UNIVERSITÀ DI PISA

Department of Computer Science, Pisa
PhD Thesis

Modeling & Predicting Privacy Risk in Personal Data

Roberto Pellungrini

Supervisor:
Prof. Anna Monreale

Supervisor:
Prof. Dino Pedreschi

Abstract

Privacy in Big Data analytics is one of the most important issues that analysts and businesses face when managing personal data. In a privacy preserving analysis process, the privacy risk on the individuals represented in the data is firstly evaluated, then the data is appropriately modified in order to preserve privacy while at the same time maintaining a certain level of data quality. In this thesis we focus on privacy risk assessment, proposing new models and algorithms to deal with this fundamental part of privacy aware systems. We propose some extensions to an existing state-of-the-art privacy risk assessment framework, to improve on existing literature. Then, we propose a classification based methodology to predict privacy risk. We validate our proposal on three different types of real world data: human mobility, retail and social network data. Finally we propose a new model for the behavior of an adversary in human mobility data, leveraging the natural structure and constraints of this kind of data.

Acknowledgements

I'd like to thank my supervisors, Anna Monreale and Dino Pedreschi, for their continuous support and for contributing significantly to this work. I'd like also to thank Luca Pappalardo, Francesca Pratesi, Filippo Simini and Riccardo Guidotti, with whom I've collaborated several times and that contributed to the work here presented. I'd also want to acknowledge the important roles of professors Chiara Bodei and Franco Turini, internal committee, for their poignant critique and evaluation of the thesis. I'd like to thank reviewers Wendy Hui Wang and Vivenc Torra, for the care with which they reviewed and corrected the thesis. Finally, I'd like to thank professor Degano and professor Ferragina, Ph.D. presidents during my years of enrollment.

Contents

1 Introduction **6**
1.1 Organization and Contributions 8

2 Related Works **10**
2.1 European Legislation 10
2.1.1 Privacy Actors 11
2.2 Privacy preserving methodologies 12
2.2.1 Randomization based methods 12
2.2.2 Differential Privacy for sequential data 13
2.2.3 Anonymity based methods 14
2.2.4 Distributed privacy protecting methods 15
2.3 Risk analysis techniques 16
2.3.1 Risk of re-identification 16
2.3.2 Risk of attribute disclosure and inference 17
2.3.3 Risk analysis frameworks 18
2.4 Privacy In Complex Data 18

3 Data Modeling **21**
3.1 Mobility Data modeling 21
3.1.1 Mobility Data Metrics 22
3.2 Retail Data modeling 23
3.2.1 Retail Data Metrics 24
3.3 Social Network Data modeling 25
3.3.1 Social Network Data Metrics 27
3.4 Experimental Datasets 28
3.4.1 Experimental Mobility Dataset 28
3.4.2 Experimental Retail Dataset 29
3.4.3 Experimental Network Dataset 29

4 Privacy Risk & Data Quality Assessment **31**
4.1 PRUDEnce Privacy Risk Assessment Framework 31
4.1.1 PRUDEnce Extension 36
4.2 Evaluating Risk & Data Quality 39
4.2.1 Privacy Attacks on Mobility data 40
4.2.2 Mobility privacy risk assessment in scikit-mobility 44

4.2.3 Risk Distributions on Mobility data . . . 46
4.2.4 Mobility Data Quality Experiments . . . 46
4.2.5 Privacy Attacks on Retail data . . . 53
4.2.6 Risk Distributions on Retail data . . . 54
4.2.7 Individual Patterns Risk Experiments . . . 55
4.2.8 Retail Data Quality Experiments . . . 57
4.2.9 Privacy Attacks on Network data . . . 61
4.2.10 Risk Distributions on Network data . . . 64
4.2.11 Social Network Data Quality Experiments . . . 66
4.2.12 Discussion . . . 70

5 Privacy Risk Prediction **72**
5.1 Computational Complexity of PRUDEnce . . . 72
5.2 A Data Mining approach for Privacy Risk Assessment . . . 73
5.2.1 Construction of training dataset . . . 74
5.2.2 Usage of the data mining approach . . . 76
5.3 Privacy Risk Prediction Experiments . . . 76
5.3.1 Privacy Risk Prediction for Mobility Data . . . 77
5.3.2 Privacy Risk Prediction for Retail Data . . . 83
5.3.3 Privacy Risk Prediction for Social Network Data . . . 87
5.4 Discussion . . . 92

6 Modeling Adversarial Behavior Against Mobility Data Privacy **93**
6.1 Trajectory Modeling: variation . . . 93
6.2 Problem Statement . . . 95
6.3 Construction of the Adversary Trajectory . . . 98
6.3.1 Real Adversary Trajectory . . . 98
6.3.2 Synthetic Adversary Trajectory . . . 98
6.3.3 Simulated Adversary Trajectory . . . 99
6.4 Experiments . . . 104
6.4.1 Dataset of real trajectories . . . 104
6.4.2 Generation of synthetic trajectories . . . 104
6.4.3 Experimental Results . . . 105
6.4.4 Simulated Annealing Analysis . . . 107
6.4.5 Performance analysis of Simulated Annealing . . . 108
6.4.6 Discussion . . . 110

7 Conclusions and Future Works **112**

Bibliography **115**

List of Figures

4.1 The general schema of the PRUDEnce privacy framework 32
4.2 Cumulative distributions of privacy risk for Florence dataset. 47
4.3 Cumulative distributions of privacy risk for Pisa dataset. 48
4.4 Metrics distribution in mobility data Florence 49
4.5 Metrics distribution in mobility data Pisa 50
4.6 Example of MUC on mobility data 1 51
4.7 Example of MUC on mobility data 2 51
4.8 Cumulative distributions of privacy risk for retail data. 55
4.9 Metrics distribution in retail data 1 58
4.10 Metrics distribution in retail data 2 59
4.11 Examples of MUC for various attacks in retail data 60
4.12 Cumulative distributions of privacy risk for social network data. 65
4.13 Some examples of distributions of network metrics for the Neighborhood Pair attack. 67
4.14 Distributions of network metrics for the Neighborhood attack. 68
4.15 Examples of MUC for various attacks on social networks 69

5.1 Classification error per class for classifier on mobility data 79
5.2 The distribution of average importance of the mobility features for all the classifiers (Florence dataset). 81
5.3 Classification error per class for classifiers on retail data 84
5.4 The distribution of average importance of the retail features for all the classifiers. 86
5.5 Classification error per class for classifiers on social network data 89
5.6 The distribution of average importance of the social network features for all the classifiers. 91

6.1 Distribution of Average Adversary Risk for real and synthetic adversaries . 105
6.2 Cumulative distribution of Privacy Risk for individuals attacked by the best adversary 106
6.3 Variation of Average Adversary Risk in time 107
6.4 Visualization of the worst adversary trajectories 109
6.5 Variation of Average Adversary Risk by distance limit and exponential cooling rate 110
6.6 Variation of the Average Adversary Risk of the current and best solution . 111

List of Tables

3.1 The individual mobility metrics used in our work. 23
3.2 The individual retail data metrics used in our work. 25
3.3 The individual social network data metrics used in our work. 28

4.1 Numerical values of MUC for mobility data of Florence 52
4.2 Numerical values of MUC for mobility data of Pisa 53
4.3 Results of an attack with simple baskets against retail data 56
4.4 Characterization of matched individuals in the TX-means patterns against baskets attack 56
4.5 Characterization of non matched individuals in the TX-means patterns against baskets attack 57
4.6 Numerical values of MUC for retail data 61
4.7 Numerical values of MUC for social network data 70

5.1 Results of the classification experiments for the Florence and the Pisa datasets 80
5.2 The average importance of every mobility feature 82
5.3 Comparison of execution times for mobility data 82
5.4 Results of the classification experiments for the retail datase 85
5.5 Results of the classification experiments for the retail dataset with rebalancing 85
5.6 The average importance of every retail feature 87
5.7 Comparison of execution times for retail data 87
5.8 Results of the classification experiments for the social network dataset . . . 90
5.9 The average importance of every social network feature 91
5.10 Comparison of execution times of attack simulations and classification tasks on social network data. 92

6.1 Summary of five datasets characteristics 104
6.2 Mobility analysis of the most efficient adversaries Pisa 108
6.3 Mobility analysis of the most efficient adversaries Florence 108

Chapter 1

Introduction

In recent years, the so called "Big Data" have become the most sought after commodity by businesses and enterprises alike. It has been estimated that people generate 2.5 quintillion bytes of data every day.[1] This enormous quantity of data is today at the forefront of innovation and research, as they allow us to study the behavior of people on a new, much bigger scale and from different perspectives. Some examples of big data include data about purchases made by customers of business activities, data about the movement of individuals in urban areas, data about social networks etc. Big Data are generally collected automatically through various means: these can range from mobile devices, increasingly more present in our lives today, or through the recording of transactions, or simply by monitoring the behavior of an individual in his interactions with online services or applications. To deal with this huge amount of data, many techniques have been developed to extract useful information from the data itself. These techniques are used to find patterns, models and rules inside the data, in order to make predictions about the studied phenomena [108]. Data mining and machine learning are today at the pinnacle of scientific research and to keep on improving in these fields, huge amounts of data is required to train and validate machine learning models. However, it is almost always the case that data contains very personal information about the individuals represented. There exist a serious risk of privacy violations for the people involved: highly sensitive and personal information about individuals can be extracted from the data, leading to dangerous privacy leaks and with the seamless usage of mobile devices and online services, people might share private information even without realizing it. Basic data protection measures like de-identification are not enough to guarantee the privacy of individuals in some particular contexts. By linking data from different sources it is indeed possible to re-identify an individual in multiple datasets. One such event was the case of the linking of two public datasets: Internet Movie Database and the the Netflix Prize Dataset [101]. Therefore, Privacy has become an integral component of the design of business practices and analytical processes. Many privacy preserving solutions modify or transform the original data in order to mask individuals and protect them, thus distorting the original characteristics of the data in some way. The challenge in designing privacy protection methods is therefore to achieve privacy for as many individuals as possible while preserving

[1] http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/

the quality of the data, allowing meaningful analyses. This has been the focus of privacy related literature for the latest years: most of the frameworks proposed for data protection stemmed from the widely accepted paradigm of Privacy By Design, postulated by Ann Cavoukian in [29], and are mostly centralized frameworks where some condition is applied to the data in order to achieve privacy. Most privacy preserving techniques heavily modify the data in order to achieve privacy, thus greatly undermining subsequent analyses. To understand whether and which privacy preserving technique has to be applied on the data, tools have been developed to quantitatively analyze the risk of privacy violations for the individuals represented in the data. This privacy risk assessment techniques however, are mostly based on the assumption that an adversary, who wants to extract sensitive information about an individual in a dataset, has access to all the pre-existing knowledge, a worst-case scenario assumption, necessary to conduct the most damaging attack possible. This simple assumption usually leads to great overestimation of privacy risk with respect to real world data. In [18], the authors describe this situation from the perspective of businesses and enterprises: either for the fear of disclosing sensitive information, or because of the lack of mutual trust, we run the risk of misjudging privacy risk (either overestimating the risk of underestimating it) and make an improper use of the data. It is therefore necessary to design better privacy preserving processes, that strive for a balance between the protection of the privacy of individual and the utility of the data for analyses. One of the fundamental steps in any privacy preserving process is privacy risk assessment, that is the process by which we try to understand which individuals represented in the data are at risk of a privacy violation and how much this risk is, if quantifiable. Recently, the General Data Protection Regulation in Europe [1] has bestowed the on data holders the responsibility to handle data in a privacy preserving way. It is therefore fundamental, for data holders, to evaluate quantitatively the privacy risk in the data they are managing, as to better understand which privacy preserving process they can enact to protect the privacy of individuals. Many methodologies have been proposed to evaluate the privacy risk of individuals in any kind of data [35, 148, 107, 72, 91]. Our aim, in this thesis, is to provide an improvement in privacy risk assessment, by proposing new models and algorithms to efficiently assess privacy risk. We move our research on different parallel directions described in the following.

First We present the state-of-the-art privacy risk assessment framework PRUDEnce. We present the mathematical formulation for a set of privacy attacks on three types of data (i.e., mobility, retail and social network data) showing how PRUDEnce can be instantiated in three different contexts by defining and analysing the threats that may harm the privacy of individuals. We show how risk is computed and how it is distributed in our experimental data.

Second we propose two extensions to the PRUDEnce framework: the *first one* is a methodology to evaluate the risk of individual patterns, focusing on purchasing patterns in retail data. This particular kind of analyses is one of the most common practice in mining retail data [10]. Our aim is to provide a methodology based on distance based record linkage to evaluate how much the purchasing patterns can be linked to the original data from which they where extracted. The *second one* is an extension that aims at incorporating into PRUDEnce a new methodology to assess the quality of data. While PRUDEnce provides a way to evaluate data quality based on the quantity of data that

is put at risk by an attack, we propose an alternative based on the quality of metrics calculated on the data itself. We show how the trend of these metrics changes depending on the level of privacy protection that we impose, by using the distance between the original distribution of the metric and the distribution on non-risky data. We test our methods on three different kinds of experimental data: mobility, retail and social network data.

Third Facing the issue of a daunting computational complexity for directly computing privacy risk, we then move our focus to possible methodologies to predict privacy risk using data mining techniques. We focus on a classification approach that is able to predict privacy risk based on individual metrics, i.e., on metrics calculated on the data of single individuals. Our approach has the objective of being simple and quick, usable by any data holder wanting an efficient way of finding out how many and which individuals are at risk in the data. We show how this approach can be applied to different data types.

Fourth Finally, we explore possible alternative models for adversarial behavior against mobility data privacy. In particular, we focus on the fact that existing privacy frameworks work on a worst-case scenario approach: they calculate risk on an individual basis, assuming that, for each individual, there can be a specific worst adversary able to produce the maximum privacy risk for that individual. These frameworks are based on adversarial models that does not consider the process of gathering the information about the individuals. Looking at mobility data, we propose to model the adversary as a mobile agent that collects information about the individuals in the data by moving in the same geographical space and respecting the spatio-temporal constraints. Then, we define a privacy risk assessment approach based on this adversarial model while maximizing the overall privacy risk of individuals in a dataset.

1.1 Organization and Contributions

This thesis is organized as follows: in Chapter 2 we give an overview of the literature relevant to the different topics that we address in this thesis. In Chapter 3 we give the mathematical modeling for the data that we use in our work: mobility, retail and social network data. In particular, after introducing for each data type a possible mathematical formulation of their structures, we introduce the metrics most commonly used in analyzing these kinds of data. We also present the characteristics of the experimental datasets used in the validation of our methods. This chapter works as a foundation for the understanding of the three different kinds of data, their nature, limitations and possibilities. In Chapter 4 we introduce the PRUDEnce privacy risk assessment framework. We provide the mathematical formulation for several attacks and show how we can compute privacy risk for the three different kinds of data. We then propose our extensions to the PRUDEnce framework and test them on the various experimental data. This part is mainly based on the following publications:

- Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. Analyzing privacy risk in human mobility data. In *Software Technologies: Applications*

and Foundations - STAF 2018 Collocated Workshops, Toulouse, France, June 25-29, 2018, Revised Selected Papers, pages 114–129, 2018

- Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. Fast estimation of privacy risk in human mobility data. In *SAFECOMP Workshops*, volume 10489 of *Lecture Notes in Computer Science*, pages 415–426. Springer, 2017

- Roberto Pellungrini, Anna Monreale, and Riccardo Guidotti. Privacy risk for individual basket patterns. In *ECML PKDD 2018 Workshops - MIDAS 2018 and PAP 2018, Dublin, Ireland, September 10-14, 2018, Proceedings*, pages 141–155, 2018

- Luca Pappalardo, Filippo Simini, Gianni Barlacchi, and Roberto Pellungrini. scikit-mobility: a python library for the analysis, generation and risk assessment of mobility data. arxiv:1907.07062, 2019

- Luca Pappalardo, Gianni Barlacchi, Roberto Pellungrini, and Filippo Simini. Human mobility from theory to practice: Data, models and applications. In *WWW (Companion Volume)*, pages 1311–1312. ACM, 2019

In Chapter 5 we propose our data mining approach to predict privacy risk in personal data with classification methods. We discuss thoroughly our approach by testing it on different data with different risk profiles and characteristics, and we show the improvements we make with respect to direct computation. This part of the thesis is based on the publication:

- Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. A data mining approach to assess privacy risk in human mobility data. *ACM TIST*, 9(3):31:1–31:27, 2018

In Chapter 6 we propose a new, alternative way of modeling adversary behavior in mobility data, by modeling the actual process with which the adversary gathers the information she uses. We show how this alternative approach works, and evaluate its benefits. This part of the thesis is based on:

- Roberto Pellungrini, Filippo Simini, Luca Pappalardo, and Anna Monreale. Modeling adversarial behavior against mobility data privacy. *Submitted to IEEE Transactions on Intelligent Transportation Systems*, 2019

Finally Chapter 7 concludes the thesis with our final remarks about the work we have done and a discussion on the possible future developments.

Chapter 2

Related Works

In this chapter we provide an overview of the existing literature regarding privacy preserving techniques and privacy risk analysis approaches. We start discussing important aspects of the European legal framework and then, we present the scientific literature addressing the problem of privacy in personal data.

2.1 European Legislation

A comprehensive legal approach to the data privacy problem has been given by the European Union in the Data Protection Directive redacted in 1995 [2]. This directive foresees rules for the handling of personal data and it includes a number of rights for data subjects. We can summarize the principles of the directive as:

- **Notice** Data subjects should be noticed whenever data about them is being collected.
- **Purpose** Data should be used only for the purpose stated by the collector.
- **Consent** The data subject must give his consent for the data to be disclosed.
- **Security** Data collected should be kept safe from abuses of any kind.
- **Disclosure** Data subjects should be informed regarding who is collecting their data.
- **Access** Data subjects should be allowed to access their data and to modify them to correct any mistakes.
- **Accountability** Data subjects should have ways to hold data collector accountable for the disregard of any of the previous principles.

More recently, in 2012 the European Union has proposed a reform to the data protection rules in Europe. On 4 May 2014, an official text was published for both the new Regulation and the new Directives [1]. The General Data Protection Regulation was adopted on 14 April 2016, and became enforceable beginning 25 May 2018. This reform aims to address a number of issues with the previous directives:

- Eliminating discrepancies with national or regional laws.
- Enriching privacy measures and safeness for the individuals
- Updating the laws to address contemporary issues like those posed by social media, drones, big data etc.
- Reducing bureaucratic and economic cost for companies dealing with data protection authorities and laws.

Some of the most important points worth nothing are:

- Stricter conditions for consent, defined as "freely given specific, informed and explicit indication of his or her wishes by which the data subject, either by a statement or by a clear affirmative action, signifies agreement to personal data relating to them being processed".
- The addition of biometric and genetic data to the set of sensitive data as indicated in Article 9.
- The right to be forgotten and erased as outlined in Article 17.
- Where a type of processing in particular using new technologies is likely to result in a high risk to the rights and freedoms of persons, Data Controllers shall, prior to the processing, carry out an assessment of the impact of the envisaged processing operations on the protection of personal data.
- Data Controllers have to put in place appropriate technical and organisational measures to implement the data protection principles and safeguard individual rights applying the privacy by design and by default principle.

The principles of *Privacy by Design*, adopted by the GDPR, was theorized in the nineties by Ann Cavoukian. These principles provide a proactive approach to privacy related issues when dealing with personal data of any kind. The basic concept of Privacy by Design is that privacy must be embedded into networked data systems and technologies by default becoming an integral part in the design, development and organization of business and analytical processes.[29] While general in its formulation, and not addressing specific methodologies for the actual compliance to the principles, in the past years Privacy by Design has been at the center of studies both from regulators and technical developers. Cavoukian herself gave a partial interpretation in [28] to better explain how to integrate Privacy by Design in a justice system.

2.1.1 Privacy Actors

First of all it is important to know who are the actors in a privacy aware environment. There are indeed different subjects whose privacy may be important in any analytical or business process. This issue was discussed first in [38], where three main actors are

identified, each one with different related privacy issues: data respondent, data holder and user.

Data respondent is the subject that generates the data in the first place. Data respondent are considered passive in the privacy process, in the sense that they do not take any direct action towards the prevention of their privacy. The main objective in protecting the privacy of a data respondent is to avoid the disclosure of her sensitive data. For example, the customers of an insurance company, that gathers information about their driving habits, are data respondents, and their interest is that the insurance company does not reveal sensitive information about them while managing their mobility data,.

Data holder is the subject, organization or individual, that gathers and maintains the data. The name "data owner" is often used in place of data holder in the literature [38, 55], and therefore these terms are considered equivalents. According to the GDPR [1], data holders are responsible for the implementations of privacy measures in order to guarantee the privacy protection of individuals involved in the data they gather, that means ensuring no relevant information contained in the database is disclosed. For example, a supermarket that shares the data of its customers with an analyst does not want the analyst to disclose sensitive information as this could violate the privacy of the database.

User is a subject that generates data through the use of a specific service and that has a direct participation in the protection of his own privacy. The main objective in protecting a user's privacy is to assure her privacy when accessing and using a specific service or system. In such a process, the user takes an active role.

There is also a fourth actor, which is the malicious third party which tries to attack one of the aforementioned actors. An *Adversary* is a subject whose interest is to disclose some information about a respondent, holder or user. In [8] the term adversary is used equivalently to the term "attacker", and the attack that this subject conducts is usually referred to as adversarial attack or privacy attack.

2.2 Privacy preserving methodologies

Several methodologies have been developed in order to preserve the privacy of individuals. The interest in these methodologies grew thanks to the increasing capability of storing and processing large amounts of data. In this section we provide a brief overview.

2.2.1 Randomization based methods

One of the earliest methods used to assure privacy protection is to perturb the data with some random noise [11]. The perturbed data can still be used to extract patterns or machine learning models. In *additive random perturbation* some noise is drawn from a distribution and added to each record of a dataset. The original record values can not be easily guessed from the distorted data while the distribution of the dataset can be easily recovered by using one of the methods discussed in [11, 9]. So, original records are not available, while it is possible to obtain distribution only along individual dimensions describing the behavior of the original dataset. This technique however modifies the statistics required for some commonly data mining models, thus requiring specific data

mining approaches to be tackled properly. For Bayesian classifiers, the authors of [171] propose a method to build a Naive Bayesian Classifier over perturbed data. For association rule mining, [170] use a similar methodology to mitigate the effect of perturbation.

In 2006, Dwork et al. introduced *Differential Privacy* model [41] that is based on randomization approach. The fundamental idea at the base of *Differential Privacy* is that an algorithm applied to two dataset that differ only on the data of a single individual should yield *almost* the same result. This means that the individual can confidently submit her record to the dataset because nothing, or almost nothing, can be discovered from the database with her information that could not have been discovered without her information. More formally a randomized algorithm A is ϵ-differentially private if for all datasets D and D' differing only on a single record, and for all $S \subseteq range(A)$ the property $Pr[A(D) \in S] \leq e^{\epsilon} Pr[A(D') \in S]$ holds. A relaxed version of differential privacy was proposed in [17] in which the author claim that the same level of privacy protection can be achieved even when admitting a small amount of privacy loss. Formally, the differential privacy property changes then to $Pr[A(D) \in S] \leq e^{\epsilon} Pr[A(D') \in S] + \delta$. Note that, with $\delta = 0$ we have the original definition of differential privacy. Differential privacy is generally achieved by adding noise the the results of the algorithm to be computed, e.g. drawing from the Laplace distribution [42]. There are however cases in which adding noise through the Laplace distribution may not be feasible. For the analysis whose outputs are not real or make no sense after adding noise, the authors of [90] propose an exponential mechanism, selecting an output from the output domain, $r \in R$, by taking into consideration its score of a given utility function q. In [43] authors give estimates for the ϵ parameter, stating how it can still produce meaningful results when assuming values larger than 1. Further work has been done in researching possible relaxation of the differential privacy property in Since its inception many differentially private algorithms and methods have been proposed. A thorough discussion can be found in [44]. In recent years there have been many works that utilized differential privacy in various contexts, for example, [97] for movement data, or [151] for network data. Much attention has been put on classification methods under the differential privacy paradigm. A survey on decision tree classification methods with differential privacy can be found in [49].

2.2.2 Differential Privacy for sequential data

Differential privacy has been used in mobility data to various degrees of success: one of the first works for differentially private trajectory publishing can be found in [31]. However, the simple application of noise based methodologies to ensure differential privacy on mobility data may lead to unnatural trajectories, with "zig zags" and crossings. Works like [136] have used a modified methodology, combining differential privacy with sampling of the trajectory points to protect trajectories of vessels while still maintaining a natural shape of the trajectories, while the authors in [65] propose a partition based algorithm for trajectory publishing via differential privacy, leveraging an exponential mechanism to divide the trajectories for protection. For social network data, some of the earliest practical approaches to ensure differential privacy can be found in [152], where the authors discuss differentially private algorithms to tackle some widely used analyses on social networks such as triangles analysis of clustering coefficient distribution analysis. Since the noise

injection required for guaranteeing certain privacy levels under differential privacy are proportional to the size of the network, and since many social network have an enormous size, the authors in [52] propose a definition of group-based local differential privacy, based on splitting the original network into subgraphs to then apply Hierarchical Random Graph models for the extraction of features [163]. In general, the Differential Privacy model has been proven to be an effective methodology to protect data from privacy attacks, but drawbacks to this methodology still remain across all fields of applications: the effectiveness of differential privacy, being based on some sort of randomization of the original data, always depends on the kind of analyses that is performed on the data, i.e., on the sensitivity of the function to be applied, and on the privacy budget allowed. Differential privacy in essence is designed for low sensitivity queries on the data, and even more refined models such as, for example, classification may require high privacy budget to work properly: for example, in [64] the authors perform nearest neighbor classification with privacy budgets as high as 2.0. Another problem is given by the modifications made to the data. Pratesi et al. in [**?**] propose a novel privacy risk mitigation methodology based on PRUDENnce and compare it with an approach based on differential privacy: their findings show that differential privacy affects both similarity to the original data and utility with respect to data mining applications in a significant way. Many of the more interesting analyses that can be done on sequential data, such as mobility or retail data, relies on the analyses on consecutive records, i.e., where someone has been before and where that someone is going or what someone has bought before and what will that someone buy in the future. This may lead to heavy losses in utility when applying differential privacy to these kinds of data. Moreover, differential privacy looses power with consecutive queries and analyses: performing queries consecutively, an adversary may be able to exhaust the privacy guarantee, thus differential privacy relies on constant monitoring on the actual use of the data by third parties or customers to reliably work on the long run. Therefore, differential privacy remains an open field of research where improvements are made constantly.

2.2.3 Anonymity based methods

One of the most commonly used methods to achieve anonymity is the k-anonymity framework. Introduced in [147], in k-anonymity the attributes of a record are divided into sensitive attributes and quasi-identifiers. The sensitive attributes are the attributes that need to be protected. Quasi-identifiers are attributes that may be linked to external information retrieved by an adversary in a linking attack. If the adversary is able to do so it can get access to the identity of the individual and its sensitive attributes. Therefore, 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. K-anonymity is a boolean condition for privacy: a dataset either has the k-anonymity property of doesn't. There are mainly three ways of of achieving k-anonymity: *generalization* [71], i.e. reducing the granularity of the representation of quasi-identifiers, *suppression* [146], i.e. replacing the value of highly sensitive attribute with a special value, and *microaggregation*[39], a perturbative data protection method where the data is divided into small clusters and values of sensitive attribute are substituted with the values of the centroid of the clusters. The problem of achieving optimal k-anonymity has been proven

to be NP-Hard in [93]. Several heuristics have been proposed in the literature to achieve k-anonymity: a greedy partition-based algorithm algorithm was proposed in [81] while a cluster-based approach for achieving k-anonymity in mobility data was proposed in [5]. K-anonymity has some vulnerabilities: if the individuals in the anonymity set present the same value for some sensitive attribute, an adversary can easily infer the value of such attribute for the subject of his attack, especially if the adversary uses some background knowledge about the subject allowing him to reduce the number of individuals withing the k-anonymity set of the subject. To tackle these problems [88] proposes the *l-diversity* model, with the objective of maintaining a degree of diversity in the sensitive attributes of an anonymity set. If however the overall distribution of the sensitive attribute is skewed, further measures have to be taken. *T-closeness* [85] imposes that the distance between the distribution of the attribute in any equivalence class and the distribution of the attribute in the overall dataset has to be bounded by a threshold t, thus preventing this issue.

2.2.4 Distributed privacy protecting methods

In modern business model, data is often spread over multiples sources. In such distributed environment, different subjects would share data to compute collective data mining models, integrate information and produce better analyses. However, the different participants often cannot trust each other thus requiring privacy preserving measures specific for the distributed environment. There are several studies that show privacy vulnerabilities in distributed contexts. For example [58] and [26] discuss important privacy fallacies in cloud computing. [168] gives a survey on different distributed data mining techniques categorizing them into three groups such as *secure multi-party computation*, *perturbation* and *restricted query*. In general, the methods developed in this context allow to compute functions over inputs provided by multiple parties without sharing the inputs. [36] acknowledged the privacy risks related to data mining on cloud system and presented a distributed framework to remove such risks. The proposed approach involved classification, disintegration, and distribution. Although suitable against mining attacks, it added a performance overhead as client accessed the data frequently. For preserving privacy in association rules mining, the authors of [161] proposed an algorithm called PPFDM and related computation technique based on the Frequent Data Mining (FDM) to preserve privacy. The process involved the computation of total support count along with the privacy-preserved technique while ensuring the local large item-set and local support count source is covered. Thus, the time needed for the communication is saved and the distributed data privacy at each site is secured. For clustering, [116] proposed an operative algorithm to protect the secrecy distributed over K-Means cluster using Shamir's secret sharing model. The proposed approach computes the cluster mean collaboratively and prevented the role of trusted third party. Upon comparison, it is observed that the proposed framework is orders of magnitude faster as compared to oblivious polynomial evaluation [99] and homomorphic encryption techniques [12] in terms of computation cost and more reliable for huge databases.

2.3 Risk analysis techniques

Analyzing privacy risk is a fundamental part of any privacy preserving process. Since the ultimate goal is to strike the right balance between reasonable protection of the individual privacy and quality of the data, measuring the risk of the individuals is fundamental. In the following, we present the main techniques for the assessment of different types of privacy risk.

2.3.1 Risk of re-identification

The authors of [27] advocate for the importance of the assessment of the risk of re-identification. In literature, this is also referred to as identity disclosure risk. A re-identification occurs when some adversary is able to link the de-identified or otherwise protected data of an individual with some information available to her, be it public or otherwise obtained. A great overview on both the terminology and the methodologies related to risk of re-identification an its measurement can be found in [156]. In the literature, there are two main ways to measure the risk of re-identification:

- *Dataset-level risk measure*: risk is defined as the proportion of records that an adversary can re-identify out of the whole set of records he has. This approach dates back to [109]. Formally, if we denote with A a protected dataset, B a dataset in the hands of an adversary (representing its knowledge), $t : B \to A$ a function that for each $b \in B$ gives the correct record $a \in A$, $r : B \to A$ a method of re-identification used by the adversary to associate a record $b \in B$ to a record $a \in A$ and $c(t(b), r(b))$ a function that returns 1 if $t(b) = r(b)$ and 0 otherwise, then the portion of records correctly re-identified by an adversary is: $Reid(B, A) = \frac{\sum_{b \in B} c(t(b), r(b))}{|B|}$.

- *Individual risk measures*: risk is defined as the probability that a particular sample record of the adversary is recognized as corresponding to a particular individual in the dataset. This comes from the intuition that risk is non homogeneous in a dataset, and that rare combinations of sensitive attribute may lead to the re-identification of individuals [46]. Following the definition given in [50], if there are K possible combinations of key attributes, these induce a partition both in the population and in the information of an adversary. If the frequency of the k-th combination in the population was known to be F_k, then the individual disclosure risk of a record in the sample with the k-th combination of key attributes would be $\frac{1}{F_k}$.

For both measures, the main technique used by an adversary is *Data Matching*. Data matching focuses on establishing relationships between the records with the goal of identifying the records that belong to the same individual but that are in different databases. [32] gives a detailed description of the different phases of such technique. They are:

- *Data preprocessing*. In this step data files are transformed so that all attributes have the same structure and the data have the same format. In data fusion, this is said to make data commensurate. That is, data should refer to the same point in time and refer to the same position in space. The same should be done here to make data comparable.

Major steps in data preprocessing are: (i) remove unwanted characters and words (stop words), (ii) expand abbreviations and correct misspellings, (iii) segment attributes into well defined and consistent output attributes, (iv) verify correctness of attributes values (e.g., that ages are always positive).

- *Indexing.* In most data matching problems it is unfeasible to compare all pairs of records in order to know which pairs correspond to the same individual. Note that the number of comparisons is the product of the number of records in the two databases. In order to reduce the number of comparisons, only some pairs are compared. Indexing is about the determination of which are the pairs interesting to be compared.

- *Record pair comparison.* This consists in the calculation of a value for each pair of records of interest. The comparison can be either a vector of Boolean values (stating whether each pair of attributes coincide or not) or a vector of similarities (stating in a quantitative way how similar are the values of the corresponding attributes).

- *Classification.* Using the comparison we need to establish whether the two records in the pair correspond to the same object or they correspond to different objects.

- *Evaluation step.* The result of the data matching system is analyzed and evaluated to know its performance.

Methods for record pair comparison and classification generally are of two types: *distance based record linkage methods* where record pairing is evaluated based on some distance measure, for example Euclidean distance, and *probabilistic record linkage* where record pairing is based on some probabilistic model [157].

2.3.2 Risk of attribute disclosure and inference

Attribute disclosure happens when intruders can increase the accuracy of their information with respect to the value of an attribute for a particular individual. This means that attribute disclosure can take place as a side effect of re-identification or may even take place without re-identification. Approaches measuring attribute disclosure vary depending on the type of the attribute. For numerical attributes, values are ranked and a rank interval is defined around such values for each record. The ranks of values within the interval for an attribute around a record r should differ less than p percent of the total number of records and the rank in the center of the interval should correspond to the value of the attribute in record r. Then, the proportion of original values that fall into the interval centered around their corresponding protected value is a measure of disclosure risk. A 100% proportion means that an attacker is completely sure that the original value lies in the interval around the protected value.

For categorical data a suitable method is defined in [104]. This method computes attribute disclosure risk for a given attribute in terms of a particular model or classifier for this attribute constructed from the released data. The percentage of correct predictions given by the classifier is a measure of the risk.

2.3.3 Risk analysis frameworks

One of the most important work about privacy risk assessment is the LINDDUN methodology [35], a privacy-aware threat analysis framework based on Microsoft's STRIDE methodology [148], useful for modeling privacy threats in software-based systems. However, LINDDUN methodology lacks a quantitative approach for privacy evaluation. In the last years, different techniques for risk management have been proposed, such as the OWASP's Risk Rating Methodology [107], NIST's Special Publication 800-30 [105], SEI's OCTAVE [72] and Microsoft's DREAD [91]. Unfortunately, many of these works do not consider privacy risk assessment and simply include privacy considerations when assessing the impact of threats. In [158], authors elaborate an entropy-based method to evaluate the disclosure risk of personal data, trying to manage quantitatively privacy risks. The *unicity* measure proposed in [140, 6] evaluates the privacy risk as the number of records/trajectories which are uniquely identified. The authors of [20] propose an empirical risk model for the estimation of privacy risk for trajectory data and a framework to improve privacy risk estimation for mobility data, evaluating their model using k-anonymized data. [16] proposes a risk-aware framework for information disclosure which supports runtime risk assessment. In this framework access-control decisions are based on the disclosure-risk associated with a data access request and adaptive anonymization is used as risk-mitigation method. This framework is designed to work on relational datasets, as it needs to discriminate between quasi-identifiers and sensitive attributes: for sequential datasets, e.g. mobility data, quasi-identifiers and sensitive attributes are generally not easy to specify, thus requiring specific approaches. Other works in literature study the re-identification risk as privacy measure in the context of network and social media data [100, 129] or combine network data and mobile phone data to re-identify people [30].

2.4 Privacy In Complex Data

Privacy preserving methodologies and risk assessment methodologies often depend heavily on the nature of the data that they operate on. In the following, we discuss the literature regarding privacy issues in non-tabular data focusing the discussion on the type of data that we take into consideration in this thesis.

Mobility Data An overview on the problems, techniques and methodologies related to urban mobility data and urban computing can be found in [173]. Human mobility data contains personal sensitive information and can reveal many facets of the private life of individuals, leading to the possibility of a serious privacy violation. Nevertheless, in the last years many techniques for privacy-preserving analysis on human mobility data have been proposed in literature [54] showing that it is possible to design analytical mobility services where the quality of results coexists with the protection of personal data. [4] proposes the (k, δ)-anonymity model, which takes advantage of the inherent uncertainty of the moving object's whereabouts, where δ represents the location precision. Assuming that different adversaries own disjoint parts of an individual's trajectory, Terrovitis and Mamoulis [153] reduce privacy risk by relying on the suppression of the dangerous ob-

servations from each individual's trajectory. In [166], authors propose the attack-graphs method to defend against attacks, based on k-anonymity. Monreale et al. [96] illustrate a generalized approach to achieve k-anonymity. Other works like [79] propose a methodology for matching trajectories in large scale datasets: the proposed methodology is based on the cooccurence of activities between the individuals in the two dataset, i. e., on matching individuals that repeatedly find themselves in the same place at the same time as each other. Other works are based on the differential privacy model [42]. As an example, [97] proposes the application of a ϵ-differential privacy model for guaranteeing the privacy protection in a distributed aggregation framework for movement data. Cormode et al. [33] propose to publish a contingency table of trajectory data, where each cell in the table contains the number of individuals commuting from the given source location to the given destination location. [135] proposes a mobility model called Mobility Markov Chain built upon mobility traces to re-identify an individual, while [74] defines several similarity metrics which can be combined in a unified framework to provide de-anonymization of mobility data and social network data.

Retail Data Privacy for retail data has been discussed from a multiplicity of angles. The authors in [76] first proposed the Platform for Enterprise Privacy Practices which defines technology for privacy-enabled management and exchange of customer data. The methodology proposed in this article is a general framework directed to enterprises for managing the data of their customers in a privacy enabled way. Most works concentrate on on-line shopping with the aim of guaranteeing privacy in on-line purchasing transactions. One of the first studies in this field was done in [47] where the authors tackle the problem of preserving the privacy of customers of internet retailers by preventing the vendor from directly linking information gathered about the customer with identifying information usually contained in the customer's order. Other works concentrate on privacy enabling user interfaces [80], while others focus on improving the privacy trade-off for e-shopping transactions [37]. Our focus in this thesis will be on customer retail data, i.e., on data regarding the purchases made by customers of general goods retailers. Several works in this context focus on Radio Frequency Identification (RFID) [83, 141], i.e., technologies for tracking customer's purchasing behaviour in retailers. The impact of such technology for the privacy of individuals has been well investigated in several works such as [83, 141, 155]. The authors in [144] focus in particular on the privacy challenge for stationary retailers that choose to adopt RFID for their businesses. In the context of data mining, the most common analysis made on retail data is association rule mining. Sevel methods have been proposed to carry out privacy preserving association rule mining [130, 134, 48]. Some recent work focused on solving this problem from different perspectives: [167] tackles the problem of association rule mining in cloud computing, by outsourcing the association rule mining process to"semi-honest" servers that collaborate to perform the analysis on encrypted data. In [84] the authors perform association rule mining on vertically partitioned databases, using homomorphic encryption. A survey on privacy preserving association rule mining methodologies can be found in [102].

Network Data Privacy for social media networks is a high interest topic, as shown in works such as [98], where the authors highlight how privacy awareness changes the perspectives and motivations of users of a social media. As noted by [126], privacy preserving solutions for social networks are mostly ad-hoc solutions that differ wildly depending on the problem tackled, which may range from privacy in publishing social network data for the use of third-party consumers, to the privacy of users in the leakage of individuals' information to unexpected people in their social circle. In the context of privacy for online social networks Liu and Terzi [87] propose a framework for computing privacy scores for each user in the network. Such scores indicate the potential risk caused by their participation in the network. In [22] Becker and Chen propose a framework called PrivAware, a tool to detect and report unintended information loss in online social networks. In [14] Ananthula et al. discuss a "Privacy Index" (PIDX) used to measure a user's privacy exposure in a social network. They have also described and calculated the "Privacy Quotient" (PQ), i.e. a metric to measure the privacy of the user's profile using a naive approach. Pensa and Blasi in [125] have proposed a supervised learning approach to calculate a privacy score of an individual in social network data based on the actual people allowed to access the profile of the individual. The authors in [131] survey the works regarding privacy issues in decentralized social networks. Several work explore the design of possible privacy attacks for social network data: [149] proposes a privacy attack based on the friends of an individual in the social network, i.e., on the nodes directly connected to the node of the victim. In [145] the authors introduce an attack that leverages mutual friendship relationships between the neighbors of a victim node.

Chapter 3

Data Modeling

In this chapter we present the mathematical modeling for the different kinds of data that we consider in this thesis and for each data type we also describe the experimental datasets used to validate our frameworks. The chapter is divided into four main sections. The first three sections are dedicated to the modeling of the three data types that we will use throughout the thesis: mobility data, retail data and social network data. For the three data types we also introduce a set of metrics that we will exploit in several of our approaches. While the last section is dedicated to experimental data description.

3.1 Mobility Data modeling

A trajectory is a sequence of records that identifies the movements of an individual during a period of observation [174, 172]. Each record contains the following information: the identifier of the individual; the visited location expressed in coordinates (typically, latitude and longitude); a timestamp that indicates when the individual stopped in or went through that location.

Definition 1. ***Trajectory.*** *The trajectory T_u of an individual u is a temporally ordered sequence of tuples $T_u = \langle (x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n) \rangle$, where x_i and y_i are the coordinates of a geographic location and t_i is the corresponding timestamp, with $t_i < t_j$ if $i < j$ $\forall i, j \leq n$, with $n = |T_u|$.*

Definition 2. ***Mobility Dataset.*** *A mobility dataset is a set of Trajectories $D = \{T_1, T_2, \ldots, T_m\}$, where T_u $(1 \leq u \leq m)$ is the trajectory of individual u.*

In practice, trajectories may have different resolutions depending on how the mobility data are collected. For our purposes, we refer to trajectories where the coordinates of each point represent the centroid of a larger geographical area comprising the original point. Specifically, in this thesis with the term *point* or *visit* we refer to a single element of a trajectory, while with the term *location* we refer to the point's spatial information. For brevity we will also denote with $l_i = (x_i, y_i)$ the geographical information of a point. We denote by $U_{set} = \{u_1, \ldots, u_n\}$ the set of the distinct individuals represented in the mobility dataset D and by $L_{set} = \{l_1 = (x_1, y_1), \ldots, l_w = (x_w, y_w)\}$ the set of distinct locations in D.

Depending on the specific application, a raw trajectory can be aggregated into different mobility data structures. These structure will be useful when defining some metrics or privacy attacks for mobility data:

Definition 3. ***Frequency vector.*** *The frequency vector W_u of an individual u is a sequence of tuples $W_u = \langle (l_1, w_1), (l_2, w_2), \ldots, (l_n, w_n) \rangle$ where $l_i = (x_i, y_i)$ is a location, w_i is the frequency of the location, i.e., how many times location l_i appears in the individual's trajectory T_u, and $w_i > w_j$ if $i < j$. A frequency vector W_u is hence an aggregation of a trajectory T_u.*

Definition 4. ***Probability vector.*** *The probability vector P_u of an individual u is a sequence of tuples $P_u = \langle (l_1, p_1), (l_2, p_2), \ldots, (l_n, p_n) \rangle$, where $l_i = (x_i, y_i)$ is a location, p_i is the probability that location l_i appears in W_u, i.e., $p_i = \frac{w_i}{\sum_{l_i \in W_u} w_i}$, and $p_i > p_j$ if $i < j$. A probability vector P_u is hence an aggregation of a frequency vector T_u.*

3.1.1 Mobility Data Metrics

The mobility dynamics of an individual can be described by a set of metrics widely used in literature. Some of these metrics describe specific aspects of an individual's mobility; other describe an individual's mobility in relation to collective mobility.

A subset of these measures can be simply obtained as aggregation of an individual's trajectory or frequency vector. The number of visits V_{num} of an individual is the length of her trajectory, i.e., the sum of all the visits she did in any location during the period of observation [56, 115]. By dividing this quantity by the number of days in the period of observation we obtain the average number of daily visits $\overline{V_{num}}$, which is a measure of the erratic behavior of an individual during the day. *Locs* indicates the number of unique locations visited by the individual during the period of observation [56, 137]. Dividing *Locs* by the number of available locations on the considered territory we obtain $Locs_{ratio}$, which indicates the fraction of territory exploited by an individual in her mobility behavior. The maximum distance D_{max} traveled by an individual is defined as the length of the longest trip of the individual during the period of observation [162], while D_{max}^{trip} is defined as the ratio between D_{max} and the maximum possible distance between the locations in the area of observation. The sum of all the trip lengths traveled by the individual during the period of observation is defined as D_{sum} [162]. It can be also averaged over the days in the period of observation obtaining $\overline{D}_{sum}$.

Besides these simple quantities, more complex metrics can be computed based on an individual's mobility data, such as the radius of gyration [56, 111] and the mobility entropy [45]. The radius of gyration r_g is the characteristic distance traveled by an individual during the period of observation, formally defined as [56, 111, 115]:

$$r_g = \sqrt{\frac{1}{V} \sum_{i \in L} w_i (r_i - r_{cm})^2},$$

where w_i is the individual's visitation frequency of location i, V is the total number of visits of the individual, r_i is a bi-dimensional vector describing the geographical coordinates of

location i, and $r_{cm} = \frac{1}{V}\sum_{i\in L} r_i$ is the center of mass of the individual [56, 111], and finally L represents here the set of all unique locations for that individual.The mobility entropy E is a measure of the predictability of an individual's trajectory. Formally, it is defined as the Shannon entropy of an individual's movements [45] [139] :

$$E = -\sum_{i\in L} p_i \log_2 p_i,$$

where p_i is the probability of location i in an individual's probability vector. Also, for each individual we keep track of the characteristics of three different locations: the most visited location, the second most visited location and the least visited location. The frequency w_i of a location i is the number of times an individual visited location i during the period of observation, while the average frequency $\overline{w}_i$ is the daily average frequency of location i. We also define w_i^{pop} as the frequency of a location divided by the popularity of that location in the whole dataset. The quantity U_i^{ratio} is the number of distinct individuals that visited a location i divided by the total number $|U_{set}|$ of individuals in the dataset, while U_i is the number of distinct individuals that visited location i during the period of observation. Finally, the location entropy E_i is the predictability of location i, defined as:

$$E_i = -\sum_{u\in U_i} p_u \log_2 p_u,$$

where p_u is the probability that individual u visits location i. All these mobility features can be computed in linear time with respect to the size of the corresponding data structure. Some preprocessing is usually performed on mobility data so that it can be analyzed properly. These preprocessing steps are highlighted in section 3.4, and are linear with respect to the number of points of each trajectory.

Table 3.1 summarizes the metrics that we used in our work.

symbol	**name**	**symbol**	**name**
V_{num}	number of points	R_g	radius of gyration
$\overline{V_{num}}$	daily visits	E	mobility entropy
D_{max}	max distance	E_i	location entropy
D_{sum}	sum distances	U_i	individuals per location
$\overline{D}_{sum}$	D_{sum} per day	U_i^{ratio}	U_i over individuals
D_{max}^{trip}	D_{max} over area	w_i	location frequency
$Locs$	distinct locations	w_i^{pop}	w_i over overall frequency
$Locs_{ratio}$	$Locs$ over area	$\overline{w}_i$	daily location frequency

Table 3.1: The individual mobility metrics used in our work.

3.2 Retail Data modeling

Retail data is generally collected through membership programs: customers who wish to do so, voluntarily agree to such programs in order to receive some benefits through

the use of a specific membership card, the data about their purchases is subsequently collected. The raw data of each individual is represented by baskets. A basket is a set of items purchased by the individual during a shopping session. We consider baskets with no repetitions, i.e., proper sets where items can appear only once. Therefore, an individual may have multiple baskets associated to her.

Definition 5. ***Basket.*** *We define a basket (or transactions) ba as a subset of items such that* $\emptyset \subset ba_i \subseteq IT$ *where* $IT = \{it_1, \ldots, it_n\}$ *is the set of all items.*

Definition 6. ***Basket History.*** *We define the basket history* $Hs_u = \langle ba_1, \ldots, ba_n \rangle$ *as the temporally ordered sequence of n baskets (or transactions) belonging to individual u.*

Definition 7. ***Retail Dataset.*** *A retail dataset is a set of Basket Histories* $D = \{Hs_1, Hs_2, \ldots, Hs_n\}$*, where* Hs_u *(*$1 \leq u \leq n$*) is the basket history of individual u.*

Again, we refer to $U_{set} = \{u_1, \ldots, u_n\}$ the set of the distinct individuals represented in the retail dataset D. When we refer to a dataset D, whether it is a mobility or retail dataset it will be either clear from the context or non relevant.

3.2.1 Retail Data Metrics

Given a retail dataset we can extract different metrics able to describe the purchasing behaviour of people and their habits. Table 3.2 contains the description of all the metrics we defined for retail data. Let I be the total number of items purchased by a customer during the period of observation. It comprises all the shopping sessions of the customer. Consequentially we indicate with I_{unique} the number of unique items bought by an individual in the period of observation. We averaged the total number of items bought by a customer with the period of observation, expressed in days. Therefore, $I_{avg} = \frac{I}{time}$, in which *time* represents the period of observation expressed in days. Another metric we define is the I^d_{max}: it is the maximum number of items purchased by an individual during a shopping session, e.g. in a basket. Formally, $I^d_{max} = maxlen(ba), ba \in Hs_u$. We define I^d_{avg} as the average number of items bought in a shopping session: $I^d_{avg} = avglen(b), b \in B_u$. Another interesting measure is the *product entropy*, defined applying Shannon's formula [139].

$$E = -\sum_{i \in L} p_i \log_2 p_i \tag{3.1}$$

in which p_i is the probability associated to the item i. Another set of metrics defined for retail data is the *product-metric*. They are metrics based on the customer under analysis, but they also involve the characteristics of a product the individual bought during the period of observation. We evaluate each of these measures over three products for each customer: the *top product*, i.e. the product that was bought more times, the *second top product*, that is the second product that was bought with more frequency and also the *least product*, that is the product the individual purchased fewest times. For each customer and each of these three products, we define the following metrics: the *product entropy* using the Shannon formula:

$$E = -\sum_{u \in U_{set}} p_u \log_2 p_u \tag{3.2}$$

In this case, the probability p_u indicates the likelihood of the item to be bought by the individuals u in the dataset. We also define w_i as the frequency of the item i from the customer under analysis. In practice, it is a count of the number of times the item under analysis has been purchased by the costumer. Note that the computation of this measure requires a scan of the whole dataset and not only of the data of the customer under analysis. Moreover, we define w^{avg} as the measure of the number of times the item under analysis has been bought divided by the number of days in the period of observation. Technically, it corresponds to $w^{avg} = \frac{w}{time}$, where $time$ represents the number of days of the period of observation. We also define U_{i_j} as the number of users who bought the item i_j under analysis at least once. We then average this value dividing it by the total number of users in the dataset, obtaining $U_{i_j}^{avg}$. Technically, $U^{avg} = \frac{U}{users}$, where $users$ represents the total number of users in the dataset.

symbol	**name**	**symbol**	**name**
I	Total number of items	$\overline{I}_{max}^{daily}$	Maximum number of products in a day divided by the total products
I_{unique}	Total number of unique items	$\overline{I}_{avg}^{daily}$	Average number of products in a day divided by the total products
I_{avg}	Total number of items averaged over time	E_{i_j}	Product entropy
I_{max}^{d}	Maximum number of items bought in a day	w_{i_j}	Frequency of the product
I_{avg}^{d}	Average number of items bought per day	$w_{i_j}^{avg}$	Average frequency of the product
E	Purchasing entropy	U_{i_j}	Number of users who bought the product
$Locs$	Distinct locations	$U_{i_j}^{avg}$	Average number of users who bought the product
I_{unique}^{avg}	Total number of unique items averaged over time		

Table 3.2: The individual retail data metrics used in our work.

3.3 Social Network Data modeling

Networks have traditionally been modeled as graphs:

Definition 8. ***Social Network.*** *We model a social network as a simple graph* $G = (V, E, L, \Gamma)$*, where* V *is the set of vertices representing individuals,* $E \subseteq V \times V$ *is the set of edges representing the relationships between individuals,* L *is a set of labels, and* Γ *is a labeling function that maps each vertex to a subset of labels in* L*.*

We assume that edges do not have any labels. In a social network, the direction of an edge indicates the relationship between vertices and can be used to distinguish the

type of relationship: single-sided or mutual. For our purpose, we will assume that all relationships are mutual.

Starting from the social network data represented as a graph it is possible to derive other data structures representing aggregated information of the original graph. The scope of these data structures is to expose less information than the original one while enabling the computation of standard network metrics. Clearly, this data transformation helps privacy preserving analyses and the respect of data minimization principle required by the GDPR [1]. Here, we define some of these structures:

Definition 9. ***Friendship Vector.*** *The friendship vector F_v of an individual $v \in V$ is a set of vertices $F_v = \langle v_1, v_2 \ldots, v_n \rangle$ representing individuals connected to v in the social network graph.*

The friendship vector of a node v essentially represents the neighborhood of the individual v at distance 1.

Definition 10. ***Label vector.*** *The label vector of an individual v is a set of labels $La_v = \langle la_1, la_2, \ldots, la_m \rangle$. Each $la_j = (f, l)$ (with $j \in \{1, 2, \ldots, |La|\}$) is a pair composed of a feature name f and the associated label value l. The label vector of an individual can be empty.*

Each label describes a profile feature of an individual, for example *gender* : *'female'* or *'male'*, *educational information: 'Pisa University'* or *'Stanford University'*, etc.

Definition 11. ***Degree Vector.*** *The degree vector of an individual v, denoted by $Dg_v = \langle dg_{v_1}, dg_{v_2}, \ldots, dg_{v_n} \rangle$, represents the number of friends of each friend of v. Thus, each element dg_{v_i} is equal to the length of the friendship vector of the individual v_i in the social network graph, i.e., $dg_{v_i} = len(F_{v_i})$.*

Definition 12. ***Mutual Friendship Vector.*** *The mutual friendship vector of an individual v, denoted by $Mf_v = \langle mf_1, \ldots, mf_n \rangle$, represents the number of common friends of v with each one of its friends v_i. Thus, each element mf_i is equal to the cardinality of the intersection between the friendship vector of v and the one of v_i, i.e., $mf_i = \mid F_v \cap F_{v_i} \mid$.*

In the above definition, the cardinality of an intersection can be empty when the individual and her friend do not share any friend in the social network.

Taking in consideration all of the structures defined above we can define a Social Network Dataset as follows:

Definition 13. ***Social Network Dataset.*** *A social network dataset is a set of individual social network structures $D = \{S_1, S_2, \ldots, S_k\}$ where S_v $(1 \leq v \leq k)$ is the social network data structure of an individual v.*

Again, when with D we indicate a dataset, whether it is a social network, mobility or retail dataset, it will be either clear from the context or non relevant. Clearly, given the definition of the different individual social network structures, we can have different types of social network datasets. Thus, a social network dataset can be a set of friendship vectors $\{F_1,F_2,...,F_k\}$, a set of label vectors $\{LA_1,LA_2,...,LA_k\}$, a set of degree vectors $\{Dg_1,Dg_2,...,Dg_k\}$ or a set of mutual friendship vectors $\{MF_1,MF_2,...,MF_k\}$. Note that, the four sets have the same size $|V| = k$.

3.3.1 Social Network Data Metrics

Numerous metrics have been introduced to characterize and analyze networks. Graph metrics can be broadly classified in two categories: *global measures* refer to global properties of a graph and, therefore, consists of a single value for each graph; meanwhile *nodal measures* refer to properties of the nodes of a graph and, therefore, consists of a vector of numbers — one for each node of the graph [67]. Our focus will be on node level metrics as they give information about the single nodes in the graph. The most simple metric is the degree of a node that we denote with dg_v. This is simply the number of other nodes connected to v. The degree centrality is used for finding very connected individuals, popular individuals or individuals who can quickly connect with the wider network [51]. Degree Centrality is computed dividing the degree of a node by the total number of nodes minus one: $Cd(v) = dg_v/(|V|-1)$. Betweenness centrality measures the number of times a node lies on the shortest path between other nodes. This shows us which nodes act as "bridge" between nodes in a network. It is computed by identifying all the shortest paths and then counting how many times each node falls onto one. Given two nodes v_1 and v_2:

$$Cb(v) = \sum_{v_1 \neq v \neq v_2 \in V} \sigma_{v_1,v_2}(v)\sigma_{v_1,v_2}$$

where σ_{v_1,v_2} is total number of shortest paths from node v_1 to node v_2, $\sigma_{v_1,v_2}(v)$ is the number of those paths that pass through v. Closeness centrality scores each node based on their "closeness" to all other nodes within the network. This metrics calculates the shortest paths between all nodes, then assigns to each node a score based on the sum of shortest paths [51]. In a connected graph, the normalized closeness centrality (or closeness) of a node is the average length of the shortest path between the node and all other nodes in the graph. Thus the more central a node is, the closer it is to all other nodes [106]: $C(v) = (|V|-1)/\sum_{v_2 \in V} d(v, v_2)$. Another metric commonly used in graph theory is the clustering coefficient [57], a measure of the degree to which nodes in the graph tend to cluster together. It also indicates the portion of neighbors of a node that are connected. We denote it as: $Cc(v) = \frac{2N_v}{dg_v(dg_v-1)}$. Eigenvector Centrality (also called eigen centrality) is another measure of the influence of a node in a network. Relative scores are assigned to all nodes based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores [103]: Let A be the adjacency matrix, i.e. $a_{v,v_2} = 1 \forall a \in A$ if the node v is connected to node v_2, and $a_{v,v_2} = 0$ otherwise. The relative centrality score of a node v can be defined as:

$$x_v = \frac{1}{\lambda} \sum_{v_2 \in F(v)} x_{v_2} = \frac{1}{\lambda} \sum_{v_2 \in G} a_{v,v_2} x_{v_2}$$

where $F(v)$ is a set of the neighbors of v and λ is a constant. With a small rearrangement this can be rewritten in vector notation as the eigenvector equation $Ax = \lambda x$. We also use the core number of a node as a metric: a k-core is a maximal subgraph that contains nodes of degree k or more. A sub graph $H = (S, E|S)$ induced by the set S is a

k-core or a core of order k $\iff \forall s \in S : deg_H(v) \geq k \wedge S$ is a maximum sub graph with this property. The core of maximum order is also called the main core. The core number of node v is the highest order of a core that contains this node. The algorithm can be found in [21]. The eccentricity of a node v is the maximum distance between a node v and all other nodes in the graph. Eccentricity of a node can be found by looking all shortest path with all other nodes and taking the maximum one. Finally we use PageRank (PR) as a metrics: page rank is an algorithm used by Google Search to rank websites in their search engine results. It is also used to compute a ranking of the nodes in the graph based on the structure of the incoming links. PageRank can be expressed as follows:

$$x_v = \alpha \sum_v a_{v,v_2} frac x_{v_2} dg_{v_2} + \frac{1-\alpha}{|V|}$$

where α is the damping factor and a be the adjacency matrix, i.e. $a_{v,v_2} = 1$ if node v is connected to node v_2, and 0 otherwise. The differences of eigenvector centrality and PageRank are the scaling factor of dg_{v_2} and the PageRank vector is a left hand eigenvector. PageRank algorithm gives each page a rating of its importance, which is a recursively defined measure whereby a page becomes important if important pages link to it. This definition is recursive because the importance of a page refers back to the importance of other pages that link to it. Table 3.3 summarizes the network data metrics that we use in our work.

symbol	**name**	**symbol**	**name**
dg	Degree of node v	Ec	Eccentricity
Cd	Degree centrality	Ax	Eigenvector centrality
Cb	Betweenness centrality	Pg	Pagerank
C	Closeness centrality	$max(dg_H)$	Core number
Cc	Clustering coefficient		

Table 3.3: The individual social network data metrics used in our work.

3.4 Experimental Datasets

For each type of data model, presented above, in this thesis we use real-world experimental data to validate our approaches and frameworks. In the rest of the thesis we will refer to this section when discussing experiments on real-world data.

3.4.1 Experimental Mobility Dataset

For experiments on mobility data we use data provided by Octo Telematics[1] storing the GPS tracks of private vehicles traveling in the Italian region of Tuscany. We selected

[1]https://www.octotelematics.com/

different urban areas and/or trajectories from the data depending on the analysis that we want to perform. We use a cut of our experimental mobility dataset storing the GPS tracks of private vehicles traveling in Florence and Pisa, from 1st May to 31st May 2011, corresponding to 9,715 and 3,281 vehicles respectively. The GPS device embedded in a vehicle automatically turns on when the vehicle starts, and the sequence of GPS points that the device produces every 30 seconds forms the global GPS track of a vehicle. When the vehicle stops no points are logged nor sent. We exploit these stops to split the global GPS track of a vehicle into several sub-tracks, corresponding to the trips performed by the vehicle. To ignore small stops like traffic lights and gas stations, we follow the strategy commonly used in literature [111, 115] and choose a stop duration threshold of at least 20 minutes: if the time interval between two consecutive GPS points of the vehicle is larger than 20 minutes, the first point is considered as the end of a trip and the second one as the start of another trip.[2] We assign each origin and destination point of the obtained sub-tracks to the corresponding census cell according to the information provided by the Italian National Statistics Bureau (ISTAT), in order to assign every origin and destination point to a location [115]. These steps are performed in linear time with respect to the number of points in each trajectory ISTAT census cells have a variable size, depending mostly on the population density of the tessellated area, and are periodically updated by ISTAT itself in collaboration with local authorities. More information can be found at: `https://www.istat.it/it/files//2019/10/IWP_9-2019.pdf` . This allows us to describe the mobility of every vehicle in the Florence or the Pisa datasets in terms of a trajectory, in compliance with the definitions introduced in Section 3.1.

3.4.2 Experimental Retail Dataset

We use a retail dataset provided by Unicoop[3] storing the purchases of individuals in shopping centers of the coast of the region of Tuscany, in Italy, focusing on the purchases of 1000 individuals in the city of Leghorn during 2013, corresponding to 659,761 items and 61,325 baskets. Shopping data is usually collected through membership programs: customers join the program by using a membership card identification, thus providing their shopping data while receiving, in exchange, special discounts, promotions of gifts. We consider each item in the shopping sessions of individuals at the category level, representing a more general description of a specific item, e.g., "Coop-brand Vanilla Yogurt" belongs to category "Yogurt", "Corn Bread" belongs to category "Bread" and so on.

3.4.3 Experimental Network Dataset

We use the Facebook dataset provided by Stanford University's "Stanford Large Network Dataset Collection" [82]. This dataset includes node features (profiles), circles and ego networks. Nodes have been anonymized by replacing the Facebook-internal id's for each user with a new value. Feature vectors from this dataset have also been provided while

[2]We also performed the extraction of the trips by using different stop duration thresholds (5, 10, 15, 20, 30, 40 minutes), without finding significant differences in the sample of short trips and in the analyses we present in this thesis.

[3]https://www.unicooptirreno.it/

the interpretation of those features has been anonymized. After aggregating all data, we obtain a social network graph of 4039 nodes and 88,234 edges. Almost half of the all individuals have 30 friends/neighbors or less.

Chapter 4

Privacy Risk & Data Quality Assessment

In this chapter we present PRUDEnce, the privacy framework proposed in [128] and our extension that enables a systematic evaluation of data quality in terms of impact of non-risky individuals on the data analytical results. Moreover, we show how to instantiate this general framework in three scenarios (mobility, retail and social network data) defining for each one a set of adversary attacks, quantifying both the empirical privacy risk produced by each attack and the data quality preserved by the non-risky individuals. Note that, the assessment of privacy risk for any attack will be the preliminary step of our methodology of privacy risk prediction presented in Chapter 5. We will also propose an extension to PRUDEnce to provide a database level evaluation of privacy risk instead of an individual evaluation. This extension is tailor made for retail data, specifically to tackle the problem of assessing the inherent privacy risk in user purchasing profiles extracted from retail data. The results presented in this chapter have been partially published in [121], [119] and [123].

4.1 PRUDEnce Privacy Risk Assessment Framework

Several methodologies have been proposed in literature for privacy risk assessment. In this thesis we focused on the privacy framework PRUDEnce [128], which allows for the systematic data-driven assessment of the privacy risk for any type of data. The framework considers a scenario where a Data Analyst requests a Data Provider some data in order to develop an analytical service. For its part, the Data Provider has to guarantee the right to privacy of the individuals whose data are recorded. As a first step, the Data Analyst communicates to the Data Provider the data requirements for the analytical service. Assuming that the Data Provider stores a database $\mathcal{D}$, it aggregates, selects and filters the dataset $\mathcal{D}$ to meet the requirements by the Data Analyst and produces a set of datasets $\{D_1, \ldots, D_n\}$ each with a different data structure and/or aggregation of the data. The Data Provider then reiterates a four-step procedure until it considers the data delivery safe:

(1) *Identification of Attacks*: identify a set of possible attacks that a malicious adversary

might conduct in order to re-identify the individuals in the datasets $\{D_1, \ldots, D_n\}$;

(2) *Privacy Risk Computation*: simulate the attacks and compute the set of privacy risk values for every individual in the datasets $\{D_1, \ldots, D_n\}$;

(3) *Dataset Selection*: select a dataset $D \in \{D_1, \ldots, D_n\}$ with the best trade-off between the *privacy risks* of the individuals and the *quality of the data*, given a certain level of tolerated privacy risk and the data requirements by the Data Analyst;

(4) *Risk Mitigation and Data delivery*: apply a privacy-preserving transformation (e.g., generalization, randomization, etc.) on the chosen dataset D to eliminate the residual privacy risk, producing a filtered dataset D_{filt}. Deliver the dataset D_{filt} to the Data Analyst when the D_{filt} is adequately safe.

The framework is summarized in Figure 4.1.

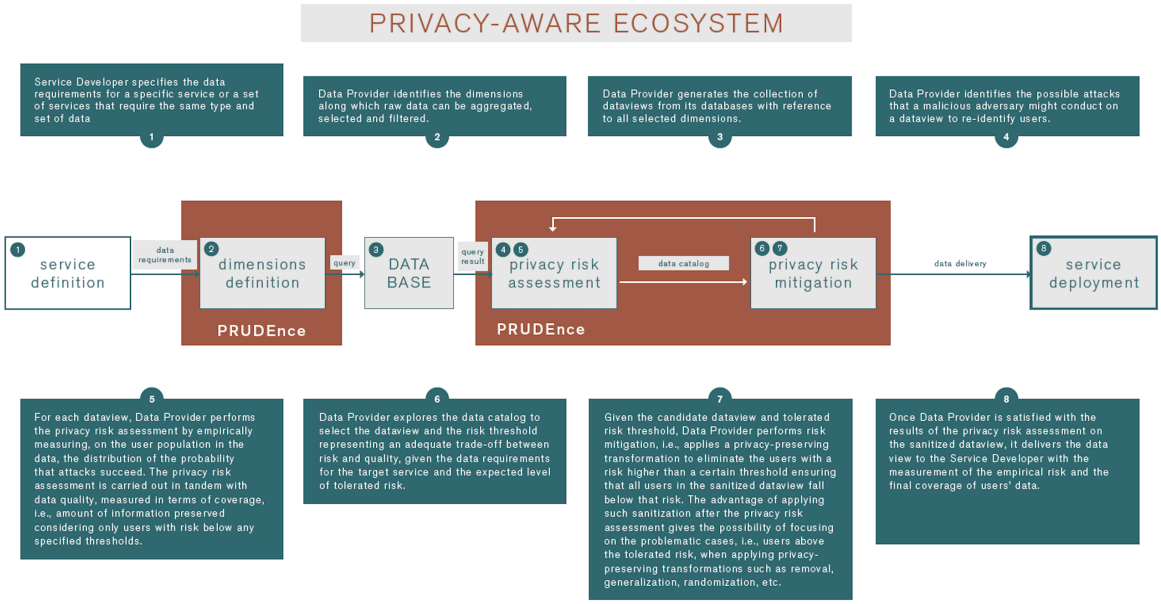


Figure 4.1: The general schema of the PRUDEnce privacy framework

Privacy Risk Computation. The privacy risk of an individual is related to her probability of re-identification in a dataset with respect to a set of re-identification attacks. A re-identification attack assumes that an adversary gains access to a dataset. On the basis of some background knowledge about an individual, i.e., the knowledge of a subset of her own data, the adversary tries to identify all the records in the dataset regarding the individual under attack. In our work we use the definition of privacy risk (or re-identification risk) introduced in [133, 132, 147] and widely used in the literature. There can be many background knowledge categories, every category may have several background knowledge configurations, every configurations have many instances.

A *background knowledge category* is a kind of information known by the adversary about a specific set of dimensions of an individual's data. Typical dimensions in mobility

data are space, time, frequency of visiting a location and probability of visiting a location. In retail data possible dimensions could be the products, or the time of a shopping session. In network data these could be the number of neighbors of a node or the degree of neighbors, for example. The number k of the elements of a category known by the adversary is called *background knowledge configuration*. An example of background knowledge configuration is the knowledge by the adversary of $k = 3$ points in the trajectory of an individual, or $k = 4$ friends of a certain individual in a network. Essentially the background knowledge configuration represents the "quantity" of knowledge that an adversary may have. Finally, an *instance of background knowledge* is the specific information known by the adversary, such as a visit in a specific location, or a specific product bought in a shopping session. We formalize these concepts as follows:

Definition 14. ***Background Knowledge Category, Configuration and Instance*** *Given a background knowledge category $\mathcal{B}$, we denote with $B_k \in \mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ a specific background knowledge configuration, where k represents the number of elements in $\mathcal{B}$ known by the adversary. We define an element $b \in B_k$ as an instance of background knowledge configuration.*

Example 1. *Suppose a trajectory $T_u = \langle (l_1, t_1), (l_2, t_2), (l_3, t_3), (l_4, t_4) \rangle$ of an individual u is present in the Data Provider's dataset D, where (l_i) is a location (x_i, y_i) and t_i the time when u visited that location, with $i = 1, \ldots, 4$ and $t_i < t_j$ if $i < j$. Based on T_u the Data Provider can generate all the possible instances of a background knowledge configuration that an adversary might use to re-identify the whole trajectory T_u. Considering the knowledge by the adversary of ordered subsequences of locations and $k = 2$, we obtain the background knowledge configuration. $B_2 = \{((l_1, t_1), (l_2, t_2)), ((l_1, t_1), (l_3, t_3)), ((l_1, t_1), (l_4, t_4)), ((l_2, t_2), (l_3, t_3)), ((l_2, t_2), (l_4, t_4)), ((l_3, t_3), (l_4, t_4))\}$. The adversary for example might know instance $b = ((l_1, t_1), (l_4, t_4)) \in B_2$ and aims at detecting all the records in D regarding individual u, in order to reconstruct the whole trajectory T_u.*

Example 2. *Suppose a basket history $Hs_u = \langle ba_1, ba_2, ba_3 \rangle$ of an individual u is present in the Data Provider's dataset D, where $ba_i = \{it_1, \ldots, it_n\}$ is a basket of items it_j, with $i = 1, \ldots, 3$. Based on Hs_u the Data Provider can generate all the possible instances of a background knowledge configuration that an adversary might use to re-identify the whole basket history Hs_u. Considering the knowledge by the adversary of the full baskets with $k = 2$, we obtain the background knowledge configuration $B_2 = \{(ba_1, ba_2), (ba_1, ba_3), (ba_2, ba_3)\}$. The adversary for example might know instance $b = (ba_1, ba_3) \in B_2$ and aims at detecting all the records in D regarding individual u, in order to reconstruct the whole basket history Hs_u.*

Let $\mathcal{D}$ be a database, D a dataset extracted from $\mathcal{D}$ as an aggregation of the data on specific dimensions (e.g., an aggregated data structure and/or a filtering on some dimension), and D_u the set of records representing individual u in D, we define the probability of re-identification as follows:

Definition 15. ***Probability of re-identification.*** *Given an attack, a function* matching(d, b) *indicating whether or not a record $d \in D$ matches the instance of background*

knowledge configuration $b \in B_k$*, and a function* $M(D, b) = \{d \in D | matching(d, b) = True\}$*, we define the* probability of re-identification *of an individual* u *in dataset* D *as:*

$$PR_D(d = u|b) = \frac{1}{|M(D, b)|}$$

that is the probability to associate a record $d \in D$ *to an individual* u*, given instance* $b \in B_k$*.*

Therefore the probability of re-identification of an individual u in a dataset D depends on two quantities: (i) $M(d, b)$, the record $d \in D$ compatible with the instance b; (ii) $M(D, b)$, the records in D compatible with the instance b. For our purposes we will assume that each individual is represented by a single data structure in any dataset D where she is represented in. The *compatibility* is expressed by a function $matching(d, b)$ which indicates whether or not a record $d \in D$ matches the instance b. The matching function depends on the background knowledge used during the attack. Note that $PR_D(d{=}u|b) = 0$ if the individual u is not represented in D. Since each instance $b \in B_k$ has its own probability of re-identification, we define the risk of re-identification of an individual as the maximum probability of re-identification over the set of instances of a background knowledge configuration:

Definition 16. ***Risk of re-identification or Privacy risk.*** *The risk of re-identification (or privacy risk) of an individual* u *given a background knowledge configuration* B_k *is her maximum probability of re-identification* $Risk(u, D) = \max PR_D(d = u|b)$ *for* $b \in B_k$*. The risk of re-identification has the lower bound* $\frac{|D_u|}{|D|}$ *(a random choice in* D*), and* $Risk(u, D) = 0$ *if* $u \notin D$*.*

We now want to summarize and clarify the functioning of the framework described: the PRUDEnce framework does not focus on the privacy risk evaluation with respect to a "specific adversary knowledge" but exploits a data-driven mechanism that allows the generation of all possible types of knowledge that one could extract and derive from the data. The assumption is that possible background knowledge known by an adversary on the user u is a subset of the entry data associated to u in the data to be shared. As stated in "Data Privacy: Foundations, New Developments and the Big Data Challenge"[156] the worst-case scenario considers an adversary knowing the same data of the shared table. In order to evaluate the trend of the risk changing the quantity of background knowledge possessed by the adversary, PRUDEnce generates for each user all the possible levels of external knowledge starting from the minimum knowledge (e.g., only one location in the example of the paper) to the maximum one (e.g., the whole set of user's locations). This assumption is due to the fact that for re-identifying a person in the dataset, the adversary needs to know a subset of the user data record or some other information that leads to the knowledge of that subset of user data. PRUDEnce to this end needs two specifications: i) the "nature" of the assumed background knowledge of the adversary. We called this "background knowledge category". This represents the dimensions of the of data that the adversary knows; ii) the quantity of information that we assume the adversary will use in its attack. We called this "background knowledge configuration". This represents the number of records that the adversary may know about the attacked

individual. Given these two types of information, PRUDEnce generates all the possible "background knowledge instances" in a systematic way, considering all the possible k-combinations of records/points belonging to an individual. These are the actual points that an adversary may use when conducting an attack. Only defining different levels of possible external knowledge, it is possible to provide organizations the possibility to reason in a systematic way on the balancing between privacy risks and data utility, for helping them in making responsible and aware decisions about the best trade-off.

In order to clarify the concepts of probability of re-identification and privacy risk we provide the following example that, given a mobility dataset D of trajectories, shows how we can compute the two matching components for a specific attack. In our example we will use timestamps with daily precision and for simplicity, we substitute x, y coordinates with names of places so that it is simpler to follow the example.

Example 3. *Let us consider a set of individuals* $U_{set}=\{u_1, u_2, u_3, u_4, u_5, u_6\}$ *and the corresponding dataset* D *of trajectories:*

$D = \{$
$T_{u_1} = \langle(Lucca, 2011/02/03), (Leghorn, 2011/02/03), (Pisa, 2011/02/03), (Florence, 2011/02/04)\rangle$
$T_{u_2} = \langle(Lucca, 2011/02/03), (Pisa, 2011/02/03), (Lucca, 2011/02/04), (Leghorn, 2011/02/04)\rangle$
$T_{u_3} = \langle(Leghorn, 2011/02/03), (Pisa, 2011/02/03), (Lucca, 2011/02/04), (Florence, 2011/02/04)\rangle$
$T_{u_4} = \langle(Pisa, 2011/02/04), (Leghorn, 2011/02/04), (Florence, 2011/02/04)\rangle$
$T_{u_5} = \langle(Pisa, 2011/02/04), (Florence, 2011/02/04), (Lucca, 2011/02/05)\rangle$
$T_{u_6} = \langle(Lucca, 2011/02/04), (Leghorn, 2011/02/04)\rangle$
$\}$

Let us assume an adversary wants to perform an attack on individual u_1 *knowing only the locations she visited (without any information about the time), with background knowledge configuration* B_2*, i.e., the adversary knows two of the locations visited by individual* u_1*. We compute the risk of re-identification of individual* u_1*, given the dataset* D *of trajectories and the knowledge of the adversary, in two steps:*

1. *We compute the probability of re-identification for every* $b \in B_2$*. Instance* $b=\{$ *Lucca, Leghorn*$\}$ *has probability of re-identification* $PR_D(d=u_1|\{Lucca, Leghorn\})=\frac{1}{4}$*, because the pair* $\{Lucca, Leghorn\}$ *appears in trajectories* T_{u_1}*,* T_{u_2}*,* T_{u_3} *and* T_{u_6}*, i.e., in a total of four trajectories. Instance* $\{Lucca, Pisa\}$ *has probability of re-identification* $PR_D(d=u_1|\ \{Lucca, Pisa\})=\frac{1}{4}$ *because the pair appears in four trajectories* T_{u_1}*,* T_{u_2}*,* T_{u_3} *and* T_{u_5}*. Instance* $\{Lucca, Florence\}$ *has probability of re-identification* $PR_D(d=u_1 \mid \{$ *Lucca, Florence* $\})=\frac{1}{3}$ *because the pair appears in three trajectories* T_{u_1}*,* T_{u_3} *and* T_{u_5}*. Analogously we compute the probability of re-identification for the other three possible instances:* $PR_D(d=u_1|\ \{Leghorn, Pisa\})=\frac{1}{4}$*,* $PR_D(d=u_1|\ \{Leghorn, Florence\})=\frac{1}{3}$*,* $PR_D(d=u_1|\ \{Pisa,\ Florence\})=\frac{1}{4}$*;*

2. *We compute the risk of re-identification of individual* u_1 *as the maximum of the probabilities of re-identification among all instances in* B_2*:* $Risk(u_1)=max(\frac{1}{4}, \frac{1}{4}, \frac{1}{3}, \frac{1}{4}, \frac{1}{3}, \frac{1}{4}) = \frac{1}{3}$.

We remark that the Data Provider does not *know in advance the instance associated to the highest probability of re-identification of individual u_1, i.e., the "best" combination of points from the perspective of the malicious adversary. The Data Provider can use the computation above in a preventive manner to identify the instance yielding the highest probability of re-identification which is, for individual u_1, instance $\{Leghorn, Florence\}$. Due to the definition of risk, which depends on both an attacked individual's structure and the structures of all the other individuals in the dataset, identifying* a priori *an attack where the adversary has access to the best k-combination of points is difficult for the Data Provider. A particular case where the Data Provider can immediately recognize the best k-combination of points is a scenario where the adversary knows a location visited only by the individual under attack. Since the Data Provider has a view of the entire dataset, she can simulate such an attack by selecting the locations visited by just an individual, i.e., with number of visits equal to 1. In such a case, computing the privacy risk for the individuals visiting those locations does not require any combinatorial computation, because the privacy risk is 1 for any value of k.*

An individual is hence associated to several privacy risks, each for every background knowledge configuration of an attack. Every privacy risk of an individual can be computed using the following procedure:

1. given an individual, define an attack based on a specific background knowledge category;
2. consider a set of m background knowledge configurations $\{B_1, \ldots, B_m\}$;
3. for every configuration $B_k \in \{B_1, \ldots, B_m\}$ compute all the possible instances $b \in B_k$ and the corresponding probability of re-identification;
4. select the privacy risk of the individual for a configuration B_k as the maximum probability of re-identification across all the instances $b \in B_k$.

Data Quality Evaluation. The *Dataset selection* of PRUDEnce process (STEP 3) is based on the evaluation of both privacy risk and data quality given a specific tolerated privacy risk threshold. PRUDEnce provides a method for measuring the data quality in terms of portion of data covered by individuals having at most a specific tolerated privacy risk. In [128] authors define the RAC_D curve as the function that for each risk value r, quantifies the percentage of records in D that are covered by individual having at most the risk r. In other words, given $U_r = \{u \in U | Risk(u, D) \leq r\}$ and let DU_r be the set of data covered by individuals in U_r this function is defines as $RAC_D(r, D) = |DU_r|/|D|$.

4.1.1 PRUDEnce Extension

Privacy Risk for Individual Patterns One of the most common analyses on personal data is customer profiling. Customer profiling is a process widely used in economy for direct marketing, service development, site selection, and customer relationship management. The process of construction and extraction of a personal data model formed

by personal patterns is generally referred to as *user profiling*. A user profile contains the systematic behaviors expressing the repetition of habitual actions, i.e., personal patterns. These patterns can be expressed as simple or complex indexes [60], behavioral rules [63], set of events [61], typical actions [159], etc. Profiles can be classified as individual or collective according to the subject they refer to [69, 59]. An *individual* profile is built considering the data of a single person. This kind of profiling is used to discover the particular characteristics of a certain individual, to enable unique identification for the provision of personalized services. We talk about *collective* data models when personal data or individual models generated by individual profiling are aggregated without distinguishing the individuals. By knowing the profile of each customer, a company can treat a customer according to her individual needs and increase the lifetime value of the customer [15]. Furthermore, customer profiling is a key element which impacts into the decisions in product life cycle cost [40].

We want to extend the PRUDEnce framework with a methodology to estimate the privacy risk of individual patterns with respect to the data from which they where extracted. To do so, we rely on a well tested methodology to evaluate privacy risk: distance based record linkage [68]. In distance based record linkage it is assumed that an adversary possesses a database with records belonging to the same individuals in the data that she wants to attack. We can apply the same idea, assuming that a malicious adversary gets access to the individual patterns extracted from a retail dataset and uses these patterns as her own database to perform the attack. Formally, we assume that the background knowledge of the malicious adversary is a dataset $\mathcal{P} = \{Pa_1, \ldots, Pa_n\}$ where Pa_u is a set of patterns representing individual u, extracted from the respective raw data. With this information the adversary tries to match the original records in dataset D with the corresponding patterns. To do so, the adversary can compute a distance between each of the patterns in her knowledge and the records in D, and then assign to each record the set of patterns with the smallest distance. For this approach, privacy risk cannot be expressed as an individual measure, i.e., from the perspective of the individuals in the data. An adversary either correctly matches the record and the patterns of the same individual or doesn't: if the distance between the records of the individual and the corresponding patterns is smaller than the distance between those same patterns and any other individual records, then the matching is successful. Therefore, following the traditional approach for distance based records linkage, risk is evaluated for the entire dataset, based on the number of individuals for which the adversary correctly guesses the matching of Basket History and set of patterns. Formally:

Definition 17. ***Dataset Risk*** *Given a distance function dist, let U_{set} be the set of all individuals of retail dataset D and $M \subseteq U_{set}$ be the set of individuals for whom $dist(Pa_u, Hs_u)$ has the minimum value $\forall Pa_i \in \mathcal{P}$. Then, we define the privacy of the dataset D as: $Risk = \frac{|M|}{|U_{set}|}$.*

This approach dates back to [142, 73]. Dataset Risk ranges between 0 to 1 and quantifies the level of success that an adversary would have when using $\mathcal{P}$ as background knowledge. In our work we focus on patterns obtained through transactional clustering, performed with the algorithm TX-Means [61]. TX-Means is a parameter-free clustering

method that follows a clustering strategy similar to X-Means [118] designed for finding clusters in the specific context of transactional data. TX-Means automatically estimates the number of clusters and it also provides the *representative basket* of each cluster, which summarizes the pattern captured by that cluster. The representative baskets correspond to the centroids of the sub-clusters and are calculated adopting the procedure described in [53]. Another example of pattern can represent the top-k frequent items, where each individual is simply represented by the items that she bought with more frequency. In any case, for retail data, a *pattern* may be modeled similarly to a set of baskets.

Definition 18. ***Retail Patterns.*** *We define as $Pa_u = \{pa_1, pa_2, \ldots, pa_m\}$ the sets of patterns of the individual u, where each $pa_i \subseteq IT$ and IT is the set of all items.*

To perform distance based record linkage with individual purchasing patterns, we have to define a distance function between basket histories and sets of patterns. Since both single baskets and single retail patterns are sets of items, we propose the use of a modified version of the Jaccard distance.

Definition 19 (**Jaccard Distance**)**.** *Let A and B be two sets. The Jaccard distance is defined as: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$.*

Both individual baskets and individual patterns are sets of items. We therefore need to extend the basic definition of the jaccard distance to operate on sets of sets.

Definition 20 (**Minimum Jaccard**)**.** *Let a and $Y = \langle b_1, b_2, \ldots, b_m \rangle$ be a set and a set of sets respectively. The Minimum Jaccard distance is defined as: $MJ(a,Y) = min_{i=1,2,\ldots,m}(J(a,b_i))$*

Definition 21 (**Best Jaccard**)**.** *Let $X = \langle a_1, a_2, \ldots, a_n \rangle$ and $Y = \langle b_1, b_2, \ldots, b_m \rangle$ be two set of sets, with $n \leq m$. The Best Jaccard distance is defined as: $BJ(A,Y) = \sum_{i=1}^{n} MJ(a_i, Y)$*

So in our context an adversary can apply the Best Jaccard distance to determine the distance between a set of individual patterns in $Pa \in \mathcal{P}$ and a basket history $Hs_u \in D$. To summarize the risk assessment process for this approach is composed of the following steps: (i) we assume that an adversary gets access to a dataset of individual patterns $\mathcal{P}$ extracted from a retail dataset D We will show how this methodology can be applied to purchasing patterns in retail data, in section 4.2.7.

Data Quality Extension Although quality measurement in PRUDENnce is important for understanding the amount of information preserved considering only non-risky individuals, we think that data quality cannot be evaluated in a vacuum, but needs to be contextualized with respect to the purpose for which the data will be used. This is important to move the evaluation towards a service quality assessment. The (big) data analytics literature provides some insights and guidelines on what kind of aggregation or metrics are useful for describing and understanding individuals habits and developing data-driven services. Indeed, as we presented in Chapter 3, there exist some data specific analytical metrics that are commonly used for (big) data analytics.

Based on this discussion, in this thesis we propose an extension of PRUDEnce that enables the assessment of the data quality in terms of quality of the analytical measures that we are able to preserve given a specific tolerated privacy risk threshold. In other words, our idea is to evaluate how much high risk individuals contribute to the distribution of such metrics, by systematically removing individuals with certain levels of risk from the data, and then computing the distribution of the metrics with the different data. Doing so, we obtain a set of distributions, one for each metric, level of risk and attack, highlighting the impact of high risk individuals on metrics. Visually analyzing how these distributions change with respect to the original one can give us only a *qualitatively* overview of the impact of high risk individuals. To *quantitatively* assess the difference between the metric distributions on the original data and the derived ones, we compute a distance between such distributions. To this end, we use the *Kolmogorov-Smirnov distance*. The Kolmogorov-Smirnov test is a parameter-free test of the equality of continuous one-dimensional probability distributions normally used to refuse the null hypothesis for two distributions, i.e., the hypothesis that some sample distributions belong to different original distributions. The Kolmogorov-Smirnov distance measures the largest absolute difference between two empirical cumulative distribution functions evaluated at any point. This suites our distributions of privacy risk quite well and allows us to perform an analysis of how much each metric is influenced by a specific attack. For our purposes we define a view on the original dataset that contains only the individuals under a certain level of risk:

Definition 22. ***Risk-r Dataset*** *Given a set of possible privacy attacks A, we define as D_r the view on dataset D containing only individual with privacy risk equal to or less than r under an attack $a \in A$: $D_r = \{u \in D | Risk_a(u, D) <= r\}$.*

Moreover we

Formally, given a set of privacy attacks A and a set of metrics M, $\forall m \in M$ we define as m_r the distribution of metric m on a Risk-r Dataset D_r. We denote with $KS(m_r, m_{r'})$ the Kolmogorov-Smirnov distance between distributions m_r and $m_{r'}$. With this setup we define the Metric Utility Curve as follows.

Definition 23. ***Metric Utility Curve.*** *The Metric Utility Curve $MUC(m, r, D)$ is the function that for each risk value r, computes the Kolmogorov-Smirnov distance between the distribution of m on D and the distribution m_r on D_r:*

$$MUC(r, D) = KS(m, m_r)$$

The metric utility curve allows us to quantify, depending on the kind of attack, background knowledge and level of risk, the changes in the distributions of our target metrics.

4.2 Evaluating Risk & Data Quality

As discussed above, PRUDEnce allows us to evaluate privacy risk and data quality in a systematic way and can be instantiated in different contexts. For instantiating PRUDEnce and correctly assessing privacy risk, we need to design privacy attacks specifically suited

for the kind of data under study. In this section, for each type of data we will provide the definition of a set of attacks. Moreover, we will analyze and discuss the resulting distributions of privacy risk produced by each attack. For each attack we will describe the background knowledge and define the matching function (Definition 15) used to find those individuals that match the background knowledge. Essentially, it is the mathematical formulation of how to check if a record of an individual contains a background knowledge instance for that type of attack. Most of the attacks we propose are well known in the literature. While the basic definition of Background Knowledge remains the same for all attacks, it assumes slightly different meaning depending on the actual attack. For example, an attack might consider the combinations of pure locations, another one also the timestamps, one attack may consider the neighbors of a node, etc.

Distribution of privacy risk is presented as a cumulative distribution function that, for every value of privacy risk, indicates the percentage of individuals that have up to the corresponding privacy risk value. Generally, a lower curve represents more individuals with risk towards 1.0, i.e., maximum risk. Once assessing how the privacy risk distributes over the population under observation, we will provide the evaluation of data quality varying the tolerated privacy risk thresholds. For testing our proposal, we use a subset of the data specific metrics introduced in Chapter 3. For all datasets and for all possible attacks we selected four thresholds of risk, then, we systematically eliminated from the original dataset those users who showed a risk beyond the thresholds, obtaining four different *derived datasets*: the original dataset D_1 and $D_{0.5}$, $D_{0.33}$, $D_{0.25}$ obtained removing individuals with risk greater than 0.5, 0.33 and 0.25 respectively. For visualizing our results we realize 3D plots: on x and y axes we can observe the different metrics m and levels of risk r. On the z axis we can see the Kolmogorov-Smirnov distance between the distribution of m_r and the original distribution m. The original distribution is omitted as, clearly, the distance would have been 0. With the intent of improving readability, the data is arranged in descending order of distance, so that it is easier to understand how the distributions evolve.

4.2.1 Privacy Attacks on Mobility data

Location Attack

In a Location attack the adversary knows a certain number of locations visited by the individual but she does not know the temporal order of the visits. Since an individual might visit the same location multiple times in a trajectory, the adversary's knowledge is a multiset that may contain more occurrences of the same location. This is similar to considering the locations as items of transactions. Similar attacks on transactional databases are used in [154], [164] and [165] with the difference that a transaction is a set of items and not a multiset. We denote with $L_{set}(T_u)$ the set of locations $l_i \in T_u$, i.e. coordinates, visited by u. The background knowledge category of a Location attack is defined as follows:

Definition 24. ***Location background knowledge.*** *Let k be the number of locations l_i of an individual u known by the adversary. The Location background knowledge is a set of*

configurations based on k locations, defined as $B_k = L_{set}(T_u)^{[k]}$. *Here* $L_{set}(T_u)^{[k]}$ *denotes the set of all the possible k-combinations of the elements in set* $L(T_u)$.

Since each instance $b \in B_k$ is a subset of locations $X_u \subseteq L_{set}(T_u)$ of length k, given a record $d \in D$ belonging to a generic individual u, we define the matching function as:

$$matching(d, b) = \begin{cases} true & b \subseteq d \\ false & otherwise \end{cases} \tag{4.1}$$

The matching of the geographical coordinates of the locations can be relaxed. A variation of this attack can be put into place where a sufficient condition for matching is that an adversary knows a point in a radius δ of the point of an individual. The matching function then becomes:

$$matching(d, b) = \begin{cases} true & \forall (x_i, y_i) \in b \exists (x_i^d, y_i^d) \in d | x_i^d - \delta \leq x_i \leq x_i^d + \delta \\ & \wedge\ y_i^d - \delta \leq y_i \leq y_i^d + \delta \\ false & otherwise \end{cases} \tag{4.2}$$

Because PRUDEnce generates the background knowledge instances for each attack from the original data, this is not needed, and we can use our original definition.

Location Sequence Attack

In a Location Sequence attack, introduced in [95], the adversary knows a subset of the locations visited by the individual and the temporal ordering of the visits. Given an individual u, we denote by $L_{seq}(T_u)$ the sequence of locations $l_i \in T_u$ visited by u. The background knowledge category of a Location Sequence attack is defined as follows:

Definition 25. ***Location Sequence background knowledge.*** *Let k be the number of locations* l_i *of a individual u known by the adversary. The Location Sequence background knowledge is a set of configurations based on k locations, defined as* $B_k = L_{seq}(T_u)^{[k]}$, *where* $L(T_u)^{[k]}$ *denotes the set of all the possible k-subsequences of the elements in set* $L_{seq}(T_u)$.

We indicate with $a \preceq b$ that a is a subsequence of b. Each instance $b \in B_k$ is a subsequence of location $X_u \preceq L_{seq}(T_u)$ of length k. Given a record $d \in D$ belonging to a generic individual u, we define the matching function as:

$$matching(d, b) = \begin{cases} true & b \preceq d \\ false & otherwise \end{cases} \tag{4.3}$$

Location Time Attack

In a Location Time attack, introduced in [4, 166, 34], an adversary knows a subset of the locations visited by the individual and the time the individual visited these locations. The background knowledge category of a Location Time attack is defined as:

Definition 26. ***Location Time background knowledge.*** *Let k be the number of points (l_i, t_i) of a individual s known by the adversary. The Location Time background knowledge is a set of configurations based on k points, defined as $B_k = T_u^{[k]}$ where $T_u^{[k]}$ denotes the set of all the possible k-subsequences of the points in trajectory T_u.*

Each instance $b \in B_k$ is a spatio-temporal subsequence X_u of length k. The subsequence X_u has a positive match with a specific trajectory if the latter supports b in terms of both spatial and temporal dimensions. Thus, given a record $d \in D$, we define the matching function as:

$$matching(d, b) = \begin{cases} true & \forall (l_i, t_i) \in b, \exists (l_i^d, t_i^d) \in d \mid l_i = l_i^d \wedge t_i = t_i^d \\ false & otherwise \end{cases} \tag{4.4}$$

Unique Locations Attack

In the Unique Locations attack the adversary knows a number of *unique* locations visited by an individual. This is similar to the Location attack with the difference that in frequency vectors a location can appear only once. As a consequence, this attack follows the same principle of [154, 164, 165], and the matching function is entirely similar

Frequency Attack

We introduce an attack where an adversary knows the locations visited by the individual, their reciprocal ordering of frequency, and the minimum number of visits of the individual in the locations. This means that, when searching for specific subsequences, the adversary must consider also subsequences containing the known locations with a greater frequency. We recall that in the case of frequency vectors we denote with visit $v \in W$ the pair composed by the frequent location and its frequency. We also recall that we denote with W_u the frequency vector of individual s. The background knowledge category of a Frequency attack is defined as follows:

Definition 27. ***Frequency background knowledge.*** *Let k be the number of elements of the frequency vector of individual u known by the adversary. The Frequency background knowledge is a set of configurations based on k elements, defined as $B_k = W_u^{[k]}$ where $W_u^{[k]}$ denotes the set of all possible k-combinations of frequency vector W_u.*

Each instance $b \in B_k$ is a frequency vector and given a record $d \in D$, we define the matching function as:

$$matching(d, b) = \begin{cases} true & \forall (l_i, w_i) \in b, \exists (l_i^d, w_i^d) \in W \mid l_i = l_i^d \wedge w_i \leq w_i^d \\ false & otherwise \end{cases} \tag{4.5}$$

Home And Work Attack

In the Home and Work attack introduced in [169] the adversary knows the two most frequent locations of an individual and their frequencies. It essentially assumes the same

background knowledge of Frequency attack but related only to two locations. This is the only attack where the background knowledge configuration is composed of just a single 2-combination for each individual. Mechanically, the matching function for this type of attack is identical to the matching function of the Frequency attack.

Proportion Attack

We introduce an attack assuming that an adversary knows a subset of locations visited by an individual and also the relative proportion between the number of visits to these locations. In particular, the adversary knows the proportion between the frequency of the most frequent known location and the frequency of the other known locations. This means that the candidate set of possible matches consists of all the set of locations with similar proportions. Given a set of visits $X \subset W$ we denote with $l1$ the most frequent location of X and with w_1 its frequency. We also denote with prp_i the proportion between w_i and w_1 for each $(l_i, w_i) \neq (l_1, w_1) \in X$. We then denote with LR a set of frequent locations l_i with their respective prp_i. The background knowledge category for this attack is defined as follows:

Definition 28. ***Proportion background knowledge.*** *Let k be the number of locations l_i of an individual u known by the adversary. The Proportion background knowledge is a set of configurations based on k elements, defined as $B_k = LR_u^{[k]}$ where $LR_u^{[k]}$ denotes the set of all possible k-combinations of the frequent locations l_i with associated proportion prp_i.*

Each adversary's knowledge $b \in B_k$ is a LR structure as previously defined. Given a record $d \in D$, we define the matching function as:

$$matching(d, b) = \begin{cases} true & \forall (l_i, prp_i) \in b, \exists (l_i^d, prp_i^d) \in LR^d \mid l_i = l_i^d \wedge prp_i \in \\ & [prp_i^d - \delta, prp_i^d + \delta] \\ false & otherwise \end{cases} \tag{4.6}$$

In the equation, δ is a tolerance factor for the matching of proportions. In our experiments, $\delta = 0.1$

Probability Attack

In a Probability attack an adversary knows the locations visited by an individual and the probability for that individual to visit each location. This attack is similar to the one introduced by [160] where the goal is to match m users with m public statistics, like empirical frequencies. However, there are some differences between the two attacks: the attack proposed in [160] works on two sets of data, called *strings*. One of the sets represents the published aggregated data of individuals, the other represents the auxiliary information known by the adversary about the individuals in the data. The two sets are equal in size and also all the strings in the two sets have the same length. Given these assumptions, [160] proposes an attack based on the minimum weight bipartite matching. Conversely, in our Probability attack we try to match a single background knowledge

instance with the set of probability vectors. Therefore, we can not rely on matching algorithms on bipartite graph, because we can not make assumptions regarding the length of the sets or the length of the data: in general the length of the probability vectors is not the same among the individuals and is greater than the length of the background knowledge configuration instances.

We recall that in the case of probability vectors we denote with $(l_i, p_i) \in Pr$ the pair composed by the frequent location and its probability. We also recall that we denote with Pr_u the probability vector of individual u. The background knowledge category for this attack is defined as follows:

Definition 29 (Probability background knowledge)**.** *Let k be the number of elements of the probability vector of individual u known by the adversary. The Probability based background knowledge is a set of configurations based on k elements, defined as $B_k = P_u^{[k]}$ where $P_u^{[k]}$ denotes the set of all possible k-combinations of probability vector P_u.*

Each adversary's knowledge $b \in B_k$ is a probability vector and given a record $d \in D$, we define the matching function as:

$$matching(d, b) = \begin{cases} true & \forall (l_i, p_i) \in b, \exists (l_i^d, p_i^d) \in d \mid l_i = l_i^d \wedge p_i \in [p_i^d - \delta, p_i^d + \delta] \\ false & otherwise \end{cases} \tag{4.7}$$

In the equation, δ is a tolerance factor for the matching of probabilities. In our experiments, $\delta = 0.1$

Discussion The set of attacks introduced in this section covers most of the dimensions that is possible to consider for human mobility data. The Location Attack, Location Sequence Attack and Location Time Attack can be performed directly on trajectories, while the other attacks require an aggregation of the original trajectory, i.e., a count of the frequency of visits of the different locations. The Location Time Attack is especially of interest: if we assume that an adversary may acquire the background knowledge from direct observation, it is reasonable then to assume that the adversary will store both geographical and temporal whereabouts of the individuals under observation. In this sense, the Location Attack and Location Sequence attack can be seen as relaxations of the Location Time Attack. For the attacks on aggregated structure, such as the Location Frequency Attack for example, we can envision a scenario where an adversary acquires the background knowledge through a source other than direct observation, such as photos, previously released aggregated data or stationary monitoring. One interesting possibility is a combined attack, where an adversary may use multiple background knowledge to conduct an attack. However, as we will show in our experimental results, privacy risk computed through single background knowledge attacks is already considerably high, therefore we think that such a possibility can be further investigated in the future.

4.2.2 Mobility privacy risk assessment in scikit-mobility

In the context of privacy risk assessment with PRUDEnce on mobility data, we developed the privacy module of the Scikit-Mobility python library [114] [110]. Scikit-mobility is

an open python library that provides a comprehensive set of methods and functions to manage and analyze mobility data. It extends the well known Pandas DataFrame library, providing a special Trajectory Dataframe to handle mobility data. In the privacy module of scikit-mobility we provide the implementation of the attack models defined for mobility data in section 4.2.1, each implemented as a python class. For example the location attack model, implemented in the ***LocationAttack*** class, implements the Location Attack that we defined in section 4.2.1. Attacks can be instantiated by providing the background knowledge length:

```
Python> import skmob
Python> from skmob.privacy import attacks
Python> at = attacks.LocationAttack(knowledge_length=2)
```

To assess the re-identification risk associated with a mobility data set, we specify it as input to the ***assess_risk*** function of an attack model. This will generate the background knowledge instances of length k and evaluate privacy risk with a worst case approach:

```
Python> tdf = TrajDataFrame.from_file(filename="privacy_sample.csv")
Python> tdf_risk = at.assess_risk(tdf)
Python> print(tdf_risk.head())
```

```
  uid        risk
0   1    0.333333
1   2    0.500000
2   3    0.333333
3   4    0.333333
4   5    0.250000
```

Since risk assessment may be time-consuming for more massive datasets, scikit-mobility provides the option to focus only on a subset of the objects with the argument ***targets***:

```
Python> tdf_risk = at.assess_risk(tdf, targets=[1,2])
Python> print(tdf_risk)
```

```
  uid        risk
0   1    0.333333
1   2    0.500000
```

During the computation, not necessarily all instances of background knowledge are evaluated when assessing the re-identification risk of an invididual: when the combination with maximum re-identification risk (e.g., risk 1) is found for a moving object, all the other combinations are not computed, so as to make the computation faster. However, if the user wants all combinations to be computed anyway, they can set the argument ***force_instances***:

```
Python> tdf_risk = at.assess_risk(tdf, targets=[2], force_instances=True)
Python> print(tdf_risk)
```

```
        lat       lon             datetime uid instance instance_elem     risk
0 43.843014 10.507994 2011-02-03 08:34:04   1        1             1 0.333333
1 43.544270 10.326150 2011-02-03 09:34:04   1        1             2 0.333333
2 43.843014 10.507994 2011-02-03 08:34:04   1        2             1 0.250000
3 43.544270 10.326150 2011-02-03 09:34:04   1        2             2 0.250000
4 43.779250 11.246260 2011-02-04 10:34:04   1        3             1 0.250000
5 43.708530 10.403600 2011-02-03 10:34:04   1        3             2 0.250000
```

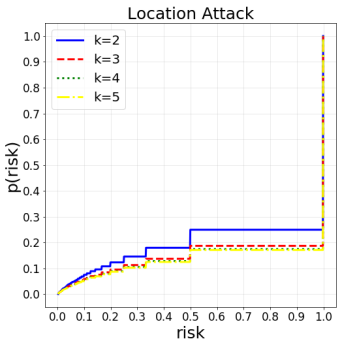
The result is a
textbfemphDataFrame that contains a reference number of each combination under the attribute ***instance*** and, for each instance, the ***risk*** and each of the locations comprising that instance indicated by the attribute ***instance_elem***. Scikit-mobility is still in development, and more functionalities will be added in the future.

4.2.3 Risk Distributions on Mobility data

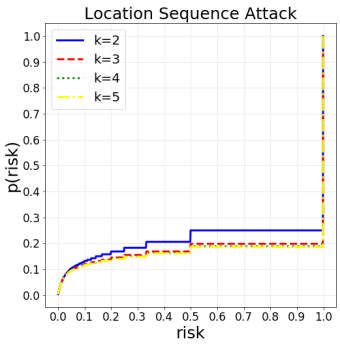
We simulated attacks using $k = 2, 3, 4, 5$ on our mobility experimental data introduced in Section 3.4.1: the cumulative distribution functions for the mobility attacks are depicted in Figure 4.2 and Figure 4.3, where we can see that the privacy risk increases not only with increasing the amount of knowledge, but also with increasing k. Since the Home&Work attack only considers the two most frequent locations, there is only a single distribution for it. It is interesting to note the evident gap between Location attack, from $k = 2$ snd $k = 3$, suggesting that, for attacks with a less dimensions in the background knowledge, increasing the size of the configuration has a greater impact than for attacks with more dimensions. For the Location Time attack, since here the background knowledge is already detailed, we can see that the increasing of k does not change so much the levels of privacy risk. The number of individuals with maximum risk of re-identification ranges from 60% for the Location attack to more that 80% for the Location Time attack, while we do have an increase in the number of individuals with risk of re-identification of 50% (or less) across the board. The cumulative distribution function of risk is quite stable for the other types of attack, varying k and the background knowledge category. This can probably be due to the fact that, with vectors, we are dealing with distinct locations for each individual, thus, since many individuals have few distinct locations, the risk remains very similar.

4.2.4 Mobility Data Quality Experiments

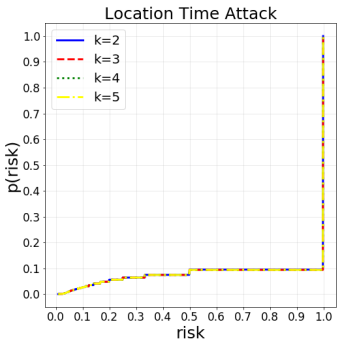
In order to analyse the data quality on our mobility datasets, we select attacks based on the background knowledge configuration with $k = 2$. In Figures 4.4 and 4.5, for a subset of metrics, we can visually analyze how their distributions change varying the different levels of risk.



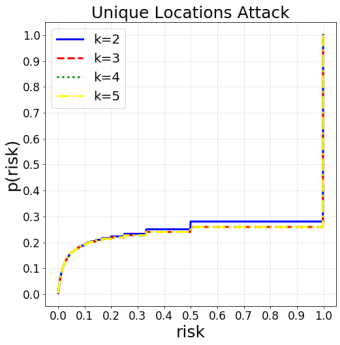
(a) Location



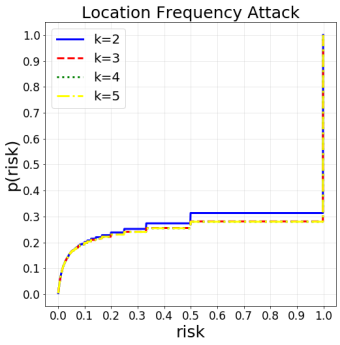
(b) Location Sequence



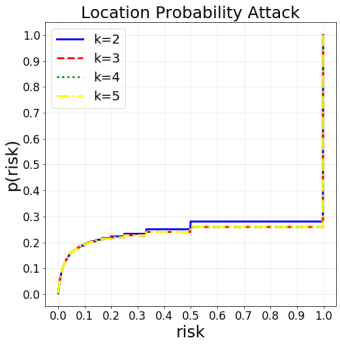
(c) Location Time (day)



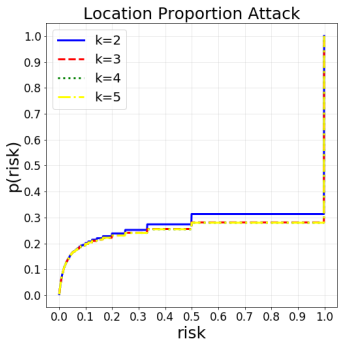
(d) Unique Locations



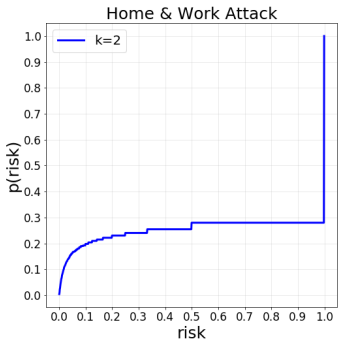
(e) Frequency



(f) Probability

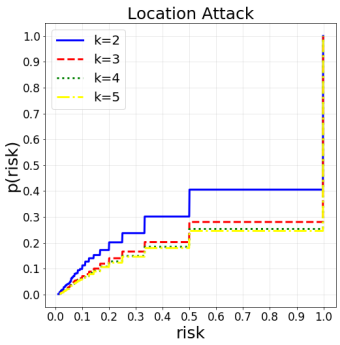


(g) Proportion

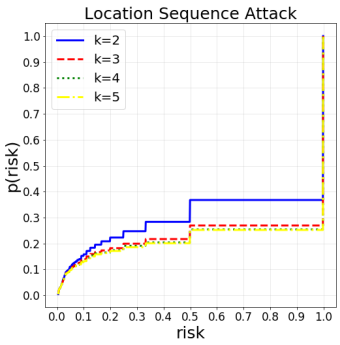


(h) Home&Work

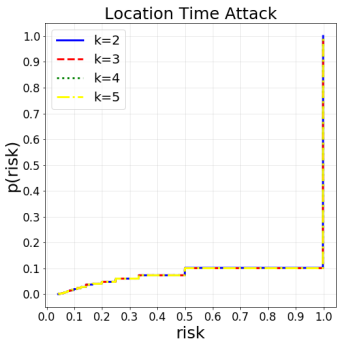
Figure 4.2: Cumulative distributions of privacy risk for Florence dataset.



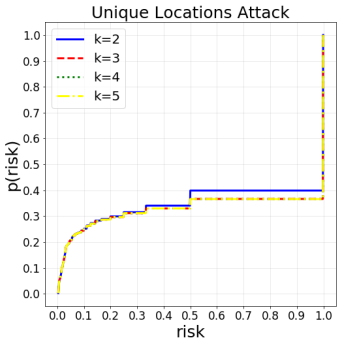
(a) Location



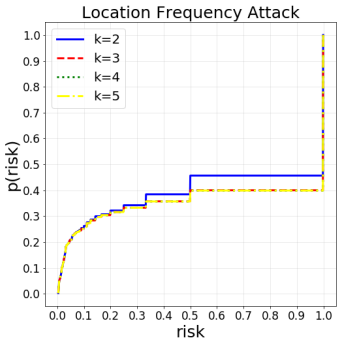
(b) Location Sequence



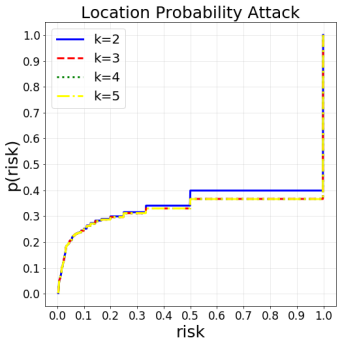
(c) Location Time (day)



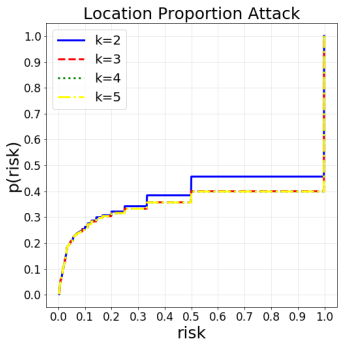
(d) Unique Locations



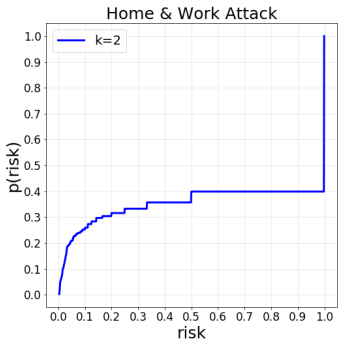
(e) Frequency



(f) Probability

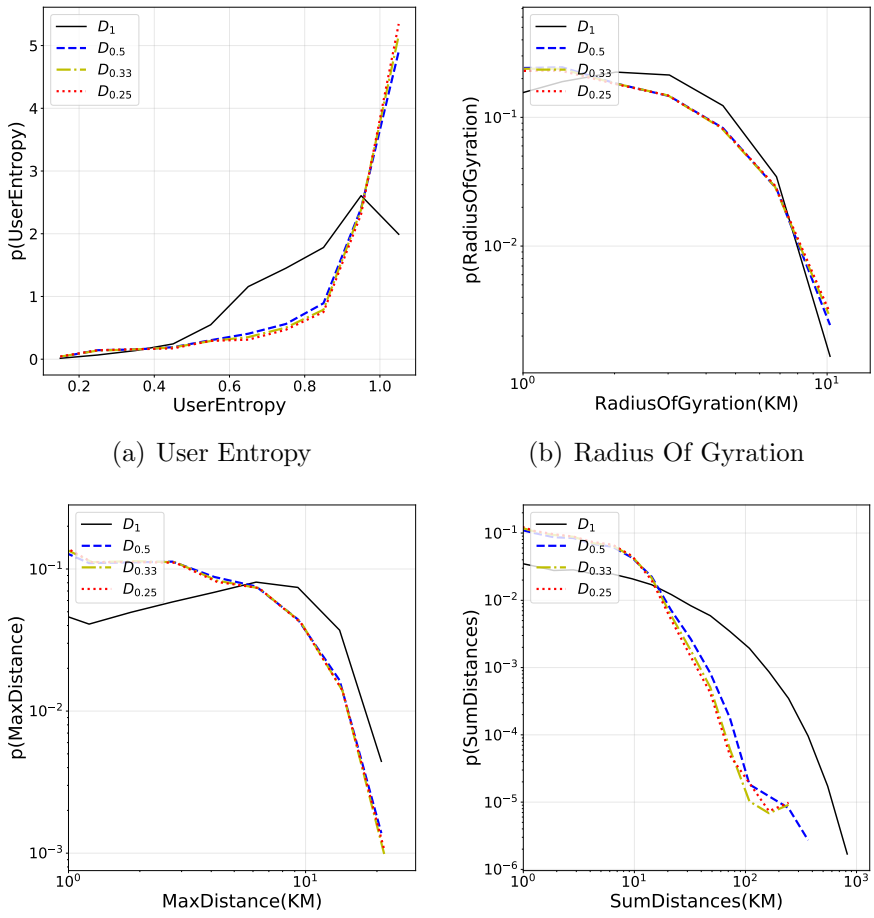


(g) Proportion



(h) Home&Work

Figure 4.3: Cumulative distributions of privacy risk for Pisa dataset.



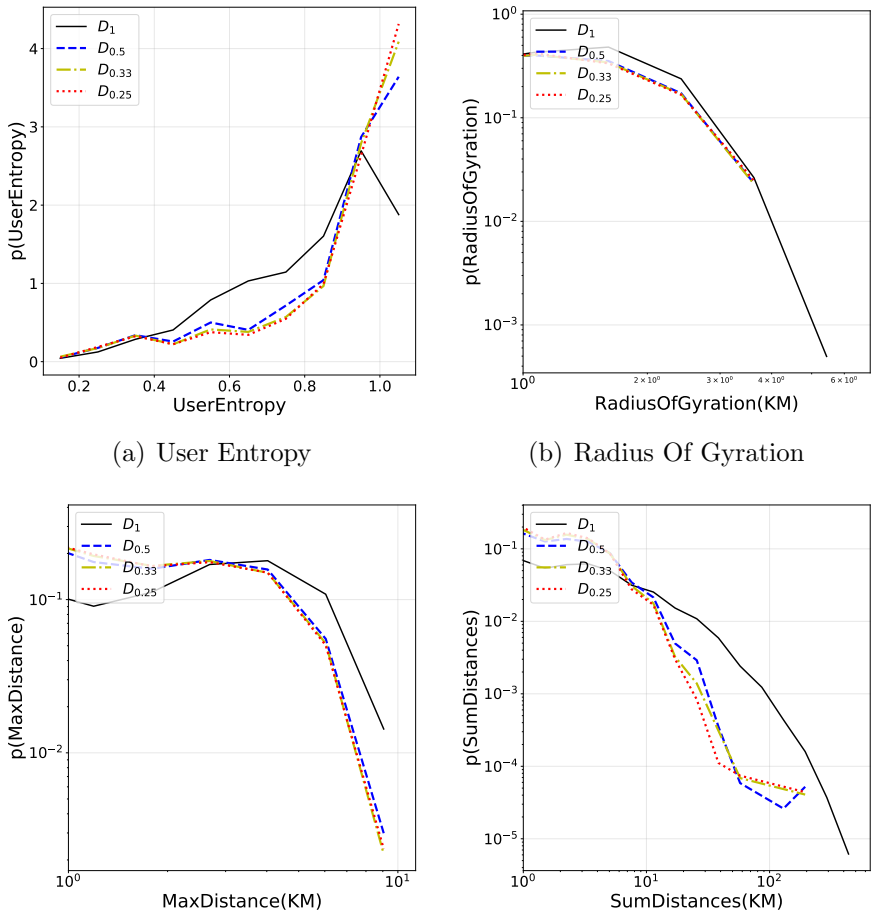
(a) User Entropy

(b) Radius Of Gyration

(c) Max Distance

(d) Sum Distances

Figure 4.4: Some examples of distributions of mobility metrics on the city of Florence for the Location Sequence attack.



(a) User Entropy

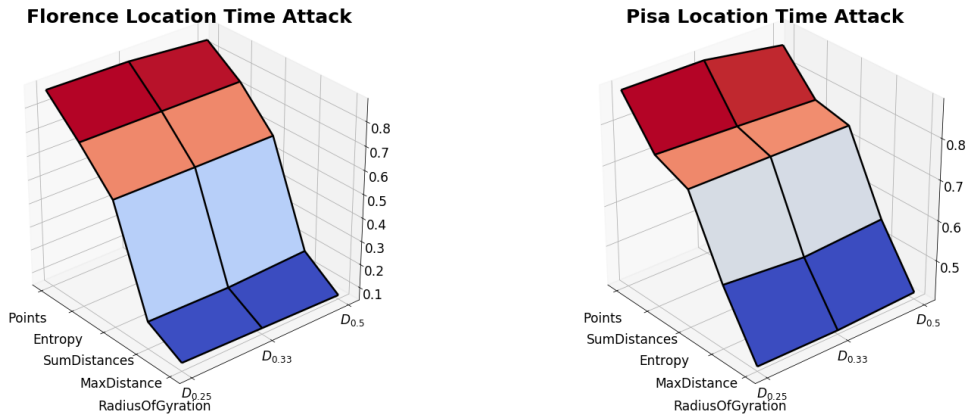
(b) Radius Of Gyration

(c) Max Distance

(d) Sum Distances

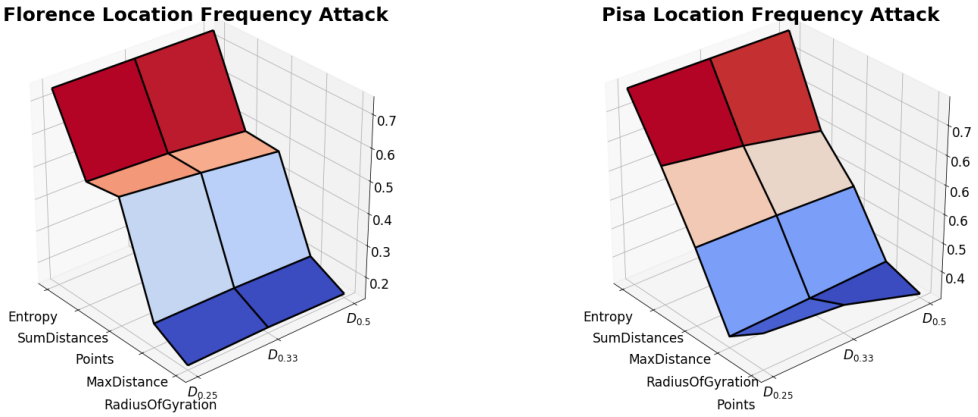
Figure 4.5: Some examples of distributions of mobility metrics on the city of Pisa for the Location Sequence attack.

We observe for example how entropy changes drastically with the deletion of high risk individuals, as it assumes value 1 for almost all remaining individuals. Radius of gyration distribution instead, remains similar in shape, while loosing some range. For the Sum of Distances we tend to loose individuals traveling a long distance. We can quantify the changes using our methodology. Figures 4.6 and 4.7 depict the results for two of the attacks defined above: Location Time attack and Location Frequency.



(a) MUC for Florence Location Time Attack (b) MUC for Pisa Location Time Attack

Figure 4.6: Examples of MUC for location time attack (daily precision) with k=2 on mobility data, considering five different metrics and three levels of risk .



(a) MUC for Florence Locaiton Frequency Attack (b) MUC for Pisa Location Frequency Attack

Figure 4.7: Examples of MUC for location frequency attack with k=2 on mobility data, considering five different metrics and three levels of risk .

It interesting to note that, the second most "affected" distribution is different for Florence (entropy) and Pisa (the sum of distances). This can also be seen in the actual distributions. We observe that generally the two mobility datasets present a similar evolution in terms of distributions: radius of gyration is consistently the metric that is least influence by the removal of high risk individuals, and the Kolmogorov-Smirnov

distance overall is pretty similar in scale regardless of the type of attack. A complete report describing the distances for all attacks with background knowledge configuration $k = 2$ and for each metric is presented in Tables 4.1 and 4.2.

Type of Attack		**Points**	**Radius Of Gyration**	**Entropy**	**Max Distance**	**Sum Distances**
Location	$D_{0.50}$	0.359677	0.167552	0.415215	0.218442	0.391691
	$D_{0.33}$	0.395061	0.200264	0.449218	0.258189	0.433416
	$D_{0.25}$	0.411943	0.216435	0.473944	0.274133	0.461593
Location Sequence	$D_{0.50}$	0.442170	0.147981	0.469349	0.211341	0.455297
	$D_{0.33}$	0.500618	0.150159	0.538881	0.212397	0.493271
	$D_{0.25}$	0.541979	0.150611	0.593628	0.214139	0.519339
Location Time	$D_{0.50}$	0.844666	0.062926	0.737904	0.169192	0.586111
	$D_{0.33}$	0.872643	0.072453	0.738757	0.157109	0.585210
	$D_{0.25}$	0.877650	0.081967	0.738512	0.163196	0.586076
Frequency	$D_{0.50}$	0.394806	0.139379	0.451632	0.190086	0.414871
	$D_{0.33}$	0.440141	0.148917	0.530524	0.203873	0.458763
	$D_{0.25}$	0.462248	0.155649	0.579349	0.208444	0.480384
Unique Locations	$D_{0.50}$	0.439574	0.150832	0.523422	0.199738	0.458949
	$D_{0.33}$	0.474875	0.155903	0.596760	0.209309	0.490806
	$D_{0.25}$	0.498499	0.155710	0.641595	0.212690	0.508671
Probability	$D_{0.50}$	0.439574	0.150832	0.523422	0.199738	0.458949
	$D_{0.33}$	0.474875	0.155903	0.596760	0.209309	0.490806
	$D_{0.25}$	0.498499	0.155710	0.641595	0.212690	0.508671
Proportion	$D_{0.50}$	0.394806	0.139379	0.451632	0.190086	0.414871
	$D_{0.33}$	0.440141	0.148917	0.530524	0.203873	0.458763
	$D_{0.25}$	0.462248	0.155649	0.579349	0.208444	0.480384
Home&Work	$D_{0.50}$	0.166596	0.044827	0.074761	0.032288	0.118579
	$D_{0.33}$	0.219245	0.043172	0.098187	0.053131	0.164503
	$D_{0.25}$	0.240914	0.053229	0.110219	0.058030	0.180603

Table 4.1: Numerical values of MUC for mobility data of Florence. Each cell shows the Kolmogorov-Smirnov distance between the distribution of the metric, computed at that risk level for that particular attack, and the original distribution of that metric. The table only shows a subset of metrics to accommodate for space.

Type of Attack		**Points**	**Radius Of Gyration**	**Entropy**	**Max Distance**	**Sum Distances**
Location	$D_{0.50}$	0.298162	0.431417	0.427972	0.515758	0.506594
	$D_{0.33}$	0.351322	0.448418	0.492403	0.551679	0.568636
	$D_{0.25}$	0.387730	0.452953	0.524274	0.565963	0.602633
Location Sequence	$D_{0.50}$	0.366946	0.432377	0.488653	0.540113	0.579720
	$D_{0.33}$	0.432894	0.430649	0.557397	0.545854	0.616439
	$D_{0.25}$	0.468165	0.430749	0.604883	0.550325	0.650423
Location Time	$D_{0.50}$	0.852873	0.417968	0.740254	0.553219	0.759847
	$D_{0.33}$	0.884619	0.410767	0.740254	0.541821	0.768367
	$D_{0.25}$	0.884619	0.412732	0.740254	0.559902	0.775720
Frequency	$D_{0.50}$	0.316738	0.425556	0.447410	0.508401	0.518250
	$D_{0.33}$	0.370089	0.434557	0.518652	0.532268	0.582983
	$D_{0.25}$	0.404861	0.430955	0.586334	0.535159	0.612783
Unique Locations	$D_{0.50}$	0.367679	0.434430	0.502292	0.525753	0.573248
	$D_{0.33}$	0.408506	0.435005	0.596708	0.534746	0.615600
	$D_{0.25}$	0.427473	0.430248	0.645288	0.539718	0.625272
Probability	$D_{0.50}$	0.367679	0.434430	0.502292	0.525753	0.573248
	$D_{0.33}$	0.408506	0.435005	0.596708	0.534746	0.615600
	$D_{0.25}$	0.427473	0.430248	0.645288	0.539718	0.625272
Proportion	$D_{0.50}$	0.316738	0.425556	0.447410	0.508401	0.518250
	$D_{0.33}$	0.370089	0.434557	0.518652	0.532268	0.582983
	$D_{0.25}$	0.404861	0.430955	0.586334	0.535159	0.612783
Home&Work	$D_{0.50}$	0.219668	0.387994	0.213467	0.435054	0.351008
	$D_{0.33}$	0.275876	0.393288	0.239865	0.447088	0.395238
	$D_{0.25}$	0.338158	0.387944	0.253886	0.443319	0.416260

Table 4.2: Numerical values of MUC for mobility data of Pisa. Each cell shows the Kolmogorov-Smirnov distance between the distribution of the metric, computed at that risk level for that particular attack, and the original distribution of that metric. The table only shows a subset of metrics to accommodate for space.

4.2.5 Privacy Attacks on Retail data

Intra-Basket Attack

In a Intra-Basket attack we assume that the adversary has as background knowledge a subset of products bought by her target in a certain shopping session. For example, the adversary once saw the subject at the workplace with some highly perishable food, that are likely bought together. Thus, each $b \in B_k$ is a subset of items. In the following, we denote with $I_{set}(ba_i)$ the set of products belonging to the basket ba_i. The background knowledge category of a Intra-Basket attack is defined as follows:

Definition 30. ***Intra-Basket Background Knowledge.*** *Let k be the number of items bought by an individual u and known by the adversary. The Intra-Basket background knowledge is a set of configurations based on k items, defined as $B_k = \bigcup_{ba_i \in Hs_u} I_{set}(ba_i)^{[k]}$. Here, $I_{set}(ba_i)^{[k]}$ denotes the set of all the possible k-combinations of items in $I_{set}(ba_i)$.*

Since each instance $b \in B_k$ is composed of a subset of purchased products $X_u \subseteq I_{set}(ba_i)$ of length k, given a record $d = Hs_u \in D$ belonging to a generic individual u, we define the matching function as:

$$matching(d, b) = \begin{cases} true & \exists\ ba_j \in d \mid b \subseteq ba_j \\ false & otherwise \end{cases} \tag{4.8}$$

Full Basket Attack

In a full basket attack we assume that the adversary knows the contents of a shopping basket of her target. For example, the adversary once gained access to a shopping receipt of her target. Note that in this case it is not necessary to establish k, i.e., the background knowledge configuration has a fixed length, given by the number of items of a specific shopping basket.

Definition 31. ***Full Basket Background Knowledge.*** *A Full Basket background knowledge instance b is an entire basket ba_i of individual u. The Full Basket background knowledge configuration is defined as $B = Hs_u$.*

Since each instance $b = ba_i \in B$ is composed of a shopping basket ba_i, given a record $d = Hs^u \in D$ belonging to an individual u, we define the matching function as:

$$matching(d, b) = \begin{cases} true & \exists\ ba_j \in d \mid b = ba_j \\ false & otherwise \end{cases} \tag{4.9}$$

Discussion Attacks on retail data are based on the assumption that the adversary can somehow recover the information of what an individual has bought. The Intra-Basket attack can represent the case where, for example, an adversary gets to directly see some of the products bought by a customer during a shopping session. The Full Basket attack instead, covers a scenario where an adversary acquires a shopping bill belonging to a customer, thus knowing all the products bought during the shopping session. Clearly, the Full Basket attack represents an extreme case of Intra-Basket attack, where the knowledge is of an entire basket, regardless of the size of such basket. The information about what products a person has bought in the past may be a very sensitive one, because for example certain particular diets may indicate diseases or religious beliefs.

4.2.6 Risk Distributions on Retail data

We simulated the two attacks on our retail experimental dataset (Section 3.4.2). For the Intra-basket attack we consider two sets of background knowledge configuration B_k with $k = 2, 3$, while for the Full Basket attack we have just one possible background knowledge configuration, where the adversary knows an entire basket of an individual.

We show in Figure 4.8 the cumulative distributions of privacy risks. For the Intra-basket attack, with $k = 2$ we have almost 75% of customers for which privacy risk is to equal 1. Switching to $k = 3$ causes a sharp increase in the overall risk: more than

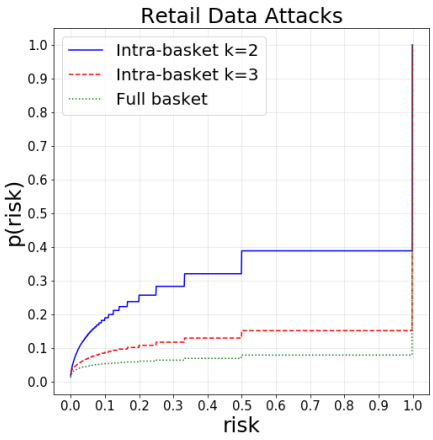


Figure 4.8: Cumulative distributions of privacy risk for retail data.

98% of individuals have maximum privacy risk (i.e., 1). The difference between the two configurations is remarkable, showing how effective an attack could be with just 3 items. We also performed a simulation with higher k obtaining risk 1 for almost all individuals; we do not show this result as it is uninformative. Stressing this fact, the Full Basket attack behaves similarly: out of 1000 individuals, only five show risk less than 1. Since most of customers are already re-identified, further increasing the quantity of knowledge (e.g., exploiting higher k or the Full Basket attack) does not offer additional gain. Similar results were obtained for movie rating dataset in [101].

4.2.7 Individual Patterns Risk Experiments

In order to extract meaningful patterns from retail data, we use a slightly different dataset than the one defined in 3.4.2: We analyzed a dataset of 2,021,414 shopping sessions, i.e., baskets, performed by 8564 individuals between the 2010 and 2012 in Leghorn province. These customers are "loyal customers", i.e., customers active in at least ten months every year. For each customer we have on average 240 baskets, containing 100 different items, and the average basket length is 8 items. For each customer we extracted her typical patterns using two approaches: first, we built simple baseline patterns made by selecting the top k most frequently bought items for each individual. Using this baseline approach for the patterns extraction, we obtained patterns considering the k-most frequent items for each person, with k ranging from 1 to 5. Then we extracted frequent purchasing patterns using Tx-means algorithms. Applying TX-Means we extracted a total of 38,068 patterns, more than 4 patterns per individual on average.

Simple Patterns Against Baskets The first experiment is based on the simulation of a patterns against baskets attack using simple patterns. We recall that for this attack risk is evaluated globally for the entire data-set and not individually. We performed distance based record linkage with simple patterns of 2, 4 and 5 items. For simple patterns of length 2 we have only 27 correct matches out of the total population of 8,564 customers.

This yields a risk of 0.003. For simple patterns of length 4 we have 298 correct matches, yielding a risk of 0.034. For patterns of length 5 we have 388 correct matches, yielding a risk of 0.045. These low values are mainly due to the fact that simple patterns are not particularly representative of the individual's baskets, as they capture only the overall frequency of the most bought items, disregarding periodicity of purchases or sequences of purchases. Also, each pattern at length k *contains* in a sense the patterns with length $\leq k$ and this diminishes the information used for the linkage. Because of how we compute distance, this implies that such distance fall in the range 0 to 1. This leads to a high number of individuals with minimum distance, i.e. 0, therefore impeding a univocal matching. We can conclude that simple patterns pose a relatively low threat when used to attack the raw data.

Length of Pattern	**Matched individuals**	**Privacy Risk**
$k = 2$	27	0.003
$k = 4$	298	0.034
$k = 5$	388	0.045

Table 4.3: Results of an attack with simple baskets against retail data

TX-means Patterns Against Baskets The second experiment is based on the simulation of a patterns against baskets attack using the patterns extracted with the TX-means clustering algorithm. As for the previous case, the risk is calculated for the entire data-set. With the TX-means patterns we have that 5,781 individuals out of the total population of 8,564 customers are correctly matched, i.e., the distance between the TX-means patterns of those individuals and their basket data is minimal. This yields a risk of 0.675. We can now characterize the individuals correctly matched, by looking at their patterns and baskets.

	Patterns: std length	Patterns: mean length	Num patterns	Num baskets	Baskets: std length	Baskets: mean length
mean	4.811004	13.049558	4.820446	244.230064	6.002396	10.897940
std	3.996948	7.899513	3.453788	201.790281	2.873166	5.264362
min	0.000000	2.200000	1.000000	10.000000	0.708363	1.744063
max	26.051631	71.000000	25.000000	1646.000000	26.411782	43.282051

Table 4.4: Characterization of matched individuals in the TX-means patterns against baskets attack

In Table 4.4 and Table 4.5 we gathered some statistics for the individuals correctly matched and those who were not matched. For each individual, we gathered the mean length of her patterns and her baskets as well as the standard deviation for such lengths and the number of patterns and baskets. In the tables we show mean, standard deviation, min value and max value for the aforementioned measures. If we compare the statistics in the two table we can see that there are not many differences. However, we observe

	Patterns: std length	Patterns: mean length	Num patterns	Num baskets	Baskets: std length	Baskets: mean length
mean	2.884773	10.653819	3.665469	219.015451	4.884745	8.000338
std	3.385043	7.122223	3.735964	220.721776	2.372840	3.951333
min	0.000000	1.000000	1.000000	10.000000	0.535428	1.221429
max	25.500000	53.000000	26.000000	2025.000000	16.146130	31.976744

Table 4.5: Characterization of non matched individuals in the TX-means patterns against baskets attack

that, for the individuals that were not re-identified by the attack, we have fewer, shorter patterns and baskets on average. This suggests us that lengthier shopping sessions or

4.2.8 Retail Data Quality Experiments

In order to analyse the data quality on our retail dataset, we considered both the attacks that we introduced above, using two different configurations ($k = 2$ and $k = 3$) for the Intra-Basket attack. We use only two configurations because, as shown in experimental results, for $k > 3$ privacy risk equals 1 for almost 98% of the population.

In Figures 4.9 and 4.10 we can visually analyze how distributions of a subset of metrics change varying the different levels of risk. In particular, Figure 4.9 showes the results for the Full Basket attack while Figure 4.10 depicts the results for the Intra-Basket attack with $k = 3$. Looking at the distributions we notice how, understandably, more knowledge leads to higher risk and hence higher risk leads to heavier distortions in the distributions. Moreover, beyond individuals with risk less than 0.5, we delete very few other individuals, and distributions stay consistent after the first round of data suppression. We can quantify the metric distribution changes using our methodology. A visualization of some of the results is shown in Figure 4.11.

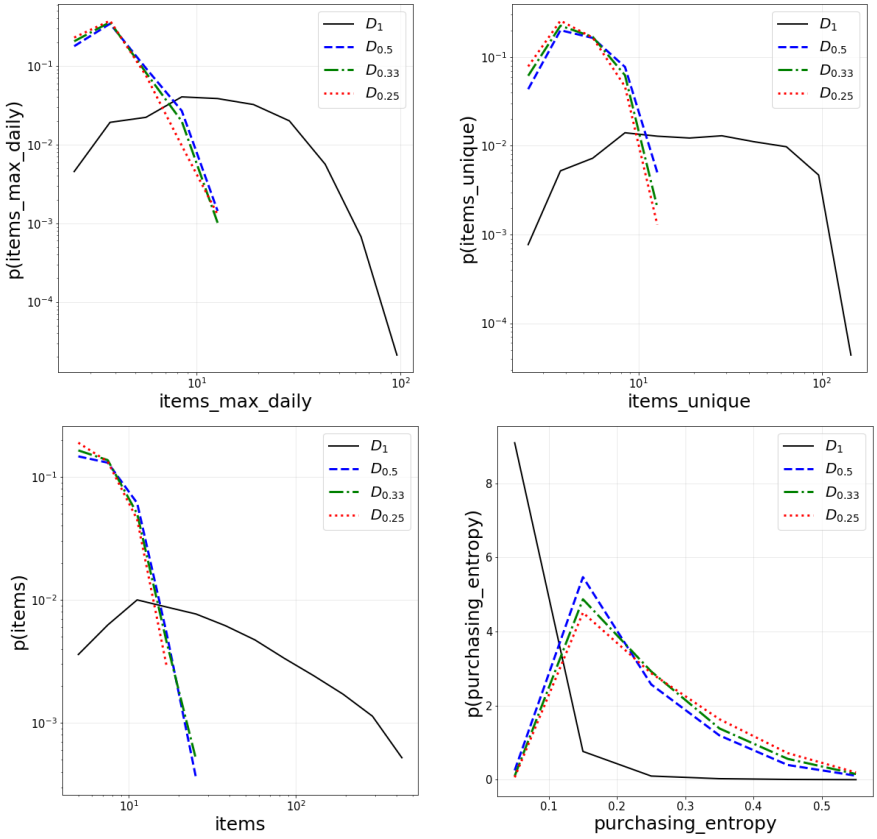


Figure 4.9: Some Distributions of retail metrics for the Full Basket attack.

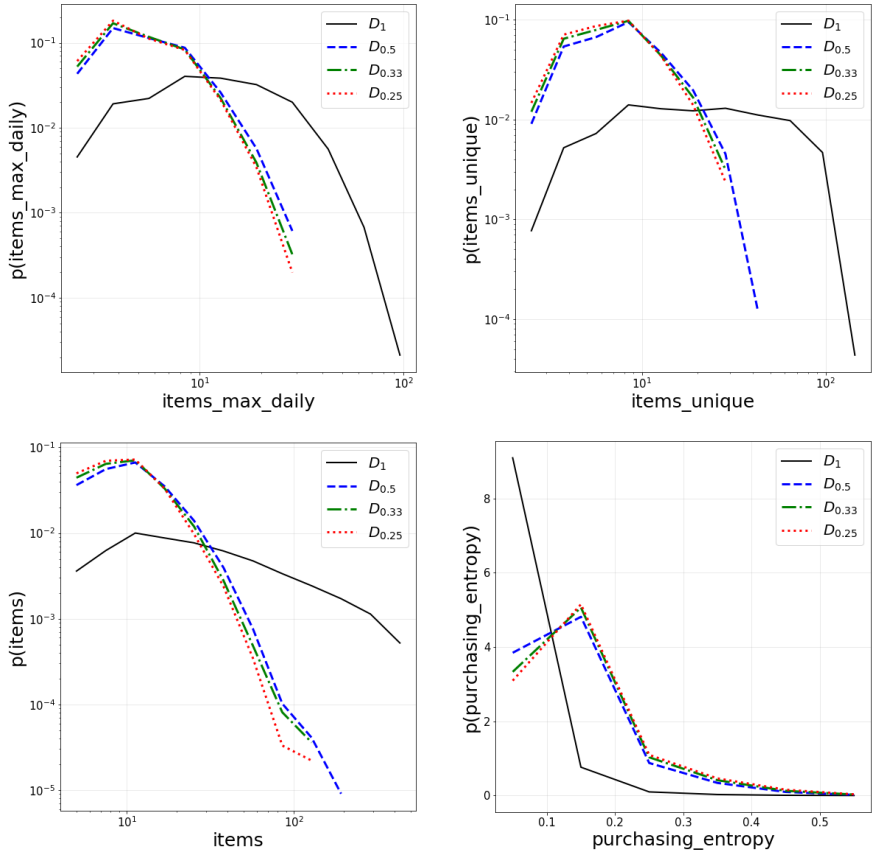
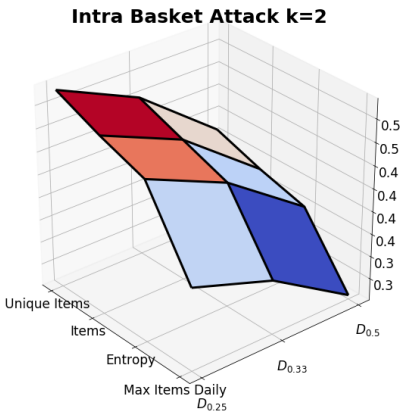
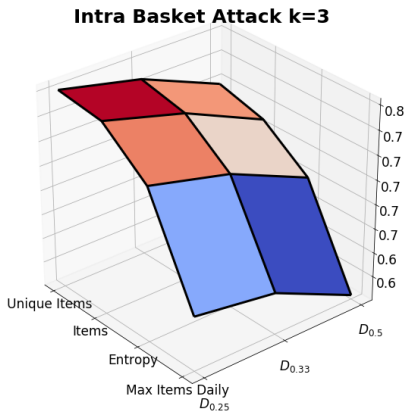


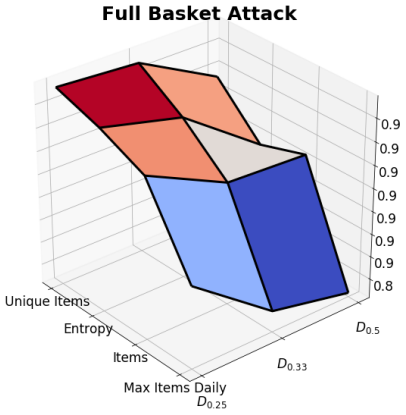
Figure 4.10: Some distributions of retail metrics for the Intra Basket attack (k=3).



(a) MUC for Intra Basket Attack k=2



(b) MUC for Intra Basket Attack k=3



(c) MUC for Full Basket Attack

Figure 4.11: Examples of MUC for various attacks with different configurations on retail data considering four different metrics and three levels of risk .

While the absolute distance assumes different values depending on the kind of attack, we can see that the overall shape of the MUC for each metric is very similar regardless of the kind of attack. The difference lies in the actual magnitude of the distance, which we can see on the z axis: distance is much higher for the Full Basket attack, as expected. This may suggest that increasing the dimension or the kind of attack has a greater impact on metrics than increasing the size of the configuration. A complete report on all the distances for all attacks and for individual measures can be seen in Table 4.6.

Type of Attack		**Items**	**Unique Items**	**Entropy**	**Max Items Daily**
Intra Basket k=2	$D_{0.5}$	0.365551	0.383618	0.351062	0.280305
	$D0.33$	0.432584	0.452118	0.413392	0.336022
	$D_{0.25}$	0.470682	0.492387	0.452525	0.367345
Intra Basket k=3	D_{05}	0.708717	0.719115	0.680781	0.603564
	$D0.33$	0.738407	0.746670	0.709889	0.634754
	$D_{0.25}$	0.756853	0.761828	0.726682	0.645237
Full Basket	D_{05}	0.893437	0.906917	0.887077	0.843385
	$D0.33$	0.895240	0.925351	0.912300	0.850966
	$D_{0.25}$	0.912196	0.927664	0.921149	0.877686

Table 4.6: Numerical values of MUC for retail data. Each cell shows the Kolmogorov-Smirnov distance between the distribution of the metric, computed at that risk level for that particular attack, and the original distribution of that metric.

4.2.9 Privacy Attacks on Network data

Neighborhood Attack

In a neighborhood attack we assume that the adversary only knows a certain number of friends/neighbors of an individual. More technically, the adversary has information about the nodes which are connected to the victim node in the social network graph. This type of attack was introduced in [175]. Background knowledge instances for this kind of attack are portions of the friendship vector F_v of an individual.

Definition 32 (Neighborhood Background Knowledge)**.** *Let k be the number of elements of the friendship vector of individual v known by the adversary. The neighborhood based background knowledge is a set of configurations based on k neighbors/friends, defined as $B_k = F_v^{[k]}$ where $F_v^{[k]}$ denotes the set of all possible k-combinations of friendship vector F_v.*

Given $b \in B_k$, an adversary neighborhood knowledge and the corresponding individual $v \in V$, we define the matching function of the neighborhood attack as follows:

$$Matching(b, F_v) = \begin{cases} true & b \subseteq F_v \\ false & \text{otherwise} \end{cases} \tag{4.10}$$

Label Pair Attack

In a label pair attack we assume that the adversary knows a certain number of pairs of labels with their values of an individual. The set of labels of a node may include the individual's demographics information (age, location, gender, occupation), interests (hobbies,movies, books, music), etc. Each label pair in key-value format $la_i = (f, l)$ is distinct in a label vector of an individual. Similar type attack has been defined in [86] by

using the label pair knowledge on two connected nodes. Background knowledge instances for this kind of attack are portions of the label pair vector La_v of an individual.

Definition 33 (Label Pair Background Knowledge)**.** *Let k be the number of elements of the label vector of individual v known by the adversary. The label pair based background knowledge is a set of configurations based on k labels, defined as $B_k = La_v^{[k]}$ where $La_v^{[k]}$ denotes the set of all possible k-combinations of label vector La_v.*

Given $b \in B_k$, an adversary label pair knowledge and the corresponding individual $v \in V$, we define the matching function of the label pair attack as follows:

$$Matching(b, La_v) = \begin{cases} true & b \subseteq La_v \\ false & \text{otherwise} \end{cases} \tag{4.11}$$

Neighborhood and Label Pair Attack

Mixing the previous two attacks, we define a new and stronger attack that we call neighborhood and label pair attack. In this case, we consider an adversary knowing a certain number of friends/neighbors and a certain number of feature labels of an individual at the same time. In other words, it combines the background knowledge of the two previous attacks. Therefore, a background knowledge instance for this kind of attack is $b = (b', b'')$, i.e., it is composed by b' that is a portion of the friendship vector F_v as well as b'' that is a portion of the label vector La_v of an individual.

Given a neighborhood and label pair knowledge $b = (b', b'')$ and the corresponding individual $v \in V$, we define the matching function of the neighborhood and label pair attack as follows:

$$Matching(b, F_v, La_v) = \begin{cases} true & b' \subseteq F_v \ \wedge \ b'' \subseteq La_v \\ false & \text{otherwise} \end{cases} \tag{4.12}$$

Friendship Degree Attack

In a friendship degree attack, the adversary knows the degree of a number of friends of the victim as well as the degree of the victim. This type of attack was introduced in [149]. A background knowledge instance for this kind of attack will be a portion of the degree vector Dg_v of an individual.

Definition 34 (Friendship Degree Background Knowledge)**.** *Let k be the number of elements of the degree vector of individual v known by the adversary. The friendship degree pair based background knowledge is a set of configurations based on k degrees, defined as $B_k = Dg_v^{[k]}$ where $Dg_v^{[k]}$ denotes the set of all possible k-combinations of degree vector Dg_v.*

Given $b \in B_k$, an adversary friendship degree knowledge and the corresponding individual $v \in V$, we define the matching function of the friendship degree attack as follows:

$$Matching(b, Dg_v) = \begin{cases} true & d_1 = len(Dg_v) \wedge d_2 \in Dg_v \forall d_2 \in b \\ false & \text{otherwise} \end{cases} \tag{4.13}$$

Mutual Friend Attack

In a mutual friend attack, the adversary knows the number of mutual friends of the victim and some of its neighbors. This type of attack was introduced in [145]. A background knowledge instance for this kind of attack will be a portion of the mutual friendship vector Mf_v of an individual.

Definition 35 (Mutual Friend Background Knowledge)**.** *Let k be the number of elements of the mutual friendship vector of individual v known by the adversary. The mutual friend based background knowledge is a set of configurations based on k mutual friends, defined as $B_k = Mf_v^{[k]}$ where $Mf_v^{[k]}$ denotes the set of all possible k-combinations of mutual friendship vector Mf_v.*

Given $b \in B_k$, an adversary mutual friend knowledge and the corresponding individual $v \in V$, we define the matching function of the mutual friend attack as follows:

$$Matching(b, Mf_v) = \begin{cases} true & b \subseteq Mf_v \\ false & \text{otherwise} \end{cases} \tag{4.14}$$

Neighborhood Pair Attack

In a neighborhood pair attack, the adversary knows subset of the friends of the victim who are friends with each other, that is a subset of F_v in which v_i and v_j are connected to each other $v_i \in F_{v_j}$, $v_j \in F_{v_i}$ and $v_i, v_j \in F_v$. For brevity, we will denote such subset as $F_{v_{pair}}$ Similar type of attack was defined in [3]. With respect to the original definition, in our work, we reduce the knowledge of the adversary by eliminating the degree of the victim. A background knowledge instance will contain pairs of connected neighbors.

Definition 36 (Neighborhood Pair Background Knowledge)**.** *Let k be the number of elements in the subset of interconnected friends of the friendship vector of individual v known by the adversary. The neighborhood pair based background knowledge is a set of configurations based on k connected friends, defined as $B_k = F_{v_{pair}}^{[k]}$ where $F_{v_{pair}}^{[k]}$ denotes the set of all possible k-combinations of the subset of friendship vector $F_{v_{pair}}^{[k]}$.*

Given $b \in B_k$, an adversary neighborhood pair knowledge and the corresponding individual $v \in V$, we define the matching function of the neighborhood pair attack as follows:

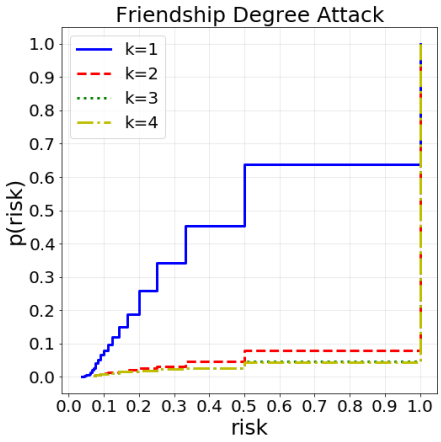
$$Matching(b, F_v) = \begin{cases} true & v_i \in F_{v_j} \wedge v_j \in F_{v_i} \wedge v_i, v_j \in F_v \forall (v_i, v_j) \in b \\ false & \text{otherwise} \end{cases} \tag{4.15}$$

Discussion Attacks on social network data exploits most of the basic topological characteristics of a network. Information such as friendships or social ties are difficult to obtain through direct observation, but are easier to obtain online. We can envision a scenario where, for example, an adversary gathers the information about the particular

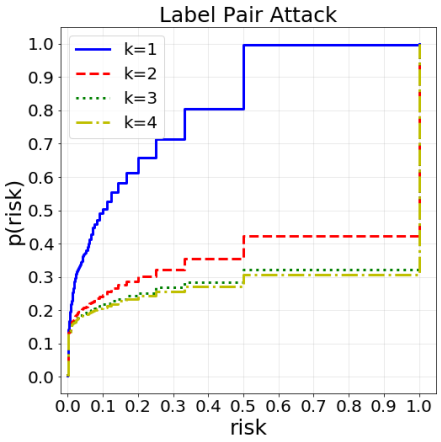
connections of an individual, possibly crawling her profiles, and then uses this information to retrieve all of his friends in the network (Friendship Attack). Social networks, especially nowadays, can be easily targeted by acquiring knowledge from other social networks: one case could be, for example, retrieving some labels from one social network and then using those labels to attack the individual in another social network.

4.2.10 Risk Distributions on Network data

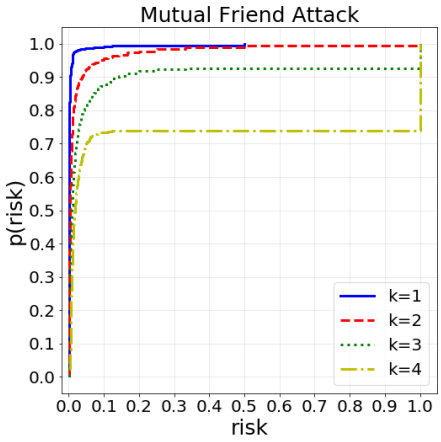
We simulated attacks using $k = 1, 2, 3, 4$ on our network experimental dataset (Section 3.4.3).



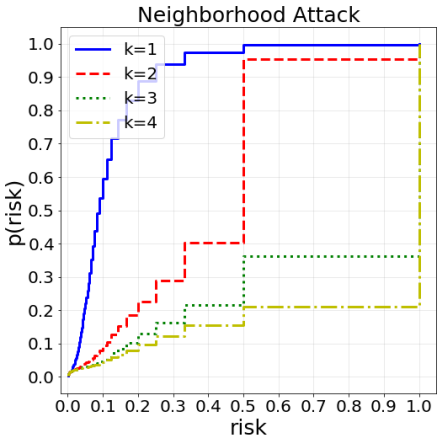
(a) Friendship Degree



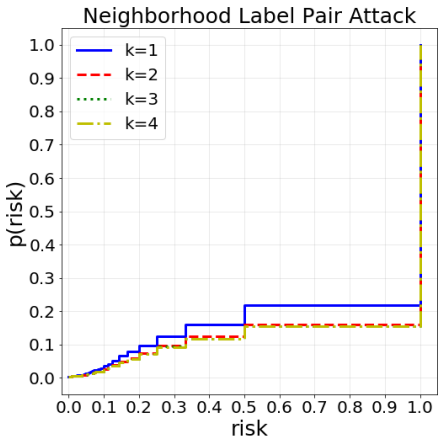
(b) Label Pair



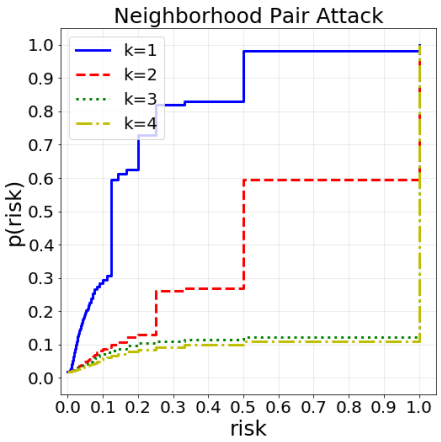
(c) Mutual Friend



(d) Neighborhood



(e) Neighborhood Label Pair



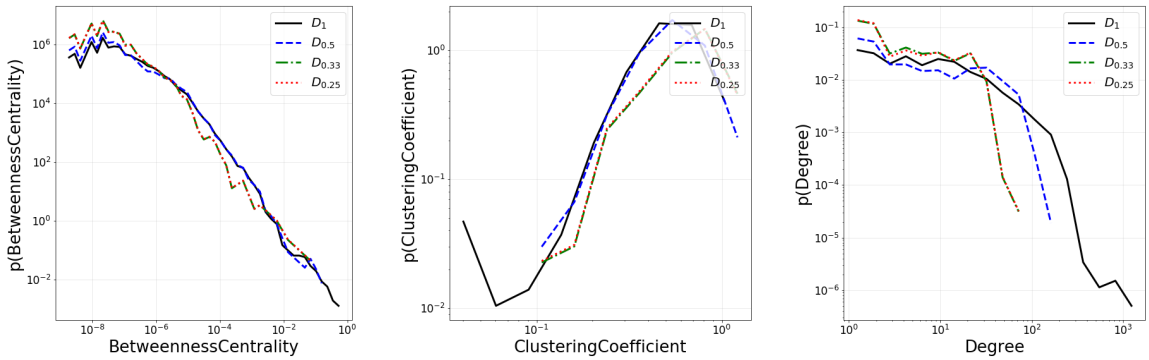
(f) Neighborhood Pair

Figure 4.12: Cumulative distributions of privacy risk for social network data.

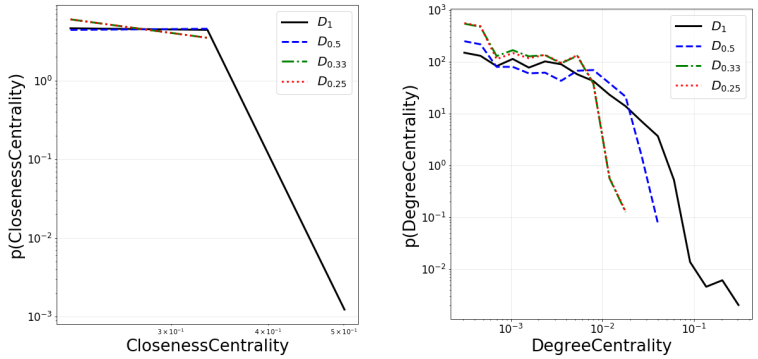
Figure 4.12 shows as privacy risk for the attacks on network data varies significantly. The most interesting results can be seen for the neighborhood label pair attack: with respect to the simple label attack or neighborhood attack, the mixed one leads to an increase of the number of high risk individuals by a great margin. The mutual friend attack is weaker with respect to all the others. Indeed, in each setup of the background knowledge configuration value k, many individuals belong to the privacy risk level $(0.0, 0.1]$. This is not surprising since the Mutual Firend attacks uses the number of mutual friends of one node, which has a pretty even distribution over the entire network.

4.2.11 Social Network Data Quality Experiments

In order to analyse the data quality on our network dataset, we select attacks based on the background knowledge configuration with $k = 3$. In Figures 4.14 and 4.13, for a subset of metrics, we can visually analyze how their distributions change varying the different levels of privacy. We can immediately notice that, for this kind of data, distortion related to individual deletion is less severe. We can also appreciate the stark difference between a more powerful attack (Neighborhood Pair attack) and a much less powerful one (Neighborhood attack). We can quantify the changes using our methodology. A visualization of some of the results is shown in Figure 4.15. It is very clear that the Neighborhood attack has a very low impact on the metrics we evaluated. Also, the scale of the z axis is worth noting: the distance between the derived distributions and the original ones is quite low, indicating a fairly high similarity. A complete report on all the distances for all attacks and for individual measures can be seen in Table 4.7.

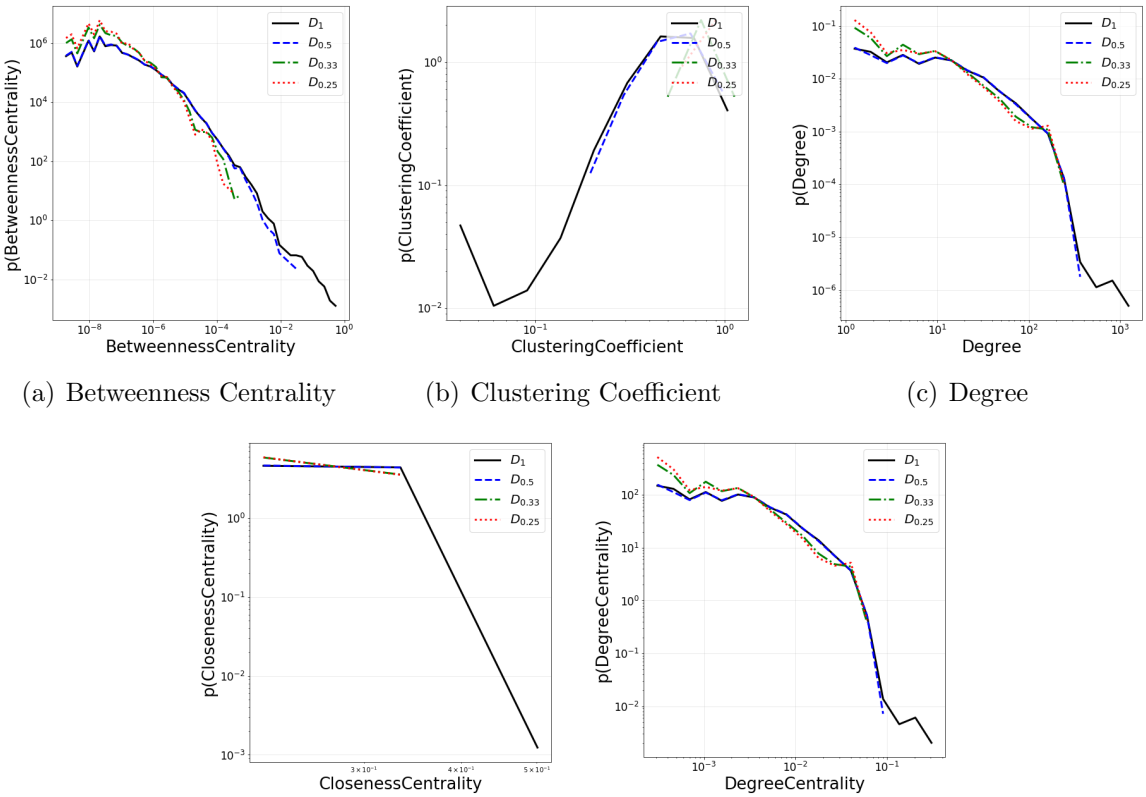


(a) Betweenness Centrality (b) Clustering Coefficient (c) Degree



(d) Closeness Centrality (e) Degree Centrality

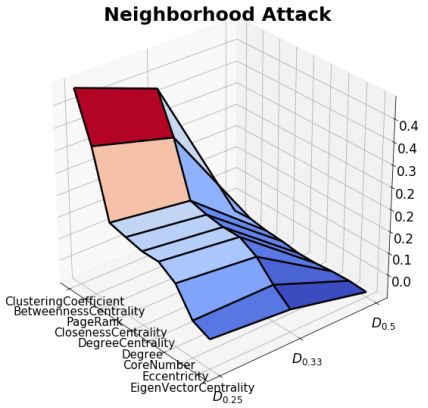
Figure 4.13: Some examples of distributions of network metrics for the Neighborhood Pair attack.



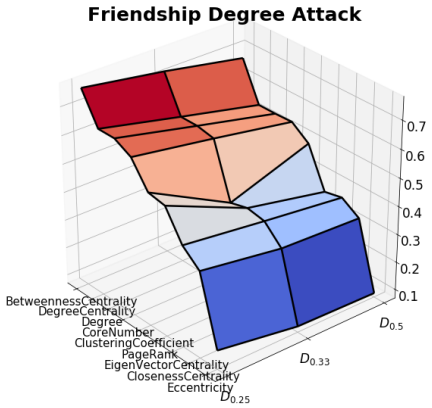
(a) Betweenness Centrality (b) Clustering Coefficient (c) Degree

(d) Closeness Centrality (e) Degree Centrality

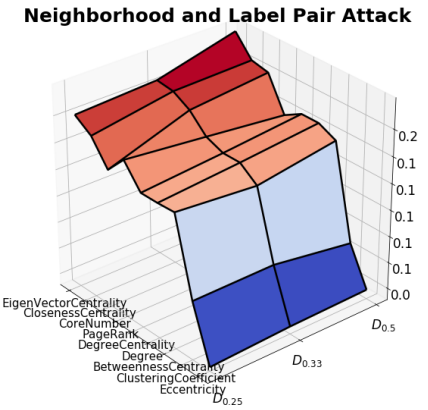
Figure 4.14: Distributions of network metrics for the Neighborhood attack.



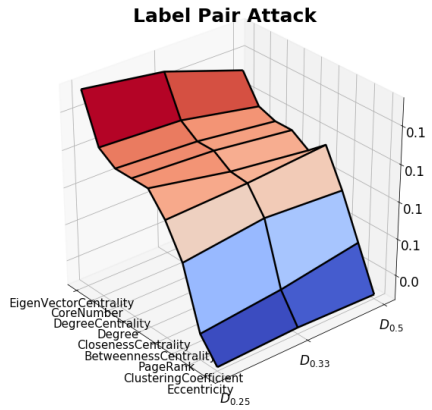
(a) MUC for NeighborhoodAttack



(b) MUC for Friendship Attack k=3



(c) MUC for Neighborhood and Label Pair Attack



(d) MUC for Label Pair Attack

Figure 4.15: Examples of MUC for various attacks with k=2 on social network data considering five different metrics and three levels of risk .

Type of Attack		**Degree**	**Degree Centrality**	**Betweenness Centrality**	**Closeness Centrality**	**Clustering Coefficient**
Neighborhood	$D_{0.5}$	0.005774	0.005774	0.021048	0.011433	0.018403
	$D_{0.33}$	0.144560	0.144560	0.283381	0.150492	0.373095
	$D_{0.25}$	0.166272	0.166272	0.333358	0.175079	0.439989
Label Pair	$D_{0.5}$	0.093174	0.093174	0.095032	0.087124	0.052200
	$D_{0.33}$	0.098663	0.098663	0.091606	0.092628	0.045214
	$D_{0.25}$	0.101080	0.101080	0.089844	0.101058	0.040794
Neigh. Label Pair	$D_{0.5}$	0.143704	0.143704	0.137463	0.164402	0.066747
	$D_{0.33}$	0.137706	0.137706	0.127882	0.162972	0.076225
	$D_{0.25}$	0.132540	0.132540	0.133198	0.152827	0.075803
Friendship Degree	$D_{0.5}$	0.530342	0.530342	0.667854	0.320770	0.487309
	$D_{0.33}$	0.593839	0.593839	0.721575	0.333070	0.388792
	$D_{0.25}$	0.658972	0.658972	0.766883	0.377360	0.540885
Mutual Friend	$D_{0.5}$	0.005669	0.005669	0.004003	0.002495	0.002201
	$D_{0.33}$	0.009708	0.009708	0.006684	0.003848	0.002127
	$D_{0.25}$	0.016338	0.016338	0.010580	0.006465	0.003587
Neighborhood Pair	$D_{0.5}$	0.133107	0.133107	0.063939	0.035011	0.057220
	$D_{0.33}$	0.133107	0.133107	0.063939	0.035011	0.057220
	$D_{0.25}$	0.406072	0.406072	0.331757	0.203764	0.262311

Table 4.7: Numerical values of MUC for social network data. Each cell shows the Kolmogorov-Smirnov distance between the distribution of the metric, computed at that risk level for that particular attack, and the original distribution of that metric. The table only shows a subset of metrics to accommodate for space.

4.2.12 Discussion

As the experimental evidence shows, privacy risk assessment with PRUDEnce can be done in systematic way by exploring all possible types of attack: PRUDEnce gives to a data provider the possibility to select which are the attacks that need to consider for a specific privacy preserving process. For each of the attacks that we defined, the experimental results show that even with relatively short background knowledge configurations, i.e., assuming that the adversary knows few information, privacy risk may be quite high for a large portion of individuals. The different types of data share high distribution of risk, especially for attacks exploiting the knowledge of more than one data dimension (e.g., the Location Time attack for mobility data and the Neighborhood and Label Pair attack for network data). Given how much the risk distribution may vary depending on the kind of attack, exploring the data quality subjected to the deletion of high risk individuals becomes even more important. Overall, the most interesting aspect of our methodology for evaluating data quality is how it can be used to thoroughly explore how each metric behaves varying both the background knowledge and the desired level of risk. The intended usage of our methodology is, for a data provider, that wants to analyze the metrics, requested by a third party, under different attacks and different levels of

risk, to better understand which individuals can be safely deleted without impacting on the quality of the requested metric or, to understand which attacks imply higher distortion to ensure privacy protection. We have seen in the experiments that, depending on the type of the data and on the background knowledge, the effects on the metrics may vary significantly. Evaluating the distance of the distribution of each metric from the original one, a data provider can devise specific protection measures to ensure that certain characteristics in the data are maintained while others are masked or deleted to ensure higher privacy. One possible interesting future development of this methodology would be to broaden the set of distances used to evaluate the changes in the distributions of the metrics, possibly finding distances specifically suited for each separate metric.

Chapter 5

Privacy Risk Prediction

As shows in Chapter 4, analyzing privacy risk may be a daunting task. Since risk can be empirically evaluated only by assuming the actual composition of the background knowledge, it is required to first define the attacks that can be conducted on a certain kind of data, and then we need to systematically simulate all the possible background knowledge instances in order to provide a worst-case scenario evaluation of privacy risk, i.e., the worst possible attack to which any single individual can be subjected to. This may be unfeasible in terms of computational resources or time available. We therefore devise a data mining approach that allow us to predict individual privacy risk based upon the specific metrics of individuals represented in the data. We initially designed this approach specifically for mobility data and we published it in [122].

5.1 Computational Complexity of PRUDEnce

The procedure of privacy risk computation introduced in Section 4.1 has a high computational complexity. We assume that the adversary uses all the information available to her when conducting a re-identification attack on an individual. Since it is unlikely that an adversary knows the complete history of an individual (i.e., all the points or all the friends or all the purchased items), we introduced the concept of background knowledge configuration B_k, which indicates the length of the portion of data k known by the adversary when performing an attack on an individual. The higher the k the more abundant is the personal data known by the adversary about the individual. The maximum possible value of k is len, the length of the data structure of an individual, be it a trajectory or a basket history etc. k-combination have been proven to be in direct relation with computational anonymity: in [143] the authors provide algorithms for achieving a relaxation of k-anonymity, whose performance is closely linked to the number of k-combination considered.

The best k-combination for the adversary is the one leading to the highest probability of re-identification of the individual under attack. However, we do not know such best combination in advance. For this reason, given k, when we simulate an attack we compute all the possible k-combinations an adversary could know. Given a combination of k elements of the data structure representing an individual, we assume that the adversary

uses *all* these k points to conduct the attack. This leads to a high overall computational complexity $\mathcal{O}(\binom{len}{k} \times N)$, since the framework generates $\binom{len}{k}$ background knowledge configuration instances and, for each instance, it executes N matching operations by applying function *matching*. In the extreme case where the adversary knows the complete data of an individual (i.e., she knows all the points of a trajectory, or all the elements of a degree vector) we have $k = len$ and the computational complexity is $\mathcal{O}(N)$. In general, in the range $k \in [1, \frac{len}{2}]$ the computational complexity of the attack simulation increases with k, while for $k \in [\frac{len}{2}, n]$ the computational complexity decreases with k. While all the $\binom{len}{k}$ possible instances must be necessarily considered since, as already stated, we cannot exclude any of them *a priori*, we can reduce the number N of matching operations between a single instance and the data structures in the dataset by eliminating unnecessary comparisons. This kind of optimization depends mostly on the kind of attack, and cannot be generalized. Although the overall worst-case complexity of the attack remains $\binom{len}{k}$, in practice optimization speeds up the execution skipping unnecessary comparisons during the matching between an instance and a data structure. However, as we will show in Section 5.3, in practice the matching optimizations do not eliminate the computational problem and the simulation of the attacks can take up to 2 weeks to compute the privacy risks of individuals in our datasets.

Example 4. *Let us consider the following scenario where an adversary attacks a mobility dataset knowing 5 locations of an individual with a trajectory of length* $len = 50$*. Computing the privacy risk of an individual with respect to the background knowledge configuration* B_5 *requires the generation of the* $\binom{50}{5} = 2,118,760$ *background knowledge instances. In a dataset of* $N = 100,000$ *individuals, each with* $len = 50$*, the overall simulation of the attack would take around 210 billions of matching operations.*

5.2 A Data Mining approach for Privacy Risk Assessment

Given its computational complexity, the procedure for Privacy Risk Computation (Section 4.1) becomes unfeasible as the size of the dataset increases, since it requires enormous time and computational costs. This drawback is even more serious if we consider that the privacy risks must be necessarily re-computed every time the dataset is updated with new data records and for every selection of individuals and specific data dimensions. In order to overcome these problems, we propose a fast and flexible data mining approach. The idea is to train a predictive model to predict the privacy risk of an individual based solely on her individual patterns and metrics that we can extract from data. The predictive model can be either a regression model, if we want to predict the actual value of privacy risk, or a classification model if we want to predict the level of privacy risk. For our context we will focus on classification: our aim is to provide a methodology that would allow a data provider to quickly understand at a glance how much at risk individuals are in the data. The training of the predictive model is made by using a training dataset where every example refers to a single individual and consists of *(i)* a vector of the individual's features and *(ii)* the privacy risk level of the individual. We define a

classification training dataset as a tuple TC = (F, C) where C is the vector of the individuals' privacy risk level (e.g., from level 0.0 indicating no risk to level 1 indicating maximum privacy risk). Given a data type, we define a possible set of features F based on the data specific metrics defined in Chapter 3. We use the repertoire of data specific attacks, introduced in Chapter 4, to assess privacy risk for each kind of data and describe how to construct the classification training dataset in Section 5.2.1. In Section 5.2.2 we describe how a Data Provider can use our approach in practice to determine the privacy risk of individuals in her database. We make our approach parametric with respect to the predictive algorithm: in our experiments we use a Random Forest classifier, but every algorithm available in literature can be used for the predictive tasks. A Random Forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [70]. We chose random forest for its particular balance of performance and intepretability: as an ensemble methods, random forest performs boosts the performance of traditional classification trees, thanks to its randomized structure and bootstrap aggregating [25]. At the same time, random forest allows us to interpret the prediction through the analysis of feature importance, which is one of the techniques used in explainability of machine learning models [7]. By looking at which features are used by the random forest for the prediction of privacy risk for each attack, we can get a basic explanation of the individual behaviors that can lead to higher privacy risk. More over, random forest helps us tackle outliers in our data. Outliers, that is individuals with data far outside the average present in the dataset, can be easily identified with our privacy risk assessment framework. For example, an individual buying a product that is very rarely bought by the collectivity will be easily re-identified. The same applies to individuals that have in their trajectories places very rarely visited by others, or that travel a lot more than other individuals in the data. Outliers have therefore a high privacy risk under our assessment framework and may be difficult to properly classify. However, in our predictive approach we compensate for outliers in the following way: we use a variety of heterogeneous features for the predictive task: for example, for retail data we use both the number of total products and the name of the most frequently bought product, the entropy and the average number of bought products over time. This helps with outlier detection especially because of the predictive model that we chose: random forests. Random forests, and tree-based methods in general, are inherently robust to outliers because of their structure: since the split of the various nodes is based on residual sum of squares, outliers usually do not impact the decision on the split in a significant way (when in reasonable quantity). Moreover, random forests randomly select a subset of the features at each split and use bagging for the overall prediction, thus reducing even further the impact of outliers.

5.2.1 Construction of training dataset

Given an attack a based on a specific background knowledge configuration B_j^a, the classification training dataset TC_j^a can be constructed by the following three-step procedure:

1. Given a dataset D, for every individual u we compute the set of individual features using some data specific metrics (Sections 4.2.1, 3.2.1 and 3.3.1) based on the data

of that individual. Every individual u is hence described by a feature vector $\overline{f}_u$. All the individuals' feature vectors compose the feature matrix $F=(\overline{f}_1, \ldots, \overline{f}_n)$, where n is the number of individuals in D;

2. For every individual we simulate the attack with background knowledge configuration B_j^a on D, in order to compute a privacy risk value for every individual. We obtain a privacy risk vector $R_j^a = (r_1, \ldots, r_n)$.

3. Construct the classification training set TC_j^a by discretizing vector R_j^a on the intervals [0.0], $(0, 0.1]$, $(0.1, 0.2]$, $(0.2, 0.3]$, $(0.3, 0.5]$, $(0.5, 1.0]$. We obtain in this way a privacy risk level vector $C_j^a = (c_1, \ldots, c_n)$. The classification training set is hence $\text{TC}_j^a = (F, C_j^a)$.

Every classification training dataset TC_j^a is used to train a predictive model M_j^a. The predictive model will be used by the Data Provider to immediately estimate the privacy risk value or the privacy risk level of *previously unseen* individuals, whose data were not used in the learning process, with respect to attack a, background knowledge configuration B_j^a and dataset D. We remark that For our particular prediction task, we obtain the ground truth directly through computation, i.e., using the PRUDEnce framework to directly compute the risk. The idea is that, once computed for a particular set of individuals, risk can be used to train the predictive model which can then be used to predict risk for other individuals instead of recomputing it again. We discretize the risk into intervals, with the aim of providing a tool for quick risk estimation, e.g., high risk vs low risk. We chose intervals [0.0], $(0, 0.1]$, $(0.1, 0.2]$, $(0.2, 0.3]$, $(0.3, 0.5]$, $(0.5, 1.0]$ because they make sense for the particular definition of risk in PRUDEnce: indeed, risk is defined as a quantity in the form $1/n$ where n is number of individuals that match a particular background knowledge. Thus, risk takes the values: $1, 0.5, 0.33, 0.25, \ldots$. This is why we chose our intervals. Let us see an example.

Example 5. *Let us consider a mobility dataset of trajectories* $D=\{T_{u_1}, T_{u_2}, T_{u_3}, T_{u_4}, T_{u_5}\}$ *corresponding to five individuals* u_1, u_2, u_3, u_4 *and* u_5*. Given an attack* a*, a background knowledge configuration* B_j^a *and dataset* D*, we construct the classification training set* TC_j^a *as follows:*

1. *For every individual* u_i *we compute the 16 individual mobility measures based on her trajectory* T_{u_i}*. Every individual* u_i *is hence described by a mobility feature vector of length 16* $\overline{m}_{u_i} = (m_1^{(u_i)}, \ldots, m_{16}^{(u_i)})$*. All the mobility feature vectors compose mobility matrix* $F=(\overline{m}_{u_1}, \overline{m}_{u_2}, \overline{m}_{u_3}, \overline{m}_{u_4}, \overline{m}_{u_5})$*;*

2. *We simulate the attack with configuration* B_j^a *on dataset* D *and obtain a vector of five privacy risk values* $R_j^a = (r_{u_1}, r_{u_2}, r_{u_3}, r_{u_4}, r_{u_5})$*, each for every individual;*

3. *Let us suppose that the actual privacy risks resulting from simulation are* $R_j^a=(1.0, 0.5, 1.0, 0.25, 0.03)$*. We discretize the values of the privacy risk vector* R_j^i *on the intervals* [0.0]*,* $[0, 0.1]$*,* $(0.1, 0.2]$*,* $(0.2, 0.3]$*,* $(0.3, 0.5]$*,* $(0.5, 1.0]$*. We hence obtain a privacy risk level vector* $C_j^i = ((0.5, 1.0], (0.3, 0.5], (0.5, 1.0], (0.2, 0.3], [0, 0.1])$ *and the classification training dataset* $TC_j^a = (F, C_j^a)$*.*

5.2.2 Usage of the data mining approach

The Data Provider can use a classifier M_j^a to determine the level of privacy risk with respect to an attack a and background knowledge configuration B_j^a for: *(i) previously unseen* individuals, whose data were *not* used in the learning process; *(ii)* a selection of individuals in the database already used in the learning process. It is worth noting that with existing methods the privacy risk of individuals in scenario *(ii)* must be recomputed by simulating attack a from scratch. In contrast, the usage of classifier M_j^a allows for obtaining the privacy risk of the selected individuals immediately. The computation of the mobility measures and the classification of privacy risk level can be done in polynomial time as a one-off procedure. To clarify this point, let us consider the following scenario. A Data Analyst requests the Data Provider for updated data about a new set of individuals with the purpose of studying some of their specific characteristic. Before releasing the data, however, the Data Provider wants to determine the level of privacy risk of the individuals with respect to some attack attack (a) and several background knowledge configurations B_j^a. The Data Provider uses classifier M_j^a previously trained to obtain the privacy risk level of the individuals. On the basis of privacy risks obtained from M_j^a, the Data Provider can immediately identify risky individuals, i.e., individuals with a high level of privacy risk. She then can decide to either filter out the risky individuals or to select suitable privacy-preserving techniques (e.g., k-anonymity or differential privacy) and transform their data in such a way that their privacy is preserved. In the next sections we present an extensive evaluation of our methodology on the experimental datasets we introduced in Chapter 3.

5.3 Privacy Risk Prediction Experiments

Since our purpose is to provide a tool to immediately discriminate between individuals with low risk and individuals with high risk, we will focus more on the the results of classification experiments. For our models, we use the implementation provided by the scikit-learn package in Python [117]. We evaluate the overall performance of a classifier by two metrics [150]: *(i)* the accuracy of classification ACC $= \frac{|\hat{f}(x_i)=f(x_i)|}{n}$, where $f(x_i)$ is the actual label of individual i, $\hat{f}(x_i)$ is the predicted label, and n is the number of individuals in the training dataset; *(ii)* the weighted average F-measure, defined as $F = \sum_{c \in C} |c| \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$, where TP, FP, FN stand for the numbers of true positives, false positives and false negatives resulting from classification, C is the set of labels and $|c|$ is the support of a label. All the experiments are performed using a k-fold cross validation procedure with k=10. We construct a classification training dataset TC_j^a for every distinct background knowledge configuration B_j^a of the attacks described in Chapter 4. Every classification dataset TC_j^a is used to train a classifier M_j^a using Random Forest [66]. We compare the performance of each classifier M_j^a with the performance of a baseline classifier which generates predictions based solely on the distribution of privacy risk labels in C_j^a. For each classifier we will also show the feature importance of the features we used. We quantify the importance of every feature in a classifier M_j^a by taking its average importance in the decision trees of the resulting random forest. The importance of a

feature in a decision tree is computed as the (normalized) total reduction of classification entropy brought by that feature in the tree [66]. It is important to note that classifying a high risk individual as a low risk individual can be a major issue. For our application the *recall* is important to evaluate the performance of a classifier: a high recall on the highest risk class $(0.5, 1.0]$ indicates that a very low number of high risk individuals are misclassified as low risk individuals. To be usable in practice classifiers need to have a high recall on the highest risk class. We will therefore show how these values behave for our classifiers. Finally, we will also provide an evaluation on the improvement in terms of execution times that we obtain with our approach with respect to the direct computation of privacy risk in PRUDEnce. The execution time of a single classification task is the sum of three subtasks: *(i)* the execution time of training the classifier on the training set; *(ii)* the execution time of using the trained classifier to predict the classes on the test set; *(iii)* the execution time of evaluating the performance of classification (i.e., computing accuracy and F-measure). The prediction approach we present was initially developed for mobility data but, as we will show, it is sufficiently general in its definition that it can be applied also to other kinds of data. We will show how different data produce different results for this methodology and compare our results.

5.3.1 Privacy Risk Prediction for Mobility Data

For all the attacks defined except the Home and Work attack we consider four background knowledge configurations B_k with $k = 2, 3, 4, 5$, where configuration B_k corresponds to an attack where the adversary knows k points of the trajectory of the individual. For the Home and Work attack we have just one possible background knowledge configuration, where the adversary knows the most frequent location and the second most frequent location of an individual. We use the mobility attacks defined in Section 4.2.1 for risk computation. Table 5.1 (columns Florence and Pisa) summarizes the results of classification tasks for both the Florence dataset and the Pisa dataset. In Table 5.1 we observe a significant gain in both accuracy and F-measure of the classifiers over the baseline. For example, in predicting the Probability privacy risk levels the classifier reaches maximum performance values of ACC = 0.95 and F-measure = 0.95 (configuration k=4, Florence), a significant improvement with respect to the baseline model. The Home and Work variable has the weakest relation with the individual mobility features, reaching the lowest performance values. The classification results for Florence and Pisa are comparable, with slightly better performances for the Florence dataset. It is worth noting that, for some attacks such as the Location Time attack, we have very similar performances in terms of both accuracy and F-measure for any k. This is due to the fact that the privacy risk distributions resulting from simulating the attack are similar for any $k \geq 2$. In contrast, for the Location Sequence attack we observe that the distribution of privacy risk for k=2 differs from the distributions of privacy risk for $k \geq 3$ (Section 4.2.3). Since the classifiers are accurate especially for the class of maximum risk $(0.5, 1]$, and since for $k \geq 3$ the number of individuals with maximum privacy risk increases, as a consequence the performance of classifiers improve.

Figure 5.1a-b show a matrix representing the classification error for every label of background knowledge configuration $k = 4$ of the Probability attack, for Florence (a) and

Pisa (b). An element i, j in the matrix indicates the fraction of instances for which the actual label j is classified as label i by the classifier. The diagonal of the matrix, hence, indicates the classifier's recall for every label. We observe that the recall of the highest risk class $(0.5, 1.0]$ is 99% for Florence and 98% for Pisa. In particular we observe that all the misclassifications of the classifiers for the highest risk class are made predicting class (0.3, 0.5], i.e., the second highest class of risk. So there is a zero probability of misclassifying high risk individuals as low risk individuals (i.e., classes [0.0] and (0.0, 0.1]). Similarly, in Figure 5.1c-d, an element i, j in the matrix indicates the fraction of instances for which the predicted label j is actually label i in the dataset. The diagonal matrix indicates in this case the classifier's precision for every label. We observe that the classifier is very precise for the two lowest (risk $\in$ [0.0] and risk $\in (0.0, 0.1]$) and the highest (risk $\in (0.5, 1.0]$) privacy risk labels: both the recall and the precision of these labels are close to 1. Even on the labels where recall and precision are lower, i.e., $(0.1, 0.2]$, $(0.2, 0.3]$, $(0.3, 0.5]$, the classifier is more prone to predict a higher level of risk than a lower level of risk. These conservative choices allow the Data Provider to limit the privacy violation of individuals: it is hence unlikely that a classifier assigns to an individual a privacy risk label that is lower than her actual privacy risk label. We have very similar results across all types of attack.

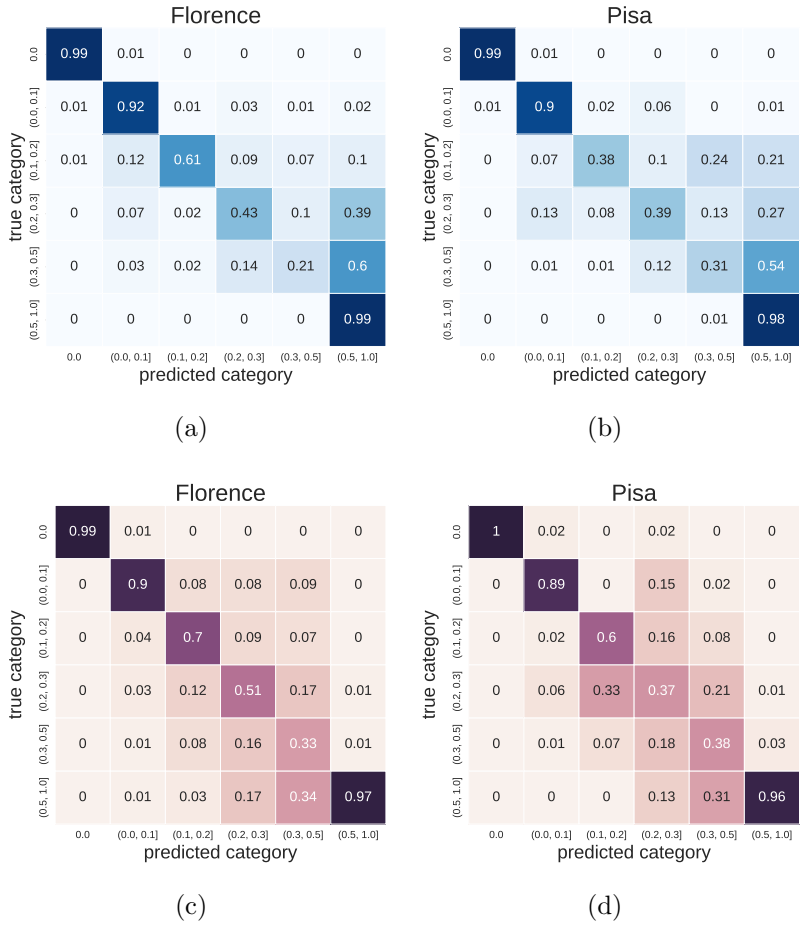


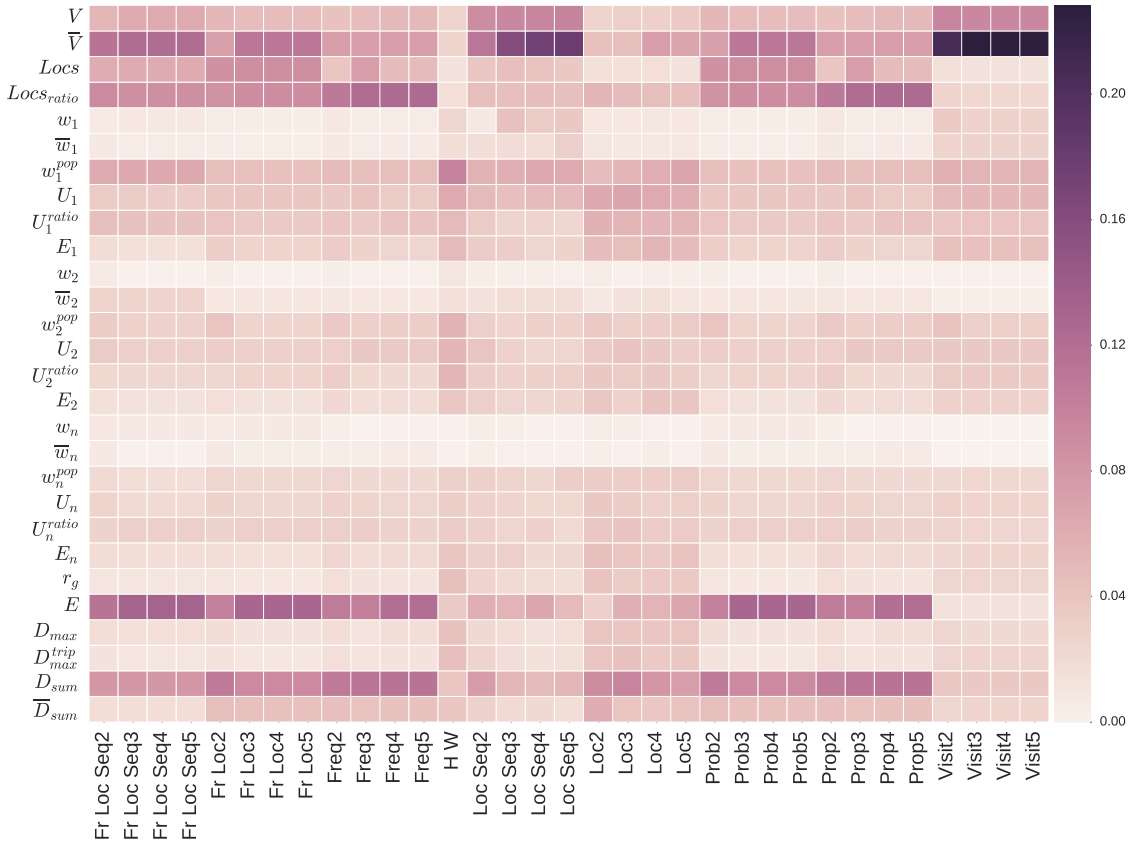
Figure 5.1: Classification error per class for classifier M_4^P (Probability attack P and background knowledge configuration B_4^P), for Florence (a, c) and Pisa (b, d). An element i, j in the matrices (a) and (b), indicates the fraction of instances for which the actual class j is classified as class i. The diagonal of the matrices (a) and (b), hence, indicate the classifier's recall for every class. An element i, j in the matrices (c) and (d) indicates the fraction of instances for which the predicted class j is actually class i in the dataset. The diagonal of matrices (c) and (d) indicate in this case the classifier's precision for every class.

In Table 5.1 (columns FI $\rightarrow$ PI and PI $\rightarrow$ FI) we also show the results of other classification experiments where we train a classifier on the Florence dataset and use it to classify the privacy risk label of vehicles in the Pisa dataset, and vice versa. Even if the two datasets cover disjoint sets of vehicles we observe good predictive performance, comparable to the performance of classifiers where the training set and the test set belong to the same original dataset.

	configuration		Florence		Pisa		FI → PI		PI → FI	
			ACC	**F**	**ACC**	**F**	**ACC**	**F**	**ACC**	**F**
Location Time	locations with timestamps	$k=2$	0.94	0.94	0.93	0.93	0.93	0.92	0.93	0.93
		$k=3$	0.94	0.94	0.93	0.93	0.93	0.93	0.93	0.93
		$k=4$	0.94	0.94	0.93	0.93	0.93	0.93	0.92	0.92
		$k=5$	0.94	0.94	0.92	0.92	0.93	0.93	0.91	0.92
	avg baseline		0.82	0.81	0.81	0.80				
Frequency	locations with frequencies	$k=2$	0.90	0.89	0.83	0.82	0.79	0.79	0.76	0.70
		$k=3$	0.94	0.93	0.89	0.89	0.84	0.86	0.83	0.79
		$k=4$	0.92	0.93	0.89	0.89	0.85	0.86	0.85	0.85
		$k=5$	0.93	0.93	0.89	0.89	0.71	0.73	0.85	0.82
	avg baseline		0.53	0.53	0.41	0.41				
HW	two most frequent locations		0.62	0.59	0.57	0.54	0.57	0.55	0.51	0.49
	avg baseline		0.37	0.37	0.28	0.29				
Locations	locations without sequence	$k=2$	0.93	0.92	0.86	0.86	0.87	0.87	0.85	0.81
		$k=3$	0.95	0.95	0.91	0.91	0.87	0.87	0.87	0.82
		$k=4$	0.95	0.95	0.91	0.91	0.89	0.89	0.89	0.86
		$k=5$	0.95	0.95	0.91	0.91	0.89	0.90	0.87	0.85
	avg baseline		0.57	0.56	0.44	0.44				
Unique Locations	locations without sequence	$k=2$	0.81	0.79	0.71	0.69	0.73	0.74	0.65	0.62
		$k=3$	0.86	0.85	0.8	0.78	0.81	0.81	0.75	0.72
		$k=4$	0.87	0.86	0.81	0.79	0.83	0.83	0.79	0.75
		$k=5$	0.87	0.87	0.81	0.8	0.82	0.83	0.78	0.75
	avg baseline		0.65	0.65	0.56	0.55				
Probability	locations with probability	$k=2$	0.93	0.92	0.86	0.86	0.86	0.85	0.82	0.80
		$k=3$	0.95	0.95	0.92	0.92	0.89	0.89	0.86	0.83
		$k=4$	0.95	0.95	0.91	0.91	0.91	0.90	0.85	0.81
		$k=5$	0.95	0.95	0.92	0.92	0.92	0.92	0.87	0.83
	avg baseline		0.56	0.56	0.45	0.44				
Proportion	locations with proportion	$k=2$	0.90	0.89	0.83	0.81	0.79	0.79	0.79	0.76
		$k=3$	0.94	0.93	0.89	0.89	0.89	0.89	0.83	0.78
		$k=4$	0.93	0.93	0.89	0.89	0.85	0.86	0.84	0.81
		$k=5$	0.93	0.93	0.89	0.89	0.83	0.84	0.83	0.77
	avg baseline		0.54	0.54	0.42	0.40				
Location Sequence	locations with sequence	$k=2$	0.88	0.86	0.79	0.77	0.83	0.82	0.78	0.74
		$k=3$	0.92	0.92	0.87	0.86	0.88	0.88	0.86	0.83
		$k=4$	0.92	0.92	0.88	0.87	0.88	0.88	0.87	0.85
		$k=5$	0.93	0.93	0.88	0.87	0.91	0.90	0.87	0.84
	avg baseline		0.64	0.64	0.55	0.54				

Table 5.1: Results of the classification experiments for the Florence and the Pisa datasets. The classification performance is evaluated by the overall accuracy (ACC) and the weighted F-measure (F) by using a k-fold cross validation with k=10. In columns FI → PI and PI → FI, where FI indicates Florence and PI indicates Pisa, we show the results of classification where we train the classifiers on the first urban area and try to predict the privacy risks of individuals in the second urban area.

Importance of mobility features Figure 5.2 shows a heatmap representing the average importance of every mobility feature to the various classifiers in Florence, where every column corresponds to a classifier and every row corresponds to a mobility feature. First, while classifiers corresponding to different configurations of the attack show similar distributions of importances, classifiers corresponding to configurations of different attacks produce different distributions. For example, in the classifiers corresponding to the four configurations of the Location Time attack the average number of points $\overline{V}$ is, not surprisingly, the most important mobility feature (Figure 5.2). In contrast, in the classifiers corresponding to the four configurations of the Proportion attack, $\overline{V}$ has a low importance while D_{sum}, E and $Locs_{ratio}$ have the highest importance. Table 5.2 shows a ranking of the average importance the mobility features have in the classifiers, for Florence and Pisa. Here we observe that individual measures (e.g., E, V, $\overline{V}$) tend to be the most important ones, while location-based features (e.g., W_i, E_i) tend to be less important.



(a)

Figure 5.2: The distribution of average importance of the mobility features for all the classifiers (Florence dataset).

	Florence		Pisa			Florence		Pisa	
	measure	**impo.**	**measure**	**impo.**		**measure**	**impo.**	**measure**	**impo.**
1	$\overline{V}$	3.66	$Locs_{ratio}$	3.24	15	U_2^{ratio}	0.96	U_2^{ratio}	0.92
2	E	2.92	D_{sum}	3.22	16	U_n	0.88	U_n	0.88
3	D_{sum}	2.75	$\overline{V}$	2.87	17	w_n^{pop}	0.83	r_g	0.87
4	$Locs_{ratio}$	2.51	E	2.62	18	E_n	0.79	E_n	0.79
5	V	1.91	V	1.69	19	E_2	0.74	E_2	0.75
6	w_1^{pop}	1.77	$Locs$	1.66	20	D_{max}	0.68	w_n^{pop}	0.73
7	$Locs$	1.67	w_1^{pop}	1.62	21	D_{max}^{trip}	0.63	D_{max}^{trip}	0.67
8	U_1	1.44	U_1	1.46	22	r_g	0.61	D_{max}	0.58
9	U_1^{ratio}	1.32	U_1^{ratio}	1.40	23	w_1	0.42	$\overline{w}_1$	0.48
10	$\overline{D}_{sum}$	1.19	U_2	1.16	24	$\overline{w}_2$	0.40	w_1	0.44
11	U_2	1.12	U_n^{ratio}	1.09	25	$\overline{w}_1$	0.36	$\overline{w}_2$	0.36
12	w_2^{pop}	1.07	w_2^{pop}	1.07	26	w_n	0.13	w_n	0.15
13	E_1	1.05	E_1	1.06	27	$\overline{w}_n$	0.12	w_2	0.13
14	U_n^{ratio}	0.99	$\overline{D}_{sum}$	0.98	28	w_2	0.10	$\overline{w}_n$	0.13

Table 5.2: The average importance of every mobility feature computed over all the classifiers for Florence and Pisa.

Execution times We show the computational improvement of our approach in terms of execution time by comparing in Table 5.3 the execution times of the attack simulations and the execution times of the classification tasks.[1] The classification tasks have constant execution times of around 10s for Pisa and 22s for Florence. Our approach can compute the risk levels for all the attacks in both Florence and Pisa in 250 seconds (less than 5 minutes), while the attack simulations require more than two weeks of computation.

Attack ($\sum_2^5 k$)	**Florence**		**Pisa**	
	simulation	**classifier**	**simulation**	**classifier**
Home and Work	149s (2.5m)	7s	5s	3s
Frequency	645s (10m)	22s	20s	10s
Proportion	900s (15m)	24s	30s	10s
Unique Locations	997s (10m)	22s	30s	10s
Probability	1,165s (20m)	22s	37s	10s
Location Time	2,274s (38m)	16s	95s (1.5m)	9s
Location Sequence	> 168h (1week)	22s	> 168h (1week)	10s
Location	> 168h (1week)	22s	> 168h (1week)	10s
total	**> 2weeks**	**172s**	**> 2weeks**	**79s**

Table 5.3: Comparison of execution times of attack simulations and classification tasks on Florence and Pisa.

[1]For a given type of attack we report the sum of the execution times of the attacks for configurations $k = 2, 3, 4, 5$.

5.3.2 Privacy Risk Prediction for Retail Data

We used a different setting for the two attacks defined on retails data: for the Intra-Basket attack we considered background knowledge configurations with $k = 2, 3$, while for the Full Basket attack we have only one configuration, where the adversary knows an entire basket of an individual. We perform the risk computation on our retail dataset for the retail attacks defined in Section 4.2.5.

Table 5.4 summarizes the results of classification tasks for the retail dataset. We observe a good gain in both accuracy and F-measure of the classifiers over the baseline for the Intra-Basket attack, while for the Full Basket attack the classifier fails to meet expected performances. This is probably due to the extremely imbalanced data: almost 98% of individuals have risk 1 with this particular attack. It is also interesting to note that our classifier performs better for the Intra-Basket attack for $k = 3$ than for $k = 2$.

Figure 5.3a-b show a matrix representing the classification error for every label of background knowledge configuration for the Full Basket and for $k = 3$ of the Intra-Basket attack respectively. An element i, j in the matrix indicates the fraction of instances for which the actual label j is classified as label i by the classifier. The diagonal of the matrix, hence, indicates the classifier's recall for every label. We observe that the recall of the highest risk class $(0.5, 1.0]$ is 99% like for mobility data. However, recall for other classes is quite low. This indicates that our classifier may be overly conservative, classifying even low risk individuals as high risk. In our context however it is acceptable as it guarantees higher protection for individuals. Similarly, in Figure 5.3c-d, an element i, j in the matrix indicates the fraction of instances for which the predicted label j is actually label i in the dataset. The diagonal matrix indicates in this case the classifier's precision for every label. Again we see that most of the time the classifier cannot properly classify individuals with lower levels of risk, while for high risk individuals in the class $(0.5, 1.0]$ we have high precision. This is due to the highly imbalanced classes, suggesting that tailor made classification methods are needed.

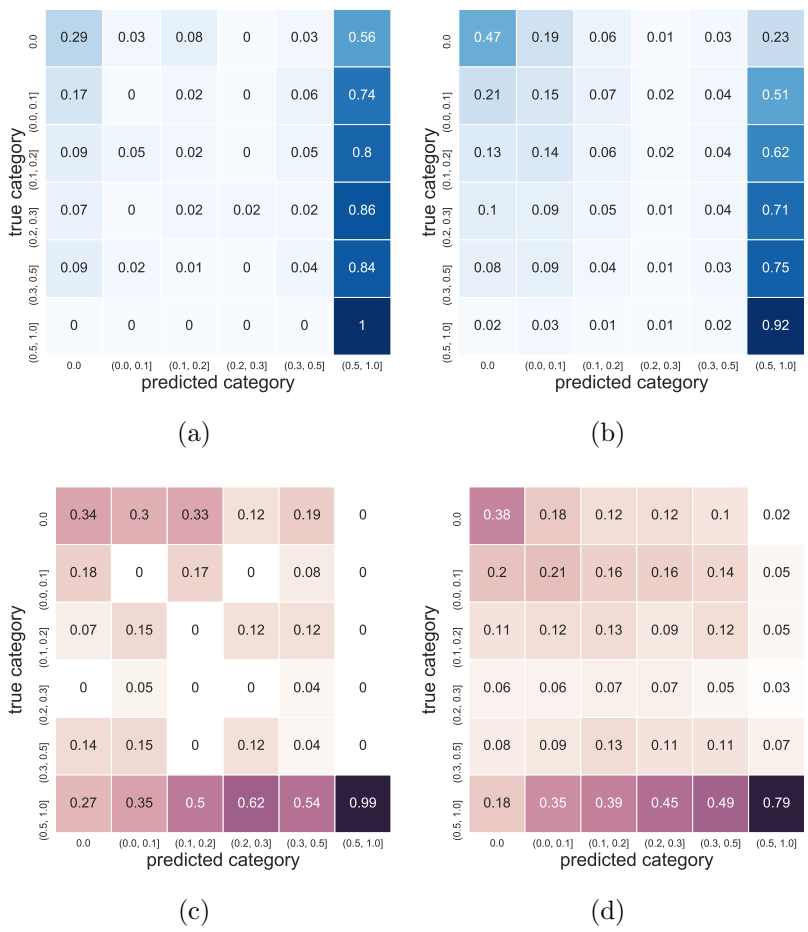


Figure 5.3: Classification error per class for classifier M^{FB} (Full Basket attack FB which has only one configuration) (a, c) and M_2^{IB} (Intra-Basket attack IB with background knowledge configuration B_2^P) (b, d). An element i, j in the matrices (a) and (b), indicates the fraction of instances for which the actual class j is classified as class i. The diagonal of the matrices (a) and (b), hence, indicate the classifier's recall for every class. An element i, j in the matrices (c) and (d) indicates the fraction of instances for which the predicted class j is actually class i in the dataset. The diagonal of matrices (c) and (d) indicate in this case the classifier's precision for every class.

	configuration		Metrics	
			ACC	**F**
Intra Basket	Items within baskets	$k=2$	0.71	0.64
		$k=3$	0.94	0.92
	avg baseline		0.68	0.64
Full Basket	Entire basket history		0.62	0.50
	avg baseline		0.97	0.97

Table 5.4: Results of the classification experiments for the retail dataset. The classification performance is evaluated by the overall accuracy (ACC) and the weighted F-measure (F) by using a k-fold cross validation with k=10.

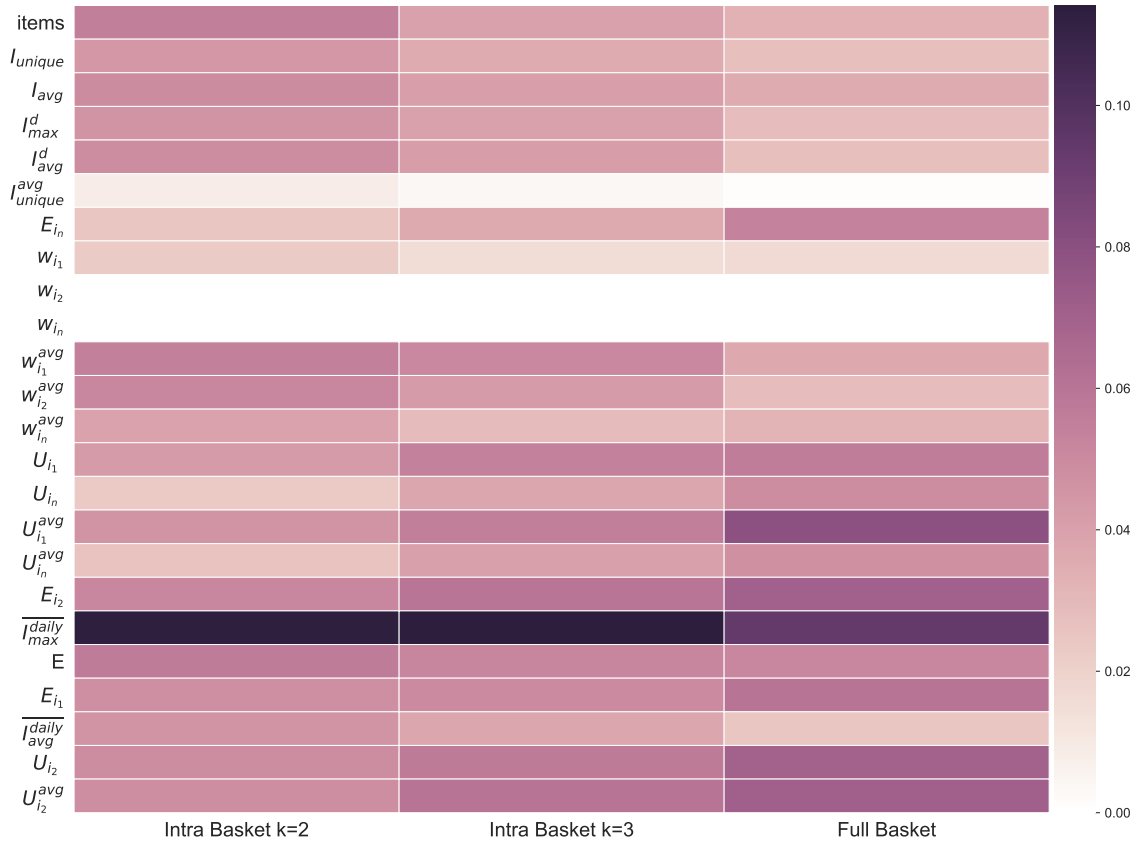
To improve the performance of the classifier for Full basket attack we can change the weight given by the classifier to the minority class, so to drive the prediction towards it. Using a balanced weighting strategy, we give to the minority class, i.e., the class of individuals with risk less than 1, a weight inversely proportional to the class frequency. With this strategy we are able to improve the performance for the Full Basket attack as shown in Table 5.5. This indicates that for high impact attacks that produce high level of risk, the classification method needs to be tailored as to compensate for the imbalance in the data.

	configuration	Metrics	
Full Basket	Entire basket history	0.98	0.97
	avg baseline	0.97	0.97

Table 5.5: Results of the classification experiments for the retail with rebalancing of the classes. The classification performance is evaluated by the overall accuracy (ACC) and the weighted F-measure (F) by using a k-fold cross validation with k=10.

Importance of retail features Figure 5.4 shows a heatmap representing the average importance of every retail feature to our classifiers, where every column corresponds to a classifier and every row corresponds to a retail feature. With respect to feature importance for mobility features we can clearly see that we have, on average, lower values of importance for retail features. We can also clearly see that some of the retail features that we selected are almost completely ignored by the classifier (the frequencies of second most bought item and least bought item w_{i_2} and w_{i_n}), thus suggesting that those features are not correlate with privacy risk. The most important feature for our classifiers is unequivocally $\overline{I_{max}^{daily}}$ which indicates the maximum number of items bought by an individual, averaged over the number of days in the period of observation. The Intra Basket attack, for which this feature is the very important in classifying individuals, is based on the knowledge of a subset of items of a basket. In all the data we have, no individual has more than one basket per day of analysis. Therefore, the average number of products bought may be a very strong indicator of risk, since it is essentially showing a mean of the number of items over the baskets. We see also that for the Full Basket attack, the

average number of individuals who bought the most popular item in the individual basket ($U_{i_j}^{avg}$) is of great importance. Clearly, since when using an entire basket as knowledge the adversary will inevitably know the most bought item of an individual, the average purchasing frequency of that item over the collectivity may influence risk, *hiding* somewhat in the crowd some purchases of the individual. We also see that, although less crucial, Entropy still has a good importance similarly to what we observed for the mobility data classifiers. Table 5.6 shows a ranking of the average importance the retail features have in the classifiers. Here we can clearly see that retail features have a lower importance on average than mobility features for predicting risk.



(a)

Figure 5.4: The distribution of average importance of the retail features for all the classifiers.

	Retail Data			Retail Data	
	measure	**impo.**		**measure**	**impo.**
1	$\overline{I_{max}^{daily}}$	0.32	13	I_{avg}^{d}	0.12
2	E_{i_2}	0.18	14	E_{i_n}	0.12
3	$U_{i_1}^{avg}$	0.18	15	I_{max}^{d}	0.12
4	$U_{i_2}^{avg}$	0.18	16	$U_{i_n}^{avg}$	0.11
5	U_{i_2}	0.18	17	U_{i_n}	0.11
6	E	0.16	18	$\overline{I_{avg}^{daily}}$	0.11
7	E_{i_1}	0.16	19	I_{unique}	0.11
8	U_{i_1}	0.15	20	$w_{i_n}^{avg}$	0.10
9	$w_{i_1}^{avg}$	0.14	21	w_{i_1}	0.05
10	$items$	0.13	22	I_{unique}^{avg}	0.01
11	I_{avg}	0.13	23	w_{i_n}	0.00
12	$w_{i_2}^{avg}$	0.12	24	w_{i_2}	0.00

Table 5.6: The average importance of every retail feature computed over all the classifiers.

Execution times We show the computational improvement of our approach in terms of execution time by comparing in Table 5.7 the execution times of the attack simulations and the execution times of the classification tasks.[2]. The classification tasks have constant execution times of around 430s in total, a little over 7 minutes, while the attack simulations require more than one day of computation. Execution times are in general longer for this kind of data, probably due to the fact that we have a lot of individual data

Attack ($\sum_2^3 k$)	**Retail Data**	
	simulation	**classifier**
Intra-Basket	>24h (1 day)	308s (5 minutes)
Full Basket	>12h	122s (2 minutes)
total	**> 1.5 days**	**430s**

Table 5.7: Comparison of execution times of attack simulations and classification tasks on retail data.

5.3.3 Privacy Risk Prediction for Social Network Data

For all the social network attacks we consider four background knowledge configurations B_k with $k = 1, 2, 3, 4$, where configuration.We use the social network attacks defined in Section 4.2.9 for risk computation. Table 5.8 summarizes the results of classification tasks for the social network dataset.

[2]For Intra-Basket attack we report the sum of the execution times of the attacks for configurations $k = 2, 3$

We observe a good gain in both accuracy and F-measure of the classifiers over the baseline across all kinds of attack and configuration. The only notable exception seems to be the Label Pair attack: our classifier still outperforms the baseline, but both accuracy and F-measure are lower than other kinds of attack.

Figure 5.5a-b show a matrix representing the classification error for every label of background knowledge configuration for the Friendship Degree attack and the Label Pair attack respectively, both with $k = 3$. An element i, j in the matrix indicates the fraction of instances for which the actual label j is classified as label i by the classifier. The diagonal of the matrix, hence, indicates the classifier's recall for every label. We can appreciate a stark difference in the performances: for the Friendship Degree attack the recall has similar characteristics to the mobility attacks, i.e., risk is correctly predicted in a conservative way, favoring high risk classes. For the Label Par attack instead, we see that the classifier performs poorly in terms of recall. Similar results can be seen for Figure 5.5c-d, where an element i, j in the matrix indicates the fraction of instances for which the predicted label j is actually label i in the dataset. The diagonal matrix indicates in this case the classifier's precision for every label. Again we see the difference between the two attacks. Results are, in general, good in terms of accuracy, F-measure, precision and recall for most of the attacks. The exceptions are the Label Pair Attacks and some configurations of the Neighborhood and Label Pair attack.

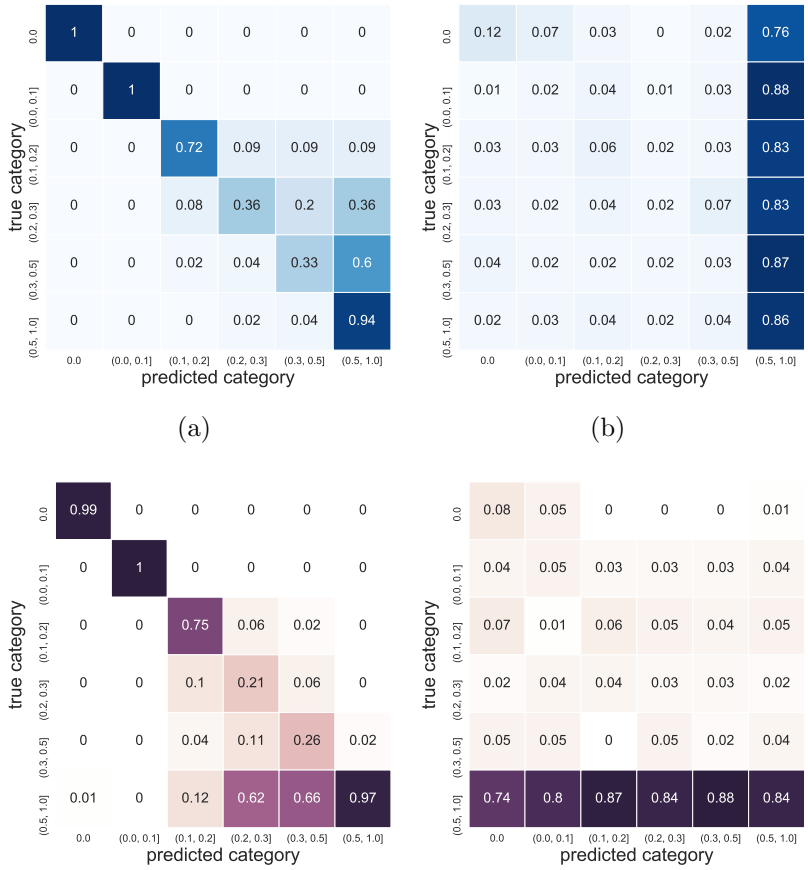


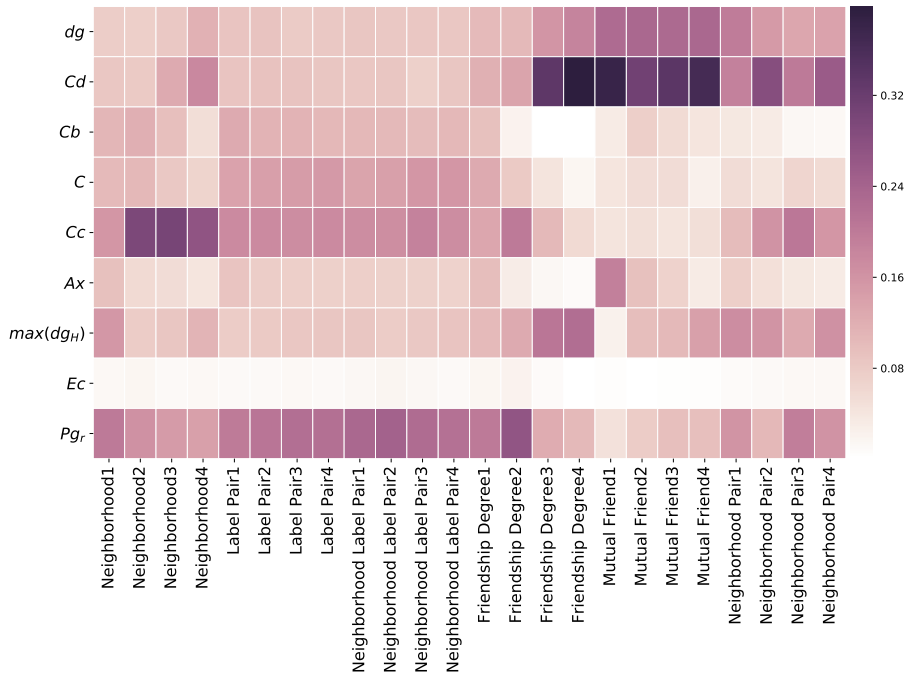
Figure 5.5: Classification error per class for classifier M_3^{FD} (Friendship Degree attack FD and background knowledge configuration B_2^{FD}) (a, c) and M_3^{LP} (Label Pair attack LP and background knowledge configuration B_2^{LP}) (b, d). An element i, j in the matrices (a) and (b), indicates the fraction of instances for which the actual class j is classified as class i. The diagonal of the matrices (a) and (b), hence, indicate the classifier's recall for every class. An element i, j in the matrices (c) and (d) indicates the fraction of instances for which the predicted class j is actually class i in the dataset. The diagonal of matrices (c) and (d) indicate in this case the classifier's precision for every class.

	configuration		Network Data	
			ACC	**F**
Neighborhood	friends of victim	$k=1$	0.74	0.73
		$k=2$	0.80	0.78
		$k=3$	0.79	0.79
		$k=4$	0.88	0.87
	avg baseline		0.48	0.48
Label Pair	label values	$k=1$	0.51	0.42
		$k=2$	0.58	0.48
		$k=3$	0.68	0.56
		$k=4$	0.68	0.58
	avg baseline		0.44	0.44
Neighborhood And Label Pair	label values and friends	$k=1$	0.79	0.70
		$k=2$	0.83	0.76
		$k=3$	0.82	0.74
		$k=4$	0.76	0.66
	avg baseline		0.66	0.66
Friendship Degree	degree of friends	$k=1$	0.61	0.60
		$k=2$	0.95	0.95
		$k=3$	0.99	0.99
		$k=4$	0.99	0.99
	avg baseline		0.72	0.72
Mutual Friend	interconnected friends	$k=1$	0.99	0.99
		$k=2$	0.98	0.98
		$k=3$	0.96	0.96
		$k=4$	0.99	0.99
	avg baseline		0.77	0.79
Neighborhood Pair	pairs of friends	$k=1$	0.88	0.88
		$k=2$	0.89	0.89
		$k=3$	0.93	0.92
		$k=4$	0.93	0.92
	avg baseline		0.53	0.53

Table 5.8: Results of the classification experiments for the social network dataset. The classification performance is evaluated by the overall accuracy (ACC) and the weighted F-measure (F) by using a k-fold cross validation with k=10.

Importance of retail features Figure 5.6 shows a heatmap representing the average importance of every social network feature to our classifiers, where every row corresponds to a retail feature. We see that the Label Pair and Neighborhood and Label Pair attacks present a more evenly distributed importance among the features, while other attacks show one distinct feature that strongly discriminates their classes. We can also clearly see that eccentricity Ec is feature with the lowest importance for classifying risk. One of the most interesting feature is degree centrality Cd: it is important for the Mutual Friend and Friendship Degree attacks. This is not surprising, since degree centrality is a measure of how connected an individual is in the network: highly connected individual tend to have more friends with high degree and more friends mutually connected.Table

5.9 shows a ranking of the average importance the social network features have in the classifiers. We see here that, even though we use fewer features for social networks, they have on average a good feature importance.



(a)

Figure 5.6: The distribution of average importance of the social network features for all the classifiers.

	Network Data	
	measure	**impo.**
1	Cd	4.50
2	Pg_r	4.08
3	Cc	3.75
4	dg	3.14
5	$max(dg\_H)$	2.77
6	C	2.21
7	Cb	1.69
8	Ax	1.59
9	Ec	0.28

Table 5.9: The average importance of every social network feature computed over all the classifiers.

Execution times We show the computational improvement of our approach in terms of execution time by comparing in Table 5.10 the execution times of the attack simulations and the execution times of the classification tasks.[3]. The classification tasks have constant execution times of around 430s int total, a little over 7 minutes, while the attack simulations require more than a day of computation. Execution times are in general longer for this kind of data, probably due to the fact that we have a lot of individual data

Attack ($\sum_1^4 k$)	**Network Data**	
	simulation	**classifier**
Friendship Degree	190s	53s
Label Pair	48min	70s
Mutual Friend	>5h	51s
Neighborhood	>7h	64s
Neighborhood Pair	>15h	56s
Neighborhood and Label Pair	240s	67s
total	**>28h**	**364s**

Table 5.10: Comparison of execution times of attack simulations and classification tasks on social network data.

5.4 Discussion

We proposed a fast and flexible data mining approach for estimating the privacy risk in personal data, which overcomes the computational issues of existing privacy risk assessment frameworks. We validated our approach on three types of real-world GPS data, showing that we can achieve accurate estimations of privacy risks. In particular, the results showed that: (i) the classifiers are accurate especially on the highest privacy risk class, which is important in order to guarantee the safeness of individuals; (ii) the classifiers have a conservative behavior, i.e., misclassified individuals are assigned more likely to classes of higher risk than to classes of lower risk with respect to the actual class of privacy risk; (iii) the importance of the features used to train the classifier can be used for understanding the causes of privacy risk: by looking at which feature discriminates an attack we can have an idea of the individual behavior that causes a certain attack to be successful or not.

[3]For all attacks we report the sum of the execution times of the attacks for configurations $k = 1, 2, 3, 4$

Chapter 6

Modeling Adversarial Behavior Against Mobility Data Privacy

We have seen in Chapter 4 how risk can be computed by mathematically generating the background knowledge that an adversary may use to conduct an attack. As we showed, the simulation of an attack always considers the worst-case scenario: amongst all possible background knowledge instance that an adversary can use to re-identify an individual, PRUDEnce selects the worst. This is a conservative approach commonly used in privacy literature [89]. Here we propose an alternative approach, tailored for mobility data: instead of evaluating risk from the individual point of view, we evaluate what damage a single adversary can produce, in terms of privacy risk, for a mobility dataset, i.e., we compute the average privacy risk of individuals in the data fixing a particular background knowledge of an adversary, based on the adversary movement. We propose a data-driven approach to realistically simulate the behavior of a malicious adversary and evaluate the privacy risk on mobility data from the perspective of an adversarial attack. First of all, we assume that the malicious adversary collects information about the attacked individuals during their movements on territory and following the natural spatio-temporal constraints of human mobility [54, 19]. Then, we present three possible alternatives: the adversary is one of the real individuals in the dataset (real adversary); the adversary is a synthetic individual that moves realistically (synthetic adversary); the adversary moves peculiarly in such a way to produce the most damage to the privacy of individuals in the dataset (simulated adversary). We implement the third alternative by designing a Simulated Privacy Annealing algorithm (SPA), based on an optimization meta-heuristic that generates movements of the adversary that maximize the average privacy risk of the individuals in the dataset.

6.1 Trajectory Modeling: variation

As already stated in Section 3.1, trajectories can be aggregated in various data structure that simplify the information that they present. While some structure, like for example the Frequency Vector, completely discard the temporal information about the points in a trajectory, other approaches can be selected in order to simplify the temporal information.

We briefly recall definition 1 of a trajectory:

Definition 37. ***Trajectory.*** *The trajectory T_u of an individual u is a temporally ordered sequence of tuples $T_u = \langle (x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n) \rangle$, where x_i and y_i are the coordinates of a geographic location and t_i is the corresponding timestamp, with $t_i < t_j$ if $i < j$ $\forall i, j \leq n$, with $n = |T_u|$.*

The period of observation of a trajectory can be discretized into time slots of a fixed length, e.g., one hour. Given a timestamp, we can map it onto a corresponding discrete time slot, for example by rounding the timestamp to the nearest hour.

Definition 38. ***Time slot.*** *Given a certain precision p, the time slot ts_i corresponding to timestamp t_i is obtained by rounding t to precision p. We denote with $Ts_{set} = \{ts_1, \ldots, ts_n\}$ the set of all different time slots in a dataset D.*

For example, timestamp 12/10/2010-23:39:46 is assigned to time slot 12/10/2010-24:00:00 if rounding to the nearest hour, or it is assigned to time slot 12/10/2010-23:30:00 if rounding to the nearest half-hour. Note that since two different timestamps t_i and t_j belonging to the same trajectory T_u may be mapped to the same time slot ts, two different locations in the trajectory, $l_i = (x_i, y_i)$ and $l_j = (x_j, y_j)$ may be associated with the same time slot ts. In such a case, typically the location with longest staying period in the time slot is selected as the location associated with that time slot [113]. When using timeslots, the discretization allows us to represent a mobility dataset D as a matrix:

Definition 39. ***Mobility Dataset Matrix.*** *A mobility dataset matrix M is a three-dimensional binary matrix $|L_{set}| \times |Ts_{set}| \times |U_{set}|$ where each element m_{ijz} is 1 if individual z was at location i during timeslot j, 0 otherwise.*

A mobility dataset matrix allows us to better visualize which individuals stayed roughly in the same place, at the same time. In the light of this definition we need to slightly modify our definitions of background knowledge and privacy risk of Section 4.1:

Definition 40. ***Background knowledge.*** *A background knowledge B represents the set of spatio-temporal points known by the malicious adversary about a set of individuals. Formally, we represent it as a $|L_{set}| \times |Ts_{set}|$ matrix where $bk_{ij} = 1$ if the adversary knows that at least one individual was at the location i during the timeslot j, and $bk_{ij} = 0$ otherwise.*

In other words, a background knowledge BK can be considered as a trajectory.

Definition 41. ***Instance of background knowledge.*** *An instance b_u of background knowledge is a specific set of spatio-temporal points known by the adversary about an individual u. Formally, we can represent it as a mobility matrix B where $\forall (i, j) b_{iju} = 1$ if the adversary knows that the specific individual u was at the location i during the timeslot j.*

We indicate with b_u the matrix derived by BK fixing u.

As stated in Chapter 4, a re-identification attack can be expressed mathematically as a matching function. In the following we will use a matching function $matching(T_w, b_u)$, which indicates whether or not a trajectory $T_w \in D$ matches the instance of background knowledge b_u We assume that, in this attack, the adversary uses both the spatial and temporal components of each point, similarly to the Location Time attack in definition 4.2.1.

$$matching(T_w, b_u) = \begin{cases} true & \forall (i,j) \in b_u, \exists (i,j) \in T_w | \\ & m_{ijw} = b_{iju} = 1 \\ false & \text{otherwise} \end{cases}$$

The matching function returns *true* if the trajectory T_w contains all the points in the background knowledge instance b_u, and *false* otherwise.

Definition 42. ***Privacy Risk.*** *The privacy risk of an individual is measured as the probability to re-identify them given a background knowledge instance b_u. We can apply the matching function to the whole dataset M and count the matching records: $F_{match}(M, b_u) = T_w \in M | matching(T_w, b_u) = True\}$. The probability of re-identification of an individual u in dataset D is defined as*

$$Risk(u, b_u, M) = PR_M(T_w = u | b_u) = \frac{1}{|F_{match}(M, b_u)|}$$

that is the probability to associate a trajectory $T_w \in M$ to an individual u, given instance b_u.

Note that, if for every $(i,j) \in b_u$, an individual $z \neq u$ has $m_{ijz} = 1$ in the dataset M, then that individual shares all the points of u in the background knowledge instance of the adversary.

Given a mobility dataset and the *privacy risk* of each individual in the data, we can define the average risk produced by an adversary as follows.

Definition 43. ***Average Adversary Risk.*** *Given the set of individuals U_{set} in the mobility dataset M, and the risk $Risk(u, b_u, M)$ posed by an adversary to each individual u using the background knowledge instance b_u, the Average Adversary Risk (AAR) is the average risk produced by the adversary: $AAR(u, b_u, M) = \frac{\sum_{u \in U_{set}} Risk(u, b_u, M)}{|U_{set}|}$.*

6.2 Problem Statement

In the literature, risk assessment methodologies aim at evaluating the privacy risk of each individual in the dataset simulating attacks that try to maximize the individual privacy risk. These methodologies assume that: (i) The malicious adversary gathers an arbitrary quantity of information, called background knowledge, about an individual they want to attack; (ii) the malicious adversary uses the background knowledge to re-identify the

attacked individual in an anonymized dataset. In the case of human mobility data, re-identification means that the malicious adversary can reconstruct the entire trajectory T_u of the attacked individual. Typically, existing privacy risk assessment frameworks (e.g., [128]) generate all the possible background knowledge that a malicious adversary may gather about an individual. For each background knowledge, they compute a re-identification probability. Finally, they define the re-identification risk for that individual as the maximum re-identification probability.

We claim that the existing frameworks do not model the process of gathering the background knowledge realistically because, for any individual, they derive the background knowledge that maximizes their risk from the available dataset. This approach is the same as considering an attacker tailored for every single individual in the data. Our claim relies on the fact that an adversary can *gather* background knowledge about a moving individual by knowing where she is at which time; this implies a *co-location* between them. Thus, the gathering of the background knowledge needs some real movements by the malicious adversary, which implies that the spatio-temporal constraints of human mobility must be taken into account during the process of background knowledge construction,e.g., the adversary cannot stay at two different locations in the same time and, cannot move at an unreasonable speed and cannot cover much distance in a single timeslot.

In this paper, we explore possible realistic ways to model the acquisition of the background knowledge by an adversary, taking into account the spatio-temporal constraints of human mobility. The main idea is to define an approach to privacy risk assessment based on an adversary that realistically gathers a background knowledge while maximizing the privacy risk of the individuals in the data.

We model the behavior of a malicious adversary as an *adversary trajectory*. We hence assume that a malicious adversary is an object that moves on the same geographic area and during the same period as the attacked individuals. While moving, the malicious adversary gathers information about the individuals they co-locate with. The malicious adversary uses the gathered background knowledge to re-identify those individuals in the mobility dataset. The adversary trajectory can refer to movements by the malicious adversary itself, or it can refer to movements by a mobile camera, such as a drone with a programmed movement that surveils an area for a specified period. Modeling the behavior of a malicious adversary as an adversary trajectory is an approach that completely departs from the literature. In traditional risk assessment methodologies, the background knowledge is built abstractly, i.e., generated by looking at the data of any single individual. In our framework, the adversary's behavior is confined within realistic spatio-temporal constraints (e.g., an adversary cannot be in two different places at the same time). The underlying assumption of our methodology is that the adversary operates on his own. In a real-world scenario, adversary may collaborate and share information. Our contribution focuses mainly on providing a different way of modeling the gathering process of the background knowledge of an adversary. We shift the perspective from a mathematically generated (or sampled) background knowledge to a background knowledge that may be acquired within the bounds of a realistic effort from the adversary. So we focused our attention on a methodology that was effective most of all in generating a trajectory that could maximize risk for the individuals in the data. Simulating multiple, collaborating adversaries is a problem that involves a number of different considerations to be made:

adversaries may cooperate in covering different regions of the map, or may collaborate in moving during different time-frames alternatively, or could use some stationary observation point to enhance their knowledge. All these scenarios are worth considering and can be very interesting future developments for our research. To formalize how the adversary gathers information through the trajectory, we use the concept of *colocation*:

Definition 44. ***Colocation*** *Let (x, y, t) and (x', y', t') be two points of two trajectories T_{u_1} and T_{u_2} respectively. The two points are considered a* colocation *if $(x = x' \wedge y = y' \wedge t = t')$. We denote by C_{u_1,u_2} the set of all colocations between trajectories T_{u_1} and T_{u_2}.*

Intuitively, a colocation indicates that whenever two trajectories intersect in a specific location during the same time slot, two individuals are at the same place at the same time.

Whenever the adversary trajectory colocates with the trajectory of an individual u, the adversary's background knowledge instance b_u expands, including the points and the time slot of the colocation. In other words, given that adversary trajectory BK and the individual trajectory T_u, the background knowledge instance b_u can be computed as $b_u = BK \cap T_u$.

Based on acquired background knowledge BK, we then simulate a re-identification attack in which the malicious adversary tries to match the points gathered about any individual in the mobility dataset.

To clarify the process of construction of the background knowledge, let us consider the following toy example, in which letters and integers substitute the geographic coordinates and time slots:

Example 6. *Let us consider a set of individuals U_{set}=$\{u_1, u_2, u_3, u_4\}$ and the corresponding mobility dataset D:*

$D = \{$
$T_{u_1} = \langle (A,1), (C,2), (A,3), (G,4) \rangle$
$T_{u_2} = \langle (G,1), (C,2), (A,3), (D,4) \rangle$
$T_{u_3} = \langle (C,1), (G,2), (D,3) \rangle$
$T_{u_4} = \langle (C,1), (G,2), (A,3) \rangle$
$\}$

Let us assume that the adversary trajectory is $T_a = \langle (A,1), (G,2), (C,3), (D,4) \rangle$. The adversary trajectory colocates with individual u_1 on $C_{a,u_1} = \{(A,1)\}$, with individual u_2 on $C_{a,u_2} = \{(D,4)\}$, with individual u_3 on $C_{a,u_3} = \{(G,2)\}$ and in with individual u_4 in point $C_{a,u_4} = \{(G,2)\}$. Therefore, the background knowledge of the adversary a is: $B = (u_1, \{(A,1)\}), (u_2, \{(D,4)\}), (u_3, \{(G,2)\}), (u_4, \{(G,2)\})$.

Based on the background knowledge B, we evaluate the privacy risk produced by a using a matching function and counting, for each individual, how many other individuals match the points in B (see Section 6.1). For example, for individual u_1 point $(A,1)$ is unique, generating a privacy risk of 1. For individual u_2, point $(D,4)$ is unique too, generating a privacy risk of 1. For individual u_3, since point $(G,2)$ is present in T_{u_4} too, the privacy

risk is equal to $\frac{1}{2}$*. Similarly, for individual* u_4*, since point* $(G, 2)$ *is also present in* T_{u_3}*, the privacy risk is equal to* $\frac{1}{2}$*.*

6.3 Construction of the Adversary Trajectory

We can construct an adversary trajectory in several ways. In this paper, we consider three possibilities: using the trajectory of a real individual, generating a realistic synthetic trajectory, or constructing a principled adversary trajectory.

6.3.1 Real Adversary Trajectory

The most straightforward approach to construct an adversary trajectory is assuming that the malicious adversary is one of the individuals represented in the mobility dataset. In this scenario, the adversary trajectory is a real individual's trajectory, that we call *Real Adversary Trajectory*. The privacy risk assessment based on this model identifies in the dataset M the adversary trajectory leading to the maximum privacy risk for individuals represented in M. To this end, for each real individual in the dataset M we use the following approach: *(i)* we consider their trajectory as a background knowledge of a malicious adversary; *(ii)* we compute the privacy risk of each individual in M against that adversary; *(iii)* we compute the privacy risk for D as average over the individual privacy risks, i.e., AAR (Definition 43). Finally, we return the privacy risk evaluation corresponding to the real adversary trajectory leading to the highest AAR.

The individual privacy risk computation at step *(ii)* works as follows. Consider a candidate real adversary trajectory a with background knowledge BK and an individual u in M. First, the approach constructs the adversary's background knowledge instance b_u, composed of the colocations between BK and the trajectory of the individual u; and then, it computes the privacy risk of u applying the $Risk(u, b_u, M)$ function (Definition 42).

6.3.2 Synthetic Adversary Trajectory

An alternative approach is to generate the adversary trajectory using generative algorithms, i.e., algorithms that generate synthetic trajectories that are realistic in reproducing the fundamental patterns of human mobility [75, 113]. We call *Synthetic Adversary Trajectory* an adversary trajectory generated in this way. In this scenario, the privacy risk assessment process generates a candidate set of adversary trajectories using a generative algorithm. This algorithm generates a population of synthetic agents moving in the same geographic area and period as the individuals in the mobility dataset. Then, the privacy risk assessment process identifies in the synthetic dataset the adversary trajectory leading to the maximum privacy risk for individuals in M. To this end, for each individual in the synthetic dataset, we use the following approach: *(i)* we consider their trajectory as a background knowledge of a malicious adversary; *(ii)* we compute the privacy risk of each individual in M; *(iii)* we compute the privacy risk for M as average over the individual privacy risks, i.e., AAR (Definition 43). Finally, we return the privacy risk evaluation

corresponding to the synthetic adversary trajectory leading to the highest average privacy risk. The individual privacy risk computation at step *(ii)* works as in the previous scenario.

6.3.3 Simulated Adversary Trajectory

The previous two approaches model the adversary as an individual whose movement is not focused on the maximization of the privacy risk of the other individuals. It represents a mobility behavior typical for common drivers.

An interesting research question is how to simulate the trajectory of an adversary that moves over the geographic area with the only goal to maximize the attack success against the set of individuals represented in the mobility dataset. Technically speaking, this is an optimization problem with a search space of *exponential* size. To clarify this point, let us assume that each trajectory consists of a number $|Ts_{set}|$ of points, one point per time slot. For each point, the number of possible locations is the set $|L_{set}|$ of locations on the geographic area of reference. Assuming that the adversary moves fast enough to reach every point of the geographical area (a reasonable assumption for small to medium-size urban areas), the number of all possible adversary trajectories is $|L_{set}|^{|Ts_{set}|}$. As a real-world example, let us consider a medium/small size city like Pisa (Italy), and let us assume that it is split into 600 geographical square cells. If the period of observation is one month, we have 720 time slots, resulting in $600^{720} \approx 1.85737791 \times 10^{2000}$ distinct possible trajectories. For such an ample search space, a brute force approach computing all possible adversary trajectories is computationally unfeasible.

We overcome this computational problem by proposing an algorithm called Simulated Privacy Annealing (SPA). Simulated Privacy Annealing is a method based on simulated annealing, a metaheuristic used for the approximation of global optimum in optimization problems. It is used for problems with very large search spaces. Simulated annealing is an adaptation of the Metropolis Hastings algorithm [92], which is a Monte Carlo algorithm used for the generation of sample states of a thermodynamic systems, such as, for example, [77]. Simulated Annealing has been applied to problems related to human mobility before, for example in [13] where the algorithm is used to tackle the problem of traffic jams by dynamically calculating optimal traffic routes. Simulated annealing requires several parameters to function properly, like for example the cooling schedule. Such parameters are largely application specific. However, general guidelines exist to guide in the selection process such as [78] for the cooling schedule, or [23] which gives a general procedure to compute the initial temperature of the simulated annealing.

Intuitively, simulated annealing starts from a solution to the problem and then explores the search space by randomly modifying the solution at each iteration. A "*temperature*" parameter controls the exploration of the solutions. Initially, the temperature is high, and the algorithm considers even solutions that do not improve on the objective function. At every successive iteration, the temperature lowers, and the algorithm is less likely to explore less optimal solutions. This exploration mechanism allows simulated annealing to avoid local minimums and to converge to near optimality, given that it explores enough solutions [94].

Algorithm 1: Simulated Annealing

input : Initial temperature $Temp_{init}$, initial solution S_0
output: Final state S

```
1  Temp ← Temp_init;
2  S ← S_0;
3  S_best ← S_0;
4  while stopping_criteria() is false do
5      Temp ← cooling_schedule(Temp);
6      S_new ← neighbor(S);
7      if P(E(S), E(S_new), Temp) ≥ random(0, 1) then
8          S ← S_new;
9      if E(S) > E(S_best) then
10         S_best ← S;
11 return S_best
```

Algorithm 1 shows the pseudocode of the simulated annealing metaheuristic. It starts with an initial solution S and an initial temperature $Temp_{init}$. The algorithm then iterates until it meets a stopping criterion (line 3 in Algorithm 1). At each iteration, the algorithm decreases the temperature according to a cooling schedule (line 4). In line 5, the algorithm generates a neighboring solution S_{new} by modifying the previous solution S. Then, the algorithm computes $E(S)$ and $E(S_{new})$, i.e., the value of the function to optimize for both the previous solution S and the neighboring solution S_{new}, respectively. $E(S)$ and $E(S_{new})$ are used alongside the current temperature $Temp$ to determine whether or not S_{new} can be accepted as the current solution. This task is done through the acceptance function $P(E(S), E(S_{new}), Temp)$, defined as:

$$P(E(S), E(S_{new}), Temp) = e^{\left(-\frac{E(S_{new})-E(S)}{Temp}\right)}.$$

If the value of the acceptance function is greater than a number generated uniformly at random in the range $[0, 1]$, the neighboring solution S_{new} becomes the new solution S; otherwise the current solution S remains unchanged. Intuitively, the acceptance function checks whether the neighboring solution S_{new} provides a significant improvement in the objective function: the more the neighboring solution improves the current one, the more likely it is to be accepted as the new solution.

We adapt simulated annealing to our problem by defining what a solution S and the objective function $E(S)$ are. Moreover, we need to implement the internal functions in Algorithm 1, i.e., *stopping_criteria*, *cooling_schedule* and *neighbor*.

For our problem, the solution S, S_0, S_{new} and S_{best} represent an adversary trajectory, while the objective function $E(S)$ must be a function that quantifies the privacy risk generated by the adversary trajectory. We use the AAR metric defined in Definition 43 as an objective function.

Simulated annealing is a minimization metaheuristic. So, to correctly model our problem, $E(S)$ will actually be $1 - AAR$ since mean risk has an upper bound of 1. We denote

with $F_{AAR}(T, M)$ the function that, given the adversary trajectory T and a Mobility Matrix M computes $1 - AAR$ over the individuals in M. So our objective function becomes simply: $F_{AAR}(T, M)$. We generate the initial adversary trajectory S by creating a random stationary trajectory: we select one location at random from the geographic area of reference and make the individual stay in that location for all the time slots. The generation of the neighboring adversary trajectory S_{new} (i.e., the implementation of the *neighbor* function) is done by selecting at random one time slot in the current adversary trajectory, and by substituting the associated location with a new location chosen at random from the set of all locations that are within a certain distance radius from the point changed. This distance parameter is needed to guarantee that the sequence of locations composing the adversary trajectory is realistic, in the sense that the adversary cannot move to seemingly unreachable locations in the span of a single time slot. To implement the *cooling_schedule* function we use the exponential cooling scheme [78]: the temperature at step $k+1$ is equal to the temperature at the previous step multiplied by a constant α between 0 and 1: $Temp_{k+1} = \alpha Temp_k$. This cooling schedule, though simple, has been proved to be effective and time efficient [127]. While in the literature the value of α is generally set somewhere between 0.95 and 0.99, in our experiments described in Section 6.4 we explore a wider range of values.

The initial temperature is usually selected in a way that the initial acceptance probability is close to a certain initial value, traditionally 80%. Ben-Ameur et al. in [24] propose a simple procedure to calculate the initial temperature. For our purposes, having a very large space of solutions we decided to select an initial temperature such that the initial acceptance probability would be 90%. This is done by simply running the annealing procedure for a small number of iterations, adjusting the temperature in the process.

Regarding the stopping criteria, two typical solutions are adopted in the literature: either simulated annealing is run on a fixed number of steps or the algorithm stops when no significant improvements are made to the solutions for a certain number of steps. We use instead the following approach: we run the algorithms at intervals of fixed number of steps. We choose to compute this number from the actual size of the area we are simulating on, i.e., as a fraction of the number of possible locations times the number of time slots. After running the algorithm for this number of steps, we evaluate the changes made to the objective function. If new solutions are accepted, temperature is still high. Moreover, if new "best solutions" are found, the function is still improving. In these two cases, we keep on running the algorithm, for the same number of steps. Instead, if no new solutions are accepted and the value of the objective function is not improving, the algorithm has sufficiently explored the space of solutions. In such a way, we make sure that every check for the stopping criteria is done after a substantial number of steps and that the possible solutions are explored thoroughly.

In summary, our Simulated Privacy Annealing works as follows:

1. **Set initial parameters**: we set the initial temperature and the initial solution.

2. **Generate a neighboring solution**: we generate a neighboring solution by changing one of the locations in the trajectory with another one at a distance no greater than a fixed limit.

3. **Evaluate current and neighboring solution**: we compute the colocations and AAR.

4. **Acceptance probability**: we either accept or reject the neighboring solution based both on the evaluation and on current temperature.

5. **Lower the temperature**: we lower the temperature according to our cooling schedule.

6. **Check for stoppage**: if a certain number of steps have been completed, check if states have been accepted or if sensible improvement has been done to the objective function.

We remark that the resulting trajectory is not chosen among the available real or synthetic trajectories in the data, but is generated modifying a random trajectory in such a way to maximize the average privacy risk for the individuals in the real data. The function that expresses privacy risk is itself dependent on the actual movement in the various cells in the time-frame of analysis. Thus, the trajectory generated does not belong to the original data but still belongs to the same geographical area and time-frame of the real trajectories. Privacy risk for the single individual, however, may depend not so much on the popularity of a particular trajectory, but rather on how the combination of points traversed by the adversary match with the trajectories of the real individuals.

Algorithm 2 shows the psuedocode of Simulated Privacy Annealing. We show in Algorithms 3, 4 and 5 how we implemented the cooling schedule, stopping criteria, and neighboring function respectively.

Algorithm 2: Simulated Privacy Annealing

input : Initial temperature $Temp_{init}$, initial adversary trajectory T_0, mobility matrix M, cooling rate α, distance limit lm

output: Final state T_{best}

$Temp \leftarrow Temp_{init}$;
$T \leftarrow T_0$;
$T_{best} \leftarrow T_0$;
$steps \leftarrow 0$;
while $stopping\_criteria(T, T_{best}, steps, M)$ is $false$ **do**
    $Temp \leftarrow cooling\_schedule(Temp, \alpha)$;
    $T_{new} \leftarrow neighbor(T, lm)$;
    if $P(AAR(T, M), AAR(T_{new}, M), Temp) \geq random(0, 1)$ **then**
        $T \leftarrow T_{new}$;
    if $F_{AAR}(T, M) > F_{AAR}(T_{best}, M)$ **then**
        $T_{best} \leftarrow T$;
    $steps \leftarrow steps + 1$;
return T_{best}

Algorithm 3: stopping_criteria

input : Current adversary trajectory T, best adversary trajectory T_{best}, number of steps *steps*, mobility matrix M
output: Stopping value *bool*

```
bool ← False;
constant ← 10;
steps_n ← |M|/constant; if steps % steps_n == 0 then
    if (T changed ∨ T_best changed)) then
        bool ← True;
return bool
```

Algorithm 4: cooling_schedule

input : Current temperature $Temp$
output: New temperature $Temp_{new}$

```
Temp_new ← αTemp;
return Temp_new
```

Algorithm 5: neighbor

input : Current adversary trajectory T, distance limit lm
output: Neighboring trajectory T_{new}

```
point ← random_choice(T);
new_point ← neighbor_point(lm);
T_new ← (T);
T_new(point) ← new_point;
return T_new
```

6.4 Experiments

6.4.1 Dataset of real trajectories

We use a slight variation our mobility dataset introduced in Section 3.4.1 splitting the GPS tracks into more urban areas, each pertaining to cities in Tuscany, spanning from small/medium size cities to large urban areas. We thus obtained five datasets corresponding to the cities of Florence, Pisa, Livorno, Siena and the urban area comprising Pistoia and Prato. For each of the five datasets we perform two further preprocessing steps. First, we assign each stop of each trajectory to the coordinates of the nearest geographical census cell according to the Italian Bureau of Statistics (ISTAT). Second, we discretize the temporal information of the trajectories obtaining the Mobility Dataset Matrix introduced in Section 6.1. Table 6.1 summarizes the characteristics of the five created datasets.

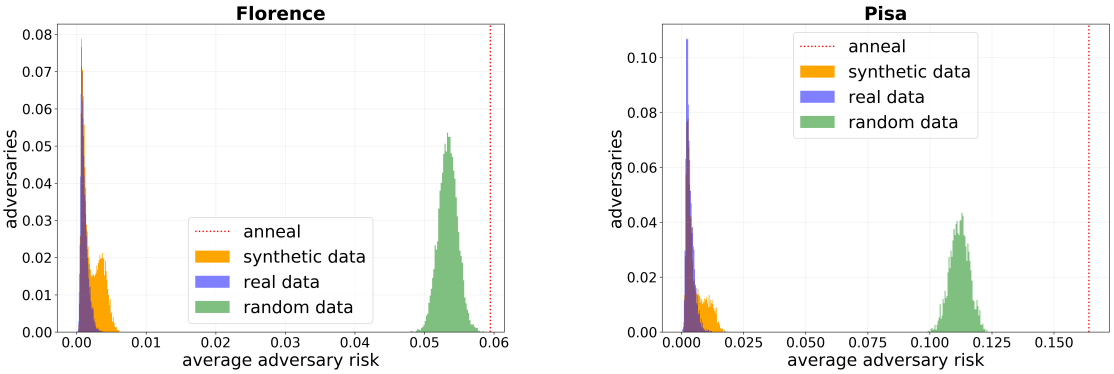
City	Trajectories	Total stops	Mean stops per Individual
Pisa	3281	54295	16.548308
Siena	3463	90850	26.234479
Prato_Pistoia	8651	275729	31.872500
Livorno	2068	28507	13.784816
Florence	9296	143040	15.387263

Table 6.1: Summary of five datasets characteristics

In the following we show results only for the cities of Florence and Pisa, as results for the other three cities are very similar.

6.4.2 Generation of synthetic trajectories

Generative models of individual mobility aim at generating synthetic individual trajectories. One of the most widely accepted individual generative models is the Exploration and Preferential Return (EPR) model [138]. This model is based on the probability that, at any given time, an individual can either explore a new location or return to a previously visited location. While the model is accurate in reproducing basic spatial statistics, it is not able to capture in a realistic way the temporal regularities of human mobility. Several improvements have been proposed on the EPR model, such as d-EPR [112], which modifies the spatial selection of EPR using the collective Gravity model to instruct the generative mechanism on the choice of locations. In our paper, we use DITRAS [113, 114] a modelling framework for generating synthetic mobility trajectories. DITRAS separates the generative procedure in two parts: first, a Markov-chain to generate the temporal component of a trajectory, then the d-EPR model for the spatial component. DITRAS has been proved to be able to capture a large portion of the characteristics of human mobility. We use DITRAS to generate the synthetic trajectories needed for the analysis of the risk produced by a synthetic adversary. To run the generative model, we use the spatial tessellation of Tuscany according to ISTAT cells and its origin-destination matrix. Having roughly 50,000 trajectories in the original data, we simulate the trajectories



(a) Florence

(b) Pisa

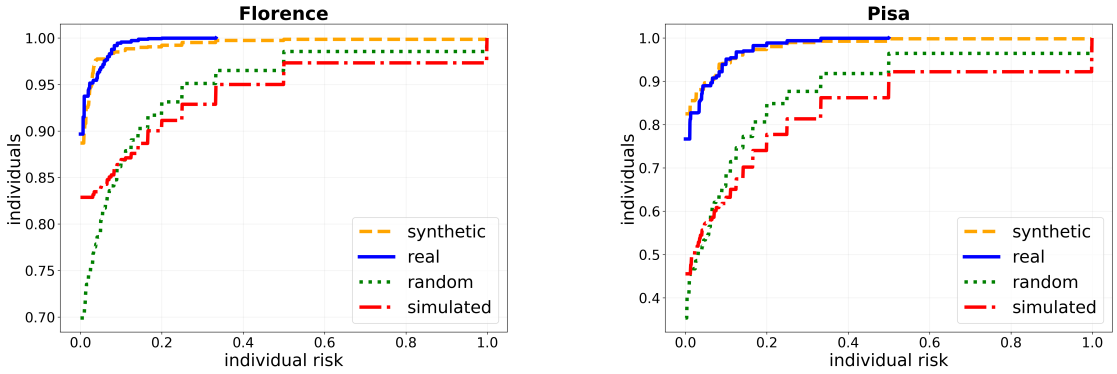
Figure 6.1: Distribution of Average Adversary Risk for real and synthetic adversaries in Pisa and Florence compared to the Average Adversary Risk of a simulated adversary. In blue we see the AAR for real adversaries, in orange we see the same value for synthetic adversaries and in green we see the AAR for randomly generated adversaries. The vertical red line indicates the AAR for the simulated adversary.

of 50,000 agents for one month, using a time slot duration of one hour. Then, we cut the synthetic trajectories obtained to fit them in the five urban areas we use for our experiments.

6.4.3 Experimental Results

For two of the three scenarios that we propose, the real adversary trajectory and the synthetic adversary trajectory, we select the adversary with the highest AAR from a population of possible adversaries. In both cases we have a number of possible adversaries equal to the number of real trajectories. Therefore, to understand how Simulated Privacy Annealing performs with respect to the other two scenarios, we first look at the distribution of the AAR for all possible real and synthetic adversaries, comparing it with the AAR achieved by the simulated approach. As a baseline control we use a set of randomly generated adversaries: random adversary trajectories are generated by selecting, for each timestamp, a random location. We generate a number of random adversary trajectories equal to the number of real and synthetic adversary trajectories.

Figure 6.1 shows that the AAR generated by the simulated adversary is considerably higher than the AAR generated by real, synthetic and random adversaries. These results are consistent across the five urban areas and demonstrate that an adversary that moves similarly to real individuals does not raise particular privacy concerns. On the contrary, an adversary that moves by optimizing the probability of co-location with real individuals yields a significantly higher privacy risk. We can also see how real adversaries, on average, have a slightly lower AAR than synthetic adversaries and both have a much lower AAR than random adversaries. This result suggests that, in order to gather a truly damaging background knowledge, a malicious adversary would need to move in a way that is much



(a) Florence

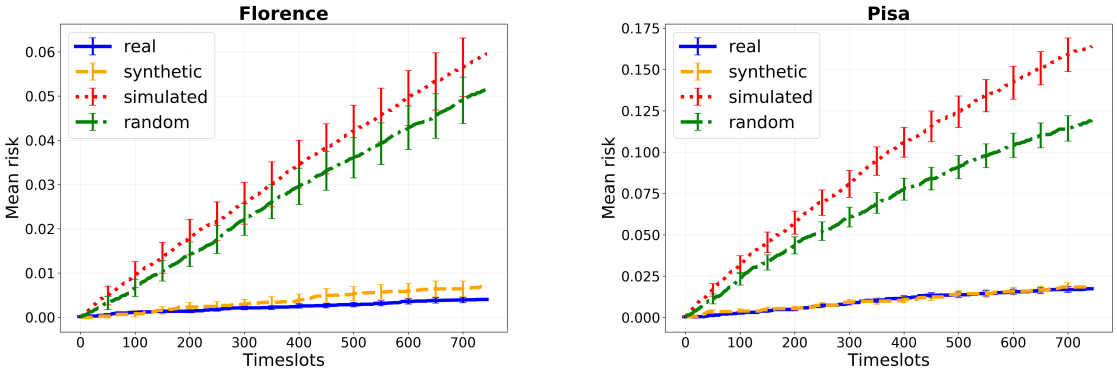
(b) Pisa

Figure 6.2: Cumulative distribution of Privacy Risk for individuals attacked by the best adversary for the three movement scenarios: real, synthetic and simulated. In blue we see the cumulative distribution of privacy risk for the best real adversary, in orange we see the same value for the best synthetic adversary and in green we see the cumulative distribution of privacy risk for the best randomly generated adversaries.

different than real individuals or *likely* individuals, like the ones generated synthetically. Another interesting observation is that the difference between the AAR of the simulated adversary and the AAR of the real, synthetic and random adversaries decreases as the size of the data set increases (Florence is a bigger than Pisa). This difference will be also highlighted in successive experiments.

We then look at how privacy risk distributes for individuals under attack. To do this, we select the best adversary trajectory for each of the three scenarios that we introduced in section 6.2. For real and synthetic adversary trajectories, we take the best performing trajectories out of the possible population of adversaries (T_{real} and T_{synth}). For the simulated adversary trajectory we consider the result of our direct simulation (T_{sim}) using the simulated privacy annealing procedure as explained in Section 6.3.3.

Figure 6.2 shows the cumulative distribution of privacy risk for the individuals in the real data subjected to the attack of the best adversary trajectories for our scenarios. We recall that privacy risk ranges in the interval $[0, 1]$ and that it is essentially the reciprocal of integers $(1/2, 1/3, ...)$. The cumulative distribution of risk can be read as the portion of individuals under a certain level of risk: the lower a curve, the higher the privacy risk overall, as more individuals have higher privacy risk. We see that T_{real} does not re-identify completely any individual: values beyond certain levels of risk are completely lacking. Again, we observe that the simulated adversary T_{sim} presents a lower cumulative distribution of privacy risk with respect to T_{real}, T_{synth} and the random baseline. Again, the difference in overall risk decreases as the dimension of the data set increases. These results clearly show that Simulated Privacy Annealing allows us to generate an adversary trajectory with an AAR higher than any other possible adversary, be it real, synthetic, or random. Moreover, for bigger data sets, we have overall lower levels of privacy risk because trajectories move over a more sparse and large territory. This suggests us that



(a) Florence

(b) Pisa

Figure 6.3: Variation of Average Adversary Risk in time for the most effective attackers of each scenario. The risk is calculated as time passes and the trajectory of the corresponding adversary grows in terms of stops and, therefore, background knowledge elements. In blue we see the average produced risk for the most effective real adversary, in orange for the most effective synthetic adversary, in red for the simulated adversary and in green for an adversary generated with a completely random movement. Standard error is also shown for each adversary every 50 timeslots.

the bigger the territory the harder it is for a malicious adversary to pose a threat to the privacy of individuals represented in the mobility data set.

6.4.4 Simulated Annealing Analysis

The simulated adversary, though unrealistic in their movement, serves as a baseline for our experiments. We find that the simulated adversary produces an AAR higher than the real and synthetic adversaries, throughout all time slots and regardless the observation period (Figure 6.3). The simulated adversary is hence an upper bound for AAR, meaning that it is the worst possible single adversary for a mobility dataset. Moreover, we generated a completely random trajectory and confronted the resulting AAR with the one produced by a simulated adversary. We find that, while significantly higher than real or synthetic adversaries, a completely random trajectory does not yield the same risk as a simulated trajectory specifically obtained with the objective of maximizing average risk.

As Tables 6.2 and 6.3 show, the best simulated adversary's trajectory (T_{sim}) has a peculiar structure that significantly differs from the structure of real (T_{real}) and synthetic (T_{synth}) adversary's trajectories. In particular, in T_{sim}, the mover changes the location at every time slot and visits a large number of locations, as witnessed by the value of the mobility entropy, which is much higher than the values of T_{real} and T_{synth}. In other words, the simulated approach, while it is more realistic than the worst-case scenario approach traditionally employed by privacy risk assessment frameworks, and while producing the highest AAR, generates an adversary trajectory that is inconsistent with real human mobility trajectories. As Figure 6.3 shows, although the trajectory obtained with simulated

Pisa	real	synthetic	simulated
number of trips	113.000000	100.000000	744.000000
mean distance	4.693582	2.209392	3.733869
unique locs	49.000000	41.000000	426.000000
entropy	4.157489	3.101694	8.560665
radius	3.758526	2.035889	3.069468

Table 6.2: Mobility analysis of the most efficient adversaries, real and synthetic, in comparison with the simulated adversary, for the Pisa dataset.

Florence	real	synthetic	simulated
number of trips	96.000000	109.000000	744.000000
mean distance	3.528668	1.646238	4.537649
unique locs	24.000000	34.000000	590.000000
entropy	2.191098	3.714275	9.102039
radius	2.764602	1.806907	3.757542

Table 6.3: Mobility analysis of the most efficient adversaries, real and synthetic, in comparison with the simulated adversary, for the Florence dataset.

annealing may seem random, we show that randomly generated trajectories do not produce the same risk as a simulated one. A visualization of the different best adversary trajectories can be seen in Figure 6.4.

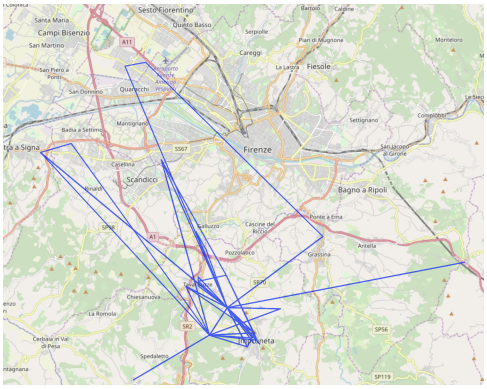
6.4.5 Performance analysis of Simulated Annealing

We find that the simulated annealing approach is robust with respect to both the limits we impose on the movements of the adversary and the cooling rate used to decrease the temperature (Figure 6.5(a) and Figure 6.5(b)).

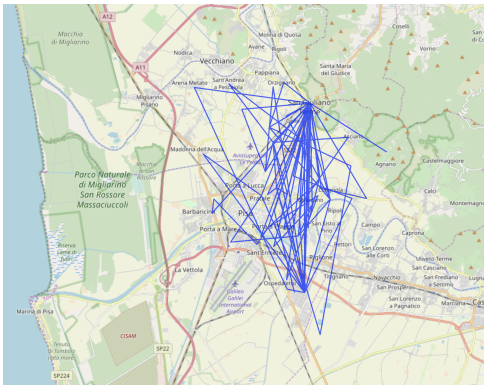
Regarding the cooling rate, we test values ranging from 0.90 to 0.98. This relatively low decreasing rate allow us for a broad exploration of the space of solutions. For both the urban areas considered and varying the cooling rate, the risk produced by the simulated adversary remains stable.

Regarding the distance limit, we test values ranging from 0.5 kilometers to 5 kilometers. These are relatively strict limits, considering that in an urban area and 1 hour, an agent can potentially cover a larger distance. We find that the risk produced by the simulated adversary is stable. Again, for both urban areas and varying the distance limit, the risk produced by the simulated adversary remains stable.

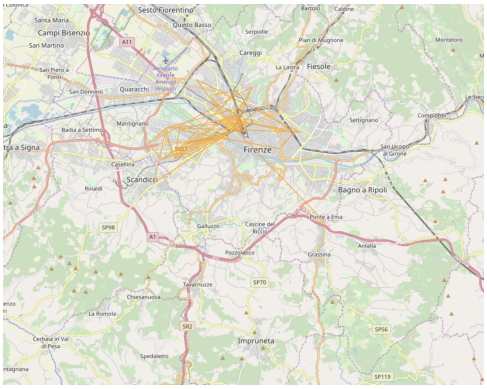
In Figure 6.6(a) and Figure 6.6(b), we investigate the evolution of the risk produced by the simulated adversary's trajectory as time goes by. To be completed, the annealing process requires roughly 26 minutes for the smaller data set (Pisa), and more than 2 and a half hours for the larger data set (Florence). Interestingly, for the larger data set, the improvement emerges early in the annealing process. Conversely, for the smaller data set, the improvements are evenly spread during the running time of the procedure. This useful information can be exploited by an analyst to understand when the annealing process can



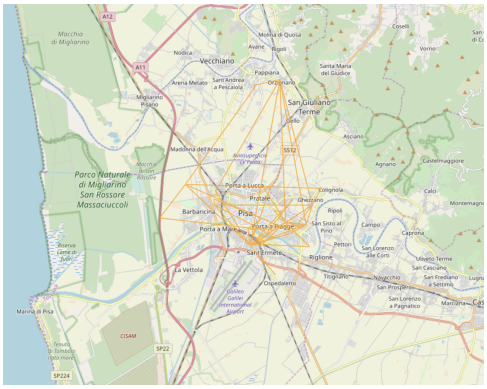
(a) Florence real trajectory



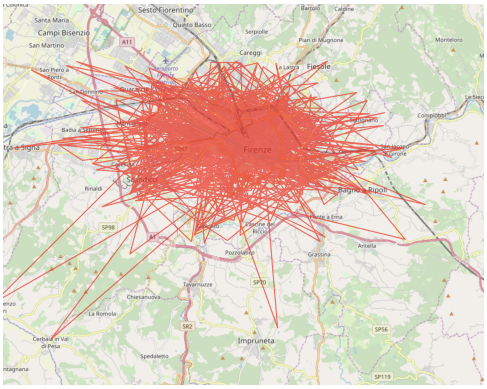
(b) Pisa real trajectory



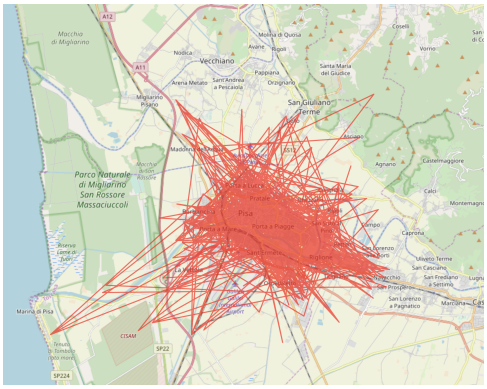
(c) Florence synthetic trajectory



(d) Pisa synthetic trajectory

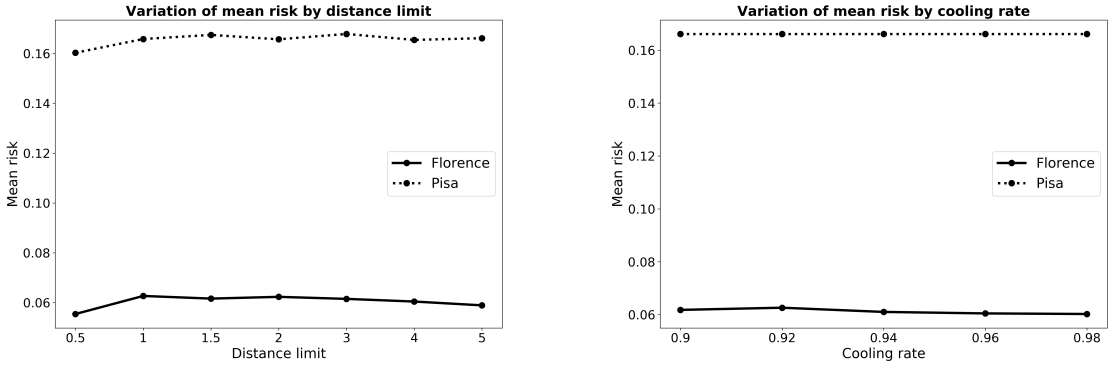


(e) Florence simulated trajectory



(f) Pisa simulated trajectory

Figure 6.4: Visualization of the worst adversary trajectories for the three scenarios. In blue real trajectories, in orange synthetic trajectories, in red simulated trajectories.



(a) Distance limit

(b) Cooling rate

Figure 6.5: Variation of Average Adversary Risk by distance limit and exponential cooling rate, for the cities of Florence and Pisa. Both parameters do not have a high impact on the final result of the annealing procedure.

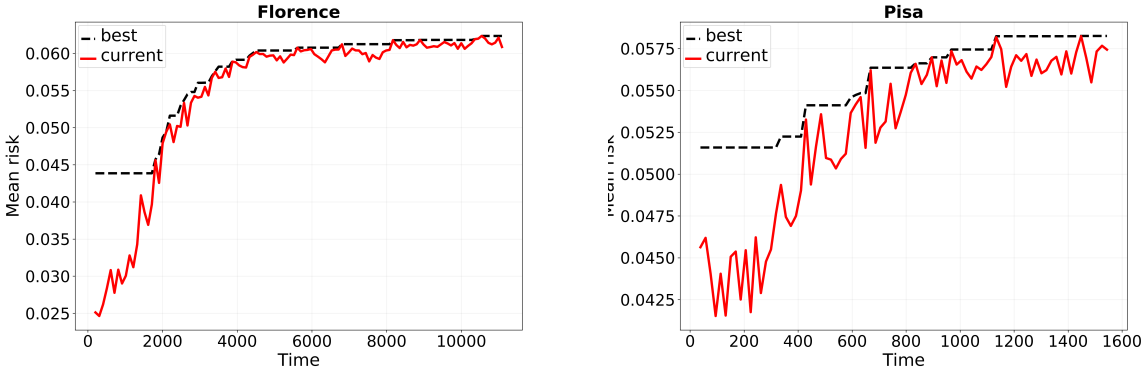
be stopped and to adjust the stopping criteria if time is a constraint in risk analysis.

6.4.6 Discussion

Our experiments show how it is possible to understand what damage a single malicious adversary could do to a mobility data set. To provide a different perspective with respect to existing and worst-cast privacy frameworks, we simulate the actual movement of a potential adversary in several different ways, thus generating a more realistic background knowledge. From our experiments, we conclude that Simulated Privacy Annealing provides a robust evaluation to the privacy risk that a malicious adversary can produce: the AAR obtained with Simulated Privacy Annealing is much higher than the one obtained with real or synthetic trajectories. In real-world scenarios, in which an adversary moves similarly to a real individual, the people's privacy risk would be lower than the risk estimated by traditional frameworks.

Although Simulated Privacy Annealing complies with the natural spatio-temporal constraints of human mobility, the simulated adversary trajectory vastly differs from the realistic and the synthetic adversary trajectories. This difference emerges from both a visual inspection of the trajectories and the analysis of their mobility patterns (Figure 6.4(f)). Simulated Privacy Annealing is stable concerning the input parameters: both the distance limit and the cooling rate do not impact significantly on the final performance of the simulated annealing. The main drawback of our approach is the high cost in terms of execution times. While the Simulated Privacy Annealing procedure may take several hours to complete, our findings seem to indicate that, for bigger data sets, convergence is reached quicker with a reasonably efficient solution.

In summary, our aim was to tackle the issue of the generation of an adversary's background knowledge in privacy risk assessment process by proposing a more realistic approach, tailored for human mobility data. Our proposal was to represent the behavior



(a) Florence

(b) Pisa

Figure 6.6: Variation of the Average Adversary Risk of the current solution and the best solution in time for the cities of Florence and Pisa.

of an adversary as a trajectory, and we envisioned three possible scenarios with three different methods for generating such trajectory: a scenario where such trajectory is a real one, a scenario where such trajectory is generated with a generative model and one scenario where such trajectory is built with the Simulated Privacy Annealing algorithm, with the specific objective of maximizing average risk.

The drawbacks of our method can be traced to how time-consuming the simulated annealing procedure can be depending on the size of the data set. While simulated annealing gives us the advantage of proven optimality and, therefore, gives us an upper bound to the privacy risk produced for individuals, our experiments also show that a random trajectory may produce acceptable results in far less time. This may suggests that further improvements could be done with our method, speeding up the computation time by further tuning the algorithm.

Another possible improvement may come from different functions to evaluate privacy risk: we chose the average adversary risk as it represents a fair way to synthesize the risk for all the individuals involved, but other functions may be tested in order to evaluate risk under different perspectives. Finally, our approach is tailored for human mobility data: it would be interesting to develop a realistic approach for the generation of background knowledge also to other kinds of data such as retail or network data.

Chapter 7

Conclusions and Future Works

The privacy of personal data is currently one of the most discussed topics in data analytics. Some analysts have gone as far as calling data "the new oil" [1]. This valuable resource attracts many interests and because of that the risk of a privacy violation for the people represented in the data is growing day by day. At the same time, the interest of companies, enterprises and analysts is completely justified as data allow them to discover more about human dynamics, and thus our society. Business decisions, new discoveries, and improvement to social well being are all important objectives that can be achieved through the proper analysis of big data. It is therefore in the interest of all parties involved to find suitable methodologies to protect individual privacy while at the same time allowing for meaningful analyses of personal data. We think that privacy risk assessment is one of the fundamental steps in building a privacy aware ecosystem. Through the use of privacy risk assessment techniques, data holders can quantitatively evaluate privacy risk for the data that they are managing and understand which individuals are at risk of a privacy violation. Traditional privacy framework evaluate privacy risk on a worst-case scenario, i.e., assuming that an adversary knows everything that he possibly can to re-identify an individual. The more recent state-of-the-art PRUDEnce framework moves forward in an interesting direction, allowing for the systematic evaluation of privacy risk, mathematically generating background knowledge and therefore allowing data holders and providers to analyse privacy risk with different levels of background knowledge. While this is a considerable step in the right direction, our aim with this thesis was to address some of the drawbacks of PRUDEnce and other existing privacy risk assessment frameworks, by proposing new models and algorithms for privacy risk evaluation. We showed how PRUDEnce can be used in practice to assess privacy risk, by providing the mathematical models for a number of privacy attacks on three different kinds of data. We introduced two extension to the PRUDEnce framework. The first to integrate distance based record linkage for retail data in prudence. The second to expand the data quality assessment of PRUDEnce, by evaluating the changes in the distribution of data specific metrics in a quantitative way. We then focused on one of the most important drawback for privacy risk assessment: assessment time. Computational complexity of PRUDEnce is one of the drawbacks of this framework. To tackle this shortcoming, we proposed a data mining

[1] https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data

approach for the prediction of privacy risk based on data specific individual features and validated our approach on three different kinds of data, achieving good results in terms of prediction performances and execution time. Finally, we departed from the PRUDEnce framework, with the aim of proposing a different approach to privacy risk evaluation for mobility data. Our goal was to provide a methodology to simulate the process with which a malicious adversary gathers the information she will use to conduct a privacy attack. We devise our model on mobility data, leveraging the natural spatio-temporal constraints of this kind of data.

Our work can be extended in several interesting directions. For data quality assessment we proposed a solution that at its core is flexible and applicable with different distance evaluation functions. An interesting development could be a study of different functions to compute the change in the distributions of data metrics with respect to the original data under different privacy protection methods. For the prediction of privacy risk, we adopted an approach based on individual features extracted from the data. Domain knowledge is therefore paramount to correctly select the right features for the task, but a domain expert may not be always available when evaluating privacy risk. Therefore, a possible direction in which to extend our research is to devise a featureless approach, that would enable data holders to predict privacy risk directly from raw data without having to select and compute data specific features. The featureless approach should also provide an explanation to the predicted risk, exploiting modern explanation techniques for black box algorithms [62]. This goes in the direction of evolving privacy-by-design in ethics-by-design. To fully comply with the current state of legislation (GDPR [1], data analysis processes will have to be verified for compliance with a broad set of ethical values, including privacy, unfairness, bias and discrimination detection. The hope is to use the result of the assessment not only as awareness tool for users but also for guiding the ethical design of the analysis process. The methods to describe and assess ethical values can exploit both the formal specification of ethical values and the additional knowledge derived by the ethicality assessment. These steps are of fundamental importance, as we proceed into a future where data analysis and AI design will drive innovation and social progress. Finally, for closing the gap between thorough privacy risk assessment and realistic privacy attack simulation, our work on the modeling of adversarial behavior in mobility data can be further improved, by looking at alternative ways to compute privacy risk. Our model, which searches for the most efficient behaviour that a malicious adversary could maintain to maximize risk with respect to a dataset, can be easily extended with different risk functions to be optimized. Further development of this model could help a data holder build a plethora of tools to quickly and efficiently assess privacy risk under different assumption.

To conclude, we have proposed a set of models and algorithms to tackle the problem of privacy risk assessment and improve on existing frameworks. We show that efficient and quick privacy risk assessment is possible and can be conducted in a data driven way, by considering the features and natural constraints of the data.

Special thanks

The Ph.D. has been an adventure in which I was thrust almost by chance, and what a chance it has been. To grow, to learn, to travel.

No man can truly say to be the sole maker of his fortunes and there are many incredible people that helped me in this journey that I feel the need to thank.

First and foremost, thanks to Anna Monreale, my supervisor, for always believing in me, even when I was undeserving. To her I owe all the opportunities that have been given to me.

To Dino Pedreschi, my supervisor, for his guidance, patience and wisdom.

To the KDD Lab, all wonderful people, that welcomed me with open arms. In particular I'd like to thank Francesca Pratesi and Luca Pappalardo, with whom I feel I developed a friendship that goes beyond the working place.

To Filippo Simini, my host for my period abroad, who welcomed me to a foreign country and offered me new perspectives on my work.

And then, there are those outside academia, that stood by my side countless times, whose support has been immense and invaluable.

To Andrea, my friend since more than twenty years, for being there, all along.

To Alice, Alessandro, Valentina and Stefano, as they cared for me in the times of need, for being true friends even when I was insufferable.

To Fabrizia, for the joy she brought into my life.

Finally, to the most important people, the family that any good man would love to have, my father Daniele, my mother Teodora and my brother Luca.

Bibliography

[1] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *OJ*, L119:1–88, May 2016.

[2] Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *OJ*, L281/31:1–9, October 1995.

[3] Jemal H. Abawajy, Mohd Izuan Hafez Ninggal, and Tutut Herawan. Vertex re-identification attack using neighbourhood-pair properties. *Concurrency and Computation: Practice and Experience*, 28(10):2906–2919, 2016.

[4] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE '08*, pages 376–385, 2008.

[5] Osman Abul, Francesco Bonchi, and Mirco Nanni. Anonymization of moving objects databases by clustering and perturbation. *Information Systems*, 35(8):884–910, December 2010.

[6] Jagdish Prasad Achara, Gergely Ács, and Claude Castelluccia. On the unicity of smartphone applications. In *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society, WPES 2015, Denver, Colorado, USA, October 12, 2015*, pages 27–36.

[7] Ajaya Adhikari, David M. J. Tax, Riccardo Satta, and Matthias Faeth. LEAFAGE: example-based and feature importance-based explanations for black-box ML models. In *FUZZ-IEEE*, pages 1–7. IEEE, 2019.

[8] Charu C. Aggarwal and Philip S. Yu, editors. *Privacy-preserving data mining: models and algorithms*. Number 34 in Advances in database systems. Springer, New York, NY, 2008. OCLC: 255823401.

[9] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '01*, pages 247–255, Santa Barbara, California, United States, 2001. ACM Press.

[10] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[11] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *ACM SIGMOD Record*, 29(2):439–450, June 2000.

[12] Abdulatif Alabdulatif, Ibrahim Khalil, Mark Reynolds, Heshan Kumarage, and Xun Yi. Privacy-preserving data clustering in cloud computing based on fully homomorphic encryption. In *PACIS*, page 289, 2017.

[13] Hayder Amer, Naveed Salman, Matthew Hawes, Moumena Chaqfeh, Lyudmila Mihaylova, and Martin Mayfield. An improved simulated annealing technique for enhanced mobility in smart cities. *Sensors*, 16:1013, 06 2016.

[14] Swathi Ananthula, Omar Abuzaghleh, Navya Bharathi Alla, Swetha Prabha Chaganti, Pragna chowdary kaja, and Deepthi Mogilineedi. Measuring Privacy in Online Social Networks. *International Journal of Security, Privacy and Trust Management*, 4(2):01–09, May 2015.

[15] Henrik Andersen, MD Andreasen, and PØ Jacobsen. The crm handbook: From group to multi-individual. *Norhaven: PricewaterhouseCoopers*, 1999.

[16] Alessandro Armando, Michele Bezzi, Nadia Metoui, and Antonino Sabetta. Risk-based privacy-aware information disclosure. *Int. J. Secur. Softw. Eng.*, 6(2):70–89.

[17] Michael Backes and Sebastian Meiser. Differentially private smart metering with battery recharging. In *DPM/SETOP*, volume 8247 of *Lecture Notes in Computer Science*, pages 194–212. Springer, 2013.

[18] Jane R. Bambauer. Tragedy of the data commons. 25, 03 2011.

[19] Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R. James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J. Ramasco, Filippo Simini, and Marcello Tomasini. Human mobility: Models and applications. *Physics Reports*, 734:1 – 74, 2018. Human mobility: Models and applications.

[20] Anirban Basu, Anna Monreale, Juan Camilo Corena, Fosca Giannotti, Dino Pedreschi, Shinsaku Kiyomoto, Yutaka Miyake, Tadashi Yanagihara, and Roberto Trasarti. A privacy risk model for trajectory data. In Jianying Zhou, Nurit Gal-Oz, Jie Zhang, and Ehud Gudes, editors, *Trust Management VIII*, pages 125–140. 2014.

[21] Vladimir Batagelj and Matjaz Zaversnik. An o(m) algorithm for cores decomposition of networks. *CoRR*, cs.DS/0310049, 2003.

[22] Justin Becker and Hao Chen. Measuring privacy risk in online social networks.

[23] Walid Ben-Ameur. Computing the initial temperature of simulated annealing. *Computational Optimization and Applications*, 29:369–385, 12 2004.

[24] Walid Ben-Ameur. Computing the initial temperature of simulated annealing. *Computational Optimization and Applications*, 29:369–385, 12 2004.

[25] Leo Breiman. Bagging predictors, 1994.

[26] Sven Bugiel, Stefan Nürnberger, Thomas Pöppelmann, Ahmad-Reza Sadeghi, and Thomas Schneider. AmazonIA: when elasticity snaps back. In *Proceedings of the 18th ACM conference on Computer and communications security - CCS '11*, page 389, Chicago, Illinois, USA, 2011. ACM Press.

[27] A. Cavoukian, K.E. Emam, Information, and Privacy Commissioner/Ontario. *Dispelling the Myths Surrounding De-identification: Anonymization Remains a Strong Tool for Protecting Privacy*. DesLibris: Documents collection. Information and Privacy Commissioner of Ontario, Canada, 2011.

[28] Ann Cavoukian. Privacy design principles for an integrated justice system - working paper. 2000. urlhttps://www.ipc.on.ca/english/Resources/Discussion-Papers/Discussion-Papers-Summary/?id=318.

[29] Ann Cavoukian. Privacy by design the 7 foundational principles. August 2009.

[30] Alket Cecaj, Marco Mamei, and Franco Zambonelli. Re-identification and information fusion between anonymized CDR and social network data. *J. Ambient Intelligence and Humanized Computing*, 7(1):83–96, 2016.

[31] Rui Chen, Benjamin C. M. Fung, and Bipin C. Desai. Differentially private trajectory data publication. *CoRR*, abs/1112.2020, 2011.

[32] P. Christen. *Data Matching—Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[33] Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Thanh T. L. Tran. Differentially private summaries for sparse data. In *ICDT '12*, pages 299–311, 2012.

[34] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3:1376 EP –, 03 2013.

[35] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requir. Eng.*, 16(1):pp 3–32, 2011.

[36] Himel Dev, Tanmoy Sen, Madhusudan Basak, and Mohammed Eunus Ali. An Approach to Protect the Privacy of Cloud Data from Data Mining Based Attacks. In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, pages 1106–1115, Salt Lake City, UT, November 2012. IEEE.

[37] Jesus Diaz, Seung Geol Choi, David Arroyo, Angelos D. Keromytis, Francisco B. Rodríguez, and Moti Yung. Privacy in e-shopping transactions: Exploring and addressing the trade-offs. In *CSCML*, volume 10879 of *Lecture Notes in Computer Science*, pages 206–226. Springer, 2018.

[38] Josep Domingo-Ferrer. *A Three-Dimensional Conceptual Framework for Database Privacy*, pages 193–202. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[39] Josep Domingo-Ferrer and Vicenç Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Min. Knowl. Discov.*, 11(2):195–212, 2005.

[40] Alan S Dunk. Product life cycle cost analysis: the impact of customer profiling, competitive advantage, and quality of is information. *Management Accounting Research*, 15(4):401–414, 2004.

[41] Cynthia Dwork. Differential Privacy. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, volume 4052, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[42] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC '06*, pages 265–284, 2006.

[43] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Differential privacy – a primer for the perplexed. 2011.

[44] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013.

[45] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63:1057–1066, 2009.

[46] Mark Elliot. *Integrating File and Record Level Disclosure Risk Assessment*, pages 126–134. Springer Berlin Heidelberg, 2002.

[47] Matthias Enzmann, Thomas Kunz, and Markus Schneider. A new infrastructure for user tracking prevention and privacy protection in internet shopping. In *InfraSec*, volume 2437 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2002.

[48] Alexandre V. Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. Privacy preserving mining of association rules. *Inf. Syst.*, 29(4):343–364, 2004.

[49] Sam Fletcher and Md. Zahidul Islam. Decision tree classification with differential privacy: A survey. *ACM Comput. Surv.*, 52(4):83:1–83:33, 2019.

[50] Luisa Franconi and Silvia Polettini. *Individual Risk Estimation in μ-Argus: A Review*, pages 262–272. Springer Berlin Heidelberg.

[51] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215 – 239, 1978.

[52] Tianchong Gao, Feng Li, Yu Chen, and Xukai Zou. Preserving local differential privacy in online social networks. In *WASA*, volume 10251 of *Lecture Notes in Computer Science*, pages 393–405. Springer, 2017.

[53] Fosca Giannotti, Cristian Gozzi, and Giuseppe Manco. Clustering transactional data. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '02, pages 175–187, London, UK, UK, 2002. Springer-Verlag.

[54] Fosca Giannotti, Anna Monreale, and Dino Pedreschi. Mobility data and privacy. In E. Zimanyi C. Renso, S. Spaccapietra, editor, *Mobility Data Modeling, Management, and Understanding*, pages 174–193, 2013.

[55] Fosca Giannotti and Dino Pedreschi, editors. *Mobility, Data Mining and Privacy*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[56] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[57] Carsten Grabow, Stefan Grosskinsky, Jürgen Kurths, and Marc Timme. Collective relaxation dynamics of small-world networks. *CoRR*, abs/1507.04624, 2015.

[58] Nils Gruschka and Luigi Lo Iacono. Vulnerable Cloud: SOAP Message Security Validation Revisited. In *2009 IEEE International Conference on Web Services*, pages 625–631, Los Angeles, CA, USA, July 2009. IEEE.

[59] Riccardo Guidotti. Personal data analytics: capturing human behavior to improve self-awareness and personal services through individual and collective knowledge. 2017.

[60] Riccardo Guidotti, Michele Coscia, Dino Pedreschi, and Diego Pennacchioli. Behavioral entropy and profitability in retail. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–10. IEEE, 2015.

[61] Riccardo Guidotti, Anna Monreale, Mirco Nanni, Fosca Giannotti, and Dino Pedreschi. Clustering individual transactional data for masses of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 195–204, New York, NY, USA, 2017. ACM.

[62] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, 2019.

[63] Riccardo Guidotti, Giulio Rossetti, Luca Pappalardo, Fosca Giannotti, and Dino Pedreschi. Market basket prediction using user-centric temporal annotated recurring sequences. In *Data Mining (ICDM), 2017 IEEE International Conference on*, pages 895–900. IEEE, 2017.

[64] Mehmet Emre Gursoy, Ali Inan, Mehmet Ercan Nergiz, and Yucel Saygin. Differentially private nearest neighbor classification. *Data Mining and Knowledge Discovery*, 31(5):1544–1575, Sep 2017.

[65] Qilong Han, Zuobin Xiong, and Kejia Zhang. Research on trajectory data releasing method via differential privacy based on spatial partition. *Security and Communication Networks*, 2018:4248092:1–4248092:14, 2018.

[66] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009.

[67] Juan Alberto Martínez Hernández and Piet Van Mieghem. Classification of graph metrics. 2015.

[68] Javier Herranz, Jordi Nin, Pablo Rodríguez, and Tamir Tassa. Revisiting distance-based record linkage for privacy-preserving release of statistical datasets. *Data Knowl. Eng.*, 100:78–93, 2015.

[69] Mireille Hildebrandt. Defining profiling: a new type of knowledge? In *Profiling the European citizen*, pages 17–45. Springer, 2008.

[70] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, 1998.

[71] Sobaria Ibrahim and Amani Mahagoub Omer. Survey on k-anonymity: Methods based on generalization technique. In *ICISA*, pages 1–2. IEEE, 2013.

[72] C. S. E. Institute. Octave – (operationally critical threat, asset, and vulnerability evaluation). http://www.cert.org/octave/.

[73] Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[74] Shouling Ji, Weiqing Li, Mudhakar Srivatsa, Jing Selena He, and Raheem Beyah. *Structure Based Data De-Anonymization of Social Networks and Mobility Traces*, pages 237–254. Springer International Publishing, Cham, 2014.

[75] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella. Human mobility models for opportunistic networks. *IEEE Communications Magazine*, 49(12):157–165, December 2011.

[76] Günter Karjoth, Matthias Schunter, and Michael Waidner. Privacy-enabled management of customer data. *IEEE Data Eng. Bull.*, 27(1):3–9, 2004.

[77] A. Khachaturyan, S. Semenovsovskaya, and B. Vainshtein. The thermodynamic approach to the structure analysis of crystals. *Acta Crystallographica Section A*, 37(5):742–754, Sep 1981.

[78] Scott Kirkpatrick, D. Gelatt Jr., and Mario P. Vecchi. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983.

[79] Dániel Kondor, Behrooz Hashemian, Yves-Alexandre Montjoye, and Carlo Ratti. Towards matching user mobility traces in large-scale datasets. *IEEE Transactions on Big Data*, PP, 09 2017.

[80] Ulrich König. Primelife checkout - - A privacy-enabling e-shopping user interface. In *PrimeLife*, volume 352 of *IFIP Advances in Information and Communication Technology*, pages 325–337. Springer, 2010.

[81] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Mondrian Multidimensional K-Anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 25–25, Atlanta, GA, USA, 2006. IEEE.

[82] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[83] Haifei Li, Patrick C. K. Hung, Jia Zhang, and David Ahn. Designing privacy policies for adopting RFID in the retail industry. In *IEEE SCC*, pages 251–252. IEEE Computer Society, 2005.

[84] Lichun Li, Rongxing Lu, Kim-Kwang Raymond Choo, Anwitaman Datta, and Jun Shao. Privacy-preserving-outsourced association rule mining on vertically partitioned databases. *IEEE Trans. Information Forensics and Security*, 11(8):1847–1861, 2016.

[85] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, Istanbul, April 2007. IEEE.

[86] Chenyang Liu, Dan Yin, Hao Li, Wei Wang, and Wu Yang. Preserving privacy in social networks against label pair attacks. In *Wireless Algorithms, Systems, and Applications - 12th International Conference, WASA 2017, Guilin, China, June 19-21, 2017, Proceedings*, pages 381–392, 2017.

[87] Kun Liu and Evimaria Terzi. A framework for computing the privacy scores of users in online social networks. *TKDD*, 5(1):6:1–6:30, 2010.

[88] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24, Atlanta, GA, USA, 2006. IEEE.

[89] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 126–135.

[90] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103. IEEE Computer Society, 2007.

[91] J.D. Meier and Microsoft Corporation. *Improving Web Application Security: Threats and Countermeasures*. Patterns & practices. Microsoft, 2003.

[92] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *jcp*, 21:1087–1092, June 1953.

[93] Adam Meyerson and Ryan Williams. On the complexity of optimal K-anonymity. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '04*, page 223, Paris, France, 2004. ACM Press.

[94] Debasis Mitra, Fabio Romeo, and Alberto Sangiovanni-Vincentelli. Convergence and finite-time behavior of simulated annealing. *Advances in Applied Probability*, 18(3):747–771, 1986.

[95] Noman Mohammed, Benjamin Fung, and Mourad Debbabi. Walking in the crowd: Anonymizing trajectory data for pattern analysis. pages 1441–1444, 01 2009.

[96] Anna Monreale, Gennady L. Andrienko, Natalia V. Andrienko, Fosca Giannotti, Dino Pedreschi, Salvatore Rinzivillo, and Stefan Wrobel. Movement data anonymity through generalization. *TDP*, 3(2):91–121, 2010.

[97] Anna Monreale, Wendy Hui Wang, Francesca Pratesi, Salvatore Rinzivillo, Dino Pedreschi, Gennady Andrienko, and Natalia Andrienko. *Privacy-Preserving Distributed Movement Data Aggregation*, pages 225–245. Lecture Notes in Geoinformation and Cartography. Springer International Publishing, 2013.

[98] Basilisa Mvungi and Mizuho Iwaihara. Associations between privacy, risk awareness, and interactive motivations of social networking service users, and motivation prediction from observable features. *Computers in Human Behavior*, 44:20–34, 2015.

[99] Moni Naor and Benny Pinkas. Oblivious polynomial evaluation. *SIAM J. Comput.*, 35(5):1254–1281, 2006.

[100] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *30th IEEE Symposium on Security and Privacy (S&P 2009), 17-20 May 2009, Oakland, California, USA*, pages 173–187.

[101] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *CoRR*, abs/cs/0610105, 2006.

[102] Geeta S. Navale and Suresh N. Mali. Survey on privacy preserving association rule data mining. *IJRSDA*, 4(2):63–80, 2017.

[103] M. E. J. Newman. *Mathematics of Networks*, pages 1–8. Palgrave Macmillan UK, London, 2016.

[104] Jordi Nin, Javier Herranz, and Vicenç Torra. *Using Classification Methods to Evaluate Attribute Disclosure Risk*, pages 277–286. Springer Berlin Heidelberg.

[105] NIST. Risk management guide for information technology systems, special publication 800-30. url=http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf.

[106] Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. Ranking of closeness centrality for large-scale social networks. In *FAW*, volume 5059 of *Lecture Notes in Computer Science*, pages 186–195. Springer, 2008.

[107] OWASP. Risk rating methodology. url=http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf.

[108] Nour E. Oweis, Suhail S. J. Owais, Waseem George, Mona G. Suliman, and Václav Snásel. A survey on big data, mining: (tools, techniques, applications and notable uses). In *ECC*, volume 370 of *Advances in Intelligent Systems and Computing*, pages 109–119. Springer, 2015.

[109] Gerhard Paass. Disclosure Risk and Disclosure Avoidance for Microdata. *Journal of Business & Economic Statistics*, 6(4):487, October 1988.

[110] Luca Pappalardo, Gianni Barlacchi, Roberto Pellungrini, and Filippo Simini. Human mobility from theory to practice: Data, models and applications. In *WWW (Companion Volume)*, pages 1311–1312. ACM, 2019.

[111] Luca Pappalardo, Salvatore Rinzivillo, Zehui Qu, Dino Pedreschi, and Fosca Giannotti. Understanding the patterns of car travel. *The European Physical Journal Special Topics*, 215(1):61–73, 2013.

[112] Luca Pappalardo, Salvatore Rinzivillo, and Filippo Simini. Human mobility modelling: Exploration and preferential return meet the gravity model. volume 83, 05 2016.

[113] Luca Pappalardo and Filippo Simini. Data-driven generation of spatio-temporal routines in human mobility. *Data Mining and Knowledge Discovery*, 32(3):787–829, May 2018.

[114] Luca Pappalardo, Filippo Simini, Gianni Barlacchi, and Roberto Pellungrini. scikit-mobility: a python library for the analysis, generation and risk assessment of mobility data. arxiv:1907.07062, 2019.

[115] Luca Pappalardo, Filippo Simini, Salvatore Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-Laszlo Barabasi. Returners and explorers dichotomy in human mobility. *Nature Communications*, 6, 09 2015.

[116] Sankita Patel, Sweta Garasia, and Devesh Jinwala. An Efficient Approach for Privacy Preserving Distributed K-Means Clustering Based on Shamir's Secret Sharing Scheme. In Theo Dimitrakos, Rajat Moona, Dhiren Patel, and D. Harrison McKnight, editors, *Trust Management VI*, volume 374, pages 129–141. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[117] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[118] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 727–734, 2000.

[119] Roberto Pellungrini, Anna Monreale, and Riccardo Guidotti. Privacy risk for individual basket patterns. In *ECML PKDD 2018 Workshops - MIDAS 2018 and PAP 2018, Dublin, Ireland, September 10-14, 2018, Proceedings*, pages 141–155, 2018.

[120] Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. Fast estimation of privacy risk in human mobility data. In *SAFECOMP Workshops*, volume 10489 of *Lecture Notes in Computer Science*, pages 415–426. Springer, 2017.

[121] Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. Analyzing privacy risk in human mobility data. In *Software Technologies: Applications and Foundations - STAF 2018 Collocated Workshops, Toulouse, France, June 25-29, 2018, Revised Selected Papers*, pages 114–129, 2018.

[122] Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. A data mining approach to assess privacy risk in human mobility data. *ACM TIST*, 9(3):31:1–31:27, 2018.

[123] Roberto Pellungrini, Francesca Pratesi, and Luca Pappalardo. Assessing privacy risk in retail data. In *Personal Analytics and Privacy. An Individual and Collective Perspective - First International Workshop, PAP 2017, Held in Conjunction with ECML PKDD 2017, Skopje, Macedonia, September 18, 2017, Revised Selected Papers*, pages 17–22, 2017.

[124] Roberto Pellungrini, Filippo Simini, Luca Pappalardo, and Anna Monreale. Modeling adversarial behavior against mobility data privacy. *Submitted to IEEE Transactions on Intelligent Transportation Systems*, 2019.

[125] Ruggero G. Pensa and Gianpiero di Blasi. A semi-supervised approach to measuring user privacy in online social networks. In *Discovery Science - 19th International Conference, DS 2016, Bari, Italy, October 19-21, 2016, Proceedings*, pages 392–407, 2016.

[126] Vu Viet Hoang Pham, Shui Yu, Keshav Sood, and Lei Cui. Privacy issues in social networks and analysis: a comprehensive survey. *IET Networks*, 7(2):74–84, 2018.

[127] M. J. D. Powell. Nonconvex minimization calculations and the conjugate gradient method. In David F. Griffiths, editor, *Numerical Analysis*, pages 122–141, Berlin, Heidelberg, 1984. Springer Berlin Heidelberg.

[128] Francesca Pratesi, Anna Monreale, Roberto Trasarti, Fosca Giannotti, Dino Pedreschi, and Tadashi Yanagihara. Prudence: a system for assessing privacy risk vs utility in data sharing ecosystems. *Transactions on Data Privacy*, 11(2):139–167, 2018.

[129] Arthi Ramachandran, Yunsung Kim, and Augustin Chaintreau. "i knew they clicked when i saw them with their friends": identifying your silent web visitors on social media. In *Proceedings of the second ACM conference on Online social networks, COSN 2014, Dublin, Ireland*, pages 239–246.

[130] Shariq Rizvi and Jayant R. Haritsa. Maintaining data privacy in association rule mining. In *VLDB*, pages 682–693. Morgan Kaufmann, 2002.

[131] Andrea De Salve, Paolo Mori, and Laura Ricci. A survey on privacy in decentralized online social networks. *Computer Science Review*, 27:154–176, 2018.

[132] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13(6):1010–1027, November 2001.

[133] Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *PODS*, page 188. ACM Press, 1998.

[134] Yücel Saygin, Vassilios S. Verykios, and Ahmed K. Elmagarmid. Privacy preserving association rule mining. In *RIDE*, pages 151–158. IEEE Computer Society, 2002.

[135] Miguel Nunez del Prado Cortez Sebastien Gambs, Marc-Olivier Killijian. Deanonymization attack on geolocated data. *Journal of Computer and System Sciences*, 80, 2014.

[136] Dongxu Shao, Kaifeng Jiang, Thomas Kister, Stéphane Bressan, and Kian-Lee Tan. Publishing trajectory with differential privacy: A priori vs. a posteriori sampling mechanisms. In Hendrik Decker, Lenka Lhotská, Sebastian Link, Josef Basl, and A. Min Tjoa, editors, *Database and Expert Systems Applications*, pages 357–365, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[137] Chaoming Song, Tal Koren, Pu Wang, and Albert-Laszlo Barabasi. Modelling the scaling properties of human mobility. *Nat Phys*, 6(10):818–823, 10 2010.

[138] Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. Modelling the scaling properties of human mobility. *Nature Physics*, 6:818 EP –, 09 2010.

[139] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Lszl Barabsi. Limits of predictability in human mobility. *Science*, 2010.

[140] Yi Song, Daniel Dahlmeier, and Stéphane Bressan. Not so unique in the crowd: a simple and effective algorithm for anonymizing location data. In *Proceeding of the 1st International Workshop on Privacy-Preserving IR: When Information Retrieval Meets Privacy and Security co-located with 37th Annual International ACM SIGIR conference, PIR@SIGIR 2014, Gold Coast, Australia, July 11, 2014.*, pages 19–24.

[141] Sarah Spiekermann. Privacy enhancing technologies for RFID in retail- an empirical investigation. In *UbiComp*, volume 4717 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2007.

[142] N. Spruill. The confidentiality and analytic usefulness of masked business microdata. *Proceedings of the Section on Survey Research Methods, 1983*, pages 602–607, 1983.

[143] Klara Stokes. On computational anonymity. In Josep Domingo-Ferrer and Ilenia Tinnirello, editors, *Privacy in Statistical Databases*, pages 336–347, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[144] Jens Strüker, Rafael Accorsi, and Günter Müller. On providing one-to-one marketing with customers' privacy in stationary retail. In *CEC/EEE*, pages 44–49. IEEE Computer Society, 2008.

[145] Chong-Jing Sun, Philip S. Yu, Xiangnan Kong, and Yan Fu. Privacy preserving social network publication against mutual friend attacks. *Trans. Data Privacy*, 7(2):71–97, 2014.

[146] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[147] Latanya Sweeney. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, October 2002.

[148] Frank Swiderski and Window Snyder. *Threat Modeling*. O'Reilly Media, 2004.

[149] Chih-Hua Tai, Philip S. Yu, De-Nian Yang, and Ming-Syan Chen. Privacy-preserving social network publication against friendship attacks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1262–1270, 2011.

[150] Pang-Ning Tan, Michael S. Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[151] C. Task and C. Clifton. A guide to differential privacy theory in social network analysis. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 411–417, 2012.

[152] Christine Task and Chris Clifton. A guide to differential privacy theory in social network analysis. In *ASONAM*, pages 411–417. IEEE Computer Society, 2012.

[153] Manolis Terrovitis and Nikos Mamoulis. Privacy preservation in the publication of trajectories. In *MDM*, pages 65–72, 2008.

[154] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. Privacy-preserving anonymization of set-valued data. *Proc. VLDB Endow.*, 1(1):115–125, August 2008.

[155] Frédéric Thiesse, Christian Floerkemeier, and Elgar Fleisch. Assessing the impact of privacy-enhancing technologies for RFID in the retail industry. In *AMCIS*, page 223. Association for Information Systems, 2007.

[156] Vicen Torra. *Data Privacy: Foundations, New Developments and the Big Data Challenge*. Springer Publishing Company, Incorporated, 1st edition, 2017.

[157] Vicenç Torra and Josep Domingo-Ferrer. Record linkage methods for multi-database data mining. 123, 2003.

[158] Slim Trabelsi, Vincent Salzgeber, Michele Bezzi, and Gilles Montagnon. Data disclosure risk evaluation. In *CRiSIS '09*, pages 35–72, 2009.

[159] Roberto Trasarti, Riccardo Guidotti, Anna Monreale, and Fosca Giannotti. Myway: Location prediction via mobility profiling. *Information Systems*, 64:350–367, 2017.

[160] Jayakrishnan Unnikrishnan and Farid Movahedi Naini. De-anonymizing private data by matching statistics. In *51st Annual Allerton Conference on Communication, Control, and Computing, Allerton 2013, Allerton Park & Retreat Center, Monticello, IL, USA, October 2-4, 2013*, pages 1616–1623, 2013.

[161] Hua-jin Wang, Chun-an Hu, and Jian-sheng Liu. Distributed Mining of Association Rules Based on Privacy-Preserved Method. In *2010 Third International Symposium on Information Science and Engineering*, pages 494–497, Shanghai, China, December 2010. IEEE.

[162] Nathalie E. Williams, Timothy A. Thomas, Matthew Dunbar, Nathan Eagle, and Adrian Dobra. Measures of human mobility using mobile phone records enhanced with GIS data. *CoRR*, abs/1408.5420, 2014.

[163] Qian Xiao, Rui Chen, and Kian-Lee Tan. Differentially private network data release via structural inference. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 911–920, New York, NY, USA, 2014. Association for Computing Machinery.

[164] Yabo Xu, Benjamin C. M. Fung, Ke Wang, Ada Wai-Chee Fu, and Jian Pei. Publishing sensitive transactions for itemset utility. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 1109–1114, 2008.

[165] Yabo Xu, Ke Wang, Ada Wai-Chee Fu, and Philip S. Yu. Anonymizing transaction databases for publication. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 767–775, 2008.

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

[167] Xun Yi, Fang-Yu Rao, Elisa Bertino, and Athman Bouguettaya. Privacy-preserving association rule mining in cloud computing. In *AsiaCCS*, pages 439–450. ACM, 2015.

[168] Liu Ying-hua, Yang Bing-ru, Cao Dan-yang, and Ma Nan. State-of-the-art in distributed privacy preserving data mining. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 545–549, Xi'an, China, May 2011. IEEE.

[169] Hui Zang and Jean Bolot. Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 145–156, New York, NY, USA, 2011. ACM.

[170] Justin Zhan, Stan Matwin, and LiWu Chang. Privacy-preserving collaborative association rule mining. *Journal of Network and Computer Applications*, 30(3):1216–1227, August 2007.

[171] Peng Zhang, Yunhai Tong, Shiwei Tang, and Dongqing Yang. Privacy Preserving Naive Bayes Classification. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Xue Li, Shuliang Wang, and Zhao Yang Dong, editors, *Advanced Data Mining and Applications*, volume 3584, pages 744–752. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[172] Yu Zheng. Trajectory data mining: An overview. *ACM TIST*, 6(3):29:1–29:41, 2015.

[173] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.*, 5(3):38:1–38:55, September 2014.

[174] Yu Zheng and Xiaofang Zhou, editors. *Computing with Spatial Trajectories*. Springer, 2011.

[175] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE*, pages 506–515. IEEE Computer Society, 2008.