Università di Pisa
Facoltà di Scienze Matematiche Fisiche e Naturali

Corso di laurea specialistica in Tecnologie Informatiche
Tesi Specialistica

Privacy by Design in Distributed Mobility Data

Candidato:
Francesca Pratesi

Relatori:
Prof. Dino Pedreschi
Dr. Anna Monreale
Controrelatore:
Prof. Maurizio Bonuccelli

Anno Accademico 2011/2012

Abstract

Movement data are sensitive, because people's whereabouts may allow re-identification of individuals in a de-identified database and thus can potentially reveal intimate personal traits, such as religious or sexual preferences. In this thesis, we focus on a distributed setting in which movement data from individual vehicles are collected and aggregated by a centralized station. We propose a novel approach to privacy-preserving analytical processing within such a distributed setting, and tackle the problem of obtaining aggregated traffic information while preventing privacy leakage from data collection and aggregation. We study and analyze three different solutions based on the differential privacy model and on sketching techniques for efficient data compression. Each solution achieves different a trade-off between privacy protection and utility of the transformed data. Using real-life data, we demonstrate the effectiveness of our approaches in terms of data utility preserved by the data transformation, thus bringing empirical evidence to the fact that the "*privacy-by-design*" paradigm in *big data* analysis has the potential of delivering high data protection combined with high quality even in massively distributed techno-social systems.

Riassunto

I dati di mobilità sono da considerarsi dati sensibili, perché la conoscenza dei luoghi visitati può permettere la re-identificazione degli individui anche in un database privato degli identificatori, rivelando caratteristiche potenzialmente intime e personali, come la religione o le preferenze sessuali. In questa tesi ci concentriamo su un ambiente distribuito in cui i movimenti di veicoli sono raccolti e aggregati da una stazione centrale. Proponiamo infatti un approccio per l'elaborazione di analisi che preservi la privacy in un ambiente distribuito, affrontando cioè il problema di ottenere informazione aggregata sul traffico evitando al contempo perdite di privacy dovute alle fasi di raccolta e di elaborazione. Analizziamo tre soluzioni diverse, tutte basate sul modello della differential privacy e su tecniche di sketching per la compressione dei dati. Ogni soluzione permette di ottenere un diverso bilanciamento tra protezione della privacy individuale e utilità dei dati trasformati. Valutiamo inoltre l'efficacia delle nostre soluzioni in termini di mantenimento dell'utilità usando dati della vita reale, fornendo una dimostrazione empirica del fatto che il paradigma di "*privacy-by-design*" nell'analisi di *big data* riesce sia a fornire un'elevata protezione che a mantenere una buona qualità dei dati, anche in sistemi sociali fortemente distribuiti.

Contents

Introduction

Over the last few years, the technique of analysis and knowledge discovery, that allow the extraction of valuable knowledge from databases, have become increasingly central. These processes acquired great importance thanks to the availability of a large and ever-growing quantity of data, that are usually provided by users while using different kinds of services. These data are more and more complex, and they are called *big data*, to summarize their main intrinsic characteristics: the data are very large and have a very fine level of detail, making it harder to perform analyses. In addition, data are rarely available in a single or few centralized structures, but often they are distributed among users, so it is necessary to find a way to gather them. On the other hand, *big data* offer many new opportunities to understand our society because they describe in detail the activities of the population.

Example of complex and *big data* are: the traces of the goods purchased by people, stored by automatic payment systems; the query-logs, stored by search engines; the information, held by social networks, about the personal relationships as friendships, partnerships, etc...

Often in our society many decisions are taken based on the knowledge represented in these datasets; therefore, sophisticated techniques for analysis have been developed, to have the opportunity to gather, save and analyze more and more complex data. These techniques are able to extract patterns, models, profiles and general rules that describe the behavior of a community. Indeed, through the analyses of personal data with sophisticated tools, we have created new chances for understanding complex phenomena, such as to comprehend the mobility in an urban area, and to foresee the diffusion of an economic crisis or the spread of epidemics and viruses.

In this thesis we consider movement data describing the mobility behavior of a population in a territory. The widespread availability of low

cost GPS devices enables the collection of these data at a large scale. Understanding of the human mobility behavior in a city could be extremely useful to improve the use of city space and accessibility of various places and utilities, to manage the traffic network, and to reduce traffic jams. Generalization and aggregation of individual movement data can provide an overall description of traffic flows in a given time interval and their variation over time. Intuitively, movement data of multiple individual devices can be collected and aggregated by a central station. However, this centralized setting entails two important problems: a) the amount of information to be collected and processed may exceed the capacity of the storage and computational resources; and b) the raw data describe the mobility behavior of the individuals in such great detail that they could enable the inference of very sensitive information related to the private personal sphere.

Some recent works [63, 41, 12] have investigated how to aggregate distributed mobility data efficiently. For instance, Andrienko et al. [12] propose a method for generalization and aggregation of movement data that requires all individual moving trajectories be transformed into aggregate flows between areas. Though these works consider releasing statistic information instead of raw trajectories to the central station, there still may exist privacy leakage. For instance, the analyses of low-density aggregate traffic flows (e.g., in rural areas) may still reveal the identity of the vehicles involved in these flows.

In order to solve these problems, in this thesis we propose a privacy-preserving *distributed* analytical processing framework for the aggregation of movement data. We assume that on-board location devices in vehicles continuously trace the positions of the vehicles and periodically send statistical information about their movements to a central station. The central station, which we call *coordinator*, will store the received statistical information and compute a summary of the traffic conditions of the whole territory, based on the information collected from individual vehicles. Since the coordinator can be untrusted, we design privacy-preserving methods for each individual participant vehicle that provide formal privacy guarantee, meaning that the statistic information revealed to the coordinator will not be swayed too much by whether or not a specific individual participant. The basic idea behind our approach is that even radical forms of data randomization, capable of yielding strong protection of personal mobility data for

each participant vehicle, can be adopted in our setting while still allowing a correct reconstruction of aggregated traffic information on the coordinator side. The results presented in this thesis show how the application of the *privacy-by-design* paradigm in this complex system, characterized by highly distributed *big data*, allow us to maintain under control the utility of data, with the aim to perform important collective mobility analyses, while providing an high level of protection for each individual by using the differential privacy model.

The Privacy-by-Design Paradigm

One of the most hot topics in the data privacy field has been *Privacy-by-Design*. This concept was coined in the '90s by Ann Cavoukian, the Information and Privacy Commissioner of Ontario, Canada. In brief, *privacy-by-design* refers to the philosophy and approach of embedding privacy into the design, operation and management of information processing technologies and systems. This innovative paradigm is also introduced by the European Commission in the proposal of the reform, on January 25, 2012, of the data protection rules.

Privacy-by-design promises a quality leap in the conflict between data protection and data utility. The principle of "by design" was applied to the data mining domain in [60], where Monreale showed that higher protection and quality can be better achieved in a goal-oriented approach. In such an approach, the data mining process is designed with assumptions about: (a) the sensitive personal data that are the subject of the analysis; (b) the attack model, i.e., the purpose of a malicious party that has an interest in discovering the sensitive data of certain individuals; (c) the category of analytical queries that are to be answered with the data.

These assumptions are fundamental for the design of a privacy-preserving framework for various reasons.
First of all, the techniques for privacy preservation strongly depend on the nature of the data that we want to protect. For example, many proposed methods are suitable for continuous variables but not for categorical variables, while other techniques employed to anonymize sequential data such as tabular data are not appropriate for moving object datasets.
Second, a valid framework for privacy protection has to define the back-

ground knowledge of the adversary, that strongly depends on the context and on the kind of data. Different assumptions on the background knowledge of an attacker entail different defense strategies. Clearly, the assumption that the background knowledge of an adversary depends on the context allows to realize frameworks that guarantee *reasonable* levels of privacy according to the privacy expectation.
Finally, a privacy-preserving strategy should find an acceptable trade-off between data privacy on one side and data utility on the other side. In order to reach this goal it is fundamental to take into account during the design of the framework the analytical questions that are to be answered with the transformed data. This means designing a transformation process capable to preserve some data properties that are necessary to preserve the results obtained by specific analytical and/or mining tasks.

In this thesis, we propose the use of the *privacy-by-design* paradigm in a novel setting, where it is necessary to take into account other important aspects that such as the data distribution and the communications from the nodes and the central station. In particular, in a distributed context, a suitable privacy-preserving framework must try to reduce the amount of information to be transmitted. Clearly, to this end we can use summarization techniques, but we have to pay attention because these techniques can introduce further approximation on the data that could lead to a more degradation on the data utility. As a consequence, the distributed setting adds a novel challenge in the application of the *privacy-by-design* paradigm; here, a valid privacy-aware framework has to keep under control the trade-off among three important aspects: privacy protection, data quality and performance of the overall system.

Contribution and Organization of the Thesis

The main question addressed in this thesis is the following:

How to design a privacy-preserving framework for distributed mobility data analytics

- *while guaranteeing high level of individual privacy*
- *while reducing the amount of information to be transmitted, and*
- *without sacrificing the quality of data utility?*

Transforming the data in such a way as to protect sensitive information is increasingly hard but our belief is that the research results reported in this thesis brings evidence to the fact that the "*privacy-by-design*" paradigm in *big data* analytics has the potential of delivering high data protection combined with high quality even in massively distributed techno-social systems. With a clear analytical goal to realize, e.g., the continuous monitoring of traffic flows, it is possible to design a privacy-preserving process that, as in our study, solves the problem delivering results with a bounded (small) quality-loss within a framework where the risk of privacy leakage is also bounded (and very small). The validity of our privacy-preserving framework is shown both by theoretical results and by a deep experimentation on real-life data.

We have the following contributions. First, to protect individual privacy, we propose three data transformation methods based on the well-known differential privacy model; each solution is characterized by a different trade-off between privacy and data utility. Second, to further reduce the amount of information that each vehicle communicates to the central station, we propose to apply sketching techniques to the differentially private data to obtain a compressed representation. The central station is able to reconstruct the movement data represented by the sketched data that, although transformed for guaranteeing privacy, preserve some important properties of the original data that make them useful for mobility analyses. We validate the robustness and efficiency of our privacy-preserving data aggregation methods by extensive experiments on large, real GPS data.

The remainder of the thesis is organized as follows.

Chapter 1 and Chapter 2 discuss the most relevant research work related to the contribution of this thesis. Specifically, Chapter 1 presents an overview of the work in literature on the individual privacy protection addressed by the data mining and the statistics community, while in Chapter 2 we describe the system architecture that we use in the work presented in this thesis and the sketching algorithms that are used in our privacy-preserving framework.

Chapter 3 introduces background information and definitions that are very important for the deep understanding of the details of our framework and states the problem addressed in this thesis.

Chapter 4 is the core of the thesis, in fact here we introduce a privacy-

preserving framework for distributed mobility data analytics that guarantees strong individual privacy protection, while preserving the quality of the transformed data. This framework is based on the notion of differential privacy that is a very strong privacy model.

In Chapter 5 we present and discuss experimental results obtained from the application of our methods to real-world data.

Lastly, Chapter 5.4.4 concludes the thesis.

Part of the results of the studies described in this thesis are presented in the following works:

> Anna Monreale, Wendy Hui Wang, Francesca Pratesi, Salvatore Rinzivillo, Dino Pedreschi, Gennady L. Andrienko, and Natalia V. Andrienko. *Privacy-preserving Distributed Movement Data Aggregation.* Accepted for publication in AGILE, 2013.
>
> Anna Monreale, Wendy Hui Wang, Francesca Pratesi, Salvatore Rinzivillo, Dino Pedreschi, Gennady L. Andrienko, and Natalia V. Andrienko. *Differential Privacy in Distributed Mobility Analytics.* Submitted in PVLDB, 2013.

Chapter 1

Privacy-Preserving Data Publishing and Mining

In the last years, the importance of the privacy protection is rising thanks to the availability of large amounts of data. These data collections can be gathered from various channels. Typically, the data collector or data holder releases these data to data miners and analysts who can conduct on them statistical and data mining analyses. The published data collections could contain personal information about users and their individual privacy could be compromised during the analytical process.

In recent years, individual privacy has been one of the most discussed jurisdictional issues in many countries. Citizens are increasingly concerned about what companies and institutions do with their data, and ask for clear positions and policies from both the governments and the data owners. Despite this increasing need, there is not a unified view on privacy laws across countries. The European Union regulates privacy by Directive 95/46/EC (Oct. 24, 1995) and Regulation (EC) No 45/2001 (December 18, 2000). The European regulations, as well as other regulations such as the U.S. rules on protected health information (from HIPAA), are based on the notion of "non-identifiability". The regulation on privacy in the EU was recently revised by the comprehensive reform of the data protection rules proposed on Jan. 25, 2012 by the European Commission, that is still under discussion. The problem of protecting the individual privacy when disclosing information is not trivial and this makes the problem scientifically attractive. It has been studied extensively in two different communities: in data mining,

under the general umbrella of privacy-preserving data mining, and in statistics, under the general umbrella of statistical disclosure control.

In this chapter we provide an overview of the most important results achieved so far in the field of data privacy; we also present a very recent model, called *differential privacy*, that will be one of the most important notions in our work.

1.1 Anonymity by Randomization

Randomization methods are used to modify data to preserve the privacy of sensitive information. These techniques try to "hide" information by randomly perturbing the data [49].
The algorithms belonging to this group of techniques first of all modify the data by using randomization techniques. Then, from the perturbed data it is still possible to extract patterns and models. There are two kinds of randomization: *additive* and *multiplicative*.

By using the additive random perturbation, the distorted dataset is obtained drawing independently, from a probability distribution, a noise quantity and adding it to each record of the original dataset.
The original record values cannot be easily guessed from the distorted data as the variance of the noise distribution is assumed large enough. On the contrary, the distribution of the original data can be easily recovered, subtracting the noise distribution from the distribution of the perturbed dataset. A typical assumption is that both distributions are known: the first one is public and the second one is easily obtainable by analyzing the perturbed data [3]. It is important to note that it is possible to reconstruct only the distribution and not the values of individual records [7].

For privacy-preserving data mining, multiplicative random perturbation techniques can be also used.
There are three macro-categories of multiplicative random perturbation [22]:

- **rotation perturbation.** It refers to the techniques based on the notion of matrix rotation. This category does not include only traditional rotations, but also all orthonormal perturbations. The property of this kind of perturbation is its capability to keep the dimensionality of

dataset unchanged, while preserving both the distance between records and the geometric shapes of data.

- **projection perturbation.** It refers to the technique of projecting a set of data points from a high-dimensional space to a randomly chosen lower-dimensional subspace, but it does not strictly guarantee the preservation of distance/inner product, which may downgrade the model accuracy.

- **sketch-based approach.** It aims at perturbing high-dimensional data (and reducing them). It is very suitable to approximate inner queries and dot-product estimation (see Section 2.2 for further details)

The main advantage of the randomization method is that it can be implemented at data-collection time [4], because it is very simple and does not require knowledge of the distribution of other records in the data for the data transformation. This means that the data transformation process does not need a trusted server containing all the original records.

The problem of the randomization (with the exception of sketches, which can provide a uniform measure across different record [22]) is that it does not consider the local density of the records and thus all records are handled equally. Outlier records can be compared to records in denser regions in the data and this can make an attack easier. Another weakness of a randomization framework is that it does not provide guarantees in case of re-identification attack conducted using public information. Indeed, if an attacker has no background knowledge over the data, then the privacy can be difficult to compromise; nevertheless, in [6], authors showed that the randomization approach is unable to effectively guarantee privacy in high-dimensional cases. Moreover, they provide an analysis revealing that the use of public information makes this method vulnerable.

In [49] Kargupta et al. challenged the effectiveness of randomization methods, showing that the original data matrix can be obtained from the randomized data matrix using a random matrix-based spectral filtering technique.

1.2 Differential Privacy

A recent model of randomization, though based on different assumptions, is *Differential Privacy*. This is a privacy notion introduced in [35] by Dwork. The key idea is that the privacy risks should not increase for a respondent as a result of occurring in a statistical database; differential privacy ensure, in fact, that the ability of an adversary to inflict harm should be essentially the same, independently of whether any individual opts in to, or opts out of, the dataset.

This privacy model is called ϵ-differential privacy, due to the level of privacy guaranteed ϵ. It assures a record owner that any privacy breach will not be a result of participating in the database since anything, or almost nothing, that is learnable from the database with his record is also learnable from the one without his data. Moreover, in [35] is formally proved that ϵ-differential privacy can provide a guarantee against adversaries with arbitrary background knowledge. This strong guarantee is achieved by comparison with and without the record owner's data in the published data.
It is important to note that the parameter ϵ is public [34].

The choice of ϵ is essentially a social question, even if some works (like [54]) tried to suggest how to instantiate it in a practical example.

There are two popular mechanisms to achieve differential privacy: Laplace mechanism that supports queries whose outputs are numerical [36] and exponential mechanism that works for any queries whose output spaces are discrete [57]. The basic idea of the Laplace mechanism is to add noise to aggregate queries (e.g., counts) or queries that can be reduced to simple aggregates. The Laplace mechanism has been widely adopted in many existing works for various data applications. For instance, [82, 26] present methods for minimizing the worst-case error of count queries; [14, 32] consider the publication of data cubes; [45, 84] focus on publishing histograms; and [58, 52] propose the methods of releasing data in a differential private way for data mining. On the other hand, for the analysis whose outputs are not real or make no sense after adding noise, the exponential mechanism selects an output from the output domain, $r \in R$, by taking into consideration its score of a given utility function q in a differentially private manner. It has been applied for the publication of audition results [57], coresets [37],

frequent patterns [18] and decision trees [39]. Recently much attention is paid to distributed private data analysis. In this setting, n parties (each holding some sensitive data) wish to compute some aggregate statistics over all parties' data with or without a centralized coordinator. [15, 19] prove that when computing the sum of all parties' inputs without a central coordinator, any differentially-private multi-party protocol with a small number of rounds and small number of messages must have large error. To the best of our knowledge, Rastogi et al. [69] and Chan et al. [79] were the first ones to consider the problem of privately aggregating sums over multiple time periods. Both of them consider the untrusted coordinator, malicious coordinator in particular, and use both encryption and differential privacy for the design of privacy-preserving data aggregation methods. Compared with their work, we focus on semi-honest coordinator, with the aim of designing privacy-preserving techniques by adding meaningful noises to improve data utility, which is an issue that is rarely discussed in both [69, 79]. We agree that our methods can be further enforced against the malicious coordinator by applying the encryption methods in [69, 79].

There are some works on publishing differentially private spatial data. Chen et al. [24] propose to release a prefix tree of trajectories with injected Laplace noise. Each node in the prefix tree contains a doublet in the form of $< tr(v), c(v) >$,where $tr(v)$ is the set of trajectories of the prefix v, and $c(v)$ is a version of $|tr(v)|$ with Laplace noise. Compared with our system, the prefix tree in [24] is *data-dependent*, i.e., it should have a different structure when the underlying database changes. In our work, the frequency vector is data-independent. Cormode et al. present a solution to publish differentially private spatial index (e.g., quadtrees and kd-trees) to provide a private description of the data distribution [26]. Its main utility concern is the accuracy of multi-dimensional range queries (e.g., how many individuals fall within a given region). Therefore, the spatial index only stores the count of a specific spatial decomposition. It does not store the movement information (e.g., how many individuals move from location i to location j) as in our work. In another paper, Cormode et al. [27] propose to publish a contingency table of trajectory data. The contingency table can be indexed by specific locations so that each cell in the table contains the number of people who commute from the given source to the given destination. The contingency table is very similar to our frequency vector structure. How-

ever, [27] has a different focus from ours: we investigate how to publish the frequency vector in a differential privacy way, while [27] addresses the sparsity issue of the contingency table and presents a method of releasing a compact summary of the contingency table with Laplace noise.

1.3 Anonymity by Indistinguishability

When the requirement of performing the data transformation at collection-time is not necessary, a good choice is to apply methods that reduce the probability of record identification by public information. In literature three techniques have been proposed: k-anonymity, l-diversity and t-closeness.

k-anonymity. One approach to preserve privacy in data publishing is the suppression of some of the data values, while releasing the remaining data values exactly. However, suppressing only the identifying attributes is not enough to protect privacy because other kinds of attributes, that are available in public database, such as age, zip-code and sex can be used in order to accurately identify the records. This kind of attributes are known as quasi-identifiers [80]. For example, in [81] it has been observed that for 87% of the population in the United States, the combination of Zip Code, Gender and Date of Birth corresponded to a unique person.
Therefore, it is evident that it is possible to use information derived from different sources (e.g., by database cross-reference) to obtain additional knowledge. We can define the *linking attack* as an attack in which an intruder (attacker) gains access to a database of personal data, in order to make inferences on the basis of background knowledge which enables the re-identification of the user(s).
The goal of k-anonymity is to guarantee that every individual object is hidden in a crowd of size k. A dataset satisfies the property of k-anonymity if each released record has at least (k-1) other records also visible in the release whose values are indistinct over the quasi-identifiers.
In k-anonymity techniques, methods such as generalization and suppression are usually employed to reduce the granularity of representation of quasi-identifiers. The first one generalizes the attribute values to a range in order to reduce the granularity of representation (e.g., a city could be generalized to its region). The method of suppression, instead, removes the value of an

attribute.

***l*-diversity.** Unfortunately, the k-anonymity framework can, in some case, be vulnerable. Suppose that we have a k-anonymous dataset containing a group of k entries with the same value for the sensitive attributes. In this case, although the data are k-anonymous, the value of the sensitive attributes can be easily inferred (*homogeneity attack*). Another problem happens when an attacker knows information useful to associate some quasi-identifiers with some sensitive attributes. In this case the attacker can reduce the number of possible values of the sensitive attributes (*background knowledge attack*).

In order to eliminate these weaknesses of the k-anonymity, the technique of l-diversity was proposed [55]. The aim of l-diversity is to maintain the diversity of sensitive attributes. In particular, the main idea of this method is that every group of individuals that can be isolated by an attacker should contain at least l well-represented values for a sensitive attribute.

***t*-closeness** l-diversity is insufficient to prevent attacks when the overall distribution is skewed. The attacker can know the global distribution of the attributes and use it to infer the value of sensitive attributes. In this case, the t-closeness [53] method can be used. This technique requires that the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the overall table. The distance between the two distributions should be no more than a threshold t.

1.4 Anonymity in Mobility Data

In Section 1.3 we have already said that, even in simple cases, suppressing the identifier of individuals is not enough for privacy-preserving purposes.

One of the most effective methods that are studied in literature to ensure privacy is k-anonymity; unfortunately, the traditional k-anonymity approach focuses on relational tables.

A big problem of spatio-temporal data is that there is no longer a clear distinction between *qi* (quasi-identifiers) and *sa* (sensitive attributes): a hospital could be a *qi* for some users (e.g., for doctors and nurses this is the workplace), while for all other users it is probably a *sa*. Therefore, protect-

ing private information in this context is a significant challenge.

Many existing works about anonymity of moving objects have been mainly developed in the context of *Location Based Services* (LBS). LBS refer to those information services that deliver differentiated information based on the user's location at the time of the request. Thus, the user location information necessarily appears in a request sent to the service provider [56]. Clearly, also in this context, the k-anonymity is applicable: each user avoids providing his exact location sending to the service provider a generalized area that includes his location and the location of other $k-1$ users. Although the idea is the same as in the tabular case, generally traditional techniques used for tabular datasets cannot be directly applied to this kind of data, so k-anonymity must be adjusted appropriately.

As written in Riboni et al. [71], a possible technique to enforce anonymity in LBS is to generalize precise location data in a request to an area including a set (called anonymity set [68]) of other potential issuers. However, they observe that, since we cannot define quasi-identifiers exactly, a large amount of context data must be generalized in order to enforce anonymity. As a consequence, the granularity of generalized context data released to the service provider could be too coarse to provide the service at an acceptable quality level.
Riboni et al. proposed a combined approach to address the issue of privacy in context awareness [70]. In particular, they use obfuscation of sensitive information and anonymity, generalizing precise location data in a request to an area including an anonymity set of other potential issuers. The key idea is that if even users who did not issue any request are potential issuers with respect to the attacker's external knowledge, then they belong to the anonymity set.

In [17], Bettini et al. introduce the concept of historical k-anonymity, that is based on the spatio-temporal pattern definition (for example, the trip from the home to the workplace), and on the spatio-temporal generalization. Indeed, an attacker may guess that two requests have been issued by the same user, simply relying on proximity of locations or on the fact that requests from the same issuer may be correlated. Given the set of requests issued by a certain user, it satisfies historical k-anonymity if there exist $k-1$ history of locations belonging to $k-1$ different users such that they

are location-time consistent, i.e., undistinguishable.

Another example of k-anonymity in spatial data can be found in Mascetti et al. [56]. Here, if the attacker does not know the generalization function, the generalized location is computed as the Minimum Bounding Rectangle (MBR) of the locations of the users in this set at the time when the request has been issued. Otherwise, the algorithm imposes the partitioning function to be independent from the issuers location, iteratively restricting the areas (then, the anonymity sets) until any of the blocks contains less than k users. A good point is that, if the degree of anonymity desired by each user at the time of a request is not known by the attacker, algorithms remain safe even when different values of k are admitted.

The main weakness of such solutions (not only of these three works, but also in general) is that the scenario assumes the existence of a *Location-aware Trusted Server* (LTS); the LTS receives the LBS requests from the users, it performs the appropriate generalization (also hiding explicitly identifying values), and it forwards the generalized request to the target service provider. A LTS is actually an anonymizer, and the use of anonymizers may not always be practical. Even if it were trusted, as stated in [48], an anonymizer may itself present security, performance, and privacy problems. For example, the anonymizer represents a *single-point-of-attack* for hackers; furthermore, the anonymizer may become a bottleneck because of the large number of users to be served.

A survey of location privacy techniques that work in traditional client-server architectures without any trusted components other than the client's mobile device can be found in the work of Jensen et al. [48]. For instance, in [85] we can find iPDA, an example of query enlargement technique, i.e. a technique where each client enlarges its exact position into a region before sending it to the server. iPDA uses a cloacking technique implemented on the client side, and it is suitable for issuing repeated queries, as in case of mobility data, because it enlarges the region at each request. In [50] users generate several false position data (dummies) to be sent to service providers along with the real position data. So, the service provider cannot distinguish the true position data from the set of all the received position data. The service provider creates service answers that respond to all the received position data and sends them to the user, who selects the true response. Lastly, in [40] we can find a kind of cryptographic transformation,

which uses both a Voronoi and a grid partition. The user finds the cell that contains him, and utilizes Private Information Retrieval to request all points within the region; the server does not know which region was retrieved, as if it received a number of requests equal to the total number of cells.

Although the LBS context is very relevant to the problem of anonymity in spatio-temporal data, there is another kind of problem to be considered: anonymity in a static *Moving Objects Databases* (MOD). The main difference is the fact that LBS consider data points (requests) as continuously arriving, and thus they provide on-line anonymity; instead, in MOD context the information about the whole history of trajectories is available, thus we can use more effective (and off-line) methods. Another difference is in the goal of these two contexts: in LBS we must provide the service, so learning the user's exact position is not a requisite, and the data can also be forgotten once that the service was provided (we can say that LBS is service-centric); whereas in MOD we must preserve not only the anonymity of the individuals, but also the quality of the data (for this reason we say that MOD is data-centric) [1].

We provide a quick overview on works which tackle the problem of k-anonymity of moving objects by the perspective of privacy aware publishing.
In [61], Monreale et al. focus on the choice of granularity of the spatial generalization and especially on the research for a method of division of the territory into sub-areas, that depends directly on the input trajectory dataset. The *privacy-by-design* concept (see Introduction) is widely used in this work.
In [59], Monreale et al. introduce a new privacy notion, called c-safety, which provides an upper bound c to the probability of inferring that a given person, observed in a sequence of non-sensitive places, has also stopped in any sensitive location. They also implement an algorithm, called CAST, which finds the best trajectory grouping in the dataset, constructing a c-safe version of the input dataset.
In [64], Nergiz et al. use a grouping based approach in order to obtain cluster trajectories, but they publish a reconstructed MOD, instead of a generalized one. Indeed, they claim that the use of MBR discloses uncontrolled information about the exact location of the points, so they apply a reconstruction approach (previously studied in the string alignment prob-

lem), which releases atomic trajectories sampled randomly from the area covered by anonymized trajectories.
In [86], Yarovoy et al. deeply analyze the problem of quasi-identifiers in mobility data: they show that the anonymization groups may not be disjoint, thus there may exist objects that can be identified explicitly by combining different anonymization groups. They suggest that *qi* may be provided directly by personal settings or found by means of statistical data analysis.
In [2], Abul et al. propose the notion of (k, δ)-anonymity for moving object databases, where δ represents the possible location imprecision. This is an innovative concept of k-anonymity based on co-localization, which takes advantage of the inherent uncertainty of the whereabouts of the moving objects. The authors also proposed an approach, called Never Walk Alone, based on trajectory clustering and spatial translation, and they present its improvement, Wait for Me, in [1]. This method is very similar to the previous one, but it is based on EDR distance [23] (instead of Euclidean distance), which is time-tolerant, so Wait for Me can recognize similar trajectories even if they are (slightly) shifted in time.
Finally, in [33], Domingo Ferrer and Trujillo-Rasua show a solution based on perturbation and micro-aggregation: this method k-anonymizes each location independently, using the whole set of trajectories. Particularly, the algorithm creates clusters of locations (close in time and in space) in such a way that the locations in each group belong to k different trajectories. The result of this transformation is that the probability that a location of a true trajectory appears in its anonymized version is at most $\frac{1}{k}$, while guaranteeing that the anonymized trajectories are suitable for range query for every value of k.

1.5 Secure Multi-Party Computation

A *Secure Multi-party Computation* (SMC) problem [87, 44] deals with computing a certain function on multiple inputs, in a distributed network. The problem in this case is to compute any probabilistic function on inputs that are distributed among the participants in the network while ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to participants in the computation, which can be computed from a single participant or a coalition of participants.

As noted in [44], a trivial centralized solution would be to assume a trusted center exists, and that all users send their inputs to this trusted center for the computation of their respective outputs. A preferable option is a distributed solution where trust is distributed.
SMC is often used in distributed environment, but regrettably it allows only some kinds of computations.

One of the first techniques is shown in [21], where participants can share secrets, even if one third of the participants deviate from the protocol (that is based on not leaking secret information and on sending the correct messages).
A more recent solution can be found in [42], where Gilburd et al. propose a new privacy model, k-privacy, for real-world large-scale distributed systems. They use a relaxed privacy model implementing efficient cryptographically secure primitives that do not require all-to-all communications.
Another example is the work of Sanil et al. [78], where they implement a privacy-preserving algorithm of computing regression coefficients, that permits (honest or semi-honest) agencies to obtain the global regression equation as well as to perform rudimentary goodness-of-fit diagnostics without revealing their data.

Chapter 2

Sketching in Distributed Stream Systems

In this chapter we provide a description of the system architecture that we will use in our work, and we introduce the sketches, that are (quite recently developed) data structures for summarizing large data streams.

2.1 Distributed Stream System Architecture

We consider a system architecture as the one described in [30]. In particular, we assume a distributed-computing environment comprising a collection of k (trusted) remote sites (nodes) and a designated (unnecessarily trusted) coordinator site. A representation of our system is shown in Figure 2.1.

Streams of data updates arrive continuously at remote sites, while the coordinator site is responsible for generating answers to periodic user queries posed over the unions of remotely-observed streams across all sites. Each remote site exchanges messages only with the coordinator, providing it with state information on its (locally observed) streams. There is no communication between remote sites.

In this general distributed streaming model, each update at remote site j is a triple of the form $<i,v,\pm 1>$, denoting an insertion (+1) or deletion (-1) of element $v \in [U_i]$ in the $f_{i,j}$ frequency distribution. All frequency distribution vectors $f_{i,j}$ change dynamically over time; handling delete operations allows us to effectively handle tracking over sliding windows of the streams, by issuing implicit delete operations for expired stream items no longer in

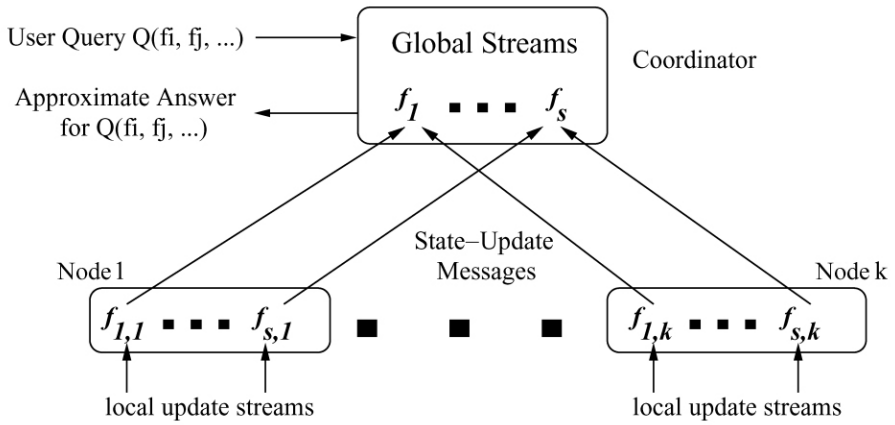

Figure 2.1: System representation

the window of interest at remote site).

It is important to observe that each of the dimensions of this problem (continuous and distributed) induce specific technical bottlenecks. Fortunately, for the first problem, when tracking statistical properties of large-scale systems, answers that are precise to the last decimal are typically not needed; instead, *approximate query answers* (with reasonable guarantees on the error) are often sufficient, since we are typically looking for indicators or patterns, rather than precisely-defined events. Concerning the challenge of being in a distributed environment, we must try to reduce as much as possible the amount of communications.

2.2 Sketching of Streams

When data sets reach considerable size, it may be necessary to transform data into a more compact form. If we are satisfied with an approximated answer for a problem, synopses of a massive data set [25] (like samples, histograms, walvelets and sketches) are solutions to be considered: they capture vital proprieties of the original data while occupying much less space. In particular, sketches are relatively recent tools (most of the algorithms have been presented in the years 2000s [25]), but they allow to receive an accurate estimate of the answer. Moreover, they are particularly appropriate for streaming data.

Sketches are essentially a kind of linear transformation of the input.
There are two kind of sketches: frequency based sketches, concerned with summarizing the frequency distribution of a data set, and sketches for distinct value queries, that count the number of distinct values in a given set. We focus on the first category, because frequency based sketch are suitable for a very large number of queries, like finding the most frequent items, estimating the size of joins between relations, approximating range queries, and, above all, extracting precise estimates of individual frequencies of items.

All the considered sketches have parametric size, depending on the values that are chosen by the user. Typically, α indicates the accuracy (i.e. the approximation error), and γ represents the probability of exceeding the accuracy bounds.
Note that, in order to decrease the error of the estimator, the size of the sketch vector has to be increased.
As already mentioned in Section 1.1, the main strenght of this technique is its use of the parameters α and γ: each user can specify a different accuracy (security) level, and therefore he use a different sketch size.

Let $\mathcal{U}$ be the domain, and consequently $|\mathcal{U}|$ the domain size. We can think about data input as a vector of size $M = |\mathcal{U}|$. In the following, we use this notation and we denote by $f[i]$ the frequency of $i\text{-}th$ item, and $\tilde{f}[i]$ the approximated frequency of that item. Furthermore, we use C for the sketch vector.

2.2.1 AGMS Sketch

The AGMS sketch was first presented, in a very primitive appearance, in the work of Alon et al. [10], with the aim to estimate the sum of the squares of the frequencies. With AGMS sketches, a data structure is mapped on a (hopefully much) smaller vector.

The sketch consists of an array C of r counters. We need a hash function g_i, which maps $\mathcal{U}$ uniformly onto $\{-1, +1\}$. This function must be four-wise independent, i.e. it must appear independent when considering sets of four items together; as written in [25], a family of four-wise independent hash

functions is given by $h(x) = ax^3 + bx^2 + cx + d \, mod \, p$ for a, b, c and d chosen uniformly from $[p]$ with p prime (for more details see [76, 74]). The sketch is built as follows.

$$\forall j,\ 1 \leq j \leq |C|,\ C[j] = \sum_{i=1}^{M} f[i] * g_j[i]$$

AGMS sketch was designed to estimate (self-)join, so there are not many works that use this kind of sketches to estimate the single items. To the best of our knowledge, [5] is one of them. Aggarwal and Yu explain how to estimate any individual value.
Let $E_i^k = C[j] * g_i[j]$. We compute $|C|$ values of E^k (one for each component of the sketch) and then we compute the mean of these E^k (because the expected contribution to the error is zero [25]).
Therefore:

$$\tilde{f}[i] = E[E^k]$$

Setting $r = O(\frac{1}{\alpha^2} \log \frac{1}{\gamma})$ ensures that the estimation of $f[i]$ has error at most $\alpha \cdot n$ with probability at least $1 - \gamma$.

A great strength of the AGMS sketch is that it allows cancellations, i.e. negative frequencies. Some drawbacks are the strong independent guarantees for the hash function, and the fact that each update affects all entries, so its complexity is $O(N)$, where N is the sketch size. An improvement of this sketch is Fast-AGMS proposed by Cormode and Garofalakis [30]; Fast-AGMS requires time sublinear at the cost of introducing a second (pair-wise independent) hash function set.

2.2.2 Count-Min Sketch

Count-Min has been introduced for the first time by Cormode and Muthukrishnan in [31].
Subsequently, it was thoroughly studied [28, 29, 25] because of its simplicity. The sketch consists of an array C of $d \times w$ counters and for each of the d rows a pair-wise independent hash function h_j, that maps items onto $[w]$. Each item is mapped onto d entries in the array, by adding to the previous

value with the new item. So each position of the sketch vector is:

$$C[j,k] = \sum_{1 \leq i \leq M : h_j(i) = k} f[i]$$

Given a sketch representation of a vector we can estimate the original value of each component of the vector by the following function:

$$\tilde{f}[i] = \min_{1 \leq j \leq d} C[j, h_j(i)]$$

Multiple keys may hash to the same bucket and thus the count of a bucket may overestimate (if frequencies are always positive) the true size of a key. For this reason the estimation procedure returns the minimum value of the counters a key is hashed to.

The estimation of each component j is affected by an error, but it is shown that the overestimate is less than n/w, where n is the number of components. So, setting $d = \log \frac{1}{\gamma}$ and $w = O(\frac{1}{\alpha})$ ensures that the estimation of $f[i]$ has error of at most $\alpha \cdot n$ with probability at least $1 - \gamma$.

The advantages of Count-Min sketch are that it requires only pairwise independent hash functions and that its update time is significantly sublinear. The main disadvantage is that it requires only positive values.

2.2.3 Count Sketch

The Count sketch [20, 25, 29] has the same structure of the Count-Min sketch, but it requires an additional pair-wise independent hash function family. One of these is required for the choice of the bucket (exactly as in the Count-Min sketch), while the other one is required to encode the value of item, like in the AGMS sketch. The sketch is defined by:

$$C[j,k] = \sum_{1 \leq i \leq M : h_j(i) = k} g_j[i] * f[i]$$

The estimate of *i-th* item is:

$$\tilde{f}[i] = \operatorname*{median}_{1 \leq j \leq d} C[j, h_j(i)] * g_j[i]$$

The median is chosen, instead of the mean, since it is less sensitive to extreme values [20]. A good comparison between some kinds of sketch and

between estimators (mean, median or minimum) can be found in the works of Rusu and Dobra [73, 75].

The dimensions of sketch are $d = O(\log \frac{1}{\gamma})$ and $w = O(\frac{1}{\alpha^2})$. The error of estimation of $f[i]$ is at most $\alpha \cdot \sqrt{F_2}$, where F_2 is the sum of the squares of the frequencies $\sum_{i=1}^{M} f[i]^2$, with probability at least $1 - \gamma$.

The advantages of Count sketch are its update time and the ability to handle negative values. The main disadvantage is that it requires two sets of hash functions.

Chapter 3

Reference Model

In this chapter we define the basic concepts useful for describing the problem and our proposed solutions. We introduce the key concept of mobility data, how we represent them and how they are used in our framework. Then, we provide the formal definitions of Differential Privacy; finally, we describe our privacy model.

3.1 Movement Data Representation

The starting point of our work is to define the concept of trajectory:

Definition 3.1.1 (Trajectory)**.** *A Trajectory or spatio-temporal sequence is a sequence of triplets* $T =< l_1, t_1 >, \ldots, < l_n, t_n >$*, where* t_i *(*$i = 1 \ldots n$*) denotes a timestamp such that* $\forall_{1 \leq i < n}\ t_i < t_{i+1}$ *and* $l_i = \langle x_i, y_i \rangle$ *are points in* $\mathbf{R}^2$*.*

Intuitively, each pair $\langle l_i, t_i \rangle$ indicates that the object is in the position $l_i = \langle x_i, y_i \rangle$ at time t_i.
In a time interval τ, each moving object can have multiple trajectories. We do not require that each trajectory is *complete*, i.e., locations may be missing at some timestamps. We allow the re-occurrence of some sub-trajectories (i.e., the object may move between locations l_i and l_j back and forth for multiple times). For example, a vehicle can have two trajectories: $T_1 = \{< \langle a, b \rangle, t_1 >, < \langle b, c \rangle, t_2 >, < \langle c, a \rangle, t_3 >\}$ and $T_2 = \{< \langle a, b \rangle, t_4 >, < \langle c, d \rangle, t_5 >\}$.

We assume that the territory is subdivided in cells $C = \{c_1, c_2, \ldots, c_p\}$ which compose a partition of the territory. For this partition we can use an

existing division of the territory (e.g., census sectors, road segments, etc.) or we can determine a data-driven partition as discussed in Section 5.2. During a travel a user goes from a cell to another cell. We use g to denote the function that applies the spatial generalization to a trajectory. Given a trajectory T this function generates the generalized trajectory $g(T)$, i.e. a sequence of *moves* with temporal annotations, where a *move* is a pair (l_{c_i}, l_{c_j}), which indicates that the moving object moves from the cell c_i to the *adjacent* cell c_j. Note that l_{c_i} denotes the pair of spatial coordinates representing the the centroid of the cell c_i; in other words $l_{c_i} = \langle x_{c_i}, y_{c_i} \rangle$. The *temporal annotated move* is the quadruple $(l_{c_i}, l_{c_j}, t_{c_i}, t_{c_j})$ where l_{c_i} is the location of the origin, l_{c_j} is the location of the destination and the t_{c_i}, t_{c_j} are the time information associate to l_{c_i} and l_{c_j}. As a consequence, we define a generalized trajectory as follows.

Definition 3.1.2 (Generalized Trajectory)**.** *Let $T = \langle l_1, t_1 \rangle, \ldots, \langle l_n, t_n \rangle$ be a trajectory. Let $C = \{c_1, c_2, \ldots, c_p\}$ be the set of the cells that compose the territory partition. A generalized version of T is a sequence of temporal annotated moves*

$$T_g = \{< l_{c_1}, l_{c_2}, t_{c_1}, t_{c_2} >< l_{c_2}, l_{c_3}, t_{c_2}, t_{c_3} > \ldots < l_{c_{m-1}}, l_{c_m}, t_{c_{m-1}}, t_{c_m} >\}$$

where $m <= n$.

More details on the generalization of trajectories are given in Section 4.2.1.

Now, we show how a generalized trajectory can be represented by a frequency distribution vector. First, we define the function *Move Frequency* (MF) that, given a generalized trajectory T_g, a move (l_{c_i}, l_{c_j}) and a time interval τ, computes how many times the move appears in T_g by considering the temporal constraint. More formally, we define the *Move Frequency* function as follows.

Definition 3.1.3 (Move Frequency)**.** *Let T_g be a generalized trajectory and let (l_{c_i}, l_{c_j}) be a move. Let τ be a temporal interval. The* Move Frequency *function is defined as:*

$$MF(T_g, (l_{c_i}, l_{c_j}), \tau) = |\{(l_{c_i}, l_{c_j}, t_i, t_j) \in T_g | t_i \in \tau \wedge t_j \in \tau\}|.$$

For any move (l_{c_i}, l_{c_j}), $MF(T_g, (l_{c_i}, l_{c_j}), \tau)$ can be any non-negative integer. For instance, given a trajectory $T = \{< (a, b), t_1 >, < (c, d), t_2 >,$

$< (a,b), t_3 >, < (e,f), t_4 >\}$ and assuming (c,d) and (e,f) locations are both positioned in the cell c_2, the generalized trajectory is $T_g = \{< l_{c_1}, l_{c_2}, t_1, t_2 >, < l_{c_2}, l_{c_1}, t_2, t_3 >, < l_{c_1}, l_{c_2}, t_3, t_4 >\}$ and, e.g., $MF(T_g, (l_{c_1}, l_{c_2}), [t_1, t_4]) = 2$. This function can be easily extended to take into consideration a set of generalized trajectories $\mathcal{T}^{\mathcal{G}}$. In this case, the information computed by the function represents the total number of movements from the cell c_i to the cell c_j in a time interval in the set of trajectories.

Definition 3.1.4 (Global Move Frequency)**.** *Let $\mathcal{T}^{\mathcal{G}}$ be a set of generalized trajectories and let (c_i, c_j) be a move. Let τ be a time interval. The* Global Move Frequency *function is defined as:*

$$GMF(\mathcal{T}^{\mathcal{G}}, (c_i, c_j), \tau) = \sum_{\forall T_g \in \mathcal{T}^{\mathcal{G}}} MF(T_g, (c_i, c_j), \tau).$$

The number of movements between two cells computed by either the function MF or GMF describes the amount of traffic flow between the two cells in a specific time interval. This information can be represented by a frequency distribution vector.

Definition 3.1.5 (Vector of Moves)**.** *Let $C = \{c_1, c_2, \ldots, c_p\}$ be the set of the cells that compose the territory partition. The* vector of moves M *with $s = |\{(c_i, c_j) | c_i$ is adjacent to $c_j\}|$ positions is the vector containing all possible moves. The element $M[z] = (l_{c_i}, l_{c_j})$ is the* move *from the cell c_i to the adjacent cell c_j.*

Now we are ready to define the frequency vector.

Definition 3.1.6 (Frequency Vector)**.** *Let $C = \{c_1, c_2, \ldots, c_p\}$ be the of the cells that compose the territory partition and let M be the vector of moves. Given a set of generalized trajectories in a time interval τ $\mathcal{T}^{\mathcal{G}}$. The corresponding* frequency vector *is the vector f with size $s = |\{(c_i, c_j) | c_i$ is adjacent to $c_j\}|$ where each $f[i] = GMF(\mathcal{T}^{\mathcal{G}}, M[i], \tau)$.*

The definition of *frequency vector of a trajectory set* is straightforward; it requires to compute the function GMF instead of MF.
Clearly, the frequency vector of a generalized trajectory is the local data vector computed by a node by using the local function MF.

Note that the above definitions are based on the assumption that consecutive locations can be in the same cell or in adjacent cells. In some cases (for example, because of GPS problems) this fact could not be true. We have to choose what to do in case of these illegal moves (moves that are not present in the Frequency Vector); a reasonable solution is to try to reconstruct the missing part of the trajectories, e.g. by interpolation.

3.2 System Architecture

We use as reference architecture the distributed system described in Section 2.1. In our scenario, we want to allow analysts to better understand the mobility behaviour in a city or territory, to monitor the traffic of vehicles, and to take advantage of this knowledge to improve the infrastructure management enabling them, for example, to reduce traffic jams
The coordinator is responsible for computing the aggregation of movement data on a territory by combining the information received by each node. In order to obtain the aggregation of the movement data in the centralized setting, we need to generalize all the trajectories by using the cells of a partition of the territory. In our distributed setting we assume that the partition of the territory, i.e., the set of cells $C = \{c_1, \ldots, c_p\}$ used for the generalization, is both known by all the nodes and the coordinator.
In a given time interval, each node, that represents a vehicle that moves in this territory, collects a set of spatio-temporal points; these points compose one or more trajectories (Definition 3.1.1). The node generalizes these locations (Definition 3.1.2), and computes the *frequency vector* (Definition 3.1.3), thus contributing to the computation of the *global frequency vector* (Definition 3.1.4) representing the movement data aggregation.

Formally, each remote node V_j (with $j \in \{1, \ldots, k\}$) observes local update streams that incrementally render a distinct frequency distribution vector f^{V_j} over data elements; that is, $f^{V_j}[v]$ denotes the frequency of the element v observed locally at remote node V_j. Then, the coordinator computes the global frequency distribution vector $F = \sum_{j=1}^{k} f^{V_j}$.

3.3 Differential Privacy Model

In Section 1.2 we introduced Differential Privacy and we provided an intuitive definition of it.

Let a database D include a private data record d_i about an individual i. By querying the database, it is possible to obtain certain information I(D) about all data and information I(D-d_i) about the data without the record d_i. The difference between I(D) and I(D-d_i) may enable to infer some private information about the individual i. Hence, it must be guaranteed that I(D) and I(D-d_i) do not significantly differ for any individual i.

The formal definition [35] is the following. We recall that the parameter ϵ specifies the level of privacy guaranteed.

Definition 3.3.1 (ϵ-differential privacy)**.** *A privacy mechanism A gives ϵ-differential privacy if for any dataset D_1 and D_2 differing on at most one record, and for any possible output D' of A we have*

$$Pr[A(D_1) = D'] \leq e^{\epsilon} \times Pr[A(D_2) = D']$$

where the probability is taken over the randomness of A.

The fundamental concept of this technique is the global sensitivity of a function mapping underlying datasets to (vectors of) reals.

Intuitively the global sensitivity represents how much the result of a query can change when it is performed on the dataset or on a dataset close to it.

Definition 3.3.2 (Global Sensitivity)**.** *For any function $f : D \rightarrow R^d$, the sensitivity of f is*

$$\Delta f = max_{D_1,D_2} ||f(D_1) - f(D_2)||_1$$

for all D_1, D_2 differing in at most one record.

The mechanism of Differential Privacy works by adding appropriately chosen random noise to the answer a=f(D), where f is the query function and D is the database. We already said in Section 1.2 that the Laplace mechanism is especially used when data are real, so in this work we focus on it. Instead of returning the true answer, we return f(D) + Lap($\frac{\Delta f}{\epsilon}$).

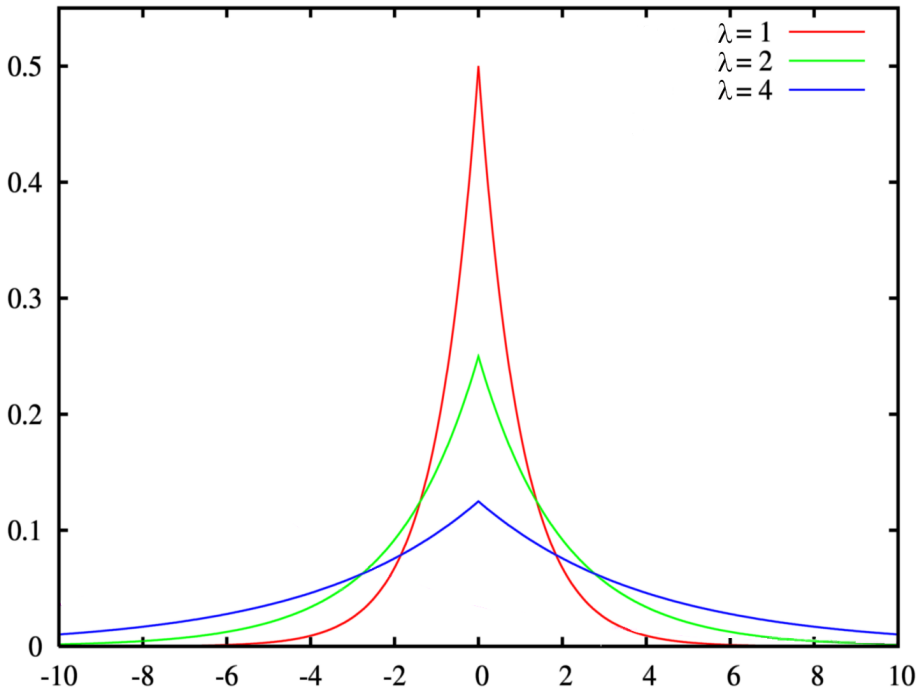

Figure 3.1: Probability Density Function as the scale changes

Note that decreasing ϵ, a publicly known parameter, flattens out the Lap($\frac{\Delta f}{\epsilon}$) curve; when ϵ is fixed, functions f with high sensitivity yield again flatter curves.
The magnitude of the noise drawn from a Laplace distribution with the probability density function $p(x|\lambda) = \frac{1}{2\lambda}e^{-|x|/\lambda}$, where λ is the *scale* factor, depends on both the global sensitivity of f and the desired privacy level ϵ (i.e. $\lambda = \frac{\Delta f}{\epsilon}$). In general, when λ increases, the curve becomes flatter, thus the peak is lower but the spread is greater (see Figure 3.1). This yields higher expected noise magnitudes.

Formally the following result holds.

Theorem 3.3.1. *[35, 36] For any function $f : D \to R^d$ over an arbitrary domain D, the mechanism A $A(D) = f(D) + Laplace(\Delta f/\epsilon)$ gives ϵ-differential privacy.*

A relaxed version of differential privacy allows claiming the same privacy level as Definition 3.3.1 in the case there is a small amount of privacy loss (due to a variation in the output distribution for the privacy mechanism A).

This is discussed in [13] and its definition is the following:

Definition 3.3.3 ((ϵ, δ)-differential privacy)**.** *A privacy mechanism A gives ϵ-differential privacy if for any dataset D_1 and D_2 differing on at most one record, and for any possible output D' of A we have*

$$Pr[A(D_1) = D'] \leq e^{\epsilon} \times Pr[A(D_2) = D'] + \delta$$

where the probability is taken over the randomness of A.

Note that, if $\delta = 0$, $(\epsilon, 0)$-differential privacy is ϵ-differential privacy.

3.4 Privacy Model

We consider as sensitive information any information from which the typical mobility behavior of a user may be inferred. This information is considered sensitive for two main reasons: 1) typical movements can be used to identify the drivers who drive specific vehicles even when a simple de-identification of the individual in the system is applied; and 2) the places visited by a driver could identify specific sensitive areas such as clinics, hospitals and routine locations such as the user's home and workplace.

In our setting, we assume that each node in our system is honest; in other words we do not consider attacks at the node level. We also assume that the coordinator is untrusted. There are two types of untrusted coordinators: (i) *semi-honest* coordinator who will try to infer the sensitive mobility information from the inputs of nodes, but otherwise follows the protocol correctly, and (ii) *malicious* coordinator who may have arbitrary auxiliary information to help break the protocol. For example, the coordinator may be able to obtain real mobility statistic information from other sources, such as from public datasets on the web, or through personal knowledge about a specific participant [79]. In this paper, we focus on designing a privacy-preserving technique to defend against a semi-honest coordinator. With this weaker assumption about the coordinator's reliability, we aim at designing privacy-preserving techniques that can provide meaningful data utility.

Unfortunately, releasing frequency of moves instead of raw trajectory data to the coordinator is not privacy-preserving, as the intruder may still infer the sensitive typical movement information of the driver. As an example, the attacker could learn the driver's most frequent move; this information can be very sensitive because such move usually corresponds to a

user's transportation between home and work place. Therefore, our goal is to compute a distributed aggregation of movement data for a comprehensive exploration of them while preserving privacy. In particular, we aim to find effective privacy mechanisms to protect the frequency information associated to each move. For this purpose, we use the Differential Privacy, the paradigm formally described in Section 3.3.

Our problem can be defined formally as the following.

Definition 3.4.1 (Problem Definition)**.** Given a set of cells $C = \{c_1, \ldots, c_p\}$ and a set $V = \{V_1, \ldots V_k\}$ of vehicles, the *privacy-preserving distributed movement data aggregation problem* (DMAP) consists in computing, in a specific time interval τ, the

$$f^{\tau}_{DMAP}(V) = [f_1, f_2, \ldots, f_s]$$

(where $f_i = GMF(\mathcal{T}^{\mathcal{G}}, M[i], \tau)$ and $s = |\{(c_i, c_j)|c_i \; is \; adjacent \; to \; c_j\}|$) while preserving privacy. Here, $\mathcal{T}^{\mathcal{G}}$ is the set of generalized trajectories related to the k vehicles V in the time interval τ and M is the vector of moves defined on the set of cells C.

Chapter 4

Privacy-Aware Distributed Mobility Data Analytics

In this chapter, we provide a detailed description of three privacy-aware data transformation methods we propose to protect the individual privacy of each user participating to our distributed analytical process. Each solution is characterized by a different trade-off between privacy and data utility. Moreover, we formally study the privacy guarantees of the various methods.

4.1 Approach Overview

In this thesis, we propose different privacy-preserving solutions based on differential privacy, which is a strong privacy model independent of the background knowledge of an adversary. Each of our solutions is characterized by a different trade-off between privacy and data utility. In the following, we describe the key ideas of these three solutions, including the computation by each node and by the coordinator respectively. The node computation mainly involves transforming data to achieve the desired privacy guarantee. We present three privacy-preserving data transformation approaches. The first one, named *UniversalNoise*, is based on the classical ϵ-differential privacy. It can provide strong privacy guarantees but high loss of data utility, due to the generation of negative flows and noise of very high magnitude. These two issues are managed in the second solution, named *BoundedNoise*, by relaxing the privacy guarantee to (ϵ, δ)-differential privacy, where δ measures the privacy loss. We will show that: (1) the *BoundedNoise* approach

can improve data utility significantly, and (2) in some cases, the *Bounded-Noise* approach may provide a low level of guaranteed privacy in practice. Indeed we can show that sometimes the privacy loss can be high. As a consequence, we propose a third solution named *BalancedNoise* that tries to maintain the balance between privacy and utility under control by setting appropriate values of ϵ and δ. The mechanism allows the nodes to specify the level of privacy ϵ and the maximum privacy loss δ and find the best solution that is capable to minimize the noise magnitude and the possible negative flows, so that it can achieve good utility. Besides the design of the privacy-preserving data transformation methods, we also design sketching approaches to reduce the communication between nodes and the coordinator. In Chapter 5 we will validate our theoretical analyses with an extensive set of experiments on large, real mobility data.

4.2 Privacy-Aware Node Computation

We assume that each node represents a vehicle that moves in a specific territory. Each vehicle in a given time interval observes sequences of spatio-temporal points (trajectories) and computes the corresponding frequency vector that is to be sent to the coordinator. The node computation is composed of two main steps, described in Algorithm 1: (a) the computation of a privacy-preserving frequency vector and (b) the vector sketching that compresses the information to be communicated with the coordinator.

Algorithm 1: NodeComputation(ϵ, τ, M, T^G, w, d)

Data: A privacy budget ϵ, a time interval τ, the vector of the moves M, a set of trajectories T^G, the sketch dimensions w and d

Result: The sketch vector representing the privacy-preserving frequency vector $sk(\tilde{f}^{V_j})$

// **Privacy-Preserving Computation (Sec. 4.2.1-4.2.3)**;

$\tilde{f}^{V_j} = PrivacyTransformation(\epsilon, M, T^G, \tau)$;

// **Data Compression (Sec. 4.2.4)**;

$sk(\tilde{f}^{V_j}) = SketchingAlgorithm(\tilde{f}^{V_j}, w, d)$;

return $sk(\tilde{f}^{V_j})$;

The first step, described in detail in Algorithm 2, is the challenging step

Algorithm 2: PrivacyTransformation(ϵ, M, T^G, τ)

Input: A privacy budget ϵ, a time interval τ, the vector of the moves M, a set of trajectories T^G

Output: The privacy-preserving frequency vector $\tilde{f}^{V_j}$

forall *observed trajectory* $T \in T^G$ **do**

  // **Trajectory Generalization (Sec. 4.2.1)**;

  $T_g = TrajectoryGeneralization(M, T)$;

  // **Update of the Frequency Vector** f^{V_j} **(Sec. 4.2.2)**;

  forall *move* $(l_{c_i}, l_{c_j}) \in T_g$ **do**

    $n = MF(T_g, (l_{c_i}, l_{c_j}), \tau)$;

    $f^{V_j}[(l_{c_i}, l_{c_j})] += n$;

// **Transformation for achieving DP (Sec. 4.2.3)**;

$\tilde{f}^{V_j} = AchievingDP(f^{V_j}, \epsilon, T^G)$;

return $\tilde{f}^{V_j}$;

because it has to transform data to achieve privacy without destroying too much of the data utility. It is composed of three phases: (1) trajectory generalization; (2) frequency vector construction; and (3) frequency vector transformation to achieve differential privacy. We describe the details of these three phases in Section 4.2.1 - 4.2.3 respectively, and discuss the details of the vector sketching step in Section 4.2.4.

4.2.1 Trajectory Generalization

Given a specific division of the territory, a trajectory is generalized in the following way. We apply a place-based division of the trajectory into segments. The area c_1 containing its first point l_1 is found. Then, the second and following points of the trajectory are checked for being inside c_1 until we find a point l_i not contained in c_1. For this point l_i, the containing area c_2 is found.

The trajectory segment from the first point to the *i-th* point is represented by the vector (c_1, c_2). Then, the procedure is repeated: the points starting from l_{i+1} are checked for containment in c_2 until finding a point l_k outside c_2, the area c_3 containing l_k is found, and so forth up to the last point of the trajectory.

In the result, the trajectory is represented by the sequence of moves

$(c_1, c_2, t_1, t_2)(c_2, c_3, t_2, t_3) \ldots (c_{m-1}, c_m, t_{m-1}, t_m)$. Here, in a specific quadruple, t_i is the time moment of the last position in c_i and t_j is the time moment of the last position in c_j. There may also be cases when all points of a trajectory are contained in one and the same area c_1. If this is the case, the whole trajectory is represented by the sequence $\{c_1\}$. Since globally we want to compute aggregation of moves, we discard this kind of trajectories. Moreover, as most of the methods for analysis of trajectories are suited to work with positions specified as points, the areas $\{c_1, c_2, \ldots, c_m\}$ are replaced, for practical purposes, by the sequence $l_{c_1}, l_{c_2}, \ldots, l_{c_m}$ consisting of the centroids of the areas $\{c_1, c_2, \ldots, c_m\}$.

4.2.2 Frequency Vector Construction

After the generalization of a trajectory, the node computes the *Move Frequency* function (Definition 3.1.3) for each move (l_{c_i}, l_{c_j}) in that trajectory and updates its frequency vector f^{V_j} associated to the current time interval τ. Intuitively, the vehicle populates the frequency vector f^{V_j} according to the generalized trajectory observed. Therefore, at the end of the time interval τ, the element $f^{V_j}[i]$ contains the number of times that the vehicle V_j moved from m to n in the given time interval τ, if $M[i] = (m, n)$.

4.2.3 Privacy-preserving Vector Transformation

As we stated in Section 3.4, if a node sends the original frequency vector without any data transformation to the coordinator, the intruder may still be able to infer the sensitive typical movements of the vehicle represented by the node. Clearly, the generalization step can help to protect the privacy of drivers but it depends on the density of the area. Specifically, if the area is not so dense, the attacker could identify a few candidates of the locations that the driver has been to. In this case, the privacy is at high risk to be breached, though it is possible to use some precaution by obtaining a suitable tessellation of the territory taking into account the density of areas (see Section 5.2 for more details). An attacker could also infer if during a trip a user went from a location a to a location b and how many times. The questions are, *how can we hide the event that the user moved from a location* a *to a location* b *during a trip in the time interval τ? And how can we hide the real count of moves in that time window?* To answer these ques-

tions, we propose three solutions based on a rigorous privacy model named ϵ-differential privacy (Section 3.3). Each solution provides a different balance between privacy and data utility.

4.2.3.1 Computation of Sensitivity

The key point of the entire differential privacy model is the definition of the sensitivity. Recall that in our setting each trajectory is transformed into a generalized one and a vehicle can go from cell a to cell b more than once during a trajectory. Therefore, the frequency count of each move can be any arbitrary non-negative integer number. We also recall that the frequency count of move (l_a, l_b) by node n_j is equal to

$$f = \sum_{\forall T_{g_i}} MF(T_{g_i}, (l_{c_a}, l_{c_b}), \tau),$$

where T_{g_i} is one of the generalized trajectories of n_j in the time interval τ and l_{c_a} and l_{c_b} denote the pair of spatial coordinates representing the centroids of the cells that l_a and l_b locate in respectively.
If we only want to hide the real value of single moves, we treat the flow of each *move* separately; in this case, adding or removing a single movement from a to b influences the count of the move (a, b) (and thus the response to an hypothetical query performed on the data containing that element or not) exactly by 1. Therefore, in this *move-based* reasoning, the sensitivity is set to 1. On the other hand, we might look for greater protection, and then a possible solution is to reason in terms of moves in a trajectory (we call this approach *trajectory-based* reasoning). In particular, we want to capture the following case: how does the move frequency count (for any single user) change if an entire trajectory (for that user) is present or not in the data? Obviously, the sensitivity of a move frequency count depends on the occurrence of that move in each user trajectory. In a time interval τ for a given vehicle (node) we can have different trips or trajectories (we have a trajectory when the user starts from a location and stops at another). We argue that adding or deleting one trajectory of n_j can affect the count of move (l_a, l_b) by at most $\max_{i=1,\ldots,q}(MF(T_{g_i}, (l_{c_a}, l_{c_b}), \tau))$. Therefore, let q be the number of trajectories and τ be the time interval, then the sensitivity of

move (l_{c_a}, l_{c_b}) is:

$$\Delta f = \max_{i=1,\ldots,q} MF(T_{g_i}, (l_{c_a}, l_{c_b}), \tau). \tag{4.1}$$

Note that the frequency count f of move (l_{c_a}, l_{c_b}) always satisfies that $f \geq \Delta f$, as $f = \sum_{i=1,\ldots,q} (MF(T_{g_i}, (l_{c_a}, l_{c_b}), \tau))$.

Given the sensitivity (either fixed to 1 or computed by Equation 4.1) we can define a differential private mechanism in various ways. In the following, we generically refer to the method *Compute Sensitivity*, thus indicating that sensitivity can be indifferently a *move-based sensitivity* or a *trajectory-based sensitivity*. We present three solutions, each one corresponding to a different implementation of the function *AchievingDP* in Algorithm 2.

4.2.3.2 *UniversalNoise* Approach

Our first approach, named *UniversalNoise*, is based on the classic ϵ-differential privacy model. In particular, at the end of the time interval τ, before sending the frequency vector to the coordinator, each node adds the Laplace noise $Lap(\frac{\Delta f}{\epsilon})$, where Δf is defined as explained in Section 4.2.3.1, to each element in the frequency vector the value in that position of the vector. At the end of this step the node transforms f^{V_j} into $\tilde{f}^{V_j}$. This process is described in Algorithm 3.

Algorithm 3: $UniversalNoise(f^{V_j}, \epsilon, T^G)$

Input: A frequency vector f^{V_j}, a privacy budget ϵ, a set of trajectories T^G
Output: The privacy-preserving frequency vector $\tilde{f}^{V_j}$
forall *vector element* $f^{V_j}[k]$ **do**
  // **Compute Sensitivity (Sec. 4.2.3.1)**;
  $\Delta f = ComputeSensitivity(\mathcal{T}^{\mathcal{G}}, M[k])$ //M = moves-vector of f^{V_j};
  $noise = Laplace(\frac{\Delta f}{\epsilon})$;
  $\tilde{f}^{V_j}[k] = f^{V_j}[k] + noise$;
return $\tilde{f}^{V_j}$;

Privacy Analysis. We are ready to show that Algorithm 2 with the privacy transformation presented just now satisfies ϵ-differential privacy.

Theorem 4.2.1. *Given the total privacy budget ϵ, for each frequency value x,* UniversalNoise *approach ensures ϵ-differential privacy.*

The correctness of Theorem 4.2.1 is straightforward due to how the noise is added according to the Laplace mechanism [36].

4.2.3.3 *BoundedNoise* Approach

The *UniversalNoise* approach has a few weaknesses. First, it could lead to the destruction of the data utility because of the added noise that, although with small probability, can reach arbitrary magnitude. Second, adding noise drawn from the Laplace distribution could generate negative frequency counts of moves, which does not make sense in our setting. To fix these two problems, we propose the second approach, named *BoundedNoise* approach, that bounds the noise drawn from the Laplace distribution. In particular, for each value x of the vector f^{V_j}, we draw the noise from $Lap(\frac{\Delta f}{\epsilon})$ bounded to the interval $[-x, x]$. In other words, for any original frequency $f^{V_j}[i] = x$, its perturbed version after adding noise should be in the interval $[0, 2x]$. By doing this, we reduce the amounts of utility loss due to adding noise, as described in Algorithm 4.

Algorithm 4: $BoundedNoise(f^{V_j}, \epsilon, T^G)$

Input: A frequency vector f^{V_j}, a privacy budget ϵ, a set of trajectories T^G
Output: The privacy-preserving frequency vector $\tilde{f}^{V_j}$
forall *vector element* $f^{V_j}[k]$ **do**
  // **Compute Sensitivity (Sec. 4.2.3.1)**;
  $\Delta f = ComputeSensitivity(\mathcal{T}^{\mathcal{G}}, M[k])$ // M = moves-vector of f^{V_j};
  $noise = Laplace(\frac{\Delta f}{\epsilon})$;
  while $(noise > f^{V_j}[k])$ *or* $(noise < -f^{V_j}[k])$ **do**
    $noise = Laplace(\frac{\Delta f}{\epsilon})$;
  $\tilde{f}^{V_j}[k] = f^{V_j}[k] + noise$;
return $\tilde{f}^{V_j}$;

We are aware that using a truncated version of the Laplace distribution may lead to privacy leakage. In the following we show that the *BoundedNoise* approach satisfies (ϵ, δ)-differential privacy, where δ measures the privacy loss.

Privacy Analysis. As pointed out in [51], differential privacy must be applied with caution. The privacy protection provided by differential privacy relates to the data generating mechanism and deterministic aggregate level background knowledge. We observe that bounding the Laplace noise will lead to some privacy leakage on some values. For instance, from the noisy frequency values that are large, the attacker can infer that these values should not be transformed from small ones. To analyze the privacy leakage of our *BoundedNoise* approach, we first explain the concept of *statistical distance* [13]. Formally, given two distributions X and Y, the *statistical distance* between X and Y over a set U is defined as

$$d(X,Y) = \max_{S \in U}(Pr[X \in S] - Pr[Y \in S]).$$

[13] also shows the relationship between (ϵ, δ)-differential privacy and the statistical distance.

Lemma 4.2.1. *[13] Given two probabilistic functions F and G with the same input domain, where F is (ϵ, δ_1)-differentially private. If for all possible inputs x we have that the statistical distance on the output distributions of F and G is:*

$$d(F(x), G(x)) \leq \delta_2,$$

then G is $(\epsilon, \delta_1 + (e^\epsilon + 1)\delta_2)$-differentially private.

Let F and F' be the frequency distribution before and after adding Laplace noise. We can show that the statistical distance between F and F' can be bounded as follows:

Lemma 4.2.2. *[13] Given an (ϵ, δ)-differentially private function F with $F(x) = f(x) + R$ for a deterministic function f and a random variable R. Then for all x, the statistical distance between F and its throughput-respecting variant F' with the bound b on R is at most*

$$d(F(x) - F'(x)) \leq Pr[|R| > b].$$

[13] has the following lemma to bound the probability $Pr[|R| > b]$.

Lemma 4.2.3. *[13] Given a function F with $F(x) = f(x) + Lap(\frac{\Delta f}{\epsilon})$ for a deterministic function f, the probability that the Laplacian noise $Lap(\frac{\Delta f}{\epsilon})$ applied to f is larger than b is bounded by:*

$$Pr(|Lap(\frac{\Delta f}{\epsilon})| > b) \leq \frac{2(\Delta f)^2}{b^2\epsilon^2}.$$

This upper bound is not tight. For instance, when $\Delta f = 1$, $b = 1$, and $\epsilon = 1$, the bound $\frac{2(\Delta f)^2}{x^2\epsilon^2} = 2$. Therefore, we improve the bound by the following theorem.

Lemma 4.2.4. *Given a function F with $F(x) = f(x) + Lap(\frac{\Delta f}{\epsilon})$ for a deterministic function f, and a Laplace distribution with zero-mean, the probability that the Laplacian noise $Lap(\frac{\Delta f}{\epsilon})$ applied to f is larger than b is bounded by:*

$$Pr\Big(\Big|Lap\Big(\frac{\Delta f}{\epsilon}\Big)\Big| > b\Big) \leq e^{\frac{-b\epsilon}{\Delta f}}.$$

Proof. *Let $\lambda = \frac{\Delta f}{\epsilon}$. The probability density function is $p(x) = \frac{1}{2\lambda}e^{(-|x|/\lambda)}$ and the cumulative distribution function is*

$$D(x) = \frac{1}{2}(1 + sgn(x)(1 - e^{\frac{-|x|}{\lambda}})).$$

Therefore,

$$\begin{aligned} Pr\Big(\Big|Lap\Big(\frac{\Delta f}{\epsilon}\Big)\Big| > b\Big) &= \int_b^{\infty} \frac{1}{2\lambda} e^{-\frac{|x|}{\lambda}}\,\mathrm{d}x \qquad (4.2)\\ &= \frac{1}{2\lambda}\Big(\int_0^{\infty} e^{-\frac{|x|}{\lambda}}\,\mathrm{d}x - \int_0^{b} e^{-\frac{|x|}{\lambda}}\,\mathrm{d}x\Big)\\ &= D(\infty) - D(b)\\ &= e^{-\frac{b}{\lambda}}. \end{aligned}$$

□

Our analysis shows that $e^{\frac{-b\epsilon}{\Delta f}} \leq \frac{2(\Delta f)^2}{b^2\epsilon^2}$, i.e., our bound is tighter than that in [13]. We stress that in our approach, the bound b of each frequency value x is not fixed. Indeed, $b = x$. Therefore, each frequency value x has different amounts of privacy leakage. Our approach thus achieves different degree of (ϵ, δ)-differentially privacy guarantee on each frequency value x. Theorem 4.2.2 shows more details.

Theorem 4.2.2. *Given the privacy budget ϵ, for each frequency value x,* BoundedNoise *approach ensures $\big(\epsilon, (e^{\epsilon} + 1)e^{\frac{-x\epsilon}{\Delta f}}\big)$-differentially privacy.*

Note that the frequency vectors with Laplace noise (without truncation) satisfy $(\epsilon, 0)$-differentially privacy.

The correctness of Theorem 4.2.2 can be easily proven by Lemma 4.2.1 and Lemma 4.2.4. Note that the frequency vectors with Laplace noise (without truncation) satisfies $(\epsilon, 0)$-differentially privacy. It is easy to verify that

the privacy loss, measured as $\delta = (e^{\epsilon}+1)e^{\frac{-x\epsilon}{\Delta f}}$, can be high. More details are as following. Recall that for any frequency count x, $x \geq \Delta f$ always holds. Next we discuss by cases that $x = \Delta f$ and $x > \Delta f$. For the former case that $x = \Delta f$, $\delta = (1 + e^{-\epsilon}) > 1$, i.e., the privacy loss is always grater than 1. For the latter case that $x > \Delta f$, $\delta = e^{\left(1-\frac{x}{\Delta f}\right)\epsilon} + e^{\frac{-x\epsilon}{\Delta f}}$. In this case, $\delta > 1$ holds when $x < \frac{ln(e^{\epsilon}+1)\Delta f}{\epsilon}$. In other words, smaller frequency counts have a higher probability to get larger amounts of privacy loss. The situation improves a lot when the x value increases. As an example considering $x = 6$ with $\epsilon = 0.3$ and sensitivity $\Delta f = 1$ the privacy loss becomes $\delta = 0.16$. Although this approach is very promising for the data utility it could be not suitable for situations where very low values of frequency are frequent. As a consequence, below we present our third solution capable to better manage the very important trade-off between privacy and utility.

4.2.3.4 *BalancedNoise* Approach

As discussed above, the *UniversalNoise* approach may provide a strong privacy guarantee but poor data utility, while the *BoundedNoise* approach can improve data utility but with a possible high privacy loss. Our third approach, named *BalancedNoise*, tries to address the trade-off issue between privacy and data utility. The *BalancedNoise* approach, described in Algorithm 5, allows the user to set the desirable values for the two parameters, the privacy budget threshold ϵ and the privacy loss threshold δ. In other words, we find the smallest interval $[-b, b]$ such that the following inequality holds:

$$(e^{\epsilon} + 1)e^{\frac{-b\epsilon}{\Delta f}} \leq \delta.$$

Note that $e^{\frac{-b\epsilon}{\Delta f}}$ is the privacy loss we found in Lemma 4.2.4. This implies that

$$b \geq \frac{-\Delta f}{\epsilon} ln \frac{\delta}{e^{\epsilon} + 1}.$$

After finding the interval, for each value x of the frequency vector, the node draws the noise from $Lap(\frac{\Delta f}{\epsilon})$ bounding the noise value to the interval $[-b, b]$, where $b = \frac{-\Delta f}{\epsilon} ln \frac{\delta}{e^{\epsilon}+1}$. Note that this solution limits as much as possible the generation of noise with values of too high magnitude while it does not completely solves the problem of the negative flows. Clearly, the possibility to compute the minimum interval that better fits the user

privacy requirements also helps to limit the negative flows (as confirmed by our experiments reported in Section 5.4.3).

Privacy Analysis. Similar to the *BoundedNoise* approach, the *Balanced-Noise* approach described in Algorithm 5 satisfies (ϵ, δ)-differential privacy.

Algorithm 5: $BalancedNoise(f^{V_j}, \epsilon, T^G,)$

Input: A frequency vector f^{V_j}, a privacy budget ϵ, a set of trajectories T^G, the privacy loss δ

Output: The privacy-preserving frequency vector $\tilde{f}^{V_j}$

Compute $b = \frac{-\Delta f}{\epsilon} ln \frac{\delta}{e^{\epsilon}+1}$;

forall *vector element* $f^{V_j}[k]$ **do**

  // **Compute Sensitivity (Sec. 4.2.3.1)**;

  $\Delta f = ComputeSensitivity(\mathcal{T}^{\mathcal{G}}, M[k])$ //M = moves-vector of f^{V_j};

  $noise = Laplace(\frac{\Delta f}{\epsilon})$;

  while $(noise > b)$ *or* $(noise < -b)$ **do**

    $noise = Laplace(\frac{\Delta f}{\epsilon})$;

  $\tilde{f}^{V_j}[k] = f^{V_j}[k] + noise$;

return $\tilde{f}^{V_j}$;

4.2.4 Vector Sketching for Compact Communications

In a distributed system an important issue to be considered is the amount of data that needs to be communicated. In fact, real life systems usually involve thousands of vehicles (nodes) that are located in any place of the territory. Each vehicle has to send to the coordinator the information contained in its frequency vector that has a size depending on the number of cells that represent the partitions of the territory. The number of cells in a territory can be very huge and this can lead to large frequency vectors. Therefore, the system has to be able to handle not only a very large number of nodes but also huge amounts of informations to be communicated. These considerations make the optimization of communicated information necessary.

We propose the application of sketching methods that allow us to apply a good compression of the information to be communicated. In particular, we propose the application of *AGMS*, *Count-Min* or *Count* sketch algorithms, introduced in Section 2.2. In Chapter 5 we empirically study the effect of

the data compression obtained with each one of these algorithms on the data utility in order to identify the best one for our final goal, that is to find a good trade-off between privacy and utility of the mobility analysis. In general, these algorithms map a frequency vector f onto a more compressed vector. The general pseudocode of this step is described in Algorithm 6; each method differs from the others in implementation details, due to the structure of the sketch, as formerly described.

Algorithm 6: $SketchingAlgorithm(\tilde{f}^{V_j}, \text{W}, \text{D})$

Input: A differential-private frequency vector $\tilde{f}^{V_j}$, the number of columns w, the number of rows d
Output: The sketched frequency vector $sk(\tilde{f}^{V_j})$
generate hash functions;
forall *vector element* $\tilde{f}^{V_j}[k]$ **do**
  update $sk(\tilde{f}^{V_j})$;
return $sk(\tilde{f}^{V_j})$;

Adding this data summarization step (the last step in Algorithm 1) does not change the privacy guarantee provided by the above methods. This is due to the fact that the sketching function only accesses a differentially private frequency vector, not the underlying database. As proven by Hay et al. [45], a post-processing of differentially private results remains differentially private. Therefore, also the whole Algorithm 1 with the sketching step maintains the same privacy guarantee of Algorithm 2.

4.3 Coordinator Computation

The computation of the coordinator is composed of two main phases: 1) computation of the set of moves and 2) computation of the aggregation of global movements.

Move Vector Computation. The coordinator in an initial setup phase has to send to the nodes the *vector of moves* (Definition 3.1.5). The computation of this vector depends on the set of cells that represent the partition of the territory. This partition can be a simple grid or a more sophisticated territory subdivision such as Voronoi tessellation. The sharing of vector of moves is a requirement of the whole process because each node has to use

the same data structure to allow for the correct computation of the global flows on the coordinator's part.

Global Flow Computation. The coordinator has to compute the global vector that corresponds to the global aggregation of movement data in a given time interval τ by composing all the local frequency vectors. It receives the sketched vector $sk(\tilde{f}^{V_j})$ from each node; then it reconstructs each frequency vector from the sketched vector, by using the estimation described in Section 2.2. Finally, the coordinator computes the global frequency vector by summing the estimate vectors component by component. Clearly the estimated global vector is an approximated version of the global vector obtained by summing the local frequency vectors after the privacy transformation only.

Chapter 5

Evaluation on Real Big Data

This chapter shows the empirical results we obtained applying our approaches on a large dataset of real GPS vehicles traces, collected by Octo Telematics. We have evaluated the three methods presented in Chapter 4, from the point of view of both data utility and privacy-preservation, and we empirically show how the trade-off between these two goals changes in the different proposals, by confirming the theoretical results.

In particular, in this chapter: first, we describe some characteristics of the selected trajectories and of the tessellation; second, we introduce the utility measures analyzed; and finally, we show how our methods behave with respect to these measures.

5.1 Dataset Description

For our experiments we used GPS vehicles traces collected in a period from 1st May to 31st May 2011. In our simulation, the coordinator collects the frequency vectors (FV) from all the vehicles to determine the Global Frequency Vector (GFV), i.e. the sum all the trajectories crossing any link, at the end of each day. Thus we defined a series of time intervals τ_i, where each τ_i spans over a single day. Note that we conducted experiments on data by considering different time intervals τ: 4 hours, one day and 2 days. Since the results we found in terms of data utility are very similar, in the following we only report the results concerning τ equal to *one day*, i.e. the 25th May 2011.
The GPS traces were collected in the geographical areas around Pisa. We

randomly selected around 4,200 vehicles out of a total of about 49,000 vehicles. This led to the generation of around 15,700 trips (trajectories) in the selected day. Furthermore, we use a territory tessellation of about 2,400 cells; so, considering as possible moves only pairs of adjacent cells we obtain frequency vectors containing about 15,900 positions (moves).
Concerning the frequency vectors constructed by all users (vehicles), we have that the majority (about 99%) of the moves are zero (this fact implies that vectors are very sparse), while the effective distribution of the non-zero elements of all users is reported in Figure 5.1. We observe that a high number of these moves consists of very low flows. Indeed, the mean of non-zero moves is 1.13 and the median is 1. This fact is reasonable because taking a time window of one day, a typical user visits few places: we have chosen a working day, therefore trips shall be mostly from home to workplace and vice versa.

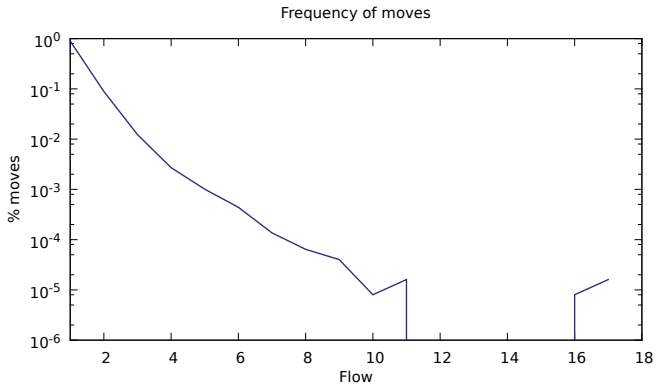

Figure 5.1: Percentage of moves that have a certain flow

Note that all the considered trajectories have at least one move and as a consequence at least one non-zero value in the frequency vector; trajectories that did not satisfy this requirement were discarded in a preprocessing phase. This is not a limitation, because in our framework these cases are discarded by the node during the *Trajectory Generalization* step, as stated in Section 4.2.1.

5.2 Spatial Tessellation

The generalization and aggregation of movement data is based on space partitioning. Arbitrary territory divisions, such as administrative districts

or regular grids, do not reflect the spatial distribution of the data. The resulting aggregations may not convey the essential spatial and quantitative properties of the traffic flows over the territory. Our method for territory partitioning extends the data-driven method suggested in paper [12]. Using a given sample of points (which may be, for example, randomly selected from a historical set of movement data), the original method finds spatial clusters of points that can be enclosed by circles with a user-chosen radius. The centroids of the clusters are then taken as generating seeds for Voronoi tessellation of the territory. We have modified the method so that dense point clusters can be subdivided into smaller clusters, so that the sizes of the resulting Voronoi polygons vary depending on the point density: large polygons in data-sparse areas and small polygons in data-dense areas. The method requires the user to set 3 parameters: maximal radius R, minimal radius r, and minimal number of points N allowing a cluster to be subdivided. In our experiments, we used a tessellation with 2661 polygons obtained with $R = 10km$, $r = 500m$, $N = 80$.

5.3 Utility Measures

To assess the information loss incurred to achieve privacy and to reduce the amount of information to be transmitted, we study how much data utility is preserved after the transformations. Since the coordinator reconstructs the flows among the zones of the tessellation, we can represent such data as a directed graph, where the nodes represent the zones and an edge between two nodes represent the flows from one zone to the other. This graph-based model allows us to analytically evaluate the resulting aggregations by means of some network-based statistics, described below. The models can also be exploited for different application scenarios and for each of them we can evaluate the quality of results after the transformations, since these mobility analyses can be performed on the transformed data too.

5.3.1 Network-based Measures

In order to assess the utility of the data collected by the coordinator we study how the distributions of general network-based measures are preserved. In particular we have considered the following measures:

Flow per Link: this measure evaluates the volume of flow in each move (edge of the network), i.e., traffic between two adjacent zones (we simply sum the traffic flows for each edge).

Flow per Zone: this measure evaluates the volume of flow in each zone (node), i.e., for each zone we sum the flows of all the incoming and outgoing flows in that zone.

Node Degree: [8, 66] this measure considers the distinct number of origins and destinations for each zone, thus focusing on the topological properties of the resulting graph, i.e, for each zone of the territory we compute the edges (with some traffic) incident to it.

Clustering Coefficient: [66] given a node the clustering coefficient is defined as the probability that two randomly selected neighbors are connected to each other. Formally,

$$CC_{c_i} = \frac{\#\ pairs\ of\ neighbors\ connected\ by\ edges}{\#\ pairs\ of\ neighbors}.$$

Node Betweenness: [38] this function is a measure of a node's centrality in a network. It computes the number of shortest paths from all nodes to all others that pass through that node. Formally,

$$NBT_{c_i} = \sum_{\substack{\forall\ (c_j, c_k),\\ c_j \neq c_i \neq c_k}} \frac{\#\ shortest\ path\ between\ c_j\ and\ c_k\ pass\ through\ c_i}{\#\ shortest\ path\ between\ c_j\ and\ c_k}.$$

Note that usually *Node Betweenness* values have high correlation with *Node Degree*, i.e., to a higher *Node Degree* corresponds a higher *Node Betweenness* value.

Edge Betweenness: [43] this function provides similar information to the previous one, but considering the edge instead of the node. In other words, it measures the edge's centrality in a network, taking into account the fraction of shortest paths between two nodes that pass through an edge, over all pairs of vertices.

5.3.2 Mobility Application Scenarios

The reconstructed GVF enables a traffic manager to evaluate the traffic condition by monitoring the status of the road network. We explored a vi-

sualization approach where the measures *Flow per Link* and *Flow per Zone* are rendered on a map. In particular, in Figure 5.2 (left) the *Flow per Link* are presented as arrows whose thickness is proportional to the amount of traffic on that link. The *Flow per Zone* (Figure 5.2 (right)) are rendered with a circle whose radius is proportional to the median value of all the zones and the color indicates if the flow is above (red) or below (cyan) the median. These two graphical representations allow to easily identify the portions of the road network with critical traffic conditions. These visualizations, when performed on the values obtained after the privacy (and sketching) transformations and compared with the original ones, allow us to qualitatively evaluate the trade-off between data privacy and data utility. Moreover, they provide examples of mobility analyses that can be done, even with the private data.

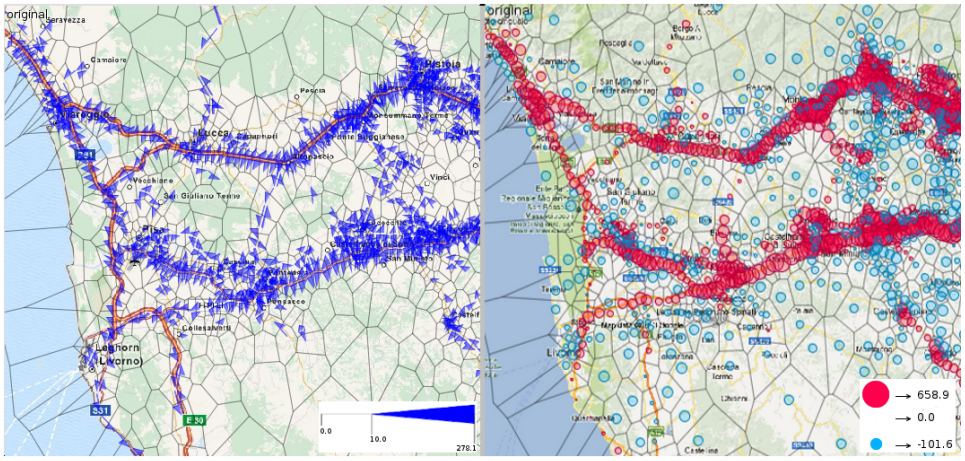

Figure 5.2: Traffic and density analysis for original data

The transformed data has also been used to study the aggregation of zones on the basis of their relative mobility, according to the approach presented in [72]. Starting from the graph-based model of flows, we apply a community discovery algorithm on the data to determine the groups of nodes strongly connected by high flows. We call such aggregation of zones as *Mobility Borders* to stress the definition of a boundary derived from mobility data. *Mobility Borders* have the aim to determine groups of regions such that the inner movements within a group are more frequent than the movements towards the other groups. In other words, this problem consists in finding areas with a dense exchange of travelers between them and a low

exchange of travelers across this set of areas, and this can then be reduced to the problem of finding clusters of nodes that are densely connected internally and sparsely connected with the rest of the network. The result of *Mobility Borders* can be rendered visually by joining the geometries of the zones into a larger polygon according to the group they belong to. Figure 5.3 shows an example of this kind of analysis.
To analytically evaluate the goodness of the resulting clusters, we consider two measures adapted from information retrieval research field: *precision* and *recall*. With precision we measure the ratio of zones in the same group in the original data that stay in the same group in the transformed one. The recall measures the contribution of several original groups to a group coming from transformed data.

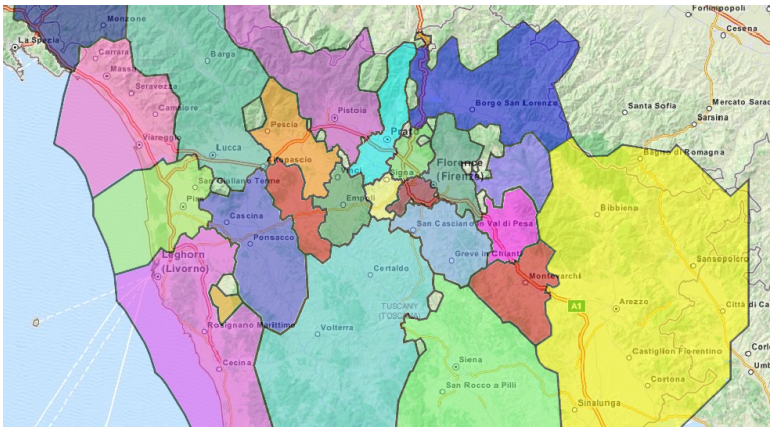

Figure 5.3: *Mobility Borders* results for original data

5.4 Analytical evaluation

We now discuss the experiments conducted on the real-world data described above. To evaluate the data quality after the transformation we compare the transformed flows with the original ones. According to the utility measures defined in Section 5.3, for each measure we compare the resulting statistics for each node and edge of the graph-model resulting from transformed data with the graph yielding from the original data.

We can use the scatter plots to highlight the differences between the transformed data and the original ones; in these plots we also report the

fitted regression line and its slope, to have a further indication of the quality of the correlation.
To present the results more formally for different comparisons of parameters and utility measures, we adopt the Pearson Correlation Coefficient (PCC) to represent analytically the amount of data perturbation introduced. The coefficient ranges from -1 to 1: it tends to 1 when the data points are close to the regression line; it tends to -1 when there is probably a reverse correlation with regards to the regression line found; finally, it tends to zero when the data points are scattered away from the line, i.e., there is no linear correlation between the variables.

5.4.1 Impact of Sensitivity on Privacy Transformations

Now, we discuss and evaluate the impact of the sensitivity on the data utility and privacy protection. In general, by increasing the sensitivity we should have a better protection (as stated in Section 4.2.3); this is because the scale factor increases when the sensitivity augments (see Section 3.3). Clearly, this leads to the generation of noise of a higher magnitude. This result is confirmed experimentally, even though the difference is not very marked (see Table 5.1). The fact that the difference is small depends on the characteristics of the dataset: several moves are equal to 1, therefore the sensitivity is often equal to 1, also considering the *trajectory-based reasoning*.

	average	*minimum*	*maximum*
move-based sensitivity	2.01656	0.0108814	9.85709
trajectory-based sensitivity	2.03846	0.0194095	10.1975

Table 5.1: Noise by varying the sensitivity, over $124,772$ values.

However, it is also important to take into account another aspect. In the *UniversalNoise* approach, considering the *move-based sensitivity* (i.e, a sensitivity always equal to 1), there is a substantial drop in the data utility, because given a node the privacy transformation adds a noise quantity to each edge, i.e., to each element of the frequency vector. The data utility improves a lot when we consider the *trajectory-based sensitivity* which does not add any noise value to edges where no flow is present. These edges have sensitivity equal to 0 as defined in Equation 4.1 in Section 4.2.3.1. This approach does not generate any privacy leak, because we do not consider as

sensitive information the fact that a user did not travel along a certain edge. In other words, our focus is on protecting the real movements because some sensitive inferences could be hidden among those.

Figure 5.4 shows the scatter plots for the *Flow per Link* measure (Figure 5.4(a)&(b)) and the *Flow per Zone* measure (Figure 5.4(c)&(d)). Here, we compare what happens when we apply the *move-based sensitivity* or the *trajectory-based sensitivity*. We observe that though we chose large ϵ (in the figure ϵ is equal to 0.9), i.e., less privacy, the correlation in the case of *move-based sensitivity* (Figure 5.4(a)&(c)) is inexistent.

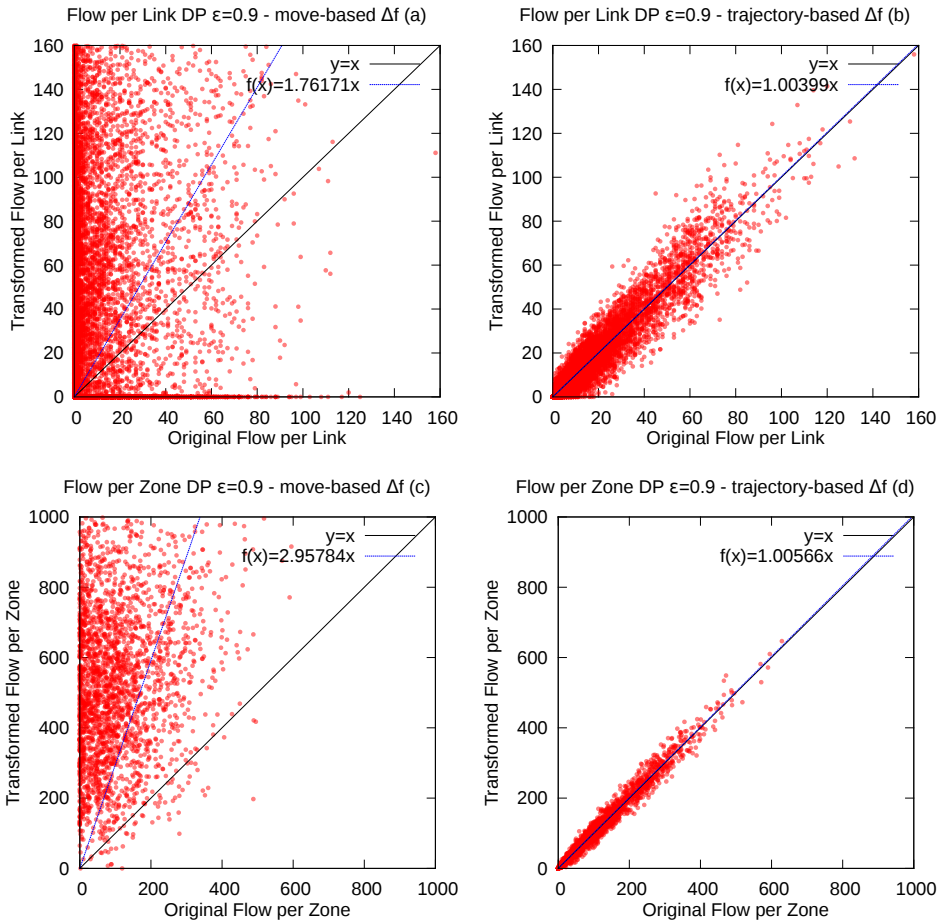

Figure 5.4: Correlations of *Flow per Link* and *Flow per Zone* by varying sensitivity in *UniversalNoise*

In Figure 5.5 are reported the visualizations of *Flow per Link* in the first row and the visualizations of *Flow per Zone* in the second row. Note that

Figures 5.5(a)&(d) illustrate the visualizations for the original data; Figures 5.5(b)&(e) show the visualizations for the perturbed data by *UniversalNoise* and the *move-based sensitivity*; and lastly, Figure 5.5(c)&(f) depict the results for the perturbed data by *UniversalNoise* and the *trajectory-based sensitivity*. Clearly, with the correlations obtained through the use of *move-based sensitivity*, the visualizations introduced in Section 5.3.2 are not meaningful.

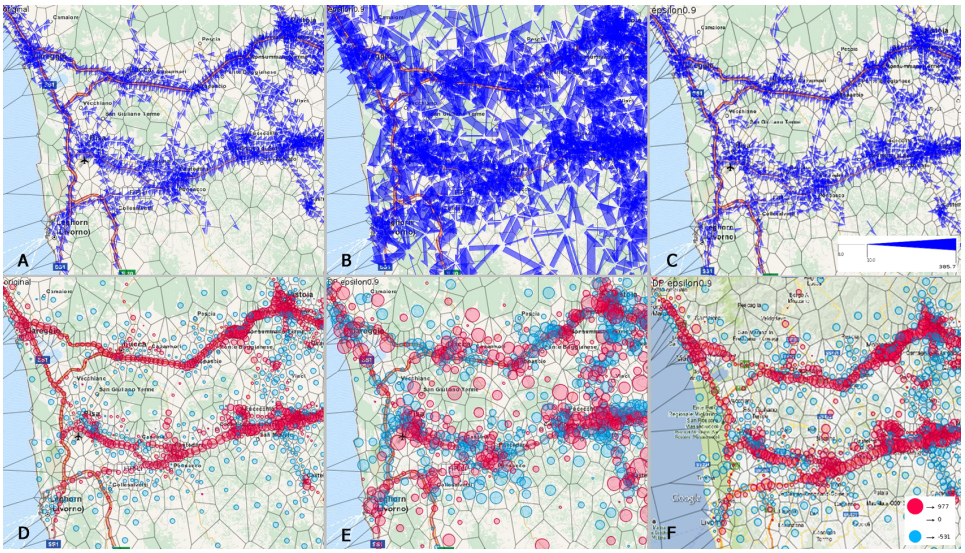

Figure 5.5: Visualizations of *Flow per Link* and *Flow per Zone* by varying sensitivity in *UniversalNoise*

Finally, with the purpose to find some property which is preserved, we compute the PCC of all the network-based measures presented above. Unfortunately, when we use the *move-based sensitivity* (Figure 5.6 (left)), the correlations are extremely low, therefore we can argue that there is no similarity between the original and the perturbed values. However, the correlations obtained using the *trajectory-based sensitivity* (Figure 5.6 (right)) are promising: this case will be discussed in detail in Section 5.4.3.
Note that this difference in terms of data utility between *move-based sensitivity* and *trajectory-based sensitivity* does not appear in the *BoundedNoise* method, because in that method the perturbed flow always lies between 0 and the double of the original flow (so the zero-moves are never altered), while it resurfaces again in the *BalancedNoise* approach.

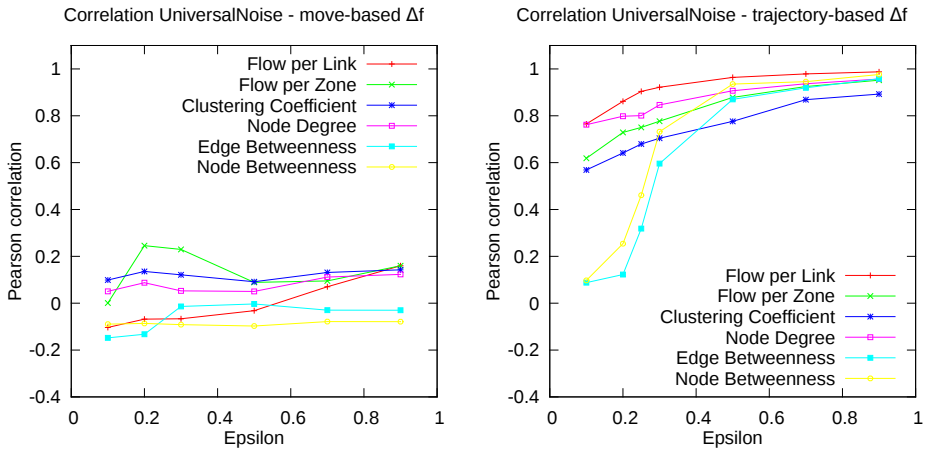

Figure 5.6: Correlations of network-based measures by varying sensitivity in *UniversalNoise*

5.4.2 Privacy and Utility of *BoundedNoise* Approach

In this section we show our evaluation of the *BoundedNoise* approach in terms of both privacy guarantee and data utility. Our experiments on real data confirm the theoretical results, described in Section 4.2.3.3, concerning the privacy loss related to the *BoundedNoise* approach. Indeed, we observed that usually in a time interval of one day each user has a high set of moves with low value in its frequency vector, because typical users go from an area to another only few times during the day. This implies that the application of the *BoundedNoise* method may lead to a too high privacy loss, due to unacceptable δ values. We might have privacy leaks if the δ values is greater than 1, and using the *move-based sensitivity* this happens in 99% of cases. Unfortunately, also using the *trajectory-based sensitivity* the situation does not highly improve: in Figure 5.7 (left) we plot the percentage of cases where we have a resulting δ higher than 1, which is unreasonable for privacy protection. We divided the events edge by edge, depending on the sensitivity value; especially, for each sensitivity value, we plot (with respect to the total number of edges that have that sensitivity) the percentage of cases where the flow is equal to the sensitivity and the percentage of cases where the flow is greater than the sensitivity value, but less than the ratio explained in the theoretical analysis. As the sensitivity increases, the percentage of the first case tends to decrease because it is likely that in these circumstances a

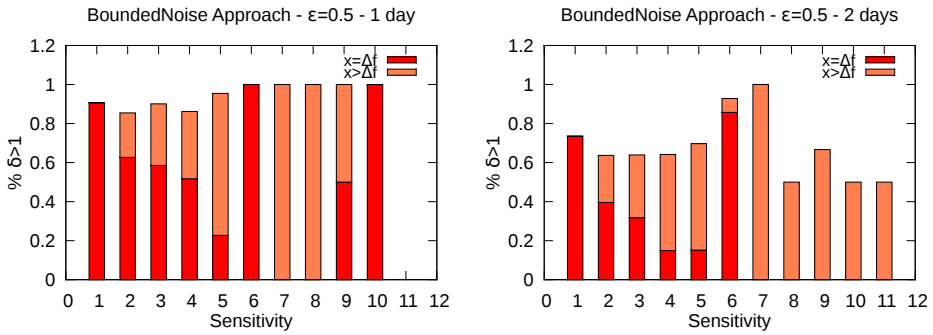

Figure 5.7: Study of the privacy loss in *BoundedNoise*

user has many trajectories, and therefore the flow on the edge in the whole time window is greater than one in a single trajectory. Note that the peak at sensitivity equal to 10 corresponds to only two people who actually have gone through an edge 10 times in a single trajectory. In Figure 5.7 (right), we also noted that when we increase the time interval τ the privacy loss decreases and this supports our hypothesis that this naive approach can give a good trade-off between privacy and data utility in scenarios where it is reasonable to have a wide time window, for example *one week*, and in contexts which are characterized by high frequencies of items.

In addition, enlarging the time window, the frequencies of the moves increase and this is confirmed by the study shown in Figure 5.8, where we can see the percentage of the moves with value 1 with respect to the total non-zero moves, selecting time windows of one day, two days and one week (respectively: red, green and blue points).

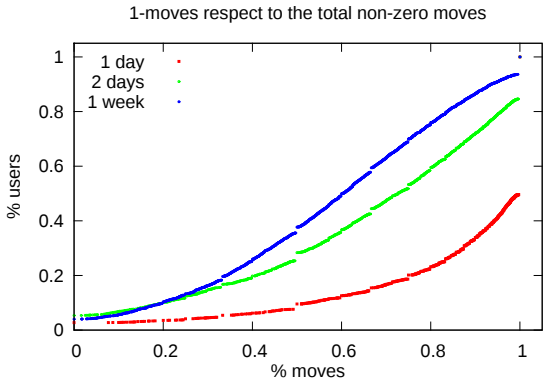

Figure 5.8: Frequencies of non-zero move

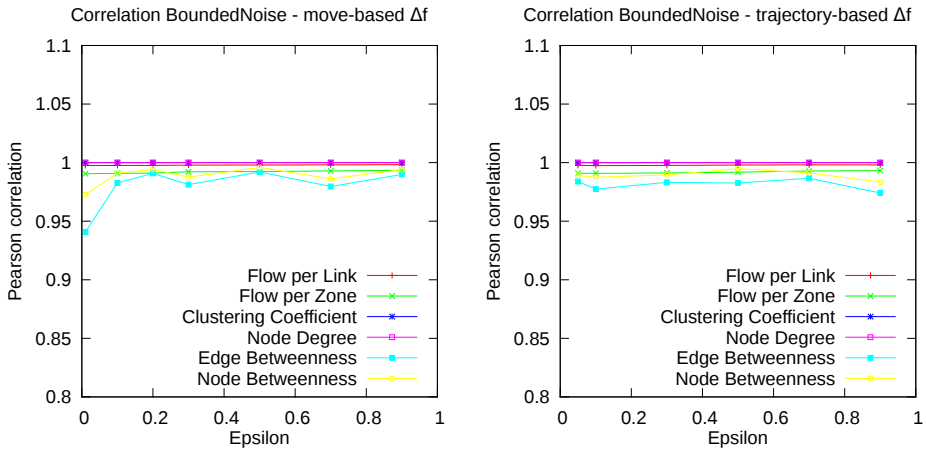

Figure 5.9: Correlation of network-based measures in *BoundedNoise*

However, the utility provided by *BoundedNoise* method is very good, as showed in Figure 5.9, where it is reported the PCC for each network-based measure computed on the for different values of ϵ. The two plots in this figure show that the values of the PCC obtained after the application of the *BoundedNoise* with either the *trajectory-based sensitivity* or the *move-based sensitivity* are substantially equivalent. For this reason in the following pictures we show the other analyses only with respect to the results regarding the utility obtained by the *trajectory-based sensitivity*. Note that the results obtained by *move-based sensitivity* are very similar. In Figure 5.10 we com-

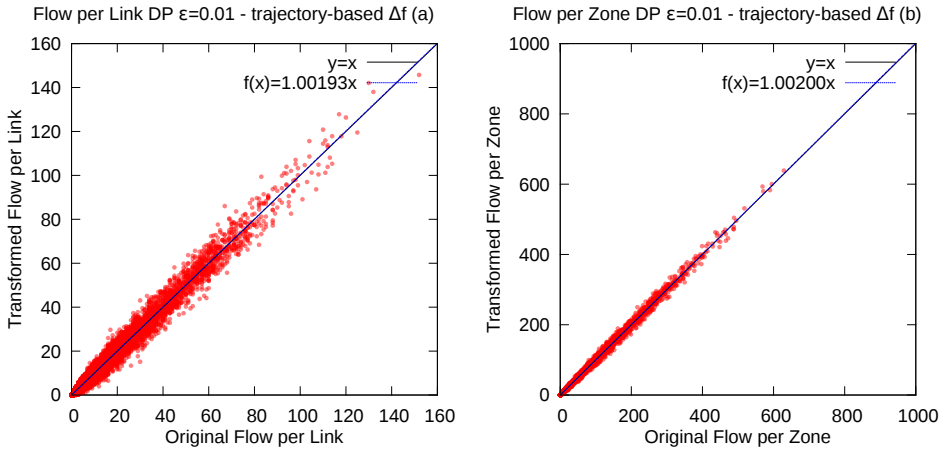

Figure 5.10: Correlations of *Flow per Link* and *Flow per Zone* in *BoundedNoise*, with $\epsilon = 0.01$

pare the values of each edge (left) and of each node (right) before and after the privacy-transformation; while in Figure 5.11 we have rendered the *Flow per Link* measure and the *Flow per Zone* measure on the map. We chose to show these results of a very low epsilon ($\epsilon = 0.01$) with the aim to emphasize the very good quality of mobility analysis that an analyst can obtain even if the data are transformed by using a very low ϵ value.

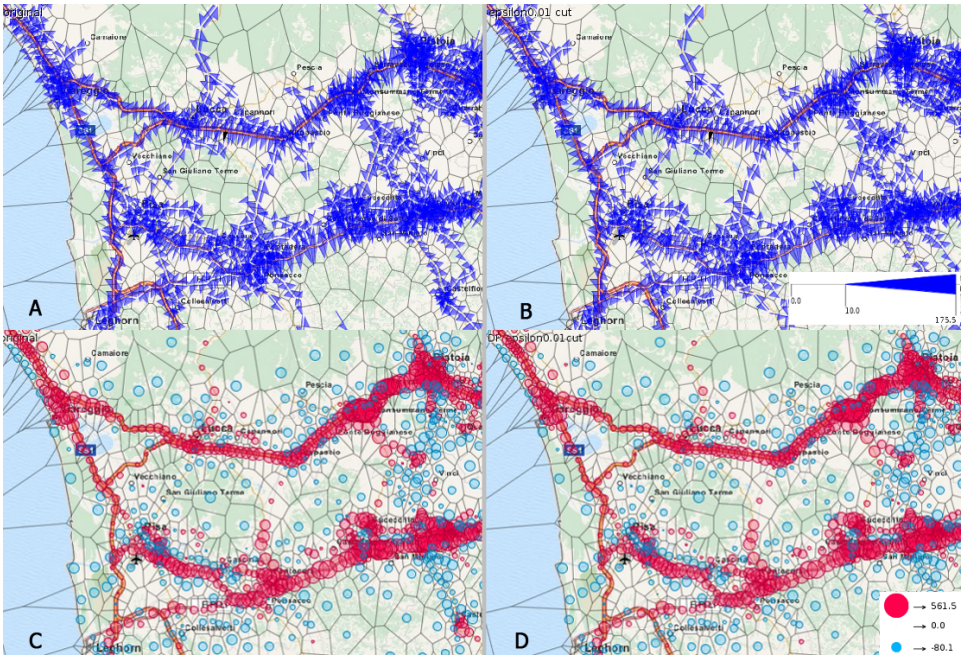

Figure 5.11: Visualization of *Flow per Link* and *Flow per Zone* in *BoundedNoise*, with $\epsilon = 0.01$

5.4.3 Data Utility for *UniversalNoise* and *BalancedNoise* Approches

In this section we analyze the two transformation methods *UniversalNoise* and *BalancedNoise*, respectively presented in Section 4.2.3.2 and Section 4.2.3.4. We have already explained in Section 5.4.1 that the *UniversalNoise* approach does not give good results when the *move-based sensitivity* is used, so now we will analyze the case in which each vehicle uses the *trajectory-based sensitivity*. In order to provide a fair comparison, the same sensitivity is also used in the analysis of the *BalancedNoise* method; we do not present the results for the *move-based sensitivity* because in terms of data utility

they are very similar to those obtained by using *trajectory-based sensitivity* and in terms of privacy the last one offers better protection.

First of all, we want to conclude the study started in the Section 5.4.1 (in Figure 5.4) by showing the scatter plots of the *Flow per Link* measure for other two transformations, namely $\epsilon = 0.5$ (Figure 5.12 (left)) and $\epsilon = 0.2$ (Figure 5.12 (right)). The scatter plots highlight the differences between the two transformations, where the more protective transformation ($\epsilon = 0.2$) perturbs the data the most, since the data points tend to go far from the fitting line.

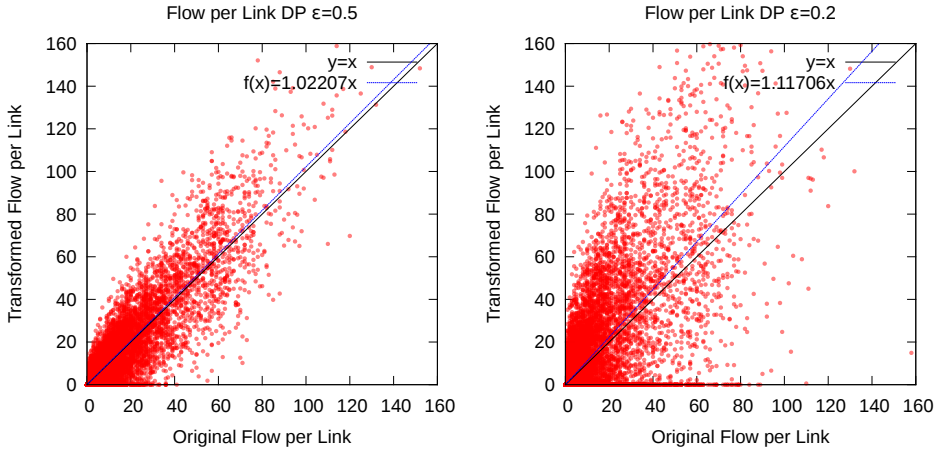

Figure 5.12: Correlations of *Flow per Link* after *UniversalNoise*

Furthermore, we want to point out that the intuition that *BalancedNoise* helps to limit the negative flow values is further confirmed by our experiments, as highlighted in Table 5.2, which shows the negative noise obtained in the global frequency vector. We fixed $\epsilon = 0.5$, but this behavior is maintained for every other ϵ. The first line simply reports the number of negative noise values obtained, while the second line presents the average of these values. As one can see, both the number and the size of the noise values obtained decrease with an increase of δ.

	UniversalNoise	*BalancedNoise*				
		$\delta = 0.05$	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.3$	$\delta = 0.5$
number	1,407	1,334	1,227	1,032	957	743
average	-3.178	-2.632	-2.223	-1.756	-1.657	-1.097

Table 5.2: Negative noise obtained with various executions, for $\epsilon = 0.5$.

After these remarks, we describe the results obtained on the basis of the measures introduced in the Section 5.3.

Network-based Measures Distributions. To assess the validity of the transformation approach, we compare the private data with the original data by varying the transformation parameters. The comparison is performed with two approaches by varying the values of ϵ and δ: we compare the resulting cumulative distribution of the utility measures and the linear correlations by means of the PCC. In the figures from Figure 5.13 to Figure 5.17 (on the left sides) we report, for each utility measure, the resulting distributions for $\epsilon = 0.1, 0.2, \ldots, 0.9$ and for the original data. From such plots it is possible to estimate the best parameters that yield a good trade-off between data protection and data utility. Furthermore, in the same figures (on the right sides) we report the distributions for original data, for data perturbed with the *UniversalNoise* method (using ϵ equal to 0.2 and 0.3) and for the data perturbed with the *BalancedNoise* technique, by fixing ϵ to 0.2 and by varying δ between 0.05 and 0.2. These comparisons are important in order to show how you can get the same quality decreasing ϵ and increasing δ, then to show, in the practice, how the *BalancedNoise* allows to manage the balance between privacy and data quality.

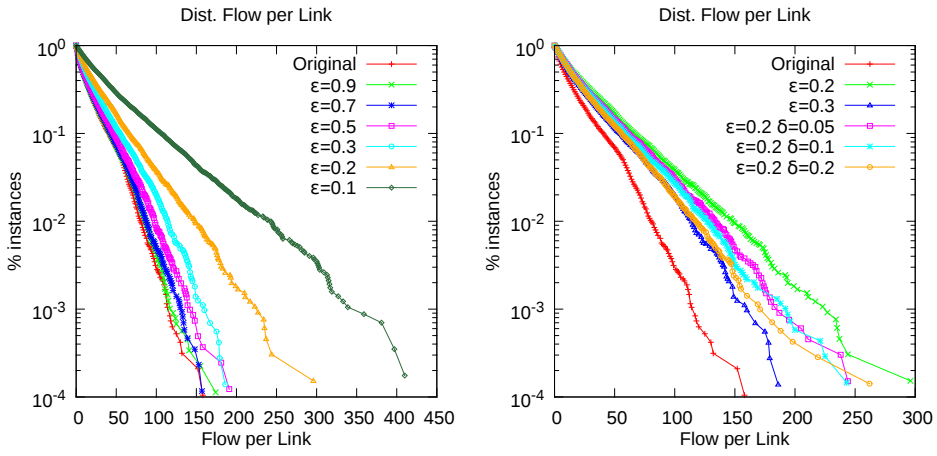

Figure 5.13: *Flow per Link* Distributions

For example, for the *Flow per Link* measure (Figure 5.13(a)), we can notice a clear discontinuity for $\epsilon = 0.2$ and $\epsilon = 0.1$, suggesting that a good value for ϵ would be 0.3. However, it is interesting to note how the δ parameter may contribute to increase data utility. In fact, considering a more protective

value for ϵ, say $\epsilon = 0.2$, it is possible increase δ to augment the resulting data utility. In Figure 5.13(b), for instance, we can see how the distributions tend to be similar to the curve for $\epsilon = 0.3$ when we increase δ. In particular, when $\delta = 0.2$ the curve is very similar to $\epsilon = 0.3$ even with a difference on the tails of the two curves.

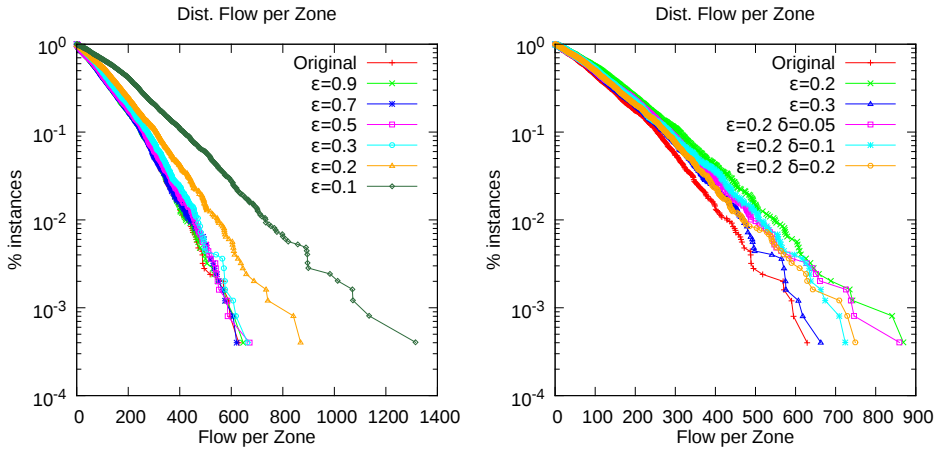

Figure 5.14: *Flow per Zone* Distributions

Similar results can be observed for *Flow per Zone* measure (Figure 5.14), where the candidate value for ϵ is again 0.3. Also in this case, the δ parameter contributes to enhance the data protection by lowering the value for ϵ to 0.2 and increasing δ to 0.2.

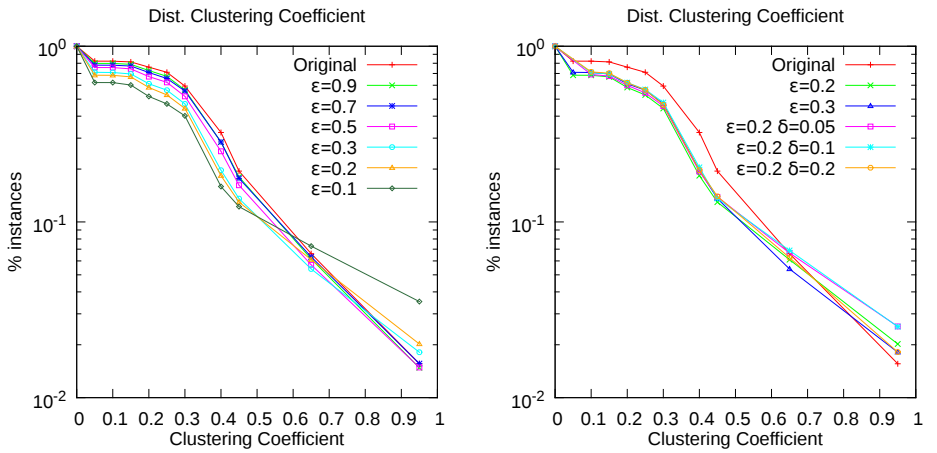

Figure 5.15: *Clustering Coefficient* Distributions

The *Clustering Coefficient* measure is very robust even for low values of ϵ (Figure 5.15): we can appreciate a different distribution only when $\epsilon = 0.1$. This property confirms that the privacy transformation may perturb the local weight of edges but in general it preserves the topology of the graph.

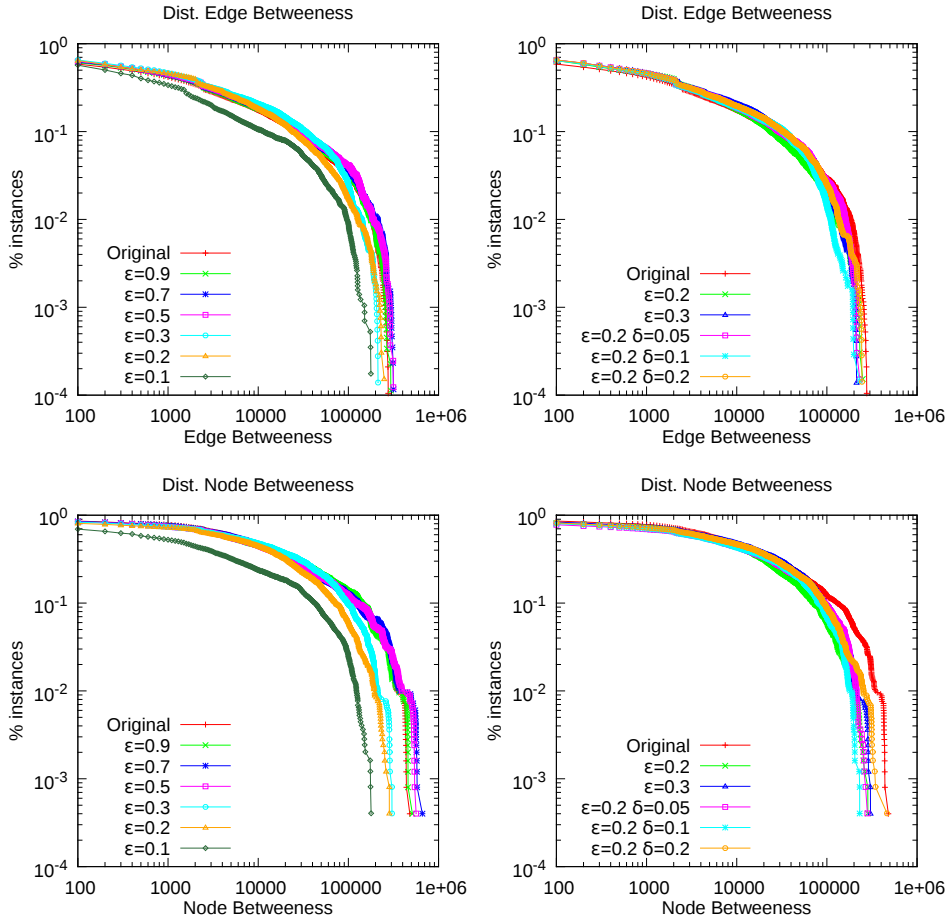

Figure 5.16: *Betweenness* Distributions

Another evidence of this phenomenon is given by the two measures of betweenness (Figure 5.16), where we can appreciate how the different parameters yield similar distribution. This means, for example, that the number of relevant edges within the graph is maintained across different transformations.

This is evident also from the distribution of the *Node Degree* measure (Figure 5.17), where we can notice how the number of neighbors for each node tend to diminish when ϵ becomes smaller. We can relate this property to

the pruning of some graph components that, however, are not relevant for the connectivity of the graph.

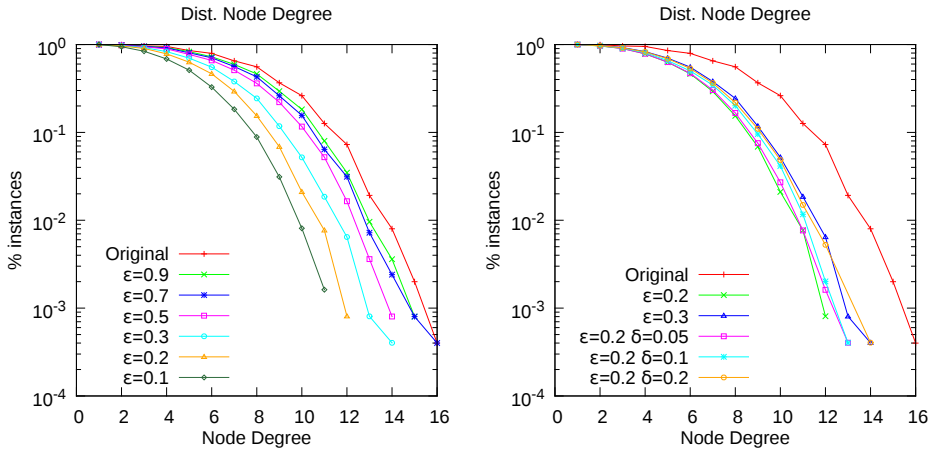

Figure 5.17: *Node Degree* Distributions

Network-based Measures Correlations. Besides the general distributions of the utility measures, we also want to determine how each component of the graph is transformed locally. Figure 5.18 shows, for each utility measure, the resulting PCC for different combination of δ and ϵ. Even at this level of details, it is possible to identify the most promising ϵ values for the transformations. In particular, let us consider the *Flow per Link* correlation in Figure 5.18(a). As already observed for the cumulative distribution, the correlation index decreases considerably when ϵ is less than 0.3. Fixed a minimum PCC threshold, we can start reasoning about the relation between ϵ and δ. Fixed a minimum value of 0.77 for PCC, we can reach a comparable quality result even if we decrease ϵ by increasing the value of δ. From the figure we can infer that the data utility provided by $\epsilon = 0.3$ is equivalent to the data utility for $\epsilon = 0.2$ and $\delta = 0.2$. Similarly, fixed a value for ϵ, say $\epsilon = 0.3$, by increasing δ it is possible to increase the data quality of the reconstructed flows. The relation between the two parameters enables the data owner to define the most suitable trade-off between data protection and data utility. The discussion for the choice of the correct ϵ parameter is even more crucial for the betweenness quality measures. Figures 5.18(e) & (f) evince that the PCC drops when the threshold is below $\epsilon = 0.3$. However, when δ is increased the quality measure performance raises.

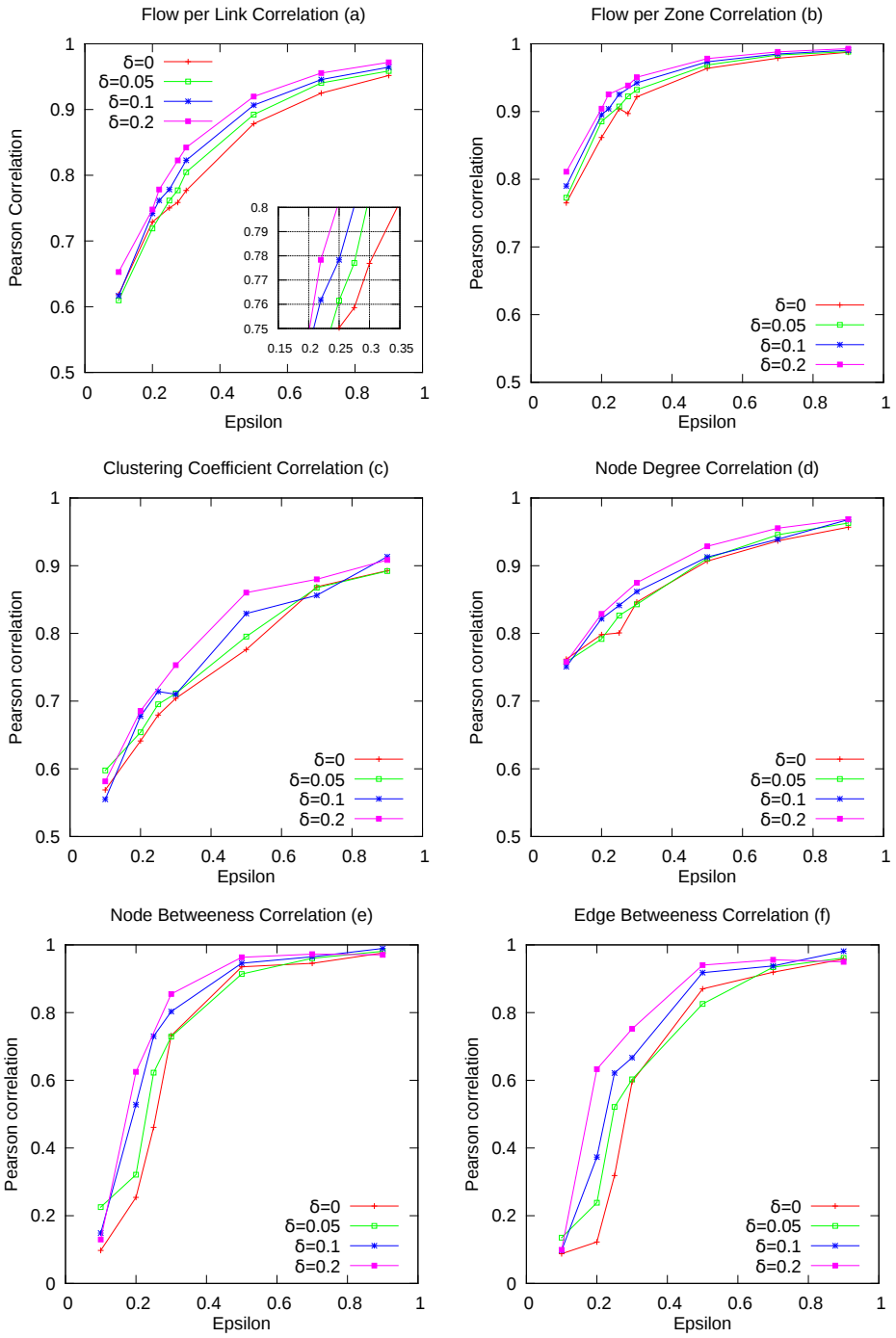

Figure 5.18: Distribution of the Pearson Correlation of the various network-based measures after the privacy transformation

Mobility Application. We also present the results obtained by these approaches with regard to mobility applications. In particular, we show that these analyses, that require the use of mobility data, can be done even using the perturbed data from the *UniversalNoise* and the *BalancedNoise* methods, which, as we have just shown, offer good guarantees on the utility.

Figures 5.19-5.22 show the reconstructed map for different parameters for privacy preservation. In particular, in Figure 5.19 and Figure 5.20 we display the results of the application of the *UniversalNoise* approach. As one can see, comparing the original map (a) with the maps obtained after the privacy transformation, we can observe that for $\epsilon = 0.5$ (b) we have high quality results, but even with low values of ϵ, e.g. $\epsilon = 0.3$ (c) and $\epsilon = 0.2$ (d), it is still possible to reason about traffic condition since the major flows for links are sufficiently preserved. As we have already seen

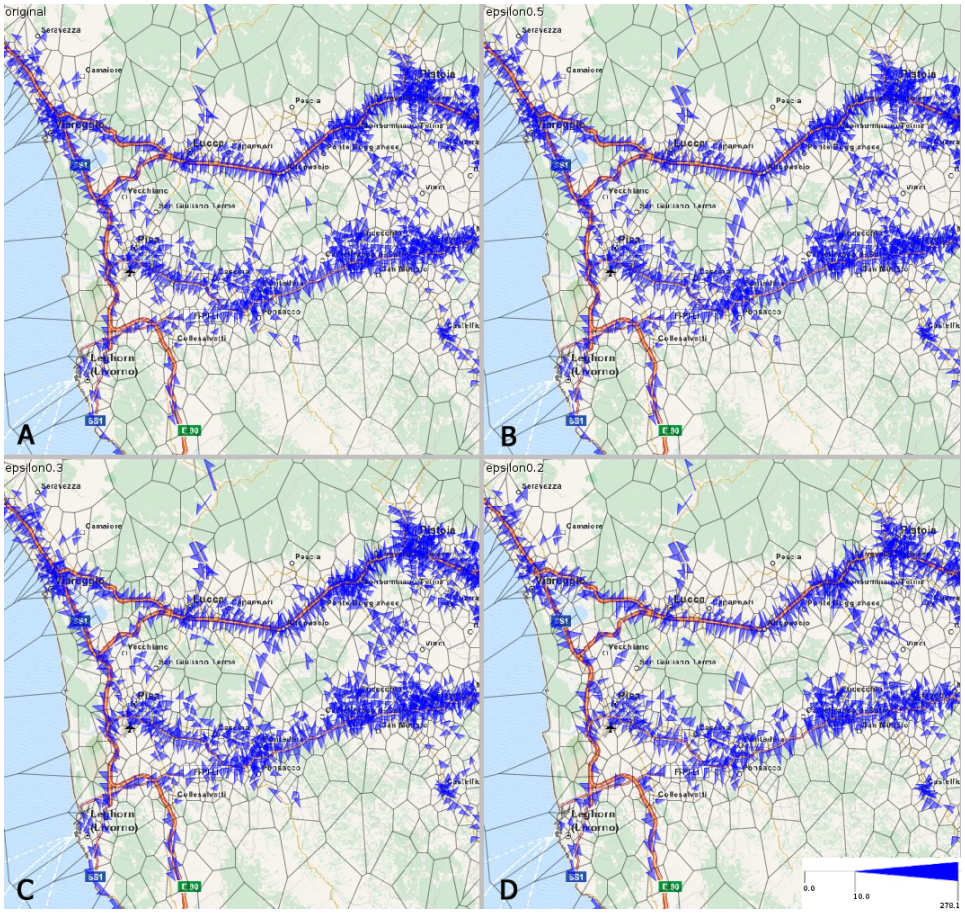

Figure 5.19: Comparison of traffic analysis with *UniversalNoise*

in different occasions, the *Flow per Zone* measure is more robust to data transformation, since the randomization is performed on the edge level and, hence, in the same zone different perturbations on incident edges tend to compensate each others; this property is further confirmed by these images, where the use of $\epsilon = 0.2$ allows to achieve a very good analysis.

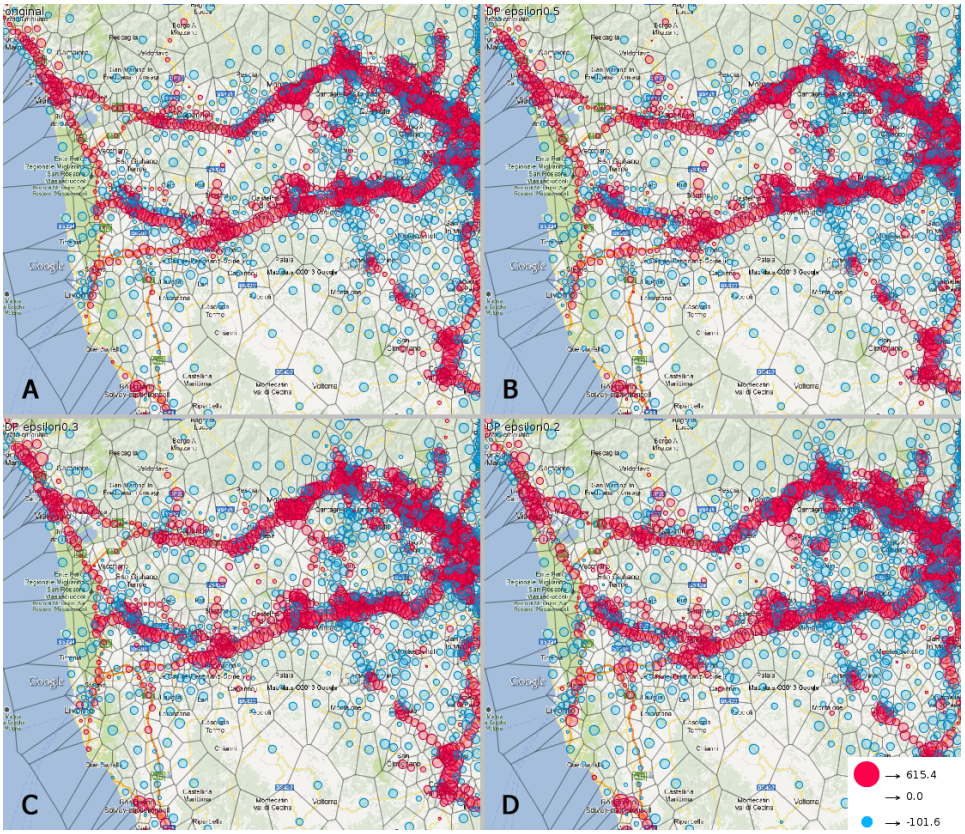

Figure 5.20: Comparison of density analysis with *UniversalNoise*

From Figure 5.21 and Figure 5.22 we can notice the influence of the δ parameter on the transformed flows. In fact, fixed a value $\epsilon = 0.2$ (reported in the figures at top-right position) the overall quality of the maps can be improved by increasing the second parameter of the *BalancedNoise* approach. In particular, it is evident how the resulting maps for $\delta = 0.1$ (c) and $\delta = 0.2$ (d) present a topology similar to the original data.

Clustering Application. Besides, we present our results with regard to the clustering analysis and *Mobility Borders*. Figure 5.23 shows a visual comparison between the resulting aggregations for different combinations of

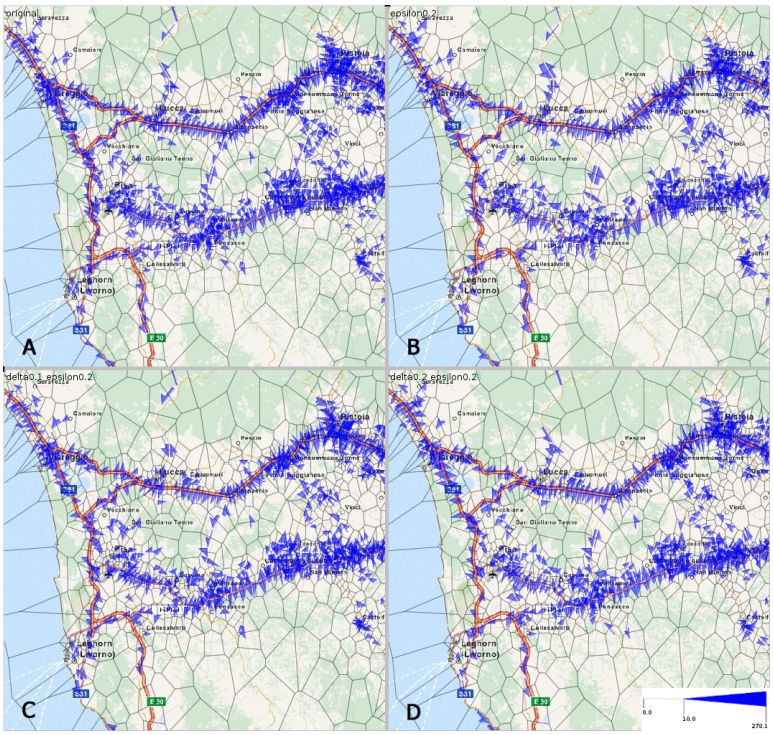

Figure 5.21: Comparison of traffic analysis with *BalancedNoise*

ϵ and δ and the aggregation resulting from the original data. The borders yielding from the original data are rendered as thicker lines to facilitate the comparison. The resulting borders for the transformed data are rendered by colors: zones in the same group are filled with the same color. The map shows the influence of the two parameters for the transformation, in particular we show the resulting maps for $\epsilon = 0.2$ and $\delta = 0.2$. We can observe that the *Mobility Borders* results are very robust to data perturbation, since the majority of the zones are preserved even for low values of ϵ. However, it is possible to identify small variation on central zones of the map with a higher density of links and connections. In general, the zones grouped for the original data tend to stay in the same group also for the transformed data. In some cases, it happens that an original group is split across two or three distinct new groups. As explained in Section 5.3.2, to analytically evaluate such behavior, we consider precision and recall. The resulting values for the two measures are showed in Figure 5.24. We can see that the precision (Figure 5.24(left)) remains very high for any value

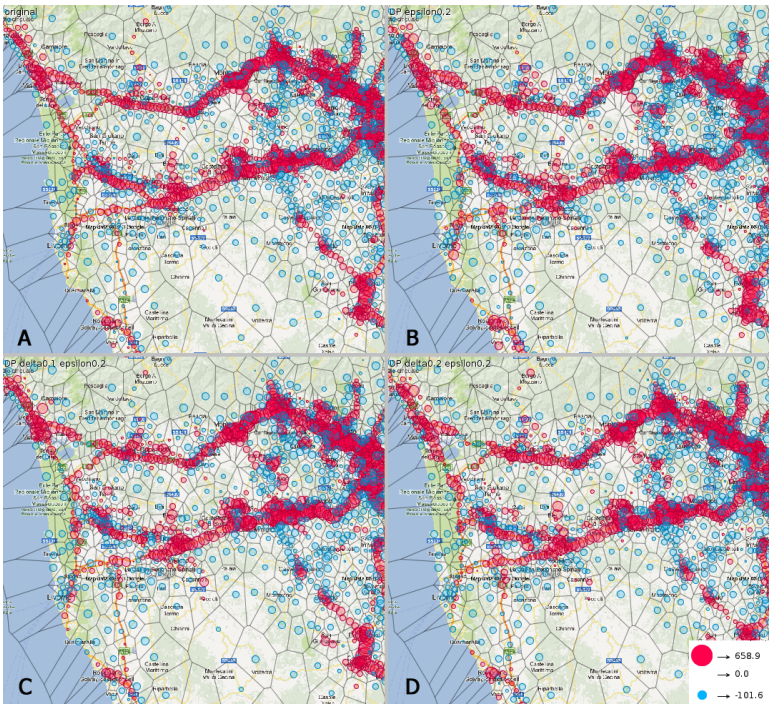

Figure 5.22: Comparison of density analysis with *BalancedNoise*

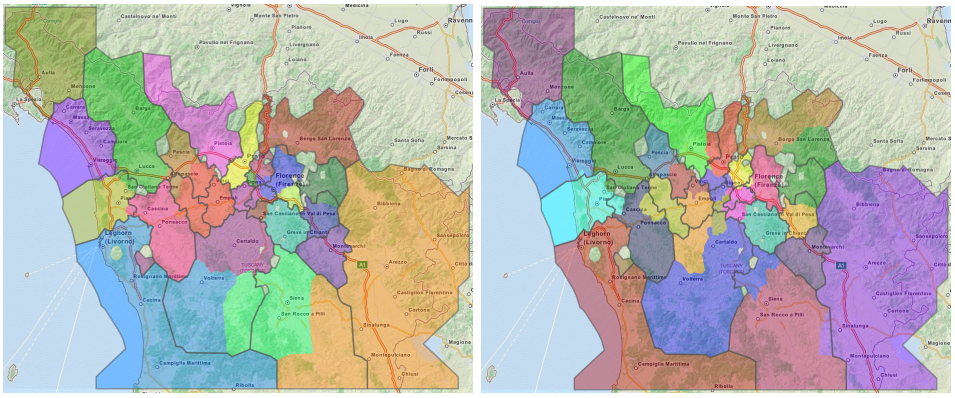

Figure 5.23: *Mobility Borders* results for $\epsilon = 0.2$ and $\langle \epsilon = 0.2, \delta = 0.2 \rangle$ compared with results from original data

of ϵ, i.e., zones in the same group in the transformed data are in the same group also in the original one, for the motivations discussed above. Recall (Figure 5.24(right)), instead, tends to decrease for $\epsilon < 0.3$, i.e., each zone of a group is no longer labeled as belonging to it, but the overall result is increased by augmenting δ to 0.2 or 0.3.

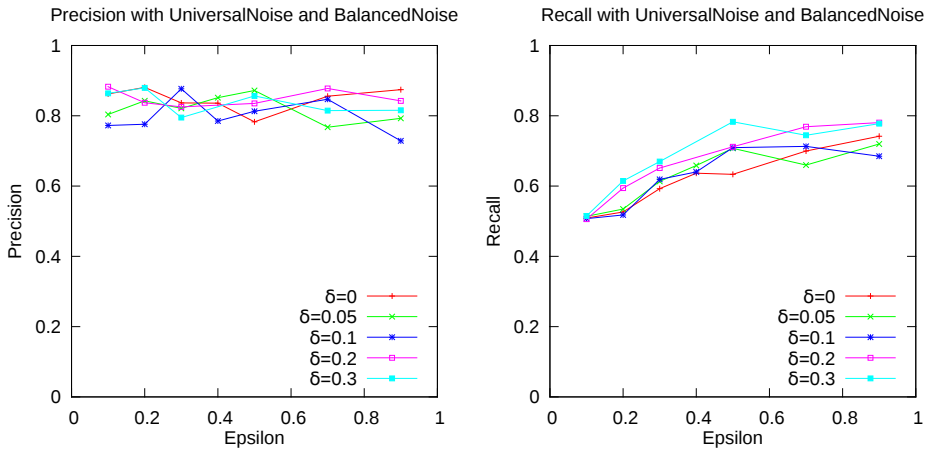

Figure 5.24: Quantitative measure for *Mobility Borders*

Spatial Distribution of the errors. We also studied the impact of the parameters of the privacy transformation in more detail by analyzing the spatial distributions of the errors, expressed as the logarithms of the ratios of the aggregated traffic values obtained from transformed data to those obtained from the original data. The use of the logarithms allowed us to reduce the impact of local outliers. The study was done using the results of 99 runs for all combinations of the values of ϵ from 0.1 to 0.9 with the step 0.1 and the values of delta 0.01, 0.02, 0.025, 0.03, 0.05, and up to the value 0.2 with the step 0.025. The corresponding 99 spatial distributions of the errors were clustered by similarity using the k-means methods. We experimented with different k and found that, starting from $k = 9$, increasing the value of k just subdivides small clusters into yet smaller ones, mostly singletons. There is one large cluster (Cluster 7) consisting of 68 distributions that preserves when k increases. This cluster consists of the spatial distributions with the best (i.e., lowest) values of the errors.

The area-wise median errors for this cluster are shown in the map in Figure 5.25 (left) by color-coding. Light yellow corresponds to values close to 0, shades of orange and red represent overestimates and shades of blue underestimates. The color legend is shown on the right of Figure 5.25. The prevalence of light yellow and light shades of orange means that the absolute values of the errors in cluster 7 are quite low. There are only a few high overestimates occurring in areas with low traffic density. Cluster 7 includes all spatial distributions for values of $\epsilon = 0.4$ and higher and values of delta from 0.01 to 0.05 and almost all spatial distributions for epsilon 0.6 and

higher irrespective of the value of δ. Hence, starting from $\epsilon = 0.6$, δ has no impact on the data quality. For comparison, the map on the right of Figure 5.25 represents the errors in another cluster, which includes the spatial situations for $\epsilon = 0.2$. Very high overestimates occur almost everywhere. For $\epsilon = 0.1$, the overestimates are even higher. This study clarifies what combinations of the parameter values should be used to obtain good results in terms of utility of the transformed data.

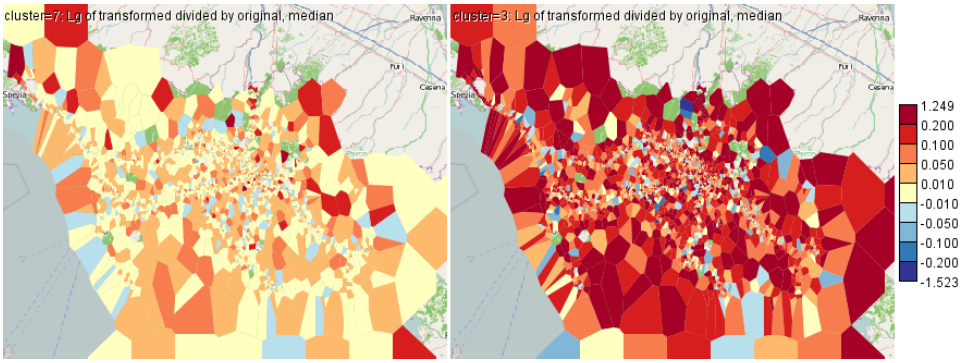

Figure 5.25: Comparison of the spatial distributions of the errors for different value combinations of ϵ and δ.

5.4.4 Evaluation of Sketching Transformations

Up to this point, we focused our study only on one step of our general approach described in Chapter 4, i.e., on the privacy-preserving transformation. Now, we want to investigate the fourth step, i.e., the sketching of the frequency vectors (Section 4.2.4). We tried to apply three types of sketches: AGMS (Section 2.2.1), Count-Min (Section 2.2.2) and Count (Section 2.2.3) sketches. Each kind of sketch was run with different combinations of parameters α and γ, i.e., different sizes. For the generation of hash functions, required by the different methods, we have relied on the implementation of Rusu and Dobra, available at [77].
We analyzed again the Pearson correlation, and we report in Figure 5.26 the values obtained for all network-based measures, starting by the use of the *UniversalNoise* approach with $\epsilon = 0.5$. Each cluster represents the size of the sketches (in Table 5.3-5.5 we show the exact parameters used) and each bar represents the kind of sketch (red means no sketch, i.e., the correlation

between the differentially private global frequency vector and the original one). We stress that, in our experiments, the frequency vectors have about 15,900 elements. We investigate the results, for each kind of sketch, in the following.

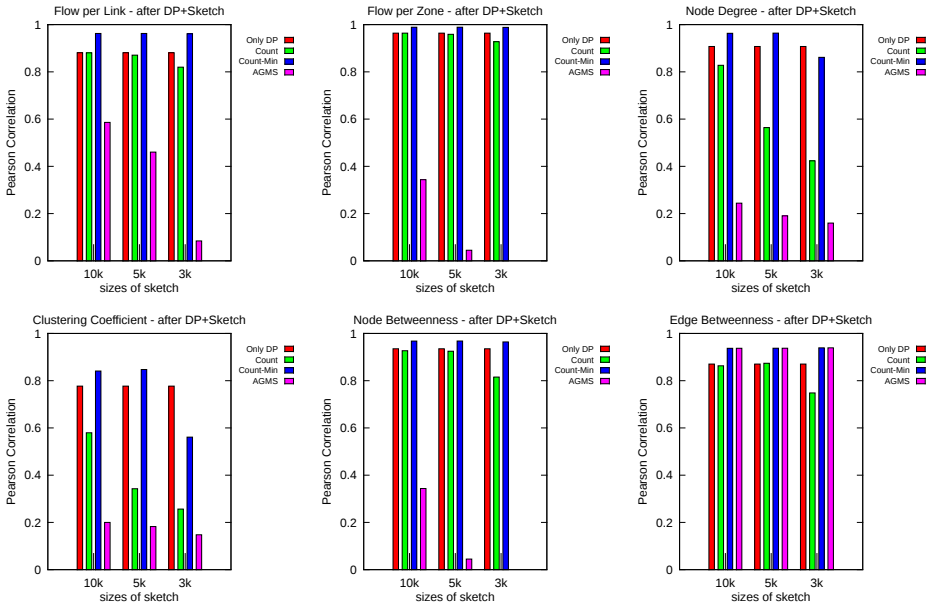

Figure 5.26: Comparison of correlation of network-based measures after the sketching transformation

AGMS sketch. Among the sketches that we considered, AGMS sketches are the most simple sketches, but they are the slowest (we recall they require linear time on the sketch size) and the ones that give worse results. As one can see, the correlation values are very low for all the measures except for the *Flow per Link* and for the *Edge Betweenness*. However, other measures strictly related to these two (respectively, *Flow per Zone* and *Node Betweenness*) are extremely different, therefore we can argue that these correlations are not very indicative. This fact is confirmed by the scatter plot reported in Figure 5.27 (a), where we can see that the values are completely not correlated.

We believe that AGMS sketches are not suitable to reduce communication while maintaining good data quality, at least in our particular setting. We must consider, however, that they have been designed for different goals, such as the estimate of the sum of the squares of the frequencies rather than

for the estimation of the single frequencies. Only in recent years they have been used for this aim.

	α	γ	$size$
$AGMS_{3k}$	0.03162	0.05	3,000
$AGMS_{5k}$	0.03162	0.01	5,000
$AGMS_{10k}$	0.03162	0.00005	10,000

Table 5.3: AGMS sketch size for different values of α and γ.

Count-Min sketch. The Count-Min sketches provide amazing results with regard to the correlations: PCC values of all measures are very close to 1. However, by further investigating, we realized that these results are only apparent: in Figure 5.27 (b) we show an example to explain the reason of these correlations. The estimated values are very compact around the regression line, but the latter is quite far away from optimal values, thus we can say that some kind of correlation with the original values exists, but there is an overestimation of the flows. The reason for this behavior is that Count-Min requires only positive values as input, so each vehicle must perform a preprocessing phase in which it sets to 0 all the negative values obtained at the end of the privacy-preserving step. Hence, a lot of values are flattened upwards and there is no longer any compensation between positive and negative values. Indeed, we have seen that when there is not the problem of managing negative values (as in the case of *BoundedNoise*) the Count-Min is the method that performs better.

	α	γ	Columns (w)	Rows (d)	$w \times d$
CM_{3k}	0.002	0.05	1,000	3	3,000
CM_{5k}	0.0008	0.01	2,500	2	5,000
CM_{10k}	0.0008	0.02	2,500	4	10,000

Table 5.4: Count-Min sketch size for different values of α and γ.

Count sketch. We can see how the Count sketch transformation preserve the PCC for the two measures *Flow per Link* and *Flow per Zone*. The former measures are well preserved since the sketch framework has been proposed for the compression of large arrays like those considered in this application. To the same extent, the second measure is well preserved when higher compression rates are reached. In Figure 5.27 we show that the val-

ues of *Flow per Link* are not so sparse, and there is no strange behavior. Moreover, this scatter plot is very similar to the one obtained by comparing the differential private and the original values, so this means that the data compression introduced a small approximation with respect to the private data. It is interesting to note how the topological properties of the graph, like *Clustering Coefficient* and *Node Degree*, are ruined after the compression. The measures of betweenness are well preserved even if they suffer for high rates of compression. In conclusion, we can state these sketches give good results and they can be used while preserving the utility.

	α	γ	Columns (w)	Rows (d)	$w \times d$
C_{3k}	0.03162	0.05	1,000	3	3,000
C_{5k}	0.03162	0.01	1,000	5	5,000
C_{10k}	0.03162	0.00005	1,000	10	10,000

Table 5.5: Count sketch size for different values of α and γ.

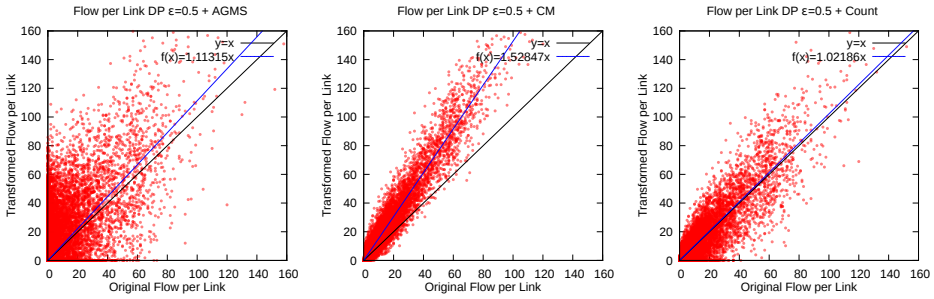

Figure 5.27: Correlation of *Flow per Link* after the sketching transformation by varying the kind of sketch

Conclusion

The issue of protecting privacy while releasing personal data is scientifically interesting and has been thoroughly studied particularly in the context of relational data. Unfortunately, only a few studies have addressed this issue in the context complex and *big data*, like spatio-temporal data, even if, as of today, we have an ever-growing diffusion of this kind of data. This new form of data are semantically rich and this makes difficult to find an efficient privacy transformation; moreover often traditional privacy-preserving techniques for relational databases are inadequate.

In this thesis we have studied the problem of protecting individual privacy in a distributed system where the goal is analyzing movement data. The data distribution makes the problem of the privacy protection more challenging. We have proposed the application of the *privacy-by-design* paradigm in the design and developing of three privacy transformation methods. They are based on the well-known (but seldom used for mobility data) notion of differential privacy that provides very effective data protection guarantees. Each solution is characterized by a different trade-off between privacy and data utility. In particular in our framework each vehicle, before sending the information about its movements within a time interval, applies a transformation to the data to achieve privacy and then it can create a summarization of the private data (by using a sketching algorithm) to reduce the amount of information to be transmitted.

This thesis presents two main contributions: (a) a framework that allows making available mobility data, while guaranteeing individual privacy for the people which the data refer to, through the application of one of the three possible methods, each one with a different guarantee; and (b) a framework that allows evaluating the proposed algorithms from the point of view of achievable utility after the privacy-preserving process. The goal reached by

our framework is double: on one hand, it allows us to provide a good level of privacy-protection; on the other hand, it allows to maintain a reasonable quality of the data, so that private data could be used for further analyses. Specifically, it gives us the opportunity to obtain private data that preserve peculiar features, as to guarantee a good quality of the analyses which can be performed on these data. For this purpose, the evaluation framework we have proposed presents: (1) functions that enable to verify if any basic statistics of the data are preserved; (2) quantitative measures that enable to verify the quality of private data; and (3) examples of analyses that can be performed on private data, compared to the ones carried out directly on the original data.

We have validated the robustness and efficiency of our privacy-preserving data aggregation methods by extensive experiments on large, real GPS data and our finding is that the proposed privacy-preserving techniques could achieve good results in terms of utility, preserving some meaningful properties of original data and keeping them usable for many analyses and applications.

Clearly, the proposed privacy-preserving framework is only one of the possible ways to address the issue we studied. As an example, different variants on the computation of the sensitivity in differential privacy are worth considering, to protect privacy on a different level. Additionally, future investigations could be directed to explore other methods to achieve differential privacy; as an example, it would be interesting to understand the impact of the use of the geometric mechanism instead of the Laplace one to achieve differential privacy.

Bibliography

[1] Osman Abul, Francesco Bonchi, and Mirco Nanni, *Anonymization of moving objects databases by clustering and perturbation*, Information Systems, Volume 35, Number 8, December 2010.

[2] Osman Abul, Francesco Bonchi, and Mirco Nanni, *Never walk alone: Uncertainty for anonymity in moving objects databases*, in ICDE, pages 376-385, 2008.

[3] Charu C. Aggarwal and Philip S. Yu, *A general survey of privacy preserving Data Mining models and algorithms*, in Charu C. Aggarwal and Philip S. Yu (edited by), *Privacy-Preserving Data Mining: models and algorithms*, Advances in Database Systems Vol. 34 Springer 2008, pages 11-52.

[4] Charu C. Aggarwal and Philip S. Yu, *A survey of randomization methods for Privacy Preserving Data Mining*, in Charu C. Aggarwal and Philip S. Yu (edited by), *Privacy-Preserving Data Mining: models and algorithms*, Advances in Database Systems Vol. 34 Springer 2008, pages 137-156.

[5] Charu C. Aggarwal and Philip S. Yu, *On Privacy-Preservation of Text and Sparse Binary Data with Sketches*, SIAM Data Mining Conference, 2007.

[6] Charu C. Aggarwal, *On randomization, public information and the curse of dimensionality*, in Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, pages 136-145, 2007.

[7] Rakesh Agrawal and Ramakrishnan Srikant, *Privacy-Preserving Data Mining*, SIGMOD international conference on Management of data, pages 439-450, 2000

[8] Réka Albert and Albert-László Barabási, *Statistical mechanics of complex networks*, CoRR cond-mat/0106096 (2001)

[9] Noga Alon, Phillip B. Gibbons, Yossi Matias, and Mario Szegedy, *Tracking Join and Self-Join Sizes in Limited Storage*, PODS 1999, pages 10-20

[10] Noga Alon, Yossi Matias, and Mario Szegedy, *The Space Complexity of Approximating the Frequency Moments*, J. Comput. Syst. Sci. 58(1), pages 137-147,1999

[11] Noga Alon, László Babai, and Alon Itai, *A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem*, J. Algorithms (JAL) 7(4), pages 567-583, 1986

[12] Natalia V. Andrienko and Gennady L. Andrienko *Spatial Generalization and Aggregation of Massive Movement Data*, IEEE Trans. Vis. Comput. Graph. 17(2),pages 205-219, 2011

[13] Michael Backes and Sebastian Meiser, *Differentially Private Smart Metering with Battery Recharging*, IACR Cryptology ePrint Archive 2012

[14] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar, *Privacy, accuracy, and consistency too: a holistic solution to contingency table release*, PODS 2007: 273-282

[15] Amos Beimel, Kobbi Nissim, Eran Omri, *Distributed Private Data Analysis: Simultaneously Solving How and What*, CRYPTO 2008, pages 451-468

[16] Claudio Bettini, Sergio Mascetti, and Xiaoyang Sean Wang *Privacy Protection through Anonymity in Location-based Services*, Handbook of Database Security 2008, pages 509-530

[17] Claudio Bettini, X. Sean Wang, and Sushil Jajodia, *Protecting privacy against location-based personal identification*, Secure Data Management 2005, pages 185-199 (Privacy and Security Support for Distributed Applications)

[18] Raghav Bhaskar, Srivatsan Laxman, Adam Smith, and Abhradeep Thakurta, *Discovering frequent patterns in sensitive data*, KDD 2010, pages 503-512

[19] T.-H. Hubert Chan, Elaine Shi, and Dawn Song, *Optimal Lower Bound for Differentially Private Multi-party Aggregation*, ESA 2012, pages 277-288

[20] Moses Charikar, Kevin Chen, and Martin Farach-Colton, *Finding frequent items in data streams*. Theor. Comput. Sci. (TCS) 312(1), pages 3-15, 2004

[21] David Chaum, Claude Crpeau, and Ivan Damgård, *Multiparty Unconditionally Secure Protocols* (Extended Abstract), STOC 1988, pages 11-19

[22] Keke Chen and Ling Liu, *A survey of multiplicative perturbation for privacy preserving Data Mining*, in Charu C. Aggarwal and Philip S. Yu (edited by), *Privacy-Preserving Data Mining: models and algorithms*, Advances in Database Systems Vol. 34 Springer 2008, pages 155-180.

[23] Lei Chen, M.Tamer Özsu, and Vincent Oria, *Robust and fast similarity search for moving object trajectories*, in Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD-05), 2005.

[24] Rui Chen, Benjamin C. M. Fung, Bipin C. Desai, and Nériah M. Sossou, *Differentially private transit data publication: a case study on the Montreal transportation system*, KDD 2012, pages 213-221

[25] Graham Cormode, Minos N. Garofalakis, Peter J. Haas, and Chris Jermaine *Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches*, Foundations and Trends in Databases 4(1-3), pages 1-294

[26] Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu, *Differentially Private Spatial Decompositions*, ICDE 2012, pages 20-31

[27] Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Thanh T. L. Tran, *Differentially private summaries for sparse data*, ICDT 2012, pages 299-311

[28] Graham Cormode and Muthu Muthukrishnan, *Approximating Data with the Count-Min Sketch*, IEEE Software (SOFTWARE) 29(1), pages 64-69, 2012

[29] Graham Cormode and Marios Hadjieleftheriou, *Methods for finding frequent items in data streams* VLDB J. (VLDB) 19(1), pages 3-20, 2010

[30] Graham Cormode and Minos Garofalakis, *Approximate continuous querying over distributed streams.* ACM Trans. on Database Systems (2008) Volume: 33, Issue: 2, Publisher: ACM, pages 1-39

[31] Graham Cormode and S. Muthukrishnan, *An improved data stream summary: the count-min sketch and its applications*, J. Algorithms 55(1), pages 58-75, 2005

[32] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li, *Differentially private data cubes: optimizing noise sources and consistency*, SIGMOD Conference 2011, pages 217-228

[33] Josep Domingo-Ferrer, and Rolando Trujillo-Rasua, *Microaggregation- and permutation-based anonymization of movement data*, Inf. Sci. 208: 55-80 (2012)

[34] Cynthia Dwork, *Differential Privacy: A Survey of Results*, TAMC 2008, pages 1-19

[35] Cynthia Dwork, *Differential privacy*, in Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (edited by), ICALP (2), volume 4052 of Lecture Notes in Computer Science, pages 1-12. Springer, 2006.

[36] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. *Calibrating noise to sensitivity in private data analysis*, in Proceedings of the Third conference on Theory of Cryptography (TCC), pages 265-284.

[37] Dan Feldman, Amos Fiat, Haim Kaplan, Kobbi Nissim, *Private coresets*, STOC 2009, pages 361-370

[38] Linton C. Freeman, *A Set of Measures of Centrality Based Upon Betweeness*, Sociometry, 40, , pages 35-41, 1977

[39] Arik Friedman, Assaf Schuster, *Data mining with differential privacy*, KDD 2010, pages 493-502

[40] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan, *Private queries in LBS: anonymizer are not necessary*, in SIGMOD, pages 121-132, 2008

[41] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, Fabio Pinelli, Chiara Renso, Salvatore Rinzivillo, and Roberto Trasarti, *Unveiling the complexity of human mobility by querying and mining massive trajectory data*, in The VLDBJ, Vol. 20, Issue 5, pages 695-719, 2011

[42] Bobi Gilburd, Assaf Schuster, and Ran Wolff, *k-TTP: a new privacy model for large-scale distributed environments*, KDD 2004, pages 563-568

[43] Michelle Girvan and Mark E. J. Newman, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA 99, pages 78217826, 2002

[44] Shafi Goldwasser, *Multi-party computations: past and present*, in Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing, Santa Barbara, CA USA, August 21-24 1997

[45] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu, *Boosting the Accuracy of Differentially Private Histograms Through Consistency*, PVLDB 3(1), pages 1021-1032 2010

[46] Shen-Shyang Ho and Shuhua Ruan, *Differential Privacy for Location Pattern Mining*, Fourth ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS (SPRINGL), Chicago, IL, 1 Nov, 2011.

[47] Zhengli Huang, Wenliang Du, and Biao Chen, *Deriving Private Information from Randomized Data*, SIGMOD 2005, pages 37-48

[48] Christian S. Jensen, Hua Lu and Man Lung Yiu, *Location privacy techniques in client-server architectures*, in Lecture Notes in Computer Science, Volume 5599/2009, pages 31-58, 2009

[49] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar, *On the privacy preserving properties of random data perturbation techniques* ICDM 2003. Third IEEE International Conference on Data Mining, 2003.

[50] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh, *An Anonymous Communication Technique using Dummies for Location-based Services*,

Proceeding ICUIMC '09 Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, pages 290-297 ACM New York, NY, USA 2009

[51] Daniel Kifer and Ashwin Machanavajjhala, *No free lunch in data privacy*, SIGMOD Conference 2011, pages 193-204

[52] Ninghui Li, Wahbeh H. Qardaji, Dong Su, and Jianneng Cao, *PrivBasis: Frequent Itemset Mining with Differential Privacy* PVLDB 5(11), pages 1340-1351, 2012

[53] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian, *t-closeness: Privacy beyond k-anonymity and l-diversity*, in Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, pages 106-115. IEEE, 2007.

[54] Yehuda Lindell and Eran Omri, *A Practical Application of Differential Privacy to Personalized Online Advertising*, IACR Cryptology ePrint Archive 2011: 152, 2011

[55] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam, *l-diversity: Privacy beyond k-anonymity*, in Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, page 24. IEEE Computer Society, 2006.

[56] Sergio Mascetti, Claudio Bettini, Dario Freni, and X. Sean Wang, *Spatial Generalization Algorithms for LBS Privacy Preservation*, Journal of Location Based Services, 2(1), 2008.

[57] Frank McSherry and Kunal Talwar, *Mechanism design via differential privacy*, in Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 94103, 2007.

[58] Noman Mohammed, Rui Chen, Benjamin C. M. Fung, and Philip S. Yu, *Differentially private data release for data mining*, KDD 2011, pages 493-501

[59] Anna Monreale, Roberto Trasarti, Dino Pedreschi, Chiara Renso, and Vania Bogorny *C-safety: a framework for the anonymization of semantic trajectories*, Transactions on Data Privacy 4(2), pages 73-101, 2011

[60] Anna Monreale, *Privacy by Design in Data Mining*, PhD thesis, University of Pisa, 2011

[61] Anna Monreale, Gennady Andrienko, Natalia Andrienko, Fosca Giannotti, Dino Pedreschi, Salvatore Rinzivillo, and Stefan Wrobel, *Movement data anonymity through generalization*, Transactions on Data Privacy, 3(2), pages 91-121, August 2010

[62] Anna Monreale, Dino Pedreschi, and Ruggero G. Pensa, *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*, in Francesco Bonchi and Elena Ferrari (edited by), *Anonymity technologies for privacy-preserving data publishing and mining*, Chapman & Hall/CRC, pages 3-33, 2010

[63] Mirco Nanni, Roberto Trasarti, Giulo Rossetti, and Dino Pedreschi, *Efficient distributed computation of human mobility aggregates through User Mobility Profiles*, in KDD Int. Workshop on Urban Computing, pages 87-94, 2012

[64] Mehmet Ercan Nergiz, Maurizio Atzori, Yücel Saygin, and Barış Güç *Towards trajectory anonymization: a generalization-based approach*, in Proceedings of ACM GIS Workshop on Security and Privacy in GIS and LBS, 2008

[65] Mehmet Ercan Nergiz, Maurizio Atzori, and Yücel Saygin, *Perturbation-driven anonymization of trajectories*, Technical Report 2007-TR-017, ISTI-CNR, Pisa, 2007

[66] Mark E. J. Newman, *The Structure and Function of Complex Networks*, SIAM Review, 45:2, pages 167-256, 2003.

[67] Ruggero G. Pensa, Anna Monreale, Fabio Pinelli, and Dino Pedreschi, *Pattern-Preserving k-Anonymization of Sequences and its Application to Mobility Data Mining*, in Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, 2008.

[68] Andreas Pfitzmann and Marit Köhntopp, *Anonymity, unobservability, and pseudonymity - a proposal for terminology*, in Designing Privacy Enhancing Technologies: International Workshop on Design Issues in

Anonymity and Unobservability, Volume 2009 of LNCS., Springer (July 2000) 1-9

[69] Vibhor Rastogi and Suman Nath, *Differentially private aggregation of distributed time-series with transformation and encryption*, SIGMOD Conference 2010, pages 735-746

[70] Daniele Riboni, Linda Pareschi, Claudio Bettini, and Sushil Jajodia, *Preserving Privacy in LBS against Attacks based on Concurrent Requests*, technical Report TR 24-07, University of Milan, 2007

[71] Daniele Riboni, Linda Pareschi, and Claudio Bettini, *Privacy in Georeferenced Context-aware services: a Survey*, Privacy in Location-Based Applications 2009, pages 151-172 PiLBA-08 Privacy in Location-Based Applications Workshop co-located with ESORICS 2008 Malaga, Spain, October 9, 2008 Proceedings

[72] Salvatore Rinzivillo, Simone Mainardi, Fabio Pezzoni, Michele Coscia, Dino Pedreschi, and Fosca Giannotti, *Discovering the Geographical Borders of Human Mobility*, KI 26(3), pages 253-260, 2012

[73] Florin Rusu and Alin Dobra, *Sketches for Size of Join Estimation*, ACM Trans. Database Syst. (TODS) 33(3), 2008

[74] Florin Rusu and Alin Dobra, *Pseudo-Random Number Generation for Sketch-Based Estimations*, ACM Transactions on Database Systems, 2007, Volume: 32, No: 2, Publisher: Article: 11

[75] Florin Rusu and Alin Dobra *Statistical analysis of sketch estimators*, SIGMOD 2007, pages 187-198

[76] Florin Rusu and Alin Dobra, *Fast Range-Summable Random Variables for Efficient Aggregate Estimation*, Proceedings of the 2006 ACM SIGMOD international conference on Management of data SIGMOD 06, 2006, Publisher: ACM Press

[77] Florin Rusu and Alin Dobra, *Sketches for Size of Join Estimation*, http://faculty.ucmerced.edu/frusu/Projects/Sketches/sketches.html

[78] Ashish P. Sanil, Alan F. Karr, Xiaodong Lin, and Jerome P. Reiter, *Privacy preserving regression modelling via distributed computation*, KDD 2004, pages 677-682

[79] Elaine Shi, T-H. Hubert Chan, Eleanor Rieffel, Richard Chow, and Dawn Song, *Privacy Preserving Aggragation of Time-Series Data*, Network & Distributed System Security Symposium (NDSS), 2011

[80] Latanya Sweeney, *k-anonymity: A model for protecting privacy*, International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems, 2002

[81] Latanya Sweeney, *Uniqueness of simple demographics in the u.s. population*, technical report, Laboratory for International Data Privacy, Carnegie Mellon University, Pittsburgh, PA, 2000

[82] Xiaokui Xiao, Guozhang Wang, Johannes Gehrke, *Differential Privacy via Wavelet Transforms*, IEEE Trans. Knowl. Data Eng. 23(8), pages 1200-1214, 2011

[83] Xiaokui Xiao and Yufei Tao, *Anatomy: Simple and effective privacy preservation*, in Proceedings of the 32nd International Conference on Very Large Data Bases, pages 139-150, ACM 2006.

[84] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, *Differentially Private Histogram Publication* ICDE 2012, pages 32-43

[85] Jianliang Xu, Jing Du, Xueyan Tang, and Haibo Hu, *Privacy-Preserving Location-based Queries in Mobile Environments*, technical Report, Hong Kong Baptist University, 2006

[86] Roman Yarovoy, Francesco Bonchi, Laks V. S. Lakshmanan, Wendy Hui Wang *Anonymizing moving objects: how to hide a MOB in a crowd?* EDBT 2009, pages 72-83

[87] Andrew Chi-Chih Yao, *Protocols for Secure Computations* (Extended Abstract), FOCS 1982, pages 160-164