead *sensitivity hull*, an independent

notion of the context of location privacy. In [118], the authors demonstrate the functionality of their system *LocLok*, which implements the concept of δ-location.

Primault et al. [93] proposed *Promesse*, an algorithm that builds on time distortion instead of location distortion when releasing trajectories. Promesse takes as input an individual's mobility trace comprising of a data set of pairs of geolocations and timestamps, and a parameter ε. The latter indicates the desired distance between the location points that will be publicly released. Initially, Promesse extracts regularly spaced locations and interpolates each one of the locations at a distance depending on the previous location and the value of ε. Then, it removes the first and last locations of the mobility trace and assigns uniformly distributed timestamps to the remaining locations of the trajectory. Hence, the resulting trace has a smooth speed, and therefore places where the individual stayed longer, e.g., home, work, etc., are indistinguishable. The algorithm needs to know the starting and ending point of the trajectory; thus, it can only apply to offline scenarios. Furthermore, it works better with fine grained data sets because in this way it can achieve optimal geolocation and timestamp pairing. Moreover, the definition of ε cannot provide versatile privacy protection since it is data dependent.

Gursoy et al. [57] designed *DP-Star*, a differential privacy framework that publishes synthetic trajectories featuring similar statistics compared to the original ones. By utilizing the *Minimum Description Length* (MDL) principle [55], DP-Star eliminates redundant data points in the original trajectories and generates trajectories containing only representative points. In this way, it is necessary to allocate the available privacy budget to far less data points, striking a balance between preciseness and conciseness. Moreover, the algorithm constructs a density-aware grid, with granularity that adapts to the geographical density of the trajectory points of the data set and preserves the spatial density despite any necessary perturbation. Then, DP-Star preserves the dependence between the trajectories' start and end points by extracting (through a first-order Markov mobility model) the trip distribution and the intra-trajectory mobility. Finally, a Median Length Estimation (MLE) mechanism approximates the trajectories' lengths, and the framework generates privacy and utility preserving synthetic trajectories. Every phase of the process consumes some predefined privacy budget, keeping the respective products of each phase private and eligible for publishing. The authors compare their design with that of [30] and [59] by running several tests, and ascertain that it outperforms them in terms of data utility. However, due to DP-Star's privacy budget distribution to its different phases, for small values of ε the framework's privacy performance is inferior to that of its competitors.

3.1.2 Infinite observation

Zhou et al. [122] introduce the problem of infinite private data publishing, and propose a randomized solution based on k-anonymity. More precisely, they continuously publish equivalence classes of size greater than or equal to k containing generalized tuples from distinct persons (or identifiers in general). To create the equivalence classes they set several desiderata. Except for the size of a class, which should be greater than or equal to k, the information loss occurred by the generalization should be minimal, whereas the delay in forming and publishing the class should be kept low as well. To achieve these requirements, they built a randomized model using the popular structure of R-trees, extended to accommodate data density distribution information. In this way, they achieve a better quality/publishing delay ratio for the released private data. On the one hand, the formed

classes contain data items that are close to each other (in dense areas), while on the other hand, classes with tuples of sparse areas are released as soon as possible so that the delay will remain low.

Gotz et al. [54] developed *MaskIt*, a system that interfaces the sensors of a personal device, identifies various sets of contexts and releases a stream of privacy-preserving contexts to untrusted applications installed on the device. A context represents the circumstances that form the setting for an event, e.g., 'at the office', 'running', etc. The individuals have to define the sensitive contexts that they wish to be protected and the desired level of privacy. The system models the individuals' various contexts and transitions between them. It captures temporal correlations and models individuals' movement in the space using Markov chains while taking into account historical observations. After the initialization, MaskIt filters a stream of individual's contexts by checking for each context whether it is safe to release it or it is necessary to suppress it. The authors define *δ-privacy* as the privacy model of MaskIt. More specifically, a system preserves δ-privacy if the difference between the posterior and prior knowledge of an adversary after observing an output at any possible timestamp is bounded by δ. After filtering all the elements of an input stream, MaskIt releases an output sequence for a single day. The system can repeat the process to publish longer context streams. The expected number of released contexts quantifies the utility of the system.

Ma et al. [84] propose *PLP* (Protecting Location Privacy), a crowdsensing scheme that protects location privacy against adversaries that can extract spatiotemporal correlations from crowdsensing data. PLP filters an individual's context (location, sensing data) stream while it takes into consideration long-range dependencies among locations and reported sensing data, which are modeled by CRFs. It suppresses sensing data at all sensitive locations while data at non-sensitive locations are reported with a certain probability defined by observing the corresponding CRF model. On the one hand, the scheme estimates the privacy of the reported data by the difference δ between the probability that an individual would be at a specific location given the supplementary information versus the same probability without the extra information. On the other hand, it quantifies the utility by measuring the total amount of reported data (more is better). An estimation algorithm searches for the optimal strategy that maximizes utility while preserving a predefined privacy threshold.

Al-Dhubhani and Cazalas [13] propose an adaptive privacy-preserving technique based on geo-indistinguishability, which adjusts the amount of noise required to obfuscate an individual's location based on its correlation level with the previously published locations. Before adding noise, an evaluation of the adversary's ability to estimate an individual's position takes place. This process utilizes a regression algorithm for a certain prediction window that exploits previous location releases. More concretely, in areas with locations presenting strong correlations, an adversary can predict the current location with low estimation error. Consequently, it is necessary to add more noise to the locations prior to their release. Adapting the amount of injected noise depending on the data correlation level might lead to a better performance, in terms of both privacy and utility, in the short term. However, alternating the amount of injected noise at each timestamp, without ensuring the preservation of the features (including correlations) present in the original data, might lead to arbitrary utility loss.

Ghinita et al. [52] tackle attacks to location privacy that arise from the linkage of maximum velocity with cloaked regions when using an LBS. The authors propose methods that

can prevent the disclosure of the exact location coordinates of an individual, and bound the association probability of an individual to a sensitive location-related feature. The first method is based on temporal cloaking and utilizes deferral and postdating. Deferral delays the disclosure of a cloaked region that is impossible for an individual to have reached based on the latest region that she published and her known maximum speed. Postdating reports the nearest previous cloaked region that will allow the LBS to return relevant results with high probability, since the two regions are close. The second method implements spatial cloaking. First, it creates cloaked regions by taking into account all of the user-specified sensitive features that are relevant to the current location (filtering of features). Then, it enlarges the area of the region to satisfy the privacy requirements (cloaking). Finally, it defers the publishing of the region until it includes the current timestamp (safety enforcement) similar to temporal cloaking. The system measures the quality of service of both methods in terms of the cloaked region size, time and space error, and failure ratio. The cloaked region size is important because larger regions may decrease the utility of the information that the LBS might return. The time and space error is possible due to delayed location reporting and region cloaking. Failure ratio corresponds to the percentage of dropped queries in cases where it is impossible to satisfy the privacy requirements. Although both methods experimentally prove to offer adequate quality of service, the privacy requirements and metrics that the authors consider do not offer substantial privacy guarantees for commercial application.

Ye et al. [119] present an l-diversity method for producing a cloaked area, based on the local road network, for protecting trajectories. A trusted entity divides the spatial region of interest based on the density of the road network, using quadtree structures, until every subregion contains at least l road segments. Then, it creates a database for each subregion by generating all the possible trajectories based on real road network information. The trusted entity uses this database, when individuals attempt to interact with an LBS by sending their current location, to predict their next locations. Thereafter, it selects the $l-1$ nearest trajectories to the individual's current location and constructs a minimum cloaking region. The resulting cloaking area covers the l nearest trajectories and ensures a minimum area of coverage. This method addresses the limitations of k-anonymity in terms of continuous data publishing of trajectories. The required calculation of every possible trajectory, for the construction of a trajectory database for every subregion, might require an arbitrary amount of computations depending on the area's features. Nonetheless, the utilization of quadtrees can limit the overhead of the searching process.

Cao et al. [25,26] propose a method for computing the temporal privacy loss of a differential privacy mechanism in the presence of temporal correlations and background knowledge. The goal of their technique is to guarantee privacy protection and to bound the privacy loss at every time point under the assumption of independent data releases. It calculates the temporal privacy loss as the sum of the backward and forward privacy loss minus the default privacy loss ε of the mechanism (because it is counted twice in the aforementioned entities). This calculation is done for each individual that is included in the original data set, and the overall temporal privacy loss is equal to the maximum calculated value at every time point. The backward/forward privacy loss at any time point depends on the backward/forward privacy loss at the previous/next instance, the backward/forward temporal correlations, and ε. The authors propose solutions to bound the temporal privacy loss, under the presence of weak to moderate correlations, in both finite and infinite data publishing scenarios. In the latter case, they try to find a value for ε for

which the backward and forward privacy loss are equal. In the former, they similarly try to balance the backward and forward privacy loss while they allocate more ε at the first and last time points, since they have higher impact to the privacy loss of the next and previous ones. This way they achieve an overall constant temporal privacy loss throughout the time series. According to the technique's intuition, stronger correlations result in higher privacy loss. However, the loss is smaller when the dimension of the transition matrix, which is extracted according to the modeling of the correlations (here it is Markov chain), is larger due to the fact that larger transition matrices tend to be uniform, resulting in weaker data dependence. The authors investigate briefly all of the possible privacy levels; however, the solutions that they propose are suitable only for the event-level. Last but not least, the technique requires the calculation of the temporal privacy loss for every individual within the data set which might prove computationally inefficient in real-time scenarios.

3.2 Statistical data

When continuously publishing statistical data, usually in the form of counts, the most widely used privacy method is differential privacy, or derivatives of it, as witnessed in Table 6. In theory differential privacy makes no assumptions about the background knowledge available to the adversary. In practice, as we observe in Table 6, data dependencies (e.g., correlations) arising in the continuous publication setting are frequently (but without it being the rule) considered as attacks in the proposed algorithms.

3.2.1 Finite observation

Kellaris et al. [69] pointed out that in time series, where users might contribute to an arbitrary number of aggregates, the sensitivity of the query answering function is significantly influenced by their presence/absence in the data set. Thus, the Laplace perturbation algorithm, commonly used with differential privacy, may produce meaningless data sets. Furthermore, under such settings, the discrete Fourier transformation of the Fourier perturbation algorithm (another popular technique for data perturbation) may behave erratically and affect the utility of the outcome of the mechanism. For this reason, the authors proposed their own method involving grouping and smoothing for one-time publishing of time series of non-overlapping counts, i.e., the aggregated data of one count does not affect any other count. Grouping includes partitioning the data set into similar clusters. The size and the similarity measure of the clusters are data dependent. Random grouping consumes less privacy budget, as there is minimum interaction with the original data. However, when using a grouping technique based on sampling, which has some privacy cost but produces better groups, the impact of the perturbation is decreased. During the smoothing phase, the average values for each cluster are calculated, and, finally, Laplace noise is added to these values. In this way, the query sensitivity becomes less dependent on each individual's data, and therefore less perturbation is required.

Chen et al. [30] exploit a text-processing technique, the *n-gram* model, i.e., a contiguous sequence of n items from a given data sample, to release sequential data without releasing the noisy statistics (counts) of all of the possible sequences. This model allows the publishing of the most common n-grams (n is, typically, less than 5) to accurately reconstruct the original data set. The privacy technique that the authors propose is suitable for count queries and frequent sequential pattern mining scenarios. In particular, one of the appli-

cations that the authors consider concerns sequential spatiotemporal data (i.e., trajectories) of individuals. They group grams based on the similarity of their n values, construct a search tree, and inject Laplace noise to each node value (count) to achieve user-level differential privacy protection. Instead of allocating the available privacy budget based on the overall maximum height of the tree, they estimate each path adaptively based on known noisy counts. The grouping process continues until the desired threshold of n is reached. Thereafter, they release variable-length n-grams with certain thresholds for the values of counts and tree heights, allowing to deal with the trade-off of shorter grams having less information than longer ones but less relative error. They use a set of consistency constraints, i.e., the sum of each node's noisy count has to be less than or equal to its parent's noisy count, and all the noisy counts of leaf nodes have to be within a predefined threshold. These constraints improve the final data utility since they result in lower values of n. On the one hand, this translates into higher counts, large enough to deal with noise injection and the inherent Markov assumption in the n-gram model. On the other hand, it enhances privacy when the universe of all grams with a lower n value is relatively small resulting in more common sequences, which, nonetheless, is rarely valid in real-life scenarios.

Hua et al. [61] use, similar to the scheme proposed in [30], the n-grams modeling technique for publishing trajectories containing a small number of n-grams, thus, sharing few or even no identical prefixes. They propose a differentially private location-specific generalization algorithm (exponential mechanism), where each position in the trajectory is one record. The algorithm probabilistically partitions the locations at each timestamp with probability proportional to their Euclidean distance from each other. They replace each partition with its centroid and therefore, they offer better utility by creating groups of locations belonging to close trajectories. They optimize the algorithm for time efficiency by using classic k-means clustering. Then, the algorithm releases the new trajectories by observing the generalized location partitions and their perturbed counts (i.e., sum of the same locations at each timestamp) with noise drawn from a Laplace distribution. The process continues until the total count of the published trajectories reaches the size of the original data set. They can limit the total number of the possible trajectories by taking into account the individual's moving speed. The authors have measured the utility of distorted spatiotemporal range queries by measuring the Hausdorff distance from the original results and concluded that the utility deterioration is within reasonable boundaries considering the offered privacy guarantees. Similar to [30], their approach works well for a small location domain. To make it applicable to realistic scenarios, it is essential to truncate the original trajectories in an effort to reduce the location domain. This results in a coarse discretization of the location area, leading to the arbitrary distortion of the spatial correlations that are present in the original data set.

Li et al. [80] focus on publishing a set of trajectories, where, contrary to [61], each one is considered as a single entry in the data set. First, using k-means clustering they partition the original locations based on their pairwise Euclidean distances. The scheme represents each location partition by their mean (centroid). A larger number of partitions, in areas where close centroids exist, results in fewer locations in each partition, and thus lower trajectory precision loss. Before adding noise, they randomly select partition centroids to generate trajectories until they reach the size of the original data set. Then, they generate Laplace noise, which they bound according to a set of constraints, and they add it to the count of locations of each point of every trajectory. Finally, they release the generalized trajectories along with the noisy count of each location point. The authors prove

experimentally that they reduce considerably the trajectory merging time at the expense of utility.

He et al. present *DPT* (Differentially Private Trajectory) [59], a system that synthesizes mobility data based on raw, speed-varying trajectories of individuals, while providing ε-differential privacy protection guarantees. The system constructs a Hierarchical Reference Systems (HRS) model to capture correlations between adjacent locations by imposing a uniform grid at multiple resolutions (i.e., for different speed values) over the space, keeping a prefix tree for each resolution, and choosing the centroids as anchor points. In each reference system, anchor points have a small number of neighboring points with increasing (by a constant factor) average distance between them and fewer children anchor points as the grid resolution becomes finer. DPT estimates transition probabilities only for the anchor points in proximity to the last observed location and chooses the appropriate reference system for each raw point, so that the consecutive points of the trajectory are either neighboring anchors or have a parent-child relationship. The system generates the transition probabilities by estimating the counts in the prefix trees. Thereafter, it chooses the appropriate prefix trees, perturbs them with noise drawn from the Laplace distribution and adaptively prunes subtrees with low counts to improve the resulting utility. DPT implements a direction-weighted sampling postprocessing strategy for the synthetic trajectories to avoid the loss of directionality of the original trajectories due to the perturbation. Nonetheless, as with all other similar techniques, the usage of prefix trees limits the length of the released trajectories, which results into an uneven spatial distribution.

Song et al. [103] propose the *Wasserstein mechanism*, a technique that applies to any general instantiation of Pufferfish (see Section 2.2.4). It adds noise proportional to the sensitivity of a query F, which depends on the worst case distance between the distributions $P(F(X)|s_i, d)$ and $P(F(X)|s_j, d)$ for a variable X, a pair of secrets (s_i, s_j), and an evolution scenario d. The Wasserstein metric function calculates the worst case distance between those two distributions. The noise is drawn from a Laplace distribution with parameter equal to the quotient resulting from the division of the maximum Wasserstein distance of the distributions of all the pairs of secrets by the available privacy budget ε. For optimization purposes, the authors consider a more restricted setting. This setting, utilizes an evolution scenario for the data correlations representation, and Bayesian networks for the correlation modeling. The authors state that in cases where Bayesian networks are complex, the Markov chains are a more efficient alternative. A generalization of the *Markov blanket* mechanism, the *Markov quilt* mechanism, calculates data dependencies. The dependent nodes of any node consist of its parents, its children, and the other parents of its children. The present technique excels at data sets generated by monitoring applications or networks, but it is not suitable for online scenarios.

Fan et al. [46] propose a real-time framework for releasing differentially private multidimensional traffic monitoring data. At every timestamp, the Perturbation module injects noise drawn from a Laplace distribution to the data. Then, the Estimation module postprocesses the perturbed data to improve the accuracy. The authors propose a temporal and spatial estimation algorithm. The former estimates an internal time series model for each location to improve the utility of the perturbation's outcome by performing a posterior estimation that utilizes Gaussian approximation and Kalman filtering [66]. The latter reduces data sparsity by grouping neighboring locations using a spatial indexing structure based on quadtree. The Modeling/Aggregation module utilizes domain knowledge, e.g., road network and density, and has a bidirectional interaction with the other two in parallel. Al-

though the authors propose the framework for real-time scenarios, they do not deal with infinite data processing/publication, which limits considerably its applicability.

In another work, Fan et al. designed *FAST* [45], an adaptive system that allows the release of real-time aggregate time series under user-level differential privacy. These were achieved by using a Sampling, a Perturbation, and a Filtering module. The Sampling module samples on an adaptive rate the aggregates to be perturbed. The Perturbation module adds noise to each sampled point according to the allocated privacy budget. The Filtering module receives the perturbed data point and the original one and generates a posterior estimate, which is finally released. The error between the perturbed and the released (posterior estimate) point is used to adapt the sampling rate; the sampling frequency is increased when data is going through rapid changes and vice-versa. Thus, depending on the adjusted sampling rate, not every single data point is perturbed, saving in this way the available privacy budget. While the system considers the temporal correlations of the processed time series, it does not attempt to deal with the privacy threat that they might pose.

Wang and Zu [110] defined Correlated Time Series Differential Privacy (*CTS-DP*). The scheme guarantees that the correlation between the perturbation that is introduced by a Correlated Laplace Mechanism (CLM), and the original time series is indistinguishable (Series-Indistinguishability). CTS-DP deals with the shortcomings of independent and identically distributed (i.i.d.) noise under the presence of correlations. I.i.d. noise offers inadequate protection, because refinement methods, e.g., filtering, can remove it. Most privacy-preserving methods choose to introduce more noise in the presence of strong correlations thus, diminishing the data utility. An original and a perturbed time series satisfy Series-Indistinguishability if their normalized autocorrelation functions are the same; hence, the two time series are indistinguishable and the published time series satisfies differential privacy as well. The authors consider the fact that, in signal processing, if an i.i.d. signal passes through a filter, which consists of a combination of adders and delayers, it becomes non-i.i.d. Hence, they design CLM, which uses four Gaussian white noise series passed through a linear system, to produce a correlated Laplace noise series according to the autocorrelation function of the original time series. Although the authors prove experimentally that the implementation of CLM outperforms the current state-of-the-art methods, they do not test its robustness against any filter, which they keep as future work.

3.2.2 Infinite observation

Chan et al. [27] designed continuous counting mechanisms for finite and infinite data processing and publishing, satisfying ε-differential privacy. Their main contribution lies in proposing the Binary and Hybrid mechanisms, which do not have any upper bound temporal requirements. The mechanisms rely on the release of intermediate partial sums of counts at consecutive timestamp intervals, called *p-sums*, and the injection of noise drawn from a Laplace distribution. The Binary mechanism constructs a binary tree where each node corresponds to a p-sum, and adds noise to each released p-sum proportional to its corresponding length. The Hybrid mechanism publishes counts at sparse time intervals, i.e., timestamps that are a power of 2. Both mechanisms offer event-level protection (pan-privacy) under single unannounced and continual announced intrusions by adding a certain amount of noise to every p-sum in memory. They can facilitate continual top-k queries in recommendation systems and multidimensional range queries. Furthermore, they are

able to support applications that require a consistent output, i.e., at each timestamp the counter increases by either 0 or 1.

Cao et al. [24] developed a framework that achieves personalized *l-trajectory* privacy protection by dynamically adding noise at each timestamp, which exponentially fades over time. Each individual can specify, in an array of size l, the desired protection level for each location of his/her trajectory. The proposed framework is composed of three components. The Dynamic Budget Allocation component allocates portions of the privacy budget to the other two components: a fixed one to the Private Approximation and a dynamic one to the Private Publishing component at each timestamp. The Private Approximation component estimates, under a utility goal and an approximation strategy, whether it is beneficial to publish approximate data or not. More precisely, it chooses an appropriate previous noisy data release and republishes it if it is similar to the real statistics planned to be published. The Private Publishing component takes as inputs the real statistics and the timestamp of the approximate data, generated by the Private Approximation component, to be republished. If the timestamp of the approximate data is equal to the current timestamp, then the current data with Laplace noise are published. Otherwise, the data at the corresponding timestamp of the approximate data will be republished. The utilized approximation technique is highly suitable for streaming processing, due to the implementation of approximation that can reduce significantly the privacy budget consumption. However, the framework does not take into account privacy leakage stemming from data dependencies, which limits considerably its applicability in real life data sets.

Bolot et al. [22] introduce the notion of *decayed privacy* in continual observation of aggregates (sums). The authors recognize the fact that monitoring applications focus more on recent events and data, therefore, the value of previous data releases exponentially fades. This leads to a schema of privacy with expiration, according to which recent events and data are more privacy sensitive than those preceding. Based on this, they apply decayed sum functions for answering sliding window queries of fixed window size w on data streams. Namely, window sum computes the difference of two running sums, and exponentially decayed and polynomial decayed sums estimate the sum of decayed data. For every consecutive w data points the algorithm generates binary trees where each node is perturbed with Laplace noise with scale proportional to w. Instead of maintaining a binary tree for every window, the algorithm considers the windows that span two blocks as the union of a suffix and a prefix of two consecutive trees. This way, the global sensitivity of the query function is kept low. The proposed techniques are designed for fixed window sizes, hence, when answering multiple sliding window queries with variable window sizes they have to distribute the available privacy budget accordingly.

Based on the notion of decayed privacy [22], Kellaris et al. [70] defined w-event privacy in the setting of periodical release of statistics (counts) in infinite streams. To achieve w-event privacy, the authors propose two mechanisms (Budget Distribution and Budget Absorption) based on sliding windows, which effectively distribute the privacy budget to sub-mechanisms (one sub-mechanism per timestamp) applied on the data of a window of the stream. Both algorithms may decide to publish a new noisy count for a specific timestamp, based on the similarity level of the current count with a previously published one. Moreover, both algorithms have the constraint that the total privacy budget consumed in a window is less than or equal to ε. The Budget Distribution algorithm distributes the privacy budget in an exponential-fading manner following the assumption that in a window most of the counts remain similar. The budget of expired timestamps becomes available for

the next publications (of next windows). The Budget Absorption algorithm uniformly distributes from the beginning the budget to the window's timestamps. A publication uses not only the by-default allocated budget but also the budget of non-published timestamps. In order to not exceed the limit of ε, adequate number of subsequent timestamps are 'silenced' after a publication takes place. Even though one can argue that w-event privacy could be achieved by user-level privacy, it is nevertheless non-practical because of the rigidity of the budget allocation that would finally render the output useless.

Wang et al. [113] propose *RescueDP* for the publishing of real-time user-generated spatiotemporal data, utilizing differential privacy with w-event-level protection. RescueDP uses a Dynamic Grouping module to create clusters of regions with small statistics, i.e., areas with a small number of samples. It estimates the similarity of the data trends of these regions by utilizing the Pearson's correlation coefficient and creates groups accordingly. The data of each group pass from a Perturbation module that injects Laplace noise to them. The grouping of the previous phase results into the increase of the sample size of each group of regions, which minimizes the error due to the noise injection. The implementation of a Kalman Filtering [66] module further increases the utility of the released data. A Budget Allocation module distributes the available privacy budget to sampling points within any successive w timestamps. RescueDP saves part of the available privacy budget by approximating the non-sampled data with previously released perturbed data. During the whole process, an Adaptive Sampling module adjusts the sampling interval according to the difference in the released data statistics over the previous timestamps while taking into account the remaining privacy budget.

Erlingsson et al. [44] presented *RAPPOR* (Randomized Aggregatable Privacy-Preserving Ordinal Response) as a solution for privacy-preserving collection of statistics. RAPPOR makes all the necessary data processing on the side of the data generators by applying the method of randomized response, which guarantees local differential privacy. The product of each local privacy-preserving processing is a report that can be represented as a bit string. Each bit corresponds to a randomized response to a logical predicate on an individual's personal data, e.g., categorical properties, numerical and ordinal values, or categories that cannot be enumerated. Initially, RAPPOR hashes a sensitive value into a Bloom filter [21]. It creates a binary reporting value, which keeps in its memory (*memoization*) and reuses for future reports (permanent randomized response). Memoization offers long-term longitudinal privacy protection for privacy-sensitive data values that do not change over time or that are not dependent. RAPPOR deals with tracking externalities by reporting a randomized version of the permanent randomized response (instantaneous randomized response). Although this adds an extra layer of randomization to the reported values, it might lead to an averaging attack that may allow an adversary to estimate the true value. Finally, the authors propose a decoding technique that involves grouping, least-squares solving, and regression. This way, they effectively make up for the loss of information due to the randomization of the previous steps and allow the extraction of useful information when observing the generated bit strings. They test their implementation with both simulated and real data and show that they can extract statistics with high utility while preserving the privacy of the individuals involved. However, the fact that the privacy guarantees of their technique are valid only for stationary individuals that produce independent data

on top of the relatively complex configuration, renders their proposal impractical for many real-world scenarios.

Le Quoc et al. [95] propose *PrivApprox*, a data analytics system for privacy-preserving stream processing of distributed data sets that combines sampling and randomized response. The system distributes the analysts' queries to clients via an aggregator and proxies, and employs sliding window computations over batched stream processing to handle the data stream generated by the clients. The clients transmit a randomized response, after sampling the locally available data, to the aggregator via proxies that apply (XOR-based) encryption. The combination of sampling and randomized response achieves *zero-knowledge* based privacy, i.e., proving that they know a piece of information without in fact disclosing its actual value. The aggregator collects the received responses and returns statistics to the analysts. The query model expresses the responses of numerical queries as counts within histogram buckets, whereas, for non-numeric queries it specifies each bucket by a matching rule or a regular expression. A confidence metric quantifies the results' approximation from the sampling and randomization. PrivApprox achieves low latency stream processing and enables a synchronization-free distributed architecture that requires low trust to a central entity. Since it implements a sliding window methodology for infinitely processing series of data sets, it would be purposeful to investigate how to achieve w-event-level privacy protection.

Li et al. [78] attempt to tackle the problem of privacy preservation in numerical data streams taking into account the correlations that may appear continuously among multiple streams and within each one of them. Firstly, the authors define the utility and privacy specifications. The utility of a perturbed data stream is the inverse of the discrepancy between the original and the perturbed measurements. The discrepancy is set as the normalized Forbenius norm, i.e., a matrix norm defined as the square root of the sum of the absolute squares of its elements. Privacy corresponds to the discrepancy between the original and the reconstructed data stream (from the perturbed one), and consists of the removed noise and the error introduced by the reconstruction. Then, correlations come into play. The system continuously monitors the data streams for trends to track correlations and dynamically perturbs the original numerical data while maintaining the trends that are present. More specifically, the Streaming Correlated Additive Noise (SCAN) module updates the estimation of the local principal components of the original data and proportionally distributes noise along the components. Thereafter, the Streaming Correlation Online Reconstruction (SCOR) module removes all the noise by utilizing the best linear reconstruction. SCOR is a representation of the ability of any adversarial entity to post-process the released data and attempt to reconstruct the original data set by filtering out any distortion. Overall, the present technique offers robustness against inference attacks by adapting randomization according to data trends, but fails to efficiently quantify the overall privacy guarantee.

Chen et al. [33] developed *PeGaSus*, an algorithm for event-level differentially private stream processing that supports different categories of stream queries (counts, sliding window, and event monitoring) over multiple stream resolutions. It consists of a Perturber, a Grouper, and a Smoother modules. The Perturber consumes the incoming data stream, adds noise ε_p, which is part of the available privacy budget ε to each data item and outputs a stream of noisy data. The data-adaptive Grouper consumes the original stream and partitions the data into well-approximated regions using, also part of the available privacy budget, ε_g. Finally, a query specific Smoother combines the independent information

produced by the Perturber and the Grouper, and performs post-processing by calculating the final estimates of the Perturber's values for each partition created by the Grouper at each timestamp. The combination of the Perturber and the Grouper follows the sequential composition and post-processing properties of differential privacy, thus, the resulting algorithm satisfies $(\varepsilon_p + \varepsilon_g)$-differential privacy.

3.3 Discussion

In the previous sections, we provided a review for each work that falls into the categories of microdata and statistical data privacy for continuous data publishing. Reviewing the algorithms and positioning them against specific characteristics, as shown in Table 5 (microdata), and Table 6 (statistical data), allow us to make the following observations on each category separately and in general.

In the Microdata category, we observe that problems with sequential data, i.e., data that are generated in a sequence and are dependent on the values of previously released data sets, are more prominent. We also encounter here the publishing of updated versions of an original data set, either vertically (schema-wise) or horizontally (tuple-wise). Naturally, in such cases the most evident attack scenarios are the complementary release ones, as in each release there is great probability that there will be an intersection of tuples with previous releases. In fact, the works based on k-anonymity in this category were designed to address attacks that are specific to the grouping approach, where groups from different timestamps may overlap in some ways. In the case of location or trajectory (which are also sequential data) publishing, many works take into account external information, and more precisely data correlations before publishing the privacy-protected version of the data.

In the Statistical Data section, all works address attacks in their more general form (linkage attacks). We notice that the data linkage is currently assumed in the bibliography as the worst case attack. For this reason, works in the Statistical Data category seem to provide a robust privacy protection solution, independent of adversarial background knowledge. The prevailing distortion operation in this category is probabilistic perturbation. This is justified by the fact that nearly all methods are based on differential privacy. The majority implements mechanisms based on the Laplacian distribution, while some of them design more sophisticated probabilistic mechanisms, depending on the data type or accounting for data dependencies that may lead to extra privacy leakage.

When data dependencies are taken into account, in either category, we observe that the privacy operation used is mainly probabilistic perturbation, if not total suppression. This is logical since by generalization the correlation between attributes would not be canceled. Generalization is used in group-based techniques to make it possible to group more tuples under the generated categories, and thus achieve privacy protection—which would still be open to dependence (and other) attacks.

As far as the publishing mode is concerned, problems with streaming processing are not the most common cases in the Microdata category. Most of the cases that include streaming scenarios are in the Statistical Data category. A technical reason behind this observation is that protecting the privacy of a raw data set as a whole may be a time-consuming process due to size and complexity, and thus not well-suited for streaming. The complexity actually depends on the number of attributes if we consider the possible combinations that may be enumerated for the generalizations. On the contrary, aggregation functions, as used in the Statistical Data category and especially in the absence of filters or other operations on

top of these aggregation functions, usually are low cost. Moreover, perturbing a numerical value (the usual result type of an aggregation function) does not add a lot in the complexity of the algorithm (depending of course on the perturbation model used). For this reason, perturbing the result of a process is more time efficient than protecting the privacy of the original data set itself and then running the aggregation process on the privacy-protected data.

4 Conclusion and open issues

In this survey, we present a comprehensive review of works regarding data privacy-preserving algorithms for continuous data publishing. In such problem settings, location-related data is prevalent due to the establishment of technologies generating continuous geo-tagged data, e.g., connected personal portable devices. We have discussed how the nature of the released data set, i.e., raw microdata or statistical results thereof, dictates the choice of the base computation privacy algorithm to use, i.e., k-anonymity or differential privacy (in the majority of the cases), and the related privacy operations. We have further categorized the relevant works based on the fashion that we observe the data (finite or infinite), and the processing modes and architectures they consider. We have listed the achieved privacy levels and attack models while paying special attention to how data correlations can be exploited by an adversary, something that inevitably appears in continuous data publishing. Our goal is to aid researchers and practitioners in the field to easily identify the proper algorithm(s) to use according to the scenarios that they have at hand, and the nature of the data that they have to process. We culminate this survey by discussing fields for further work on the subject.

Reviewing the two axes (microdata and statistical data) of privacy-preserving mechanisms side by side, we observe the prevalence of techniques based on randomization and probabilistic distributions. This prevalence is justified even more in the context of continuous data publishing because in these scenarios data might involve correlations and generate background knowledge that can be used for further privacy leakage. For the same reasons, we have observed that over the last years researchers tend to take into account correlations more than before. Certainly, there is space for progress in this field since several works mention correlations quite vaguely without computing the extra privacy loss due to the correlations or take into account only one type of correlations. More particularly, defining the appropriate privacy expectation is of critical importance. Indeed, generalizing the idea of exploiting linkage or correlations among various sources of data and privacy-protected data sets introduces a series of open research problems.

From a different perspective, we have observed that most of the existing literature emphasizes on the effectiveness of the proposed methods by focusing on their privacy guarantees and/or their impact on the data quality. Given that quantifying and balancing these remains a difficult (and in many cases application dependent) problem, the interest in this area is expected to increase in the future. In general, differential privacy guarantees an upper bound of privacy loss from the released output. However, its quality depends not only on the given privacy budget but also on the query, and the mechanism itself. Similarly for k-anonymity, k defines the worst-case protection of an individual and the quality is usually quantified by either the discrepancy between the privacy-protected and the original data set or by the difference in the utility of the two data sets. In our area of interest, we have

come across some works that employ methods to enhance the quality of the distorted answers, e.g., by smoothing the results or by specializing the originally generalized attributes (the latter being microdata specific). Further work can be done in this direction to make the results more useful in practice while not degrading the level of privacy protection. Another direction that should be more actively considered is the algorithmic efficiency, a parameter that is hardly discussed but is vital for continuous data publishing, especially for streaming scenarios, where performance is always an inherent part of the solution. Finally, comparing the techniques reviewed in Section 3.1—Microdata with the ones in Section 3.2—Statistical Data, and as observed in Table 5 and Table 6, the latter are more common for data that are generated in streaming mode and need (near) real-time processing.

Last but not least, we observe that little work has been done that combines the two main methods, i.e., k-anonymity and its derivatives, and differential privacy. In general, k-anonymity provides the possibility of sharing a whole data set, which is important in research for experimental evaluation. Nevertheless, differential privacy provides more robust privacy guarantees, especially in the presence of external information. While we acknowledge their applicability to different types of data and problems, it would be challenging and interesting to see how these methods could be integrated into a common framework to provide case- and user-specific data privacy solutions. For example, such a dynamic framework would decide, either automatically or by manual tuning, the privacy-preserving algorithm to use (or a combination thereof) based on the attributes or application domains, on the user privacy requirements and on the likelihood of finding external sources of information.

References

[1] Acxiom. https://www.acxiom.com/. Last Accessed December 1, 2019.

[2] Experian. https://www.experian.com/. Last Accessed December 1, 2019.

[3] Facebook. https://www.facebook.com/. Last Accessed January 31, 2009.

[4] Foursquare. https://www.foursquare.com/. Last Accessed December 1, 2019.

[5] Google maps. https://maps.google.com/. Last Accessed December 1, 2019.

[6] Openstreetmap. https://www.openstreetmap.org/. Last Accessed December 1, 2019.

[7] Transunion. https://www.transunion.com/. Last Accessed December 1, 2019.

[8] Twitter. https://www.twitter.com/. Last Accessed December 1, 2019.

[9] Waze. https://www.waze.com/. Last Accessed December 1, 2019.

[10] Wikipedia. https://www.wikipedia.org/. Last Accessed December 1, 2019.

[11] The world's most valuable resource is no longer oil, but data. https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data, 2016. Last Accessed December 1, 2019.

[12] ABUL, O., BONCHI, F., NANNI, M., ET AL. Never walk alone: Uncertainty for anonymity in moving objects databases. In *IEEE 24th International Conference on Data Engineering* (2008), vol. 8, pp. 376–385. doi:10.1109/icde.2008.4497446.

[13] AL-DHUBHANI, R., AND CAZALAS, J. M. An adaptive geo-indistinguishability mechanism for continuous LBS queries. *Wireless Networks 24*, 8 (2018), 3221–3239. doi:10.1007/s11276-017-1534-x.

[14] ANDRÉS, M. E., BORDENABE, N. E., CHATZIKOKOLAKIS, K., AND PALAMIDESSI, C. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (2013), ACM, pp. 901–914. doi:10.1145/2508859.2516735.

[15] ANSELIN, L. Local indicators of spatial association–LISA. *Geographical analysis 27*, 2 (1995), 93–115. doi:10.1111/j.1538-4632.1995.tb00338.x.

[16] BAUM, L. E., AND PETRIE, T. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics 37*, 6 (1966), 1554–1563. doi:10.1214/aoms/1177699147.

[17] BAYARDO, R. J., AND AGRAWAL, R. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on* (2005), IEEE, pp. 217–228. doi:https://doi.org/10.1109/icde.2005.42.

[18] BENALOH, J., CHASE, M., HORVITZ, E., AND LAUTER, K. Patient controlled encryption: ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM workshop on Cloud computing security* (2009), ACM, pp. 103–114. doi:10.1145/1655008.1655024.

[19] BITTAU, A., ERLINGSSON, Ú., MANIATIS, P., MIRONOV, I., RAGHUNATHAN, A., LIE, D., RUDOMINER, M., KODE, U., TINNES, J., AND SEEFELD, B. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles* (2017), ACM, pp. 441–459. doi:10.1145/3132747.3132769.

[20] BLOCKI, J., BLUM, A., DATTA, A., AND SHEFFET, O. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science* (2013), ACM, pp. 87–96. doi:10.1145/2422436.2422449.

[21] BLOOM, B. H. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM 13*, 7 (1970), 422–426. doi:10.1145/362686.362692.

[22] BOLOT, J., FAWAZ, N., MUTHUKRISHNAN, S., NIKOLOV, A., AND TAFT, N. Private decayed predicate sums on streams. In *Proceedings of the 16th International Conference on Database Theory* (2013), ACM, pp. 284–295. doi:10.1145/2448496.2448530.

[23] CAO, N., WANG, C., LI, M., REN, K., AND LOU, W. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on parallel and distributed systems 25*, 1 (2014), 222–233. doi:10.1109/infcom.2011.5935306.

[24] CAO, Y., AND YOSHIKAWA, M. Differentially private real-time data release over infinite trajectory streams. In *Mobile Data Management (MDM), 2015 16th IEEE International Conference on* (2015), vol. 2, IEEE, pp. 68–73. doi:10.1109/mdm.2015.15.

[25] CAO, Y., YOSHIKAWA, M., XIAO, Y., AND XIONG, L. Quantifying differential privacy under temporal correlations. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on* (2017), IEEE, pp. 821–832. doi:10.1109/icde.2017.132.

[26] CAO, Y., YOSHIKAWA, M., XIAO, Y., AND XIONG, L. Quantifying differential privacy in continuous data release under temporal correlations. *IEEE Transactions on Knowledge and Data Engineering 31*, 7 (2018), 1281–1295. doi:10.1109/tkde.2018.2824328.

[27] CHAN, T.-H. H., SHI, E., AND SONG, D. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC) 14*, 3 (2011), 26. doi:10.1145/2043621.2043626.

[28] CHATZIKOKOLAKIS, K., ELSALAMOUNY, E., PALAMIDESSI, C., AND ANNA, P. Methods for location privacy: A comparative overview. *Foundations and Trends® in Privacy and Security 1*, 4 (2017), 199–257. doi:10.1561/3300000017.

[29] CHATZIKOKOLAKIS, K., PALAMIDESSI, C., AND STRONATI, M. Geo-indistinguishability: A principled approach to location privacy. In *International Conference on Distributed Computing and Internet Technology* (2015), Springer, pp. 49–72. doi:10.1007/978-3-319-14977-6_4.

[30] CHEN, R., ACS, G., AND CASTELLUCCIA, C. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM conference on Computer and communications security* (2012), ACM, pp. 638–649. doi:10.1145/2382196.2382263.

[31] CHEN, R., FUNG, B., AND DESAI, B. C. Differentially private trajectory data publication. *arXiv preprint arXiv:1112.2020* (2011).

[32] CHEN, R., FUNG, B. C., YU, P. S., AND DESAI, B. C. Correlated network data publication via differential privacy. *The VLDB Journal–The International Journal on Very Large Data Bases 23*, 4 (2014), 653–676. doi:10.1007/s00778-013-0344-8.

[33] CHEN, Y., MACHANAVAJJHALA, A., HAY, M., AND MIKLAU, G. PeGaSus: Data-adaptive differentially private stream processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), ACM, pp. 1375–1388. doi:10.1145/3133956.3134102.

[34] CHOW, C.-Y., AND MOKBEL, M. F. Trajectory privacy in location-based services and data publication. *ACM Sigkdd Explorations Newsletter 13*, 1 (2011), 19–29. doi:10.1145/2031331.2031335.

[35] CHRISTIN, D., REINHARDT, A., KANHERE, S. S., AND HOLLICK, M. A survey on privacy in mobile participatory sensing applications. *Journal of systems and software 84*, 11 (2011), 1928–1946. doi:10.1016/j.jss.2011.06.073.

[36] DE MONTJOYE, Y.-A., HIDALGO, C. A., VERLEYSEN, M., AND BLONDEL, V. D. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports 3* (2013), 1376. doi:10.1038/srep01376.

[37] DWORK, C. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation* (2008), Springer, pp. 1–19. doi:10.1007/978-3-540-79228-4_1.

[38] DWORK, C., MCSHERRY, F., NISSIM, K., AND SMITH, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference* (2006), Springer, pp. 265–284. doi:doi.org/10.1007/11681878_14.

[39] DWORK, C., NAOR, M., PITASSI, T., AND ROTHBLUM, G. N. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing* (2010), ACM, pp. 715–724. doi:10.1145/1806689.1806787.

[40] DWORK, C., NAOR, M., PITASSI, T., ROTHBLUM, G. N., AND YEKHANIN, S. Pan-private streaming algorithms. In *ICS* (2010), pp. 66–80.

[41] DWORK, C., AND ROTH, A. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science 9*, 3–4 (2014), 211–407. doi:10.1561/0400000042.

[42] EFTHYMIOU, V., STEFANIDIS, K., AND CHRISTOPHIDES, V. Big data entity resolution: From highly to somehow similar entity descriptions in the web. In *2015 IEEE International Conference on Big Data (Big Data)* (2015), IEEE, pp. 401–410. doi:10.1109/bigdata.2015.7363781.

[43] ERDOGDU, M. A., AND FAWAZ, N. Privacy-utility trade-off under continual observation. In *IEEE International Symposium on Information Theory (ISIT)* (2015), pp. 1801–1805. doi:10.1109/isit.2015.7282766.

[44] ERLINGSSON, Ú., PIHUR, V., AND KOROLOVA, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security* (2014), ACM, pp. 1054–1067. doi:10.1145/2660267.2660348.

[45] FAN, L., AND XIONG, L. An adaptive approach to real-time aggregate monitoring with differential privacy. *IEEE Transactions on Knowledge and Data Engineering 26*, 9 (2014), 2094–2106. doi:10.1109/tkde.2013.96.

[46] FAN, L., XIONG, L., AND SUNDERAM, V. Differentially private multi-dimensional time series release for traffic monitoring. In *IFIP Annual Conference on Data and Applications Security and Privacy* (2013), Springer, pp. 33–48. doi:10.1007/978-3-642-39256-6_3.

[47] FIORE, M., KATSIKOULI, P., ZAVOU, E., CUNCHE, M., FESSANT, F., HELLO, D. L., AIVODJI, U. M., OLIVIER, B., QUERTIER, T., AND STANICA, R. Privacy of trajectory micro-data: a survey. *arXiv preprint arXiv:1903.12211* (2019).

[48] FUNG, B., WANG, K., CHEN, R., AND YU, P. Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys* (2010). doi:10.1145/1749603.1749605.

[49] FUNG, B., WANG, K., FU, A. W.-C., AND PEI, J. Anonymity for continuous data publishing. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology* (2008), ACM, pp. 264–275. doi:10.1145/1353343.1353378.

[50] GAGNIUC, P. A. *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017. doi:10.1002/9781119387596.

[51] GANTA, S. R., KASIVISWANATHAN, S. P., AND SMITH, A. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), ACM, pp. 265–273. doi:10.1145/1401890.1401926.

[52] GHINITA, G., DAMIANI, M. L., SILVESTRI, C., AND BERTINO, E. Preventing velocity-based linkage attacks in location-aware applications. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2009), ACM, pp. 246–255. doi:10.1145/1653771.1653807.

[53] GOLDREICH, O. Secure multi-party computation. *Manuscript. Preliminary version 78* (1998).

[54] GÖTZ, M., NATH, S., AND GEHRKE, J. Maskit: Privately releasing user context streams for personalized mobile applications. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (2012), ACM, pp. 289–300. doi:10.1145/2213836.2213870.

[55] GRÜNWALD, P. D. *The minimum description length principle*. MIT press, 2007. doi:10.7551/mitpress/4643.001.0001.

[56] GUALTIERI, M., CURRAN, R., KISKER, H., AND MILLER, E. Perishable insights—stop wasting money on unactionable analytics, 2016.

[57] GURSOY, M. E., LIU, L., TRUEX, S., AND YU, L. Differentially private and utility preserving publication of trajectory data. *IEEE Transactions on Mobile Computing* (2018). doi:10.1109/tmc.2018.2874008.

[58] GUT, A. *Probability: a graduate course*, vol. 75. Springer Science & Business Media, 2013. doi:10.1007/978-1-4614-4708-5.

[59] HE, X., CORMODE, G., MACHANAVAJJHALA, A., PROCOPIUC, C. M., AND SRIVASTAVA, D. DPT: differentially private trajectory synthesis using hierarchical reference systems. *Proceedings of the VLDB Endowment 8*, 11 (2015), 1154–1165. doi:10.14778/2809974.2809978.

[60] HE, Y., BARMAN, S., AND NAUGHTON, J. Preventing equivalence attacks in updated, anonymized data. In *IEEE 27th International Conference on Data Engineering* (2011), IEEE, pp. 529–540. doi:10.1109/icde.2011.5767924.

[61] HUA, J., GAO, Y., AND ZHONG, S. Differentially private publication of general time-serial trajectory data. In *Computer Communications (INFOCOM), 2015 IEEE Conference on* (2015), IEEE, pp. 549–557. doi:10.1109/infocom.2015.7218422.

[62] JAIN, P., GYANCHANDANI, M., AND KHARE, N. Big data privacy: a technological perspective and review. *Journal of Big Data 3*, 1 (2016), 25. doi:10.1186/s40537-016-0059-y.

[63] JI, Z., LIPTON, Z. C., AND ELKAN, C. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584* (2014).

[64] JIANG, K., SHAO, D., BRESSAN, S., KISTER, T., AND TAN, K.-L. Publishing trajectories with differential privacy guarantees. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management* (2013), ACM, p. 12. doi:10.1145/2484838.2484846.

[65] JOHNSON, N., NEAR, J. P., AND SONG, D. Towards practical differential privacy for SQL queries. *Proceedings of the VLDB Endowment 11*, 5 (2018), 526–539. doi:10.1145/3187009.3177733.

[66] KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering 82*, 1 (1960), 35–45. doi:10.1115/1.3662552.

[67] KAMARA, S., AND LAUTER, K. Cryptographic cloud storage. In *International Conference on Financial Cryptography and Data Security* (2010), Springer, pp. 136–149. doi:10.1007/978-3-642-14992-4_13.

[68] KATSOMALLOS, M., LALIS, S., PAPAIOANNOU, T., AND THEODORAKOPOULOS, G. An open framework for flexible plug-in privacy mechanisms in crowd-sensing applications. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on* (2017), IEEE, pp. 237–242. doi:10.1109/percomw.2017.7917564.

[69] KELLARIS, G., AND PAPADOPOULOS, S. Practical differential privacy via grouping and smoothing. In *Proceedings of the VLDB Endowment* (2013), vol. 6, VLDB Endowment, pp. 301–312. doi:10.14778/2535573.2488337.

[70] KELLARIS, G., PAPADOPOULOS, S., XIAO, X., AND PAPADIAS, D. Differentially private event sequences over infinite streams. *Proceedings of the VLDB Endowment 7*, 12 (2014), 1155–1166. doi:10.14778/2732977.2732989.

[71] KIFER, D. Attacks on privacy and deFinetti's theorem. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data* (2009), ACM, pp. 127–138. doi:10.1145/1559845.1559861.

[72] KIFER, D., AND MACHANAVAJJHALA, A. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data* (2011), ACM, pp. 193–204. doi:10.1145/1989323.1989345.

[73] KIFER, D., AND MACHANAVAJJHALA, A. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS) 39*, 1 (2014), 3. doi:10.1145/2514689.

[74] KING, J. L. Centralized versus decentralized computing: organizational considerations and management options. *ACM Computing Surveys (CSUR) 15*, 4 (1983), 319–349. doi:10.1145/289.290.

[75] LAFFERTY, J., MCCALLUM, A., AND PEREIRA, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)* (2001), pp. 282–289.

[76] LEE, J., AND CLIFTON, C. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security* (2011), Springer, pp. 325–340. doi:10.1007/978-3-642-24861-0_22.

[77] LEGENDRE, P. Spatial autocorrelation: trouble or new paradigm? *Ecology 74*, 6 (1993), 1659–1673. doi:10.2307/1939924.

[78] LI, F., SUN, J., PAPADIMITRIOU, S., MIHAILA, G. A., AND STANOI, I. Hiding in the crowd: Privacy preservation on evolving streams through correlation tracking. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on* (2007), IEEE, pp. 686–695. doi:10.1109/icde.2007.367914.

[79] LI, J., BAIG, M. M., SATTAR, A. S., DING, X., LIU, J., AND VINCENT, M. W. A hybrid approach to prevent composition attacks for independent data releases. *Information Sciences 367* (2016), 324–336. doi:10.1016/j.ins.2016.05.009.

[80] LI, M., ZHU, L., ZHANG, Z., AND XU, R. Achieving differential privacy of trajectory data publishing in participatory sensing. *Information Sciences 400* (2017), 1–13. doi:10.1016/j.ins.2017.03.015.

[81] LI, N., LI, T., AND VENKATASUBRAMANIAN, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on* (2007), IEEE, pp. 106–115. doi:10.1109/icde.2007.367856.

[82] LIU, C., CHAKRABORTY, S., AND MITTAL, P. Dependence makes you vulnerable: Differential privacy under dependent tuples. In *Network and Distributed System Security Symposium* (2016), vol. 16, pp. 21–24. doi:10.14722/ndss.2016.23279.

[83] LYON, D. Surveillance, snowden, and big data: Capacities, consequences, critique. *Big Data & Society 1*, 2 (2014), 2053951714541861. doi:10.1177/2053951714541861.

[84] MA, Q., ZHANG, S., ZHU, T., LIU, K., ZHANG, L., HE, W., AND LIU, Y. PLP: Protecting location privacy against correlation analyze attack in crowdsensing. *IEEE transactions on mobile computing 16*, 9 (2017), 2588–2598. doi:10.1109/tmc.2016.2624732.

[85] MACHANAVAJJHALA, A., GEHRKE, J., KIFER, D., AND VENKITASUBRAMANIAM, M. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on* (2006), IEEE, pp. 24–24. doi:10.1109/icde.2006.1.

[86] MCSHERRY, F., AND TALWAR, K. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on* (2007), IEEE, pp. 94–103. doi:10.1109/focs.2007.66.

[87] MCSHERRY, F. D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data* (2009), ACM, pp. 19–30. doi:10.1145/1559845.1559850.

[88] MORAN, P. A. Notes on continuous stochastic phenomena. *Biometrika 37*, 1/2 (1950), 17–23. doi:10.2307/2332142.

[89] MOTWANI, R., AND XU, Y. Efficient algorithms for masking and finding quasi-identifiers. In *Proceedings of the Conference on Very Large Data Bases (VLDB)* (2007), pp. 83–93.

[90] NARAYANAN, A., AND SHMATIKOV, V. Robust de-anonymization of large sparse data sets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on* (2008), IEEE, pp. 111–125. doi:10.1109/sp.2008.33.

[91] PARK, K. I. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer, 2018. doi:10.1007/978-3-319-68075-0.

[92] PRIMAULT, V., BOUTET, A., MOKHTAR, S. B., AND BRUNIE, L. The long road to computational location privacy: A survey. *IEEE Communications Surveys & Tutorials* (2018). doi:10.1109/comst.2018.2873950.

[93] PRIMAULT, V., MOKHTAR, S. B., LAURADOUX, C., AND BRUNIE, L. Time distortion anonymization for the publication of mobility data with high utility. In *Trustcom/BigDataSE/ISPA, 2015 IEEE* (2015), vol. 1, IEEE, pp. 539–546. doi:10.1109/trustcom.2015.417.

[94] QUINLAN, J. R. *Programs for machine learning*. Elsevier, 2014.

[95] QUOC, D. L., BECK, M., BHATOTIA, P., CHEN, R., FETZER, C., AND STRUFE, T. PrivApprox: privacy-preserving stream analytics. In *Proceedings of the 2017 USENIX Conference on Usenix Annual Technical Conference* (2017), USENIX Association, pp. 659–672. doi:10.1007/s00287-019-01206-w.

[96] ROGERS, L. C. G., AND WILLIAMS, D. *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*, vol. 2. Cambridge university press, 2000. doi:10.1017/cbo9781107590120.

[97] SATYANARAYANAN, M. The emergence of edge computing. *Computer 50*, 1 (2017), 30–39. doi:10.1109/mc.2017.9.

[98] SHANNON, C. E. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review 5*, 1 (2001), 3–55. doi:10.1145/584091.584093.

[99] SHMUELI, E., AND TASSA, T. Privacy by diversity in sequential releases of databases. *Information Sciences 298* (2015), 344–372. doi:10.1016/j.ins.2014.11.005.

[100] SHMUELI, E., TASSA, T., WASSERSTEIN, R., SHAPIRA, B., AND ROKACH, L. Limiting disclosure of sensitive data in sequential releases of databases. *Information Sciences 191* (2012), 98–127. doi:10.1016/j.ins.2011.12.020.

[101] SIMI, M. S., NAYAKI, K. S., AND ELAYIDOM, M. S. An extensive study on data anonymization algorithms based on k-anonymity. In *IOP Conference Series: Materials Science and Engineering* (2017), vol. 225, IOP Publishing, p. 012279. doi:10.1088/1757-899x/225/1/012279.

[102] SKOROKHOD, V. *Basic principles and applications of probability theory*. Springer Science & Business Media, 2005.

[103] SONG, S., WANG, Y., AND CHAUDHURI, K. Pufferfish privacy mechanisms for correlated data. In *Proceedings of the 2017 ACM International Conference on Management of Data* (2017), ACM, pp. 1291–1306. doi:10.1145/3035918.3064025.

[104] SORIA-COMAS, J., AND DOMINGO-FERRER, J. Big data privacy: challenges to privacy principles and models. *Data Science and Engineering 1*, 1 (2016), 21–28. doi:10.1007/s41019-015-0001-x.

[105] STIGLER, S. M. Francis Galton's account of the invention of correlation. *Statistical Science* (1989), 73–79. doi:10.1214/ss/1177012580.

[106] SWEENEY, L. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10*, 5 (2002), 571–588. doi:10.1142/s021848850200165x.

[107] SWEENEY, L. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10*, 5 (2002), 557–570. doi:10.1142/s0218488502001648.

[108] TANKARD, C. What the GDPR means for businesses. *Network Security 2016*, 6 (2016), 5–8. doi:10.1016/s1353-4858(16)30056-3.

[109] TOBLER, W. R. A computer movie simulating urban growth in the Detroit region. *Economic geography 46*, sup1 (1970), 234–240. doi:10.2307/143141.

[110] WANG, H., AND XU, Z. CTS-DP: publishing correlated time-series data via differential privacy. *Knowledge-Based Systems 122* (2017), 167–179. doi:10.1016/j.knosys.2017.02.004.

[111] WANG, J., LUO, Y., ZHAO, Y., AND LE, J. A survey on privacy preserving data mining. In *Database Technology and Applications, 2009 First International Workshop on* (2009), IEEE, pp. 111–114.

[112] WANG, K., AND FUNG, B. Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), ACM, pp. 414–423. doi:10.1145/1150402.1150449.

[113] WANG, Q., ZHANG, Y., LU, X., WANG, Z., QIN, Z., AND REN, K. RescueDP: Real-time spatio-temporal crowd-sourced data publishing with differential privacy. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on* (2016), IEEE, pp. 1–9. doi:10.1109/infocom.2016.7524458.

[114] WARNER, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association 60*, 309 (1965), 63–69. doi:10.1080/01621459.1965.10480775.

[115] WEI, W. W. Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*. 2006. doi:10.1093/oxfordhb/9780199934898.013.0022.

[116] XIAO, X., AND TAO, Y. M-invariance: towards privacy preserving re-publication of dynamic data sets. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (2007), ACM, pp. 689–700. doi:10.1145/1247480.1247556.

[117] XIAO, Y., AND XIONG, L. Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), ACM, pp. 1298–1309. doi:10.1145/2810103.2813640.

[118] XIAO, Y., XIONG, L., ZHANG, S., AND CAO, Y. LocLok: location cloaking with differential privacy via hidden Markov model. *Proceedings of the VLDB Endowment 10*, 12 (2017), 1901–1904. doi:10.14778/3137765.3137804.

[119] YE, A., LI, Y., XU, L., LI, Q., AND LIN, H. A trajectory privacy-preserving algorithm based on road networks in continuous location-based services. In *2017 IEEE Trustcom/BigDataSE/ICESS* (2017), IEEE, pp. 510–516. doi:10.1109/trustcom/bigdatase/icess.2017.278.

[120] ZHANG, J., CORMODE, G., PROCOPIUC, C. M., SRIVASTAVA, D., AND XIAO, X. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS) 42*, 4 (2017), 25. doi:10.1145/3134428.

[121] ZHAO, J., ZHANG, J., AND POOR, H. V. Dependent differential privacy for correlated data. In *Globecom Workshops (GC Wkshps), 2017 IEEE* (2017), IEEE, pp. 1–7. doi:10.1109/glocomw.2017.8269219.

[122] ZHOU, B., HAN, Y., PEI, J., JIANG, B., TAO, Y., AND JIA, Y. Continuous privacy preserving publishing of data streams. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology* (2009), ACM, pp. 648–659. doi:10.1145/1516360.1516435.

[123] ZHOU, B., PEI, J., AND LUK, W. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM SIGKDD Explorations Newsletter 10*, 2 (2008), 12–22. doi:10.1145/1540276.1540279.