Modeling Adversarial Behavior Against Mobility Data Privacy

Roberto Pellungrini, Luca Pappalardo, Filippo Simini, and Anna Monreale

***Abstract*— Privacy risk assessment is a crucial issue in any privacy-aware analysis process. Traditional frameworks for privacy risk assessment systematically generate the assumed knowledge for a potential adversary, evaluating the risk without realistically modelling the collection of the background knowledge used by the adversary when performing the attack. In this work, we propose Simulated Privacy Annealing (SPA), a new adversarial behavior model for privacy risk assessment in mobility data. We model the behavior of an adversary as a mobility trajectory and introduce an optimization approach to find the most effective adversary trajectory in terms of privacy risk produced for the individuals represented in a mobility data set. We use simulated annealing to optimize the movement of the adversary and simulate a possible attack on mobility data. We finally test the effectiveness of our approach on real human mobility data, showing that it can simulate the knowledge gathering process for an adversary in a more realistic way.**

***Index Terms*— Data privacy, privacy, agent-based modeling.**

I. Introduction

MOBILITY data are some of the most sought after commodity in the data analysis landscape today. With the pervasiveness of location-based services and the widespread use of mobile devices, mobility data are more abundant than ever. However, in light of national and international data privacy regulations, such as the European General Data Protection Regulation (GDPR), protecting data from privacy breaches has become a major concern, affecting all kinds of data across several fields. Privacy-preserving solutions usually modify or transform the original data to mask individuals and protect them, thus changing the characteristics of the original data set. Therefore, the challenge in designing privacy protection methods is to achieve privacy for as many individuals as possible while preserving the quality of the data, allowing meaningful analyses.

The scientific literature has proposed several tools to quantify the risk of privacy violations for the individuals represented in a mobility data set. A standard approach is to consider a *worst-case scenario* in which a malicious adversary has the maximum knowledge on any individual in the data and performs the smartest attack against them. For example, given an individual's spatio-temporal trajectory covering one month, worst-case approaches assume that the adversary knows the whole month's worth of locations and tries to identify the individual in the shared mobility dataset. This assumption led to the definition of several privacy-preserving algorithms such as differential privacy, randomization, and k-anonymity [1]–[5]. These algorithms transform the data in such a way to guarantee certain thresholds on the risk of privacy leaks. Unfortunately, the low data quality resulting from the application of the privacy transformation often inhibits the use of the mobility data. This situation is referred to as the "tragedy of data commons" [6]. Either for fear of disclosing sensitive information or because of the lack of mutual trust, we may end up misjudging privacy risk (either overestimating or underestimating the risk), make improper use of the data, or give up on them completely.

The framework used in [7] and [8] tries to mitigate the issue above performing the systematic assessment of empirical privacy risk concerning specific attacks on mobility data. In practice, the framework simulates an adversary that, for each individual, possesses the knowledge maximizing the privacy risk of that individual. To this end, the framework generates all the possible background knowledge that the adversary may know, and assesses the risk with respect to the worst one. Although this framework has advanced the state of the art considerably, it does not model the process of background knowledge gathering in a realistic way.

In this article, we propose a data-driven approach to realistically simulate the behavior of a malicious adversary in the acquisition of background knowledge for privacy attacks in mobility data. First of all, we assume that the malicious adversary collects information about the attacked individuals during their movements while satisfying the natural spatial and temporal constraints of human mobility [9], [10]. Then, we present three possible alternatives: the adversary is one of the real individuals in the data set (real adversary); the adversary is a synthetic individual that moves realistically (synthetic adversary); the adversary moves in such a way to produce the greatest damage to the privacy of individuals in the data set (simulated adversary). We implement the third alternative by designing a Simulated Privacy Annealing algorithm (SPA) based on an optimization meta-heuristic that generates the adversary's movements that maximize the average privacy risk of the individuals in the data set. We show, on large-scale

Manuscript received January 9, 2020; revised June 5, 2020 and July 30, 2020; accepted August 13, 2020. This work was supported in part by the European Project SoBigData++ RI under Grant 871042. The Associate Editor for this article was J. Blum. *(Corresponding author: Roberto Pellungrini.)*

Roberto Pellungrini and Anna Monreale are with the Department of Computer Science, Faculty of Natural Science and Mathematical Physics, University of Pisa, 56100 Pisa, Italy (e-mail: roberto.pellungrini@di.unipi.it).

Luca Pappalardo is with ISTI, CNR, 56124 Pisa, Italy.

Filippo Simini is with the Faculty of Engineering, University of Bristol, Bristol BS8 1TR, U.K.

Digital Object Identifier 10.1109/TITS.2020.3021911

mobility data, that SPA provides more realistic estimates of the privacy risk for individuals than traditional approaches, also generating an average privacy risk higher than the most efficient real and synthetic adversaries. Our results show that SPA also gives a robust upper bound to the risk that some adversary may produce for the individuals in a dataset. The approach we present in this article can be applied to the simulation of any privacy attack on mobility data based on background knowledge and can be used by a data owner to understand which individuals have a high privacy risk under these new assumptions.

The paper is organized as follows. In Section II we give a brief overview of the literature on the topics relevant to our work, in Section III we provide the mathematical definitions we use in our work, in Section IV we introduce the principles of traditional privacy risk assessment frameworks, in Section V we state our approach and we outline the different scenarios for our simulations in Section VI. In Section VII, we comment on our results. Finally, in Section VIII, we summarize our findings and give some hints for future developments.

II. Related Works

A. Privacy Risk and Mobility Data Privacy

An overview of the techniques and methodologies concerning urban mobility can be found in [11]. Human mobility data contains sensitive information and can be used to disclose private details of the lives of the individuals involved. Many privacy-preserving techniques for human mobility data have been proposed in literature [12]. For example, Giannotti amd Pedreschi [3] summarize the best practices in handling geo-located data and the standard privacy-preserving methodologies that can be applied to human mobility data. k-anonymity [1], [2] states that an individual should be indistinguishable from a group of at least $k-1$ other individuals, based on their quasi-identifiers attributes. A generalized methodology to achieve k-anonymity can be found in [13]. Poulis *et al.* [14] propose an apriori algorithm to achieve k-anonymity for trajectory data. Another widely used privacy-preserving model is differential privacy [15], which limits the impact of data aggregation algorithms on the privacy of individuals. For example, Monreale *et al.* [16] propose applying a ϵ-differential privacy model for movement data. Cavoukian and Emam [17] 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 an adversary can link the de-identified or otherwise protected data of an individual with some information available to them. In the literature, there are two main ways to measure the risk of re-identification:

- *File-level risk assessment*: risk is defined as the proportion of records that an adversary can re-identify out of the whole set of records they have [18];
- *Individual risk assessment*: risk is defined as the probability that a particular record of the adversary is recognized as corresponding to a particular individual in the data. This comes from the intuition that risk is not homogeneous in a data set, and that rare combinations of attributes may lead to the re-identification of individuals [19].

Recently works have improved on the techniques used in privacy risk assessment for mobility data. Basu *et al.* [20] propose an empirical model for the estimation of privacy risk for trajectory data. Pratesi *et al.* [7] propose a generalized privacy risk assessment framework applicable to any data. Risk is assessed based on the k-anonymity principle by systematically evaluating all the possible background knowledge of an adversary with a combinatorial worst-case approach. Pellungrini *et al.* provide the definition of a large number of attacks on mobility data [8], alongside an accurate classification approach that can reduce computational time significantly required for privacy risk assessment. Kondor *et al.* [21] present a large-scale analysis of user matchability in real mobility datasets, effectively linking mobility data based on co-occurrence, a premise similar to our definition of *colocation*.

In some existing works, attacks in mobility data are simulated using de-identification algorithms based on some learning phase. In [22], the authors propose Bayesian approaches where users are classified by the frequency of their visits. In contrast, in [23] the authors learn Markov Chains models from the data before linking the knowledge of the attacker to the data. This approach ultimately produces an estimation of the probability with which an individual can be re-identified by a particular attack. The focus of our methodology is slightly different: we focus on empirical privacy risk, i.e., the direct matching of the knowledge that an adversary may possess with the data. In particular, we propose a novel method of simulating the construction of such knowledge.

B. Generative Models of Human Mobility

The generation of synthetic trajectories that capture the salient characteristics of real mobility data is a topic of growing interest. Mobility data suitable for specific purposes may not be readily available or may not be safe to share. Therefore, several models have been proposed to generate such data synthetically. 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 [24]. 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 cannot capture the temporal regularities of human mobility realistically. Several improvements have been proposed on the EPR model, such as d-EPR [25], 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 [26], a modelling framework for generating synthetic trajectories. DITRAS separates the generative procedure into 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.

C. Simulated Annealing

Simulated annealing is a metaheuristic to approximate global optimum for optimization problems. It is used for problems with vast search spaces. Simulated annealing is an adaptation of the Metropolis-Hastings algorithm [27], which is a Monte Carlo algorithm used for the generation of sample states of a thermodynamic system, such as, for example, [28]. Simulated annealing has been applied to human mobility problems before, for example, in [29], where the algorithm is used to tackle traffic jams by dynamically calculating optimal traffic routes. Simulated annealing requires several parameters, like the cooling schedule. Such parameters are application-specific. However, general guidelines exist to guide in the selection process, such as [30] for the cooling schedule, or [31], which gives a general procedure to compute the initial temperature of the simulated annealing.

III. Data Definitions

In this article, we focus on vehicular mobility data, i.e., trajectories from private vehicles. From now on, we will refer to this kind of data as individual mobility data. A trajectory is a sequence of records that identifies the movements of an individual during a period of observation [32]–[35]. 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, with the term *point*, we refer to a single element of a trajectory, while with the term *location* we refer to the point's spatial information. We denote by $U_{set} = \{u_1, \ldots, u_m\}$ the set of the distinct individuals represented in the mobility data set 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.

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

Definition 3 (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 $Ts_{set} = \{ts_1, \ldots, ts_v\}$ the set of all different time slots in a data set 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 the longest staying period in the time slot is selected as the location associated with that time slot [26]. We can then represent a mobility dataset D as a matrix:

Definition 4 (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 visualize better which individuals stayed roughly in the same place and at the same time, and it also allows us to simulate better the behavior of an adversary for a privacy attack. The trajectory T^u of individual u is made of all the elements $m_{iju} \forall (i, j)$ in matrix M.

IV. Privacy Risk Assessment

In the literature [1], [2], [7], most of the methodologies of privacy risk assessment assume that the simulation of a privacy attack takes place in two phases. In phase 1, the malicious adversary gathers, in some way, a *background knowledge* about an individual's movements (e.g., a fragment of their trajectories). In phase 2, the malicious adversary uses the acquired background knowledge to re-identify the records of the individual in a mobility dataset. Formally, the conceptual framework of privacy attacks relies on the following definitions:

Definition 5 (Background Knowledge): A background knowledge BK 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, the adversary background knowledge BK can be considered as an *adversary trajectory* denoted by T^a.

Definition 6 (Background Knowledge Instance): A background knowledge instance B^u is a specific set of spatio-temporal points known by the adversary about an individual u. Formally, we can represent it as a 2-dimensional matrix where $\forall (i, j) B^u_{ij} = 1$ if the adversary knows that the specific individual u was at the location i during the timeslot j.

Given a set of m individuals, we denote by B the 3-dimensional mobility matrix representing the background knowledge instances of the adversary of all m individuals.

In this article, we consider a scenario in which a malicious adversary successfully gathers some points belonging to several identities and then tries to match these points against a data set. The gathered points compose the background knowledge of the adversary. The points in the background knowledge belonging to one specific individual compose the background knowledge instance of the adversary for that individual.

Example 1: Let us consider an adversary with trajectory (or background knowledge) $T^a = \{(l_1, ts_1), (l_2, ts_2),$

$(l_3, ts_3), (l_4, t_4)\}$ and an individual with trajectory $T^u = \langle (l_1, ts_1), (l_4, ts_2), (l_2, ts_3), (l_3, ts_4)\rangle$. We can represent them by the matrices:

$$T^a = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_3 \end{array} \begin{array}{c} \begin{array}{cccc} ts_1 & ts_2 & ts_3 & ts_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$T^u = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{cccc} ts_1 & ts_2 & ts_3 & ts_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

For that individual the background knowledge instance of the adversary is $\{(l_1, ts_1)\}$ (i.e, only $B^u_{l_1 ts_1} = 1$) because this is the only point in the adversary trajectory that also belongs to the trajectory of the individual, i.e., $T^a_{l_1,ts_1} = T^u_{l_1,ts_1} = 1$.

A *re-identification attack* is the process with which an adversary compares their knowledge to some mobility record. A re-identification attack can be expressed mathematically as a matching function $matching(T_w, B_u)$, which indicates whether or not a trajectory T^w matches the instance of background knowledge B^u. A match indicates that, for the adversary, the points in B^u are associated with T^w, and therefore that the two identities w and u might be the same. We assume that, in this attack, the adversary uses both the spatial and temporal components of each point. We can formalize the associated matching function as:

$$matching(T^w, B^u) = \begin{cases} true & \forall (i,j) \in B^u, \exists (i,j) \in T^w | \\ & m_{ijw} = B^u_{ij} = 1 \\ false & \text{otherwise} \end{cases}$$

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 7 (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 mobility 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 M 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 each $(i, j) \in B^u$, an individual $z \neq u$ has $m_{ijz} = 1$ in M, then the individual shares all the points of u in the adversary's background knowledge instance. Algorithm 1 details the procedure that calculates the privacy risk for a given individual u.

Given each individual's *privacy risk* in a mobility dataset, we define the average risk produced by an adversary as:

Definition 8 (Average Adversary Risk (AAR)): Given the set of individuals U_{set} in mobility dataset M, and $Risk(u, B^u, M)$, the Average Adversary Risk (AAR) is the average risk produced by the adversary:

$$\text{AAR}(u, B^u, M) = \frac{\sum_{u \in U_{set}} Risk(u, B^u, M)}{|U_{set}|}.$$

Algorithm 1 Algorithm to Compute Risk for User u Against Mobility Dataset Matrix M

```
input : Background knowledge instance B^u, mobility
        dataset matrix M
output: Privacy risk for individual u
S ← ∅; T ← ∅; F ← 1;
for (i, j) ∈ B^u do
    for (z) ∈ m_{i,j} do
        if m_{i,j,z} == 1 then
            T ← T ∪ z;
    if F == 1 then
        S ← T; F ← 0;
    else
        S ← S ⋂ T; T ← ∅;
risk ← 1/|S| return risk
```

V. Problem Statement

In the literature, risk assessment methodologies aim at evaluating the privacy risk of each individual in a data set 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 data set. 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., [7]) generate all the possible background knowledge that a malicious adversary may gather about an individual. They compute a re-identification probability for each background knowledge and define the individual's re-identification risk 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 data set. 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 they are 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 spatial and temporal constraints of human mobility must be taken into account during the process of background knowledge construction.

In this article, we explore possible realistic ways to model the acquisition of the background knowledge by an adversary, taking into account the spatial and 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 data set. 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. Modelling the behavior of a malicious adversary as an adversary trajectory is an approach that completely departs from the literature. Traditional risk assessment methodologies build the background knowledge abstractly, i.e., 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). To formalize how the adversary gathers information through the trajectory, we use the concept of *co-location*:

Definition 9 (Co-Location): Let (x, y, t) and (x', y', t') be two points of two trajectories T^u and T^w respectively. The two points are considered a *co-location* if $(x = x' \wedge y = y' \wedge t = t')$. We denote by $C_{u,w}$ the set of all co-locations between trajectories T^u and T^w.

Intuitively, a co-location 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 co-locates 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 co-location. In other words, given the adversary trajectory T^a and the individual trajectory T^u, the background knowledge instance B^u is given by intersecting the adversary and individual trajectories, that can be computed by the element-wise product between the two matrices T^a and T_u, i.e., $B^u = T^a \circ T^u$. As stated in Section III, when we discretize time into timeslots, whenever two locations end up in the same timeslot for a particular individual, we maintain only the location with the longest stay time. Therefore, in our context, it is easy to apply the concept of co-location: the mobility matrix M indicates when two individuals are in the same location at the same time slot. Based on acquired background knowledge, we simulate a re-identification attack in which the malicious adversary tries to match the points gathered about any individual in the mobility data set. We finally compute the privacy risk of each individual using Algorithm 1 (Section IV). 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 2: Let us consider a 3-dimensional mobility matrix M containing the trajectories of three users $\{u_1, u_2, u_3\}$:

$$T^{u_1} = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{ccc} ts_1 & ts_2 & ts_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \quad T^{u_2} = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{ccc} ts_1 & ts_2 & ts_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

$$T^{u_3} = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{ccc} ts_1 & ts_2 & ts_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

Given an adversary with the following trajectory:

$$T^a = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{ccc} ts_1 & ts_2 & ts_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

the co-locations between the adversary trajectory and the dataset D, is computed by performing an element-wise product between the two matrices. The resulting background knowledge instance B is the 3-dimensional matrix composed of the following individual matrices:

$$B^{u_1} = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{ccc} ts_1 & ts_2 & ts_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \quad B^{u_2} = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{ccc} ts_1 & ts_2 & ts_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

$$B^{u_3} = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \end{array} \begin{array}{c} \begin{array}{ccc} ts_1 & ts_2 & ts_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

Based on the background knowledge instance B, we evaluate the privacy risk produced by the adversary using a matching function and counting, for each individual, how many other individuals match the points in B. For example, points $B_{(l_1 ts_1 u)}, B_{(l_2 ts_3 u)}$ have value 1 only for $u = u_1$, generating a privacy risk of 1; while points $B_{(l_4, ts_2, u)}$ have value 1 for $u = u_2$ and $u = u_3$, the privacy risk is equal to $\frac{1}{2}$.

VI. Construction of the Adversary Trajectory

We can construct an adversary trajectory in several ways. We consider three possibilities: using the trajectory of a real individual, generating a realistic synthetic trajectory, or constructing a principled adversary trajectory. These three types of trajectories mimic different potential behaviors of an adversary. In the first case, we represent the scenario in which the adversary may be one of the individuals in the data. In the second case, we generate a synthetic adversary trajectory able to reproduce the fundamental mobility patterns. The goal is to simulate an adversary who is a person moving similarly to the others but is not represented in the data. Finally, the simulated trajectory is obtained by applying an optimization method to simulate the case in which the adversary tries to maximize its ability to re-identify other individuals in the data.

A. 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 data set. 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 data set M the adversary trajectory leading to the maximum privacy risk for individuals represented in M. For each real individual in the data set M, we use the following approach: *(i)* we consider their trajectory as 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 the dataset as average over the individual privacy risks, i.e., AAR (Definition 8). 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 T^a (background knowledge) and an individual u in M. First, the approach constructs the adversary's background knowledge instance B^u, composed of the co-locations between T^a and the trajectory of the individual u. Then, it computes the privacy risk of u applying the $Risk(u, B^u, M)$ function (Definition 7).

B. 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 [26], [36]. 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 data set. Then, the privacy risk assessment process identifies in the synthetic data set the adversary trajectory leading to the maximum privacy risk for individuals in M. For each individual in the synthetic data set, 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 8). Finally, we return the privacy risk evaluation related 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.

C. 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. They represent 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 specific goal to maximize the attack success against the set of individuals represented in the mobility data set. 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 splits 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. A brute force approach that computes all possible adversary trajectories is computationally unfeasible for such an ample search space.

We overcome this computational problem by proposing an algorithm called Simulated Privacy Annealing (SPA). It is a method based on simulated annealing, an optimization meta-heuristic that is an adaptation of the Metropolis-Hastings algorithm [27]. Simulated annealing is a flexible procedure and has been adapted to many different problems, including urban mobility problems [29]. We chose simulated annealing for its inherent characteristics of adapting well to problems with ample space of the solutions and the capacity to escape local optimum [37]. 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 mechanism allows simulated annealing to avoid local minimums and to converge to near optimality, given that it explores enough solutions [38].

Algorithm 2 Simulated Annealing

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

1 $Temp \leftarrow Temp_{init}$; $S \leftarrow S_0$; $S_{best} \leftarrow S_0$;
2 **while** $stopping\_criteria()$ is $false$ **do**
3 $Temp \leftarrow cooling\_schedule(Temp)$;
4 $S_{new} \leftarrow neighbor(S)$;
5 **if** $P(E(S), E(S_{new}), Temp) \geq random(0, 1)$ **then**
6 $S \leftarrow S_{new}$;
7 **if** $E(S) > E(S_{best})$ **then**
8 $S_{best} \leftarrow S$;
9 **return** S_{best}

Algorithm 2 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 2). 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 higher 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 2, 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 8 as an objective function.

Simulated annealing is a minimization metaheuristic. So, to correctly model our problem, $E(S)$ will be $1 - \text{AAR}$ since mean risk has an upper bound of 1. We denote with $F_{\text{AAR}}(T, M)$ the function that, given the adversary trajectory T and a Mobility Matrix M computes $1 - \text{AAR}$ over the individuals in M. So our objective function becomes simply: $F_{\text{AAR}}(T, M)$. 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 [30]: 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 [39]. Whereas in the literature the value of α is generally set somewhere between 0.95 and 0.99, in our experiments described in Section VII we explore a broader range of values. The initial temperature is usually selected in a way that the initial acceptance probability is close to a specific initial value, traditionally 80%. Ben-Ameur *et al.* in [31] propose a simple procedure to calculate the initial temperature. For our purposes, having a vast space of solutions, we select an initial temperature such that the initial acceptance probability would be 90%. This is done by running the annealing procedure for a small number of iterations, adjusting the temperature in the process. Regarding the stopping criteria, two common 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 the following approach instead: we run the algorithms at intervals of a 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, the 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, 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 (SPA) process 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 co-locations and AAR.
4) **Acceptance probability**: we either accept or reject the neighboring solution based both on the evaluation and on the 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.

Algorithm 3 shows the Simulated Privacy Annealing (SPA) process. We show in Algorithms 4, and 5 how we implemented the stopping criteria and neighboring function, respectively.

VII. Experiments

A. Data Set of Real Trajectories

We use mobility data provided by Octo Telematics describing the GPS tracks of private vehicles travelling in Tuscany during May 2011. When a vehicle is turned on, the GPS device

Algorithm 3 Simulated Privacy Annealing (SPA)

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**
  $Temp \leftarrow \alpha Temp$;
  $T_{new} \leftarrow neighbor(T, lm)$;
  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 4 stopping_Criteria

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

$bool \leftarrow False$; $constant \leftarrow 10$;
$steps_n \leftarrow |M|/constant$; **if** $steps \ \% \ steps_n == 0$ **then**
  if $(T$ $changed \vee T_{best}$ $changed))$ **then**
    $bool \leftarrow True$;
return $bool$

Algorithm 5 Neighbor

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

$point \leftarrow random\_choice(T)$;
$new\_point \leftarrow neighbor\_point(lm)$;
$T_{new} \leftarrow (T)$; $T_{new}(point) \leftarrow new\_point$;
return T_{new}

embedded in it starts registering the information about the vehicle's position every 30 seconds. When the vehicle stops, no points are logged nor sent. We use these stops to split each vehicle's GPS track into sub-tracks, obtaining all the trips performed by the vehicles. To recognize and eliminate small stops, such as traffic lights and traffic jams, we follow the strategy commonly used in literature [40], [41], ignoring stops shorter than 20 minutes. We further split the GPS tracks into urban areas, each pertaining to cities in Tuscany, spanning from small/medium size cities to large urban areas. We thus obtained five data sets corresponding to the cities of Florence, Pisa, Livorno, Siena and the urban area comprising Pistoia and Prato. For each of the five data sets, 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 III. Table I summarizes the characteristics of our data sets.

TABLE I
SUMMARY OF THE CHARACTERISTICS OF THE FIVE DATA SETS

City	Trajectories	Total stops	Mean stops per individual
Pisa	3281	54,295	16.548308
Siena	3463	90,850	26.234479
Prato_Pistoia	8651	275,729	31.872500
Livorno	2068	28,507	13.784816
Florence	9296	143,040	15.387263

B. Generation of Synthetic Trajectories

We use DITRAS [26], [42] to generate the synthetic trajectories needed for the analysis of the risk produced by a synthetic adversary. We run DITRAS using the spatial tessellation of Tuscany and its origin-destination matrix. Having roughly 50,000 trajectories in the original data set, we simulate the trajectories 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.

C. Experimental Results

For two of the three scenarios, 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, the number of possible adversaries is equal to the number of real trajectories. To understand how SPA 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 generate random adversary trajectories by selecting, for each timestamp, a random location. We generate as many random adversary trajectories as the number of real and synthetic adversary trajectories. Figure 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. 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 to gather truly damaging background knowledge, a malicious adversary would need to move in a much different way than real individuals or *likely* synthetic individuals. Another interesting observation is that the difference between the simulated adversary's AAR and the AAR of the real, synthetic and random adversaries decreases as the size of the data set increases. For example, Florence and Prato-Pistoia have a much lower AAR than Pisa and Leghorn, suggesting that

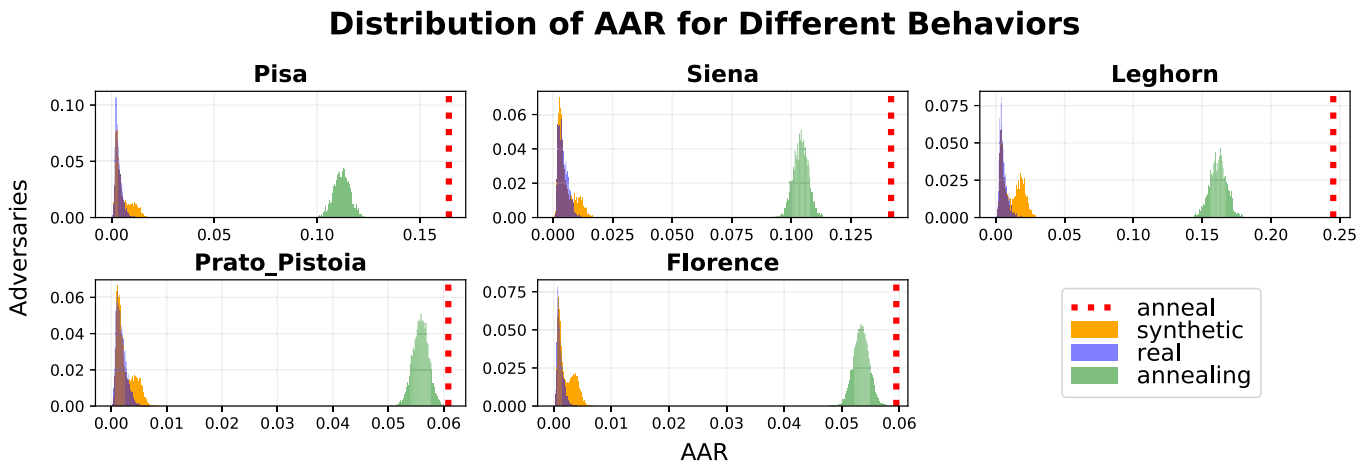


Fig. 1. Distribution of Average Adversary Risk (AAR) for real and synthetic adversaries compared to the AAR of a simulated adversary. In blue, we see the AAR for real adversaries. In orange, we see the same value for synthetic adversaries. In green, we see the AAR for randomly generated adversaries. The vertical red line indicates the AAR for the simulated adversary.

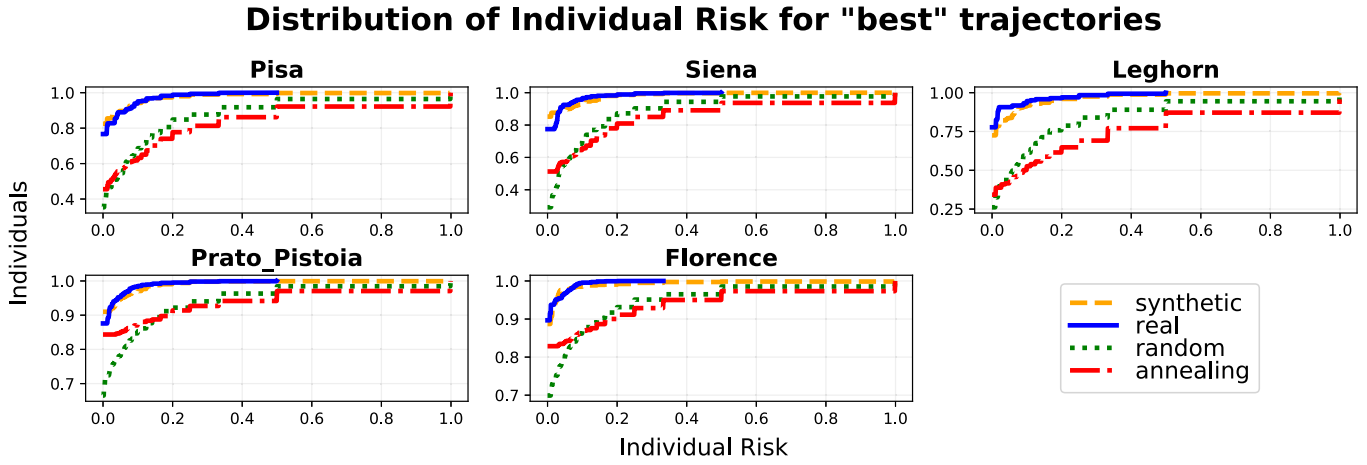


Fig. 2. Cumulative distribution of privacy risk for individuals attacked by the best adversary for the three scenarios: real, synthetic, and simulated. In blue, the cumulative distribution of privacy risk for the best real adversary. In orange, the same value for the best synthetic adversary. In green, the cumulative distribution of privacy risk for the best randomly generated adversaries.

the optimization is more effective in smaller areas. Indeed, whereas an individual is more likely to be hidden in the crowd in large data sets, in small data sets, the same individual may be easier to attack.

We then look at how privacy risk distributes over the individuals under attack. To do this, we select the best adversary trajectory for each of the three scenarios introduced in section V. 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 simulation (T_{sim}) using SPA as explained in Section VI-C.

Figure 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 $(1/2, 1/3, \ldots)$. The cumulative distribution of risk is 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 lacking. Again, we observe that the simulated adversary T_{sim} presents a lower cumulative distribution of privacy risk than T_{real}, T_{synth}, and the random baseline. The difference in overall risk decreases as the dimension of the data set increases. These results show that SPA generates an adversary trajectory with an AAR higher than any other possible adversary, be it real, synthetic, or random. Overall, for bigger data sets, we have lower levels of privacy risk because trajectories move over a more sparse and vast territory. In other words, the bigger the territory, the harder it is for an adversary to pose a threat to individuals' privacy represented in the data set.

Our results highlight the differences in the three approaches for the simulation of the adversary trajectory. An individual moving like one of the individuals represented in the data

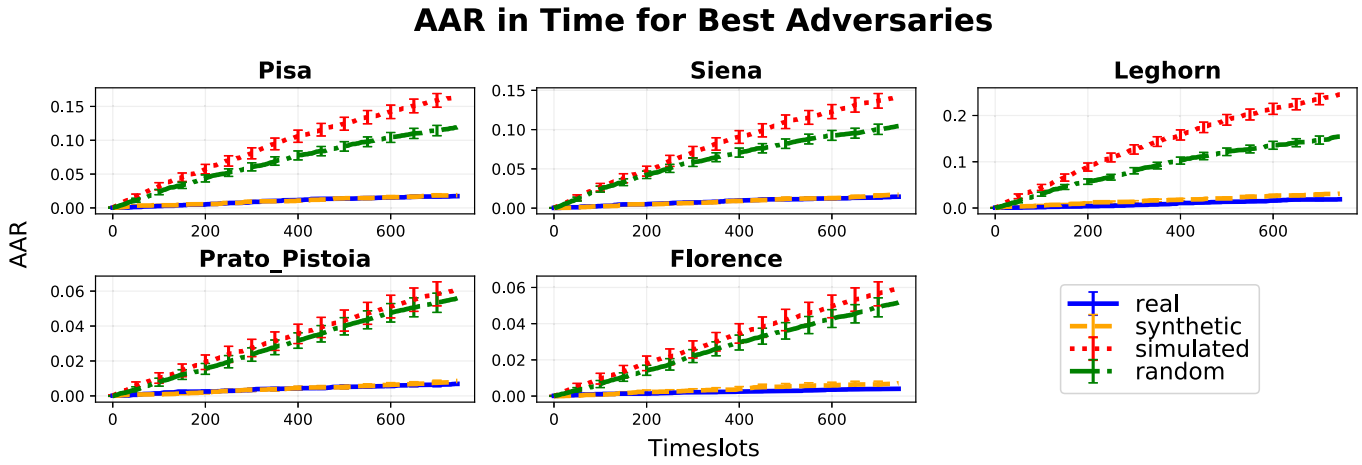


Fig. 3. Variation in time of Average Adversary Risk (AAR) for the most effective attackers of each scenario. The risk is calculated as time goes by and the trajectory of the corresponding adversary grows. In blue, we see the average produced risk for the most effective real adversary. In orange, for the most effective synthetic adversary, in red, for the simulated adversary, and in green, for an adversary generated with a completely random movement.

poses a little privacy threat. A synthetic trajectory could represent an adversary that is "crossing" the real trajectories. Since synthetic trajectories are generated through a probabilistic method, we see more erratic movements and, therefore, a higher adversary risk. The simulated trajectory, engineered to maximize average risk, produces the highest privacy threat.

D. 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 ones of real and synthetic adversaries throughout all time slots and regardless of the observation period (Figure 3). Hence, the simulated adversary is an upper bound for AAR, meaning that it is the worst possible single adversary for a mobility data set. Moreover, we generate a random trajectory and compare the resulting AAR with the one produced by a simulated adversary. We find that, while significantly higher than real or synthetic adversaries, a random trajectory does not yield the same risk as a simulated trajectory obtained explicitly to maximize average risk.

As Table II shows, the trajectory of the best simulated adversary (T_{sim}) has a peculiar structure that significantly differs from the structure of trajectories of the real (T_{real}) and synthetic (T_{synth}) adversaries. In T_{sim}, the mover changes location at every time slot, visiting many 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 used by existing privacy risk assessment frameworks, and while producing the highest AAR, generates an adversary trajectory that is inconsistent with real human mobility trajectories. As Figure 3 shows, although the trajectory obtained with simulated annealing may seem random, randomly generated trajectories do not produce the same risk as a simulated one. Figure 4 reports a visualization of the different best adversary trajectories.

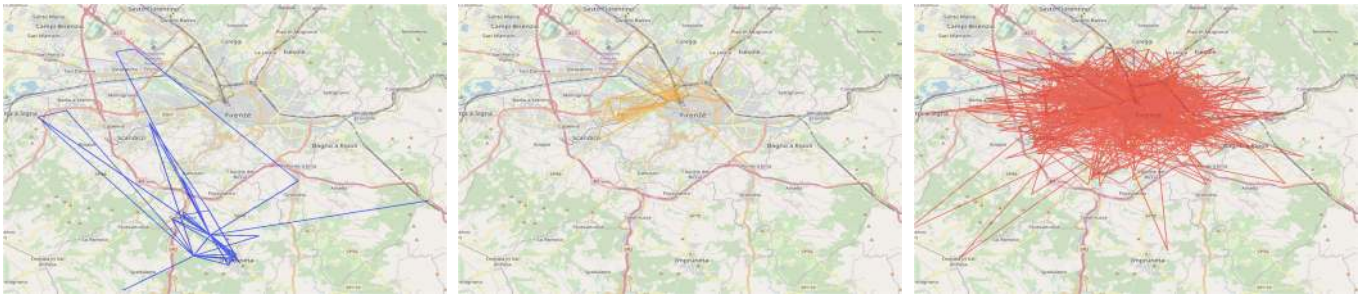
TABLE II
MOBILITY ANALYSIS OF THE BEST REAL AND SYNTHETIC ADVERSARIES IN COMPARISON WITH THE SIMULATED ADVERSARY

Area	Metric	Real	Synthetic	Simulated
Florence	number of trips	96	109	744
	mean distance	3.53	1.64	4.54
	unique locs	24	34	590
	entropy	2.19	3.71	9.10
	radius	2.76	1.80	3.75
Pisa	number of trips	113	100	744
	mean distance	4.70	2.20	3.73
	unique locs	49	41	426
	entropy	4.15	3.10	8.56
	radius	3.75	2.03	3.06
Leghorn	number of trips	137	115	744
	mean distance	6.83	2.07	5.675
	unique locs	47	48	587
	entropy	2.73	4.46	9.09
	radius	4	4.24	5.03
Leghorn	number of trips	96	109	744
	mean distance	3.52	1.64	4.53
	unique locs	24	34	590
	entropy	2.19	3.71	9.10
	radius	2.76	1.80	3.75
Siena	number of trips	50	91	743
	mean distance	2.95	2.06	3.28
	unique locs	32	31	416
	entropy	4.31	3.26	8.51
	radius	2.28	2.57	2.69

E. Performance Analysis of Simulated Annealing

We find that SPA is robust with respect to both the limits we impose on the adversary movements and the cooling rate used to decrease the temperature (Figure 5).

Regarding the cooling rate, we test values ranging from 0.90 to 0.98. This relatively low decreasing rate allows 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



(a) Florence real trajectory (b) Florence synthetic trajectory (c) Florence simulated trajectory

(d) Pisa real trajectory (e) Pisa synthetic trajectory (f) Pisa simulated trajectory

Fig. 4. Visualization of the worst adversary trajectories for the three scenarios. In blue real trajectories, in orange synthetic trajectories, in red simulated trajectories.

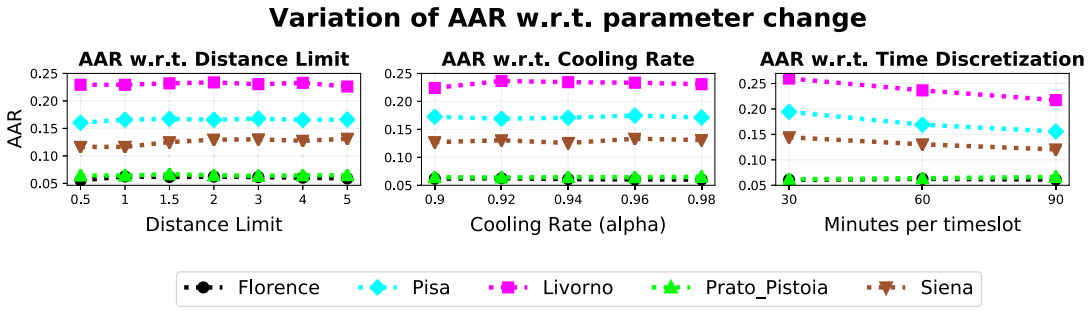


Fig. 5. Variation of Average Adversary Risk (AAR) by distance limit and exponential cooling rate, for all data sets. Both parameters do not have a strong impact on SPA's results.

kilometers to 5 kilometers. These are relatively strict limits, considering that in an urban area and 1 hour, an agent can potentially cover a greater distance. We find that, for both urban areas and varying the distance limit, the risk produced by the simulated adversary remains stable.

Since smaller timeslots imply more detailed trajectories and more precise information for the simulated adversary, small timeslots lead to high AAR values overall. Time discretization helps in "grouping" individuals in the same places at identical timeslots. Therefore, the smaller the dimension of the timeslots, the more timeslots we need to cover the entire timeframe of the analysis, and the fewer individuals are hidden within each other's movements. This effect is evident in large data sets, while for small urban areas, we observe little differences. Moreover, small timeslots slow down SPA, as shown in Table III. For the three smallest datasets (Siena, Prato Pistoia, and Leghorn), the difference in runtime is small, and performances are still good. SPA takes much longer for small discretizations in the two largest datasets (Florence and Pisa).

TABLE III
RUNTIME OF SPA VARYING THE SIZE OF THE TIMESLOTS

Dataset	30min	60min	90m
Florence	51307s	15716s	11804s
Pisa	8558s	1754s	1627s
Leghorn	7521s	3794s	1335s
Prato Pistoia	51693s	20421s	11493s
Siena	10774s	5157s	2348s

In Figure 6, we investigate the evolution of the risk produced by the simulated adversary's trajectory in time. SPA requires roughly 20 minutes for the small datasets (Pisa, Leghorn, Siena), and more than two and a half hours for the large datasets (Florence, Prato/Pistoia). For the large data sets, the improvement emerges early in the annealing process; for the smaller data set, the improvements spread evenly during the runtime of the procedure. This useful information can be exploited by an analyst to understand when the annealing process can be stopped and to adjust the stopping criteria if time is a constraint in risk analysis.

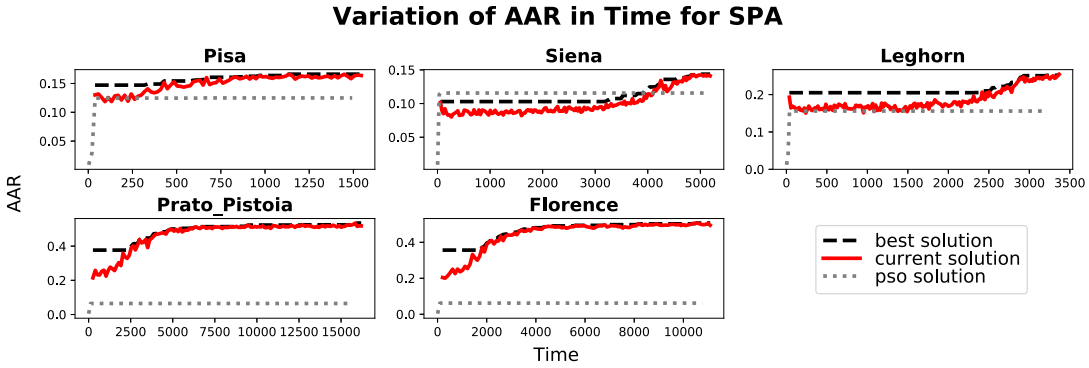


Fig. 6. Variation of the Average Adversary Risk (AAR) of the current solution and the best solution in time.

To enhance our contribution, we compared SPA with the Particle Swarm Optimization approach (PSO). PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution [43]. PSO modifies a starting group, called swarm, of randomly selected feasible solutions, called particles, by evaluating their objective function, and apply a translation that "moves" the particles of the swarm towards the position of the best particle in terms of the objective function in the swarm. The topology of the swarm, i.e., how the single particles are compared with each other, influences the algorithm's performances. Following existing approaches that apply PSO to discrete or semi-discrete problems [44], [45], we use a circular topology that helps escape local optimum, and we modify the translation formula of the particles to suit our specific problem better. To ensure feasibility, we impose the same constraints of travelled distance we use for SPA. We run PSO for the same time that SPA needed to reach termination, with a number of particles equal to 10% of the size of the data set, and compare the results in Figure 6. SPA is more effective than PSO in improving on the initial solution. PSO fails to escape local optima and that, after a small number of iterations, stops improving on the solutions, reaching a plateau. This behavior can be attributed to the comparative nature of PSO, which modifies solutions to move them towards the best one available and applies limited randomness to this movement. In contrast, SPA allows for more exploration of less than optimal solutions, thus exploring the space more effectively.

F. Discussion

We simulate a potential adversary's movement in different ways, generating realistic background knowledge. Our results show that SPA provides a robust evaluation of the privacy risk that an adversary can cause: the AAR obtained with SPA is significantly 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 existing frameworks. Although SPA complies with the natural spatial and 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 4f). SPA is stable over the input parameters: both the distance limit and the cooling rate do not significantly impact the performance of the simulated annealing. The main drawback of our approach is the high execution time. While SPA may take several hours to complete, our findings indicate that, for large data sets, convergence is reached quicker with a reasonably efficient solution.[1]

VIII. Conclusion

In this article, we aimed to tackle the issue of the generation of an adversary's background knowledge in privacy risk assessment by proposing a realistic approach, tailored for human mobility data. We represented the behavior of an adversary as a trajectory and envisioned three possible scenarios for generating it. In the first scenario, the trajectory is real; in the second scenario, it is synthetic; in the third scenario, the trajectory is generated by the Simulated Privacy Annealing (SPA) algorithm, with the specific objective of maximizing average risk. A limitation of our method is that, depending on the size of the data set, SPA may be heavily time-consuming. While simulated annealing provides a reasonable estimation of the privacy generated by an adversary, we find that a random trajectory produces acceptable results in far less time, suggesting that we may further speed up the computation by tuning the algorithm. Moreover, while we chose the average adversary risk as it represents a fair way to synthesize the risk for all the individuals involved, other functions may be tested 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 and other kinds of data such as retail or network data.

References

[1] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, pp. 1010–1027, Nov. 2001.

[1]The code used for our experiments can be found at: https://github.com/pellungrobe/SimulatedPrivacyAnnealing

[2] L. Sweeney, "K-Anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[3] F. Giannotti and D. Pedreschi, *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery*, 1st ed. Berlin, Germany: Springer-Verlag, 2008.

[4] V. Torra, *Data Privacy: Foundations, New Developments and the Big Data Challenge* (Studies in Big Data), vol. 28. Springer, 2017.

[5] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern, "Worst-case background knowledge for privacy-preserving data publishing," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 126–135.

[6] J. R. Bambauer. (Mar. 25, 2011). *Tragedy of the Data Commons*. [Online]. Available: https://ssrn.com/abstract=1789749

[7] F. Pratesi, A. Monreale, R. Trasarti, F. Giannotti, D. Pedreschi, and T. Yanagihara, "PRUDEnce: A system for assessing privacy risk vs utility in data sharing ecosystems," *TDP*, vol. 11, no. 2, pp. 139–167, 2018.

[8] R. Pellungrini, L. Pappalardo, F. Pratesi, and A. Monreale, "A data mining approach to assess privacy risk in human mobility data," *ACM TIST*, vol. 9, no. 3, pp. 31:1–31:27, 2018.

[9] F. Giannotti, L. Pappalardo, D. Pedreschi, and D. Wang, *A Complexity Science Perspective on Human Mobility*. Cambridge, U.K.: Cambridge Univ. Press, 2013, pp. 297–314.

[10] H. Barbosa *et al.*, "Human mobility: Models and applications," *Phys. Rep.*, vol. 734, pp. 1–74, Mar. 2018.

[11] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, p. 38, 2014.

[12] F. Giannotti, A. Monreale, and D. Pedreschi, "Mobility data and privacy," in *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery*, C. Renso, S. Spaccapietra, and E. Zimanyi, Eds. Springer, 2013, pp. 174–193.

[13] A. Monreale *et al.*, "Movement data anonymity through generalization," *TDP*, vol. 3, no. 2, pp. 91–121, 2010.

[14] G. Poulis, S. Skiadopoulos, G. Loukides, and A. Gkoulalas-Divanis, "Apriori-based algorithms for k^m-anonymizing trajectory data," *Trans. Data Privacy*, vol. 7, no. 2, pp. 165–194, 2014.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. TCC* 2006, pp. 265–284.

[16] A. Monreale *et al.*, "Privacy-preserving distributed movement data aggregation," in *Proc. AGILE*, in Lecture Notes in Geoinformation and Cartography, 2013, pp. 225–245.

[17] A. Cavoukian and E. K. Emam. (Jan. 2011). *Dispelling Myths Surrounding De-Identification: Anonymization Remains a Strong Tool for Protecting Privacy*. [Online]. Available: https://fpf.org/wp-content/uploads/2011/07/Dispelling%20the%20Myths%20Surrounding%20De-identification%20Anonymization%20Remains%20a%20Strong%20Tool%20for%20Protecting%20Privacy.pdf

[18] G. Paass, "Disclosure risk and disclosure avoidance for microdata," *J. Bus. Econ. Statist.*, vol. 6, no. 4, pp. 487–500, Oct. 1988.

[19] M. Elliot, "Integrating file and record level disclosure risk assessment," in *Inference Control in Statistical Databases*. Berlin, Germany: Springer, 2002, pp. 126–134.

[20] A. Basu *et al.*, "A privacy risk model for trajectory data," in *Trust Management VIII* (IFIP Advances in Information and Communication Technology), vol. 430. Springer, 2014, pp. 125–140.

[21] D. Kondor, B. Hashemian, Y.-A. de Montjoye, and C. Ratti, "Towards matching user mobility traces in large-scale datasets," *IEEE Trans. Big Data*, early access, Sep. 24, 2018, doi: 10.1109/TBDATA.2018.2871693.

[22] L. Rossi and M. Musolesi, "It's the way you check-in: Identifying users in location-based social networks," in *Proc. 2nd Ed. ACM Conf. Online Social Netw. (COSN)*, 2014, p. 215.

[23] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "De-anonymization attack on geolocated data," *J. Comput. Syst. Sci.*, vol. 80, no. 8, pp. 1597–1614, Dec. 2014.

[24] C. Song, T. Koren, P. Wang, and A.-L. Barabási, "Modelling the scaling properties of human mobility," *Nature Phys.*, vol. 6, no. 10, pp. 818–823, Oct. 2010.

[25] L. Pappalardo, S. Rinzivillo, and F. Simini, "Human mobility modelling: Exploration and preferential return meet the gravity model," in *Proc. 7th Int. Conf. Ambient Syst., Netw. Technol. (ANT)*, vol. 83. Amsterdam, The Netherlands: Elsevier, 2016, pp. 934–939.

[26] L. Pappalardo and F. Simini, "Data-driven generation of spatio-temporal routines in human mobility," *Data Mining Knowl. Discovery*, vol. 32, no. 3, pp. 787–829, May 2018.

[27] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, Jun. 1953.

[28] A. Khachaturyan, S. Semenovsovskaya, and B. Vainshtein, "The thermodynamic approach to the structure analysis of crystals," *Acta Crystallographica Sect. A*, vol. 37, no. 5, pp. 742–754, Sep. 1981.

[29] H. Amer, N. Salman, M. Hawes, M. Chaqfeh, L. Mihaylova, and M. Mayfield, "An improved simulated annealing technique for enhanced mobility in smart cities," *Sensors*, vol. 16, no. 7, p. 1013, Jun. 2016.

[30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simmulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[31] W. Ben-Ameur, "Computing the initial temperature of simulated annealing," *Comput. Optim. Appl.*, vol. 29, no. 3, pp. 369–385, Dec. 2004.

[32] Y. Zheng and X. Zhou, Eds., *Computing With Spatial Trajectories*. Cham, Switzerland: Springer, 2011.

[33] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, p. 29, 2015.

[34] G. Andrienko *et al.*, "(So) big data and the transformation of the city," *Int. J. Data Sci. Analytics*, Mar. 2020. [Online]. Available: https://link.springer.com/article/10.1007%2Fs41060-020-00207-3

[35] L. Pappalardo, G. Barlacchi, R. Pellungrini, and F. Simini, "Human mobility from theory to practice:Data, models and applications," in *Proc. Companion World Wide Web Conf.*, May 2019, p. 1311.

[36] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella, "Human mobility models for opportunistic networks," *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 157–165, Dec. 2011.

[37] D. Henderson, S. Jacobson, and A. Johnson, "The theory and practice of simulated annealing," in *Handbook of Metaheuristics*. Boston, MA, USA: Kluwer, Apr. 2006, pp. 287–319.

[38] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli, "Convergence and finite-time behavior of simulated annealing," *Adv. Appl. Probab.*, vol. 18, no. 03, pp. 747–771, Sep. 1986.

[39] M. J. D. Powell, "Nonconvex minimization calculations and the conjugate gradient method," in *Numerical Analysis*, D. F. Griffiths, Ed. Berlin, Germany: Springer, 1984, pp. 122–141.

[40] L. Pappalardo, S. Rinzivillo, Z. Qu, D. Pedreschi, and F. Giannotti, "Understanding the patterns of car travel," *Eur. Phys. J. Special Topics*, vol. 215, no. 1, pp. 61–73, Jan. 2013.

[41] L. Pappalardo, F. Simini, S. Rinzivillo, D. Pedreschi, F. Giannotti, and A.-L. Barabási, "Returners and explorers dichotomy in human mobility," *Nature Commun.*, vol. 6, no. 1, pp. 1–8, Nov. 2015.

[42] L. Pappalardo, F. Simini, G. Barlacchi, and R. Pellungrini, "Scikit-mobility: A Python library for the analysis, generation and risk assessment of mobility data," 2019, *arXiv:1907.07062*. [Online]. Available: https://arxiv.org/abs/1907.07062

[43] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[44] M. Clerc, "Discrete particle swarm optimization, illustrated by the traveling salesman problem," in *New Optimization Techniques in Engineering*. Berlin, Germany: Springer, 2004, pp. 219–239.

[45] M. Rapaic, Z. Kanovic, and Z. Jelicic, "Discrete particle swarm optimization algorithm for solving optimal sensor deployment problem," *J. Autom. Control*, vol. 18, no. 1, pp. 9–14, 2008.

Roberto Pellungrini received the master's degree in business informatics, with a thesis on Assessing Privacy Risk and Quality in Human Mobility Data, and the Ph.D. degree in computer science from the University of Pisa in 2020. He is currently a member of the KDD Lab. His main research interest includes ethical aspects related to data science, in particular regarding privacy issues. He received the Ph.D. Scholarship.

Luca Pappalardo received the Ph.D. degree in CS from the University of Pisa. He is currently a permanent Researcher with the KDD Lab, ISTI-CNR. His research interests include data science and big data analytics, with a specific focus on applications like human mobility, social network analysis, and sports analytics. He was a recipient of the 2014 Google-ISTAT Award on innovative applications of big data. He is one of the developers of scikit-mobility, a Python library for human mobility analysis.

Filippo Simini is currently a Senior Lecturer with the Department of Engineering Mathematics, University of Bristol, U.K., and a fellow of The Alan Turing institute, U.K., for data science and artificial intelligence. He has a background in statistical physics, complex systems, and data science. He is interested in interdisciplinary problems, including collective and individual human mobility, population dynamics, ecological networks, and computational social science. His work has impact and applications on the assessment of future mobility scenarios, epidemic forecasting, emergency planning, and anomaly detection.

Anna Monreale is currently an Assistant Professor with the CS Department, University of Pisa, and a member of the KDD Lab. Her research interests include big data analytics, privacy issues raising in mining social, and human sensitive data. She is interested in the evaluation of privacy risks during analytical processes and the design of privacy-by-design technologies in the era of big data.