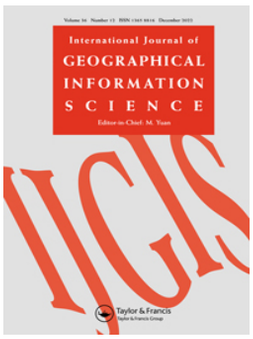

International Journal of Geographical Information Science

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tgis20

My home is my secret: concealing sensitive locations by context-aware trajectory truncation

Anna Brauer, Ville Mäkinen, Axel Forsch, Juha Oksanen & Jan-Henrik Haunert

To cite this article: Anna Brauer, Ville Mäkinen, Axel Forsch, Juha Oksanen & Jan-Henrik Haunert (2022) My home is my secret: concealing sensitive locations by context-aware trajectory truncation, International Journal of Geographical Information Science, 36:12, 2496-2524, DOI: 10.1080/13658816.2022.2081694

To link to this article: https://doi.org/10.1080/13658816.2022.2081694

Published online: 06 Jun 2022.

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
2022, VOL. 36, NO. 12, 2496–2524
https://doi.org/10.1080/13658816.2022.2081694

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

OPEN ACCESS Check for updates

My home is my secret: concealing sensitive locations by context-aware trajectory truncation

Anna Brauer[a,b], Ville Mäkinen[a], Axel Forsch[c], Juha Oksanen[a] and Jan-Henrik Haunert[c]

[a]Department of Geoinformatics and Cartography, Finnish Geospatial Research Institute, National Land Survey of Finland, Masala, Finland; [b]Department of Computer Science, University of Helsinki, Helsinki, Finland; [c]Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany

ABSTRACT

Ever since location-based services and mobile applications collecting data gathered through Global Navigation Satellite System (GNSS) positioning have become popular, concerns about location privacy have been expressed. Research has shown that human trajectory repositories containing sequences of observed locations ordered in time constitute a rich source for analyzing movement patterns, but they can also reveal sensitive personal information, such as a person's home address. In this paper, we present a mechanism that protects visits to sensitive locations by suppressing revealing parts of trajectories. Our attack model acknowledges that the course of a trajectory, combined with spatial context information, can facilitate privacy breaches even if sensitive locations have been concealed. Thus, we introduce the concept of *k*-site-unidentifiability, a specialization of *k*-anonymity, under which a sensitive location cannot be singled out from a group of at least *k* sites that the trajectory could have visited. In an experimental study, we show that our method is utility-preserving and protects sensitive locations reliably even in sparsely built environments. As it can process each trajectory independently, individuals may also use our mechanism to enhance their privacy before publishing their trajectories.

ARTICLE HISTORY
Received 14 August 2021
Accepted 12 May 2022

KEYWORDS
Location privacy; anonymity; mobility tracks; movement data; privacy-preserving data publication

1. Introduction

Location-based data connects the real and the virtual world tighter than ever before due to the ubiquity of applications run on mobile devices equipped with Global Navigation Satellite System (GNSS) receivers. Spatial trajectories (i.e., time-stamped sequences of location measurements) provide particularly valuable insights into human mobility that are highly relevant for researchers, urban planners, and businesses alike. To name a few examples, trajectory data has been utilized to investigate

CONTACT Anna Brauer anna.brauer@nls.fi Department of Geoinformatics and Cartography, Finnish Geospatial Research Institute, National Land Survey of Finland, Masala, Finland

route preferences, monitor the behavior of crowds, improve and manage transportation systems, and optimize the placement of advertisements (Fan *et al.* 2015, Mazimpaka and Timpf 2016, Lu *et al.* 2018, Zhang *et al.* 2018).

Spatial trajectories record paths of moving objects. Using positioning technologies such as GNSS, spatial trajectories are created, for example, when users interact with location-based services (LBS) or track their movements and share them online to contribute to volunteered geographical information (Goodchild 2007). The latter type, high-sampling-rate trajectories (i.e., a position recorded every 1 to 10 seconds), have been collected in large amounts by proprietary platforms (e.g., AllTrails 2021, Strava 2021) and non-profit initiatives (e.g., Bike Data Project 2021). Partly, these comprehensive trajectory sets are publicly accessible online.

Human mobility data is personal data that can reveal more than just a person's whereabouts. If an individual often cycles the same route in the morning, one may conclude that this is their commuting route—their apartment might be empty during the day. Similarly, if an individual frequently visits a cancer clinic, they might suffer from cancer. Certain points of interest (POIs), such as hospitals, places of worship, or casinos can be considered universally sensitive. Other sensitive locations are places that can be used to identify a person uniquely, notably their home and workplace. Inferring these meaningful places from human mobility data is a subject of ongoing research as it has applications in, e.g., urban planning (Chen and Poorthuis 2021). On the other hand, the privacy of individuals can be compromised when the inferred data reveals sensitive information, or when the inferred information is used to link anonymously published data to a person's identity, which is referred to as a re-identification attack.

In the literature, a frequently adapted concept to protect trajectories from re-identification attacks is *k-anonymity* (Sweeney 2002). In a *k*-anonymous dataset, each trajectory is indistinguishable from at least $k - 1$ other trajectories. Consequently, an attacker can only link an individual's trip to a group of at least *k* trajectories. To achieve *k*-anonymity for trajectory datasets, **L**ocation **P**rivacy-**P**reserving **M**echanisms (LPPMs, Shokri *et al.* 2011) have been developed that generalize, suppress and distort trajectory data (e.g., Abul *et al.* 2008, Yarovoy *et al.* 2009, Monreale *et al.* 2010, Dong and Pi 2018). On the downside, these mechanisms can take a high toll on the utility of the data. LPPMs that solely focus on the protection of sensitive locations, on the other hand, are usually more utility-preserving, and hence better suited for applications where strict *k*-anonymity is not required (e.g., Huo *et al.* 2012, Dai *et al.* 2018, Wang and Kankanhalli 2020). However, there is a gap in the literature regarding mechanisms that are able to protect locations that are not universally sensitive, but especially sensitive to certain individuals, such as their homes and workplace. Existing solutions (e.g., Krumm 2007) are simple and unaware of the spatial context.

This paper proposes a suppression-based algorithm that protects sensitive locations by applying the concept of *k*-anonymity to sites, i.e., locations with a specific semantic meaning, e.g., buildings or building complexes. Consider a trajectory whose destination is a site that is regarded as sensitive and should be protected. The goal of the algorithm is to hide the true destination among at least $k - 1$ other sites, all of which would be plausible destinations given the sanitized trajectory. In other words, an attacker could narrow possible destinations down to these *k* sites, but they could not

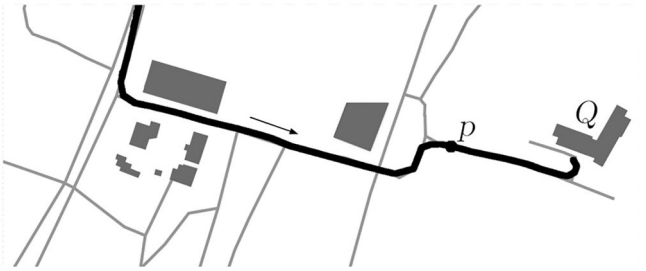

Figure 1. A trajectory (black line) ends close to a site Q (gray polygon), so that Q can be inferred as the trajectory's destination. Assuming that the trajectory is truncated at point p, the site Q can still be identified as the destination by taking the direction into account. This holds particularly if the road network (gray lines) is also considered.

determine the destination more accurately. The algorithm achieves this by truncating the trajectory; it removes the trajectory points that lead to the destination, as many as necessary to render the destination unidentifiable among $k - 1$ other site, but no more than that so as to retain as much data as possible.

Assessing the tools an adversary may have to infer a sensitive location is hard, as it is impossible to know which auxiliary information is accessible to them (Keßler and McKenzie 2018). In this work, we adopt a pragmatic approach and consider an adversary with access to publicly available spatial context information, who uses geometric techniques to infer a sensitive location. To the best of our knowledge, we are the first to consider not only the proximity of the trajectory to the sites but also its direction as a clue for inferring the trajectory's destination. The trajectory in Figure 1, for example, evidently leads to a building Q. If the trajectory is truncated at point p, the building Q is not the closest site anymore, yet its direction still unambiguously identifies Q as the destination.

Throughout the paper, we focus on protecting a trajectory's destination to present our mechanism in the simplest manner. Nevertheless, the mechanism is able to protect sensitive locations at any position of a trajectory. To protect an intermediate stay point, i.e., a location where the trajectory dwells for a considerable time (Zheng *et al.* 2009), the trajectory can be split at the stay point, and both of the resulting sub-trajectories can be truncated with our mechanism. To summarize, our main contributions are:

1. a formal privacy guarantee for the protection of sensitive locations, called *k-site-unidentifiability*, that adopts the *k*-anonymity principle for sites;
2. a suppression-based algorithm for spatial trajectories that protects sensitive locations by preserving *k*-site-unidentifiability, which does not require a global definition of sensitive locations;
3. an implementation of the aforementioned algorithm preventing the disclosure of sensitive locations through proximity- and direction-based attacks, which allows trajectories to be processed independently of each other;
4. an experimental study that documents and quantifies the utility-preserving properties of the implementation and highlights practical challenges.

The paper is structured as follows: In Section 2, we give an overview of LPPMs for trajectories. Section 3 frames the problem solved by our algorithm, which is presented in Section 4. The experimental study is presented in Section 5. Finally, we discuss our work in Section 6 and conclude it in Section 7.

2. Related work

The protection of sensitive locations has received considerable attention within the field of privacy-preserving trajectory publishing. Some location privacy-preserving mechanisms (LPPMs) target trajectories as a whole, primarily to prevent re-identification attacks. These mechanisms may protect sensitive locations implicitly. However, other LPPMs are explicitly designed to ensure that visits to sensitive locations remain private. Both categories of LPPMs are covered in the following review, with the first category split into three subcategories. For a more extensive overview of state-of-the-art LPPMs, we refer the reader to Fiore *et al.* (2020) and Primault *et al.* (2019). The latter also provides a review of LPPMs for real-time interactions with location-based services, a related topic that has been omitted here.

2.1. k-anonymity-based LPPMs

The *k*-anonymity principle, originally proposed by Sweeney (2002), is a classical privacy criterion for relational tables that have been adopted for trajectory data. Considering a pseudonymized database where each trajectory constitutes a record, *k*-anonymity requires each trajectory to be indistinguishable from at least $k - 1$ other trajectories with respect to a set of attributes called quasi-identifiers. Any part of a trajectory, from single locations to sequences of locations, can be considered a location-based quasi-identifier if it identifies an individual uniquely. Yarovoy *et al.* (2009) argue that each trajectory has a different set of locations as its quasi-identifiers. Their mechanism groups the trajectories into anonymization sets and hides the quasi-identifying locations of each anonymization set by spatial generalization. On the other hand, Nergiz *et al.* (2008) present a mechanism that considers every trajectory point a potential quasi-identifier. After clustering the trajectories into groups of at least *k* trajectories, this LPPM creates a generalized trajectory for each group: the mechanism matches the points of the trajectories and replaces them with spatial generalizations. Unmatched points are suppressed.

Further suppression-based methods are presented by Terrovitis *et al.* (2017) and Pensa *et al.* (2008). Pensa *et al.* (2008) recognize that possible quasi-identifiers are not only single locations but also sequences of locations. Their LPPM builds a prefix tree of locations and prunes all sequences that exist less than *k* times in the trajectory dataset. Terrovitis *et al.* (2017) assume that an adversary has observed portions of the trajectories and that the anonymizing party is aware of the adversary's knowledge. Through suppression and trajectory splitting, their mechanisms limit the adversary's ability to expand their knowledge. The mechanisms apply the ℓ-*diversity* paradigm (Machanavajjhala *et al.* 2007), an extension of *k*-anonymity. The concept of ℓ-diversity requires records with the same quasi-identifier, i.e. records in the same equivalence

class, to have sufficiently diverse values of sensitive attributes. This addresses a major weakness of *k*-anonymity, its vulnerability against attribute linkage attacks. Even if the *k* trajectories in one equivalence class are indistinguishable from each other, they might have the same sensitive attribute values, which allows the attacker to link the attribute value to all of the trajectories. The concept of ℓ-diversity has been extended further by *t-closeness* (Li *et al.* 2007), which requires the distribution of sensitive attribute values in each equivalence class to resemble the distribution in the whole dataset. Tu *et al.* (2019) introduce an LPPM that applies *t*-closeness for spatial trajectories. The mechanism spatially generalizes each trajectory point so that the generalized locations contain semantically diverse POIs.

A commonality between these works (Nergiz *et al.* 2008, Pensa *et al.* 2008, Yarovoy *et al.* 2009, Terrovitis *et al.* 2017, Tu *et al.* 2019); is that they are not intended for trajectories with a high sampling rate in the order of seconds. An LPPM that simplifies high-sampling-rate trajectories has been presented by Monreale *et al.* (2010). The mechanism reduces the trajectories to characteristic points and clusters them. By creating the Voronoi tessellation of the cluster centroids, the mechanism obtains generalized trajectory representations that are used to build a prefix tree from which anonymized trajectories can be generated. Abul *et al.* (2008) argued that trajectories do not have to be identical to be indistinguishable. Motivated by the uncertainty of location measurements, they proposed (k, δ)-anonymity. Under this concept, trajectories are indistinguishable if they exist within a cylindrical tube with a radius of δ. The mechanism achieves (k, δ)-anonymity through clustering and space translation.

Finally, the LPPM by Dong and Pi (2018) matches trajectories to the road network and clusters them. For each cluster, the trajectory representing the most frequently taken route is selected and published as the cluster representative.

k-anonymity prevails as the basis for many state-of-the-art LPPMs and its capability to protect from re-identification attacks is apparent. However, it has limits as a formal privacy guarantee. Besides attribute linkage attacks, it has been shown that *k*-anonymity can be breached if the attacker can combine information from several sources (Ganta *et al.* 2008). Furthermore, *k*-anonymity does not sufficiently protect from probabilistic attacks that can expand the attacker's knowledge, even if they are not entirely conclusive (e.g., Wong *et al.* 2011).

2.2. Differential privacy-based LPPMs

Differential privacy (Dwork 2008) is a stronger but more rigid concept of privacy. The result of a differentially private mechanism must not differ significantly upon the addition or removal of any single item to the database, which prevents attackers from exploiting the mechanism to expand their knowledge about an individual. Differentially private LPPMs create differentially private representations of trajectory data, for example, prefix trees (e.g., Chen *et al.* 2012, Bonomi and Xiong 2013) or probability distributions (e.g., Mir et al. 2013, Gursoy *et al.* 2019). These data models can serve as the basis for generating synthetic trajectories that retain important features for spatial data analysis (e.g., the probabilities of transitioning from one location to

another). However, most LPPMs based on differential privacy do not scale well (Fiore *et al.* 2020) and are not suitable for high-sampling-rate trajectories.

2.3. LPPMs that mitigate risks

Not all LPPMs provide a formal privacy guarantee such as *k*-anonymity or differential privacy, some only mitigate privacy risks (Fiore *et al.* 2020). Examples include traditional obfuscation techniques, such as random perturbation or grid masking, which relocate the trajectory points individually (Kwan *et al.* 2004, Krumm 2007, Seidl *et al.* 2016). Other LPPMs in this category segment trajectories and associate each segment with a different identifier (Song *et al.* 2014) or swap identifiers among trajectories (Salas *et al.* 2020). Another approach is cloaking, i.e., the reduction of the trajectories' granularity in space or time (e.g., Hoh *et al.* 2006, Rossi *et al.* 2015).

2.4. LPPMs for sensitive locations

LPPMs in this category focus solely on protecting sensitive locations, leaving the remaining parts of the trajectories untouched. Thus, these mechanisms do not necessarily aim to prevent re-identification attacks, although the risk of re-identification can be reduced by protecting revealing locations, such as the home or workplace.

Substitution-based techniques replace sensitive locations with other nearby locations that are, ideally, semantically similar to ensure that semantic patterns are preserved (Naghizade *et al.* 2014, Han and Tsai 2015, Dai *et al.* 2018). Then, the trajectories are reconstructed to contain the new location instead of the old, so that the trajectory takes a detour from its original route. However, this distorts the traffic flow, and a suitable location for substitution may not always be available. Other approaches hide sensitive locations by cloaking (Huo *et al.* 2012, Cicek *et al.* 2014, Wang and Kankanhalli 2020) or suppression (Krumm 2007). These mechanisms are the most similar to ours. Cloaking-based LPPMs replace trajectory points close to sensitive locations with generalized zones that contain several locations. Huo *et al.* (2012)'s mechanism ensures that the locations in these zones are semantically diverse. However, this mechanism is not suited for sparsely built areas, where the zones need to be very large to fulfill the diversity criterion, and it does not protect from attacks exploiting the direction in which a trajectory is headed. The LPPMs used by Cicek *et al.* (2014) and Wang and Kankanhalli (2020) bound the probability that a trajectory visits a certain sensitive location by taking human mobility patterns into account. These mechanisms solve a problem that differs from the one considered in our work. They assume that some locations are sensitive per se, i.e., for all trajectories equally. For example, hospitals or places of worship could be considered universally sensitive. As opposed to this, the simple suppression-based LPPM considered by Krumm (2007) does not require a global definition of sensitive locations and can be applied to single trajectories. It suppresses trajectory points within a circular region around the sensitive location. The circle's center is randomly placed inside a smaller circle centered at the sensitive location. However, this method is, unlike ours, ignorant of the spatial context. Finally, Primault *et al.* (2015) hide stay points by distorting the trajectories in the

dimension of time. Their mechanism recreates the trajectories with a constant speed, removing any stay points in the process. This procedure removes relevant traffic-related information and is not suited to protect the ends of a trajectory.

3. Problem formulation

In this section, we formalize the problem by defining the type of adversary we consider and presenting a privacy concept that limits the adversary's ability to infer sensitive locations from trajectory data. First, we lay out the definitions that are essential to the problem.

Definition 1 (Trajectory). A trajectory T is a sequence of triplets, i.e., $T = \langle (x_1, y_1, \tau_1), \ldots, (x_n, y_n, \tau_n) \rangle$, where $x_i, y_i, \tau_i \in \mathbb{R}$ for each $i \in 1, \ldots, n$. x_i and y_i are the coordinates of a point in the Euclidean plane, and τ_i is a time stamp with $\tau_i < \tau_{i+1}$ for each $i \in 1, \ldots, n-1$.

Definition 2 (Trajectory segment). A trajectory segment is a line bounded by two consecutive trajectory points.

Definition 3 (Endpoint). Given a trajectory $T = \langle (x_1, y_1, \tau_1), \ldots, (x_n, y_n, \tau_n) \rangle$, the endpoint is the last point of T , i.e., the point (x_n, y_n). It is denoted by $end(T)$.

Definition 4 (Site). A site s is a shape in the Euclidean plane.

Let T be a trajectory that represents the movement towards a sensitive location represented by a site $\tilde{s}$. We call $\tilde{s}$ the destination of T, regardless of whether $end(T)$ intersects with $\tilde{s}$.

3.1. Attacker model

The attacker's objective is to identify the destination site $\tilde{s}$ of a set of trajectories $\Theta = \{T_1, \ldots, T_l\}$. We assume that the attacker knows that all trajectories in Θ have a common destination $\tilde{s}$. Furthermore, we assume that the attacker knows a set of sites S which is guaranteed to contain $\tilde{s}$. In practice, S could contain all sites of a city published as open data.

To single out $\tilde{s}$, the attacker uses a number of attack functions $f_1, \ldots, f_a$. Given a trajectory T, each attack function f yields a set of candidate sites: $f(S, T) \subseteq S$. For example, an attack function may return all sites within a circular region with $end(T)$ as its center. The attacker is able to guess $\tilde{s}$ by narrowing down S using the candidate sets $\{f_j(S, T) | T \in \Theta, 1 \leq j \leq a\}$ collected by all attack functions. For example, the attacker may assume $\tilde{s}$ to be the site with the most occurrences in the candidate sets.

In the following, we define a concept under which the ability of the attacker to narrow down S is limited.

3.2. Privacy model

We define k-site-unidentifiability, a privacy concept related to k-anonymity. K-site-unidentifiability requires a destination site $\tilde{s}$ to be indistinguishable from at least $k - 1$ other site for an adversary who is trying to determine $\tilde{s}$.

Consider a database D_S as a set of records where each site s in the set of all sites S is represented by exactly one record, denoted by r_s. Each record holds its identifier and some attributes $a_1, \ldots, a_n$. We denote the value of a record r_s for an attribute a as $r_s(a)$. A subset of these attributes $Q \subseteq \{a_1, \ldots, a_n\}$ that could enable the identification of a site as the destination of a set of trajectories Θ is called a *quasi-identifier* with respect to Θ. *K*-site-unidentifiability with respect to Θ is guaranteed if there are at least $k - 1$ records that are indistinguishable from the record of the destination $\tilde{s}$ with respect to every quasi-identifier.

Definition 5 (*k*-site-unidentifiability). Let S be a set of sites and Θ be a set of trajectories with a common destination $\tilde{s} \in S$. Furthermore, let D_S be a database of sites and $\tilde{r}$ the record in D_S representing $\tilde{s}$. D_S satisfies *k*-site-unidentifiability with respect to Θ and a quasi-identifier Q if there exists a set of records $R \subseteq (D_S \backslash \{\tilde{r}\})$ so that the following two conditions hold:

$$|R| \geq k-1 \tag{1}$$

$$\forall a \in Q, \forall r \in R : \tilde{r}(a) = r(a). \tag{2}$$

In other words, if k-site-unidentifiability is satisfied, the destination $\tilde{s}$ is hidden in a set of k sites whose corresponding records have the same quasi-identifying attribute values. We refer to this set of sites as *protection set* of $\tilde{s}$ and denote it by $C(\tilde{s})$.

3.3. Problem definition

Given a set of sites S and a set of trajectories Θ with a common destination $\tilde{s}$, we aim to preserve *k*-site-unidentifiability under a set of attacks $F = \{f_1, \ldots, f_a\}$ as defined in Section 3.1. We assume that the attacker tries to identify $\tilde{s}$ by deriving quasi-identifying attributes using the attacks in F. For example, recall the attack function returning all sites within a certain radius from a trajectory's endpoint. Using this function, denoted by f_c, the attacker could construct a quasi-identifier consisting of one attribute a_c that equates to 1 only for sites occurring most frequently in the candidate sets of f_c:

$$r_s(a_c) = \begin{cases} 1, & \text{if } s \in \arg\ \max_{u \in S} occ(u) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where

$$occ(s) = |\{T | T \in \Theta, s \in f_c(S, T)\}|. \tag{4}$$

Thus, the attacker is able to identify $\tilde{s}$ as the destination if $\tilde{s}$ stands out as being most often within the vicinity of the trajectories' endpoints.

To ensure *k*-site-unidentifiability for any possible quasi-identifier derived from the attack functions in F, it is sufficient if the following condition holds:

$$\forall T \in \Theta, \forall f \in F : f(S, T) \cap C(\tilde{s}) \in \{\emptyset, C(\tilde{s})\} \tag{5}$$

Intuitively, since each attack function f returns a set of sites, the sites in the protection set $C(\tilde{s})$ are indistinguishable with respect to f if either all or none of the sites $C(\tilde{s})$ are part of the attack's result for each trajectory in Θ.

In Section 3.1, we defined that the sites in S are available to the attacker. Therefore, a mechanism that preserves k-site-unidentifiability cannot alter S. It can, however, alter the trajectories in Θ. In conclusion, the problem is defined as follows:

Given a set of sites S, a set of trajectories Θ with a common destination $\tilde{s} \in S$, and a set of attack functions F, we require a mechanism that transforms Θ into a set of trajectories Θ' that fulfills Condition 5, i.e., so that k-site-unidentifiability with respect to Θ' and the attack functions in F is preserved.

4. The S-TT algorithm

The **S**ite-dependent **T**rajectory **T**runcation algorithm (S-TT) that is presented in the following truncates trajectories in a set Θ with a common destination $\tilde{s} \in S$ so that k-site-unidentifiability with respect to a number of well-defined attack functions is guaranteed. In the following, we first outline the algorithm for a set of arbitrary functions. For the algorithm to be applicable in practice, we need to make further assumptions about possible attacks. Thus, in a second step, we present the proximity- and direction-based S-TT algorithm which prevents simple geometric attacks.

4.1. Algorithm outline

As input, the S-TT algorithm requires a set Θ of trajectories with a common destination, a set of sites S, the destination site $\tilde{s} \in S$ of the trajectories, and a set of sites $C(\tilde{s}) \subseteq S$ as the protection set of $\tilde{s}$, where $\tilde{s} \in C(\tilde{s})$ and $|C(\tilde{s})| \geq k$. Furthermore, the attack functions $F = \{f_1, \ldots, f_a\}$, defined in Section 3.1, have to be specified.

The algorithm's output is a set of trajectories Θ' where each original trajectory $T \in \Theta$ is replaced by a truncated trajectory T'. The truncated trajectory T' is obtained by iteratively suppressing the endpoint of T until none of the attack functions can be used to differentiate between the sites in $C(\tilde{s})$. Suppression thus continues until T' satisfies the following condition:

$$\forall f \in F : f(S, T') \cap C(\tilde{s}) \in \{\emptyset, C(\tilde{s})\}. \tag{6}$$

If Condition 6 holds for all $T' \in \Theta'$, so does Condition 5, and k-site-unidentifiability with respect to Θ' and F is preserved. Hence, the destination $\tilde{s}$ is indistinguishable from at least $k - 1$ site in $C(\tilde{s})$ a given F, as no site in $C(\tilde{s})$ can be part of the result of an attack using any of the functions in F unless the result includes all sites in $C(\tilde{s})$.

Algorithm 1: S-TT(Attack functions F, set of trajectories Θ, set of sites S, protection set $C(\tilde{s})$)

```
Let Θ' be a copy of Θ
foreach T' ∈ Θ' do
    while ⋁_{f∈F} f(S, T') ∩ C(s̃) ∉ {∅, C(s̃)} do
        if Number of points in T' ≥ 2 then
            Remove endpoint of T'
        else
            Θ' ← Θ' \ {T'}
        end
    end
end
return Θ'
```

4.2. Proximity- and direction-based S-TT

The S-TT algorithm decides whether to suppress a trajectory point based on the attack functions in F. In the following, we address two attacks that emulate the inference of the destination site based on geometric reasoning. By plugging these attacks into the S-TT algorithm, we obtain a mechanism that preserves k-site-unidentifiability under simple proximity- and direction-based attacks.

The proximity of a trajectory T's endpoint to the surrounding sites is a critical clue for identifying the destination site $\tilde{s}$. Thus, the first attack assumes that $\tilde{s}$ is the site in the set of all sites S that is nearest to the endpoint $end(T)$. We denote this attack by f_{p}:

$$f_{\mathrm{p}}(S,T) := \arg\min_{s \in S} \; d(end(T), s), \tag{7}$$

where the function d is the Euclidean distance.

A second aspect that can reveal the destination is the direction in which the trajectory is headed. Thus, the second attack is based on the assumption that the last segment e of a trajectory T points roughly towards the destination $\tilde{s}$. The attack considers a triangular V-shaped region $R(T)$ that is mirror-symmetric with respect to e, has a corner in $end(T)$ and is unbounded in the opposite direction (Figure 2). We call the opening angle of the region α. Assuming that $\tilde{s}$ is located within the region, we define the direction-based attack function f_{d} as follows:

$$f_{\mathrm{d}}(S,T) := \{s | s \in S, s \cap R(T) \neq \emptyset\}. \tag{8}$$

The attacks f_{p} and f_{d} are simple, yet common sense suggests that they may be fairly successful. To protect trajectories against these attacks with the S-TT algorithm, we concretize Condition 6, so that the truncated trajectory T' is now required to satisfy

$$\forall f \in \{f_{\mathrm{p}}, f_{\mathrm{d}}\} : f(S,T') \cap C(\tilde{s}) \in \{\emptyset, C(\tilde{s})\}. \tag{9}$$

In other words, T' is obtained by iteratively suppressing the endpoint of T (Figure 3a) until the site closest to the endpoint is not any site in the protection set $C(\tilde{s})$ (Figure 3b) and the last trajectory segment points towards either none or all of the sites in $C(\tilde{s})$ (Figure 3c). Hence, k-site-unidentifiability is preserved, as the k sites in the protection set $C(\tilde{s})$ are indistinguishable with respect to the attacks f_{p} and f_{d}: these attacks cannot be used to narrow down S to a set of potential destination sites that includes a subset of $C(\tilde{s})$ unless it includes $C(\tilde{s})$ as a whole.

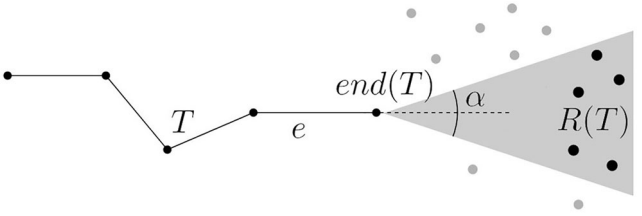

Figure 2. Schematic representation of a direction-based attack.

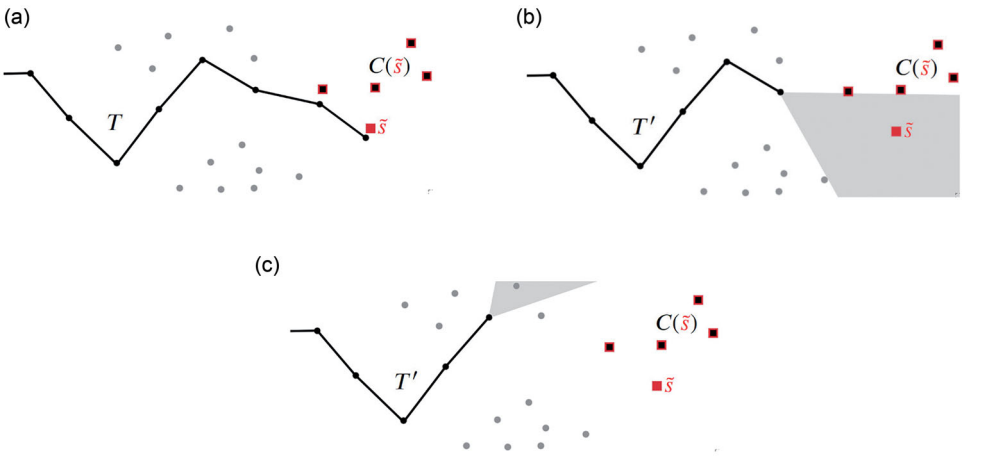

Figure 3. Schematic illustration of the proximity- and direction-based S-TT algorithm. The original trajectory T (a) leads to a destination site $\tilde{s}$ which has to be hidden among the sites in the protection set $C(\tilde{s})$, depicted as squares. Sites that are not part of $C(\tilde{s})$ are depicted as gray points. The S-TT algorithm suppresses the trajectory's endpoint iteratively until the site closest to the endpoint is not part of the protection set $C(\tilde{s})$ (b) and a V-shaped region that is aligned with the last segment of the truncated trajectory T' contains either all or none of the sites in $C(\tilde{s})$ (c).

5. Experimental study

In this section, we present experiments carried out with an implementation of the proximity- and direction-based S-TT algorithm. We investigate the algorithm's utility and impact on data properties that are relevant in common analyses. In this experimental study, we assumed that the origin and destination of the trajectories were sensitive by default and should be protected under k-site-unidentifiability.

The structure of this part is as follows. First, we provide some implementation details in Section 5.1. In Section 5.2, we present the trajectory sets and other data used in the study. Section 5.3 introduces the measures we used to quantify the utility. The study's most important findings are presented in Section 5.4.

5.1. Implementation

Recall that the proximity- and direction-based S-TT algorithm requires a trajectory T, a set of sites S, and protection set $C(\tilde{s})$ to hide the destination $\tilde{s}$. The method for determining the protection set should be chosen in consideration of the application and any auxiliary data available. The implementation used in the experimental study does not require any additional data and is suited for trajectories of varying quality where the true destination is unknown. First, a partition of sites is created where each subset contains at least k sites. These subsets serve as the basis for the protection sets of the sites. In the following, we briefly provide details on this process and some notes on the implementation of the S-TT algorithm.

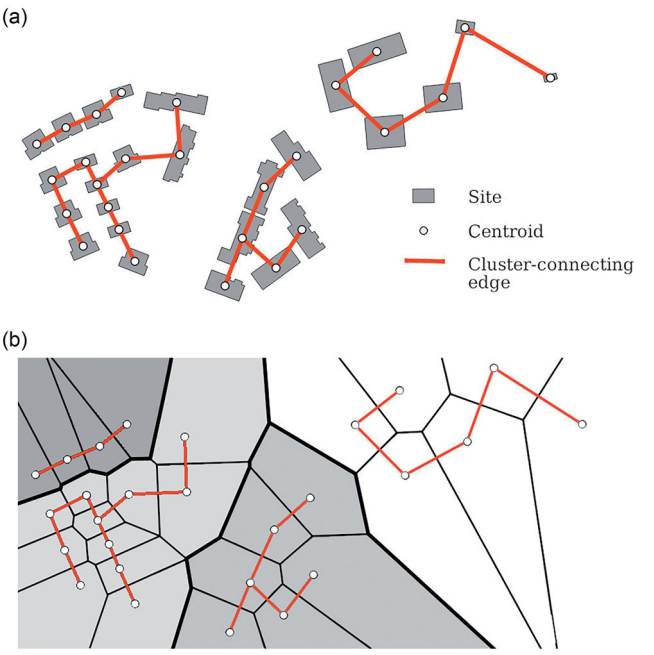

Figure 4. The implementation constructs the basis for the protection sets using a greedy clustering heuristic. (a) shows site clusters represented by minimum spanning trees. The Voronoi cells of a cluster's sites are merged to form a cluster cell. (b) depicts the cluster cells of the site clusters in (a) consisting of the Voronoi cells of the sites.

5.1.1. Obtaining site clusters

The $k-1$ sites in $C(\tilde{s})$ different from $\tilde{s}$ can theoretically be chosen arbitrarily. In practice, two requirements arise: first, the sites should be spatially close to $\tilde{s}$ to minimize the number of suppressed trajectory points; second, the way the sites are chosen should not enable the identification of $\tilde{s}$ by reverse engineering. For example, choosing the $k-1$ closest sites could enable the attacker to identify $\tilde{s}$ based on its protection set and is thus not appropriate. Both requirements can be fulfilled by computing a partition of the sites and using that partition as the basis for the protection sets of all trajectories in a dataset. S is partitioned into pairwise disjoint subsets, where each subset becomes the smallest possible protection set for all sites included in it. In other words, we define the problem as a clustering problem with minimum cluster size k and spatial compactness as the clustering criterion.

Our implementation uses a 2-approximate greedy clustering heuristic that approximates a minimum-weight spanning forest for an undirected edge-weighted graph (Imielińska *et al.* 1993). Consider a graph $G = (S, E)$ where the vertices S are the sites' centroids. As the sites are shaped in the plane, they can be represented by their centroids for simplicity. The set of edges E is constructed using the Delaunay triangulation $DT(S)$ of the vertices. The weight of an edge is its length, i.e., the Euclidean distance between the two vertices connected by the edge. Initially, each vertex is treated as a separate cluster. The greedy clustering algorithm iterates the edges of the graph sorted by their length in ascending order. Whenever an edge connects two clusters

where either cluster has less than k vertices, the clusters are joined. This process tends to produce spatially compact clusters (Figure 4a). It is also outlined in Algorithm 2. A noteworthy alternative approach was recently proposed by Haunert *et al.* (2021). It considers the layout of the road network to create spatially connected clusters of locations.

In the following, we utilize polygon representations of the site clusters called *cluster cells*. The cluster cell of a site cluster is defined as the union of the Voronoi cells of the sites in the site cluster (Figure 4b).

Algorithm 2: Site clustering(sites S, minimum cluster size k)

$C \leftarrow \{\{s\} \mid s \in S\}$ // Initialize clusters
$E \leftarrow \text{edges}(DT(S))$ // Delaunay triangulation

while exists $c \in C : size(c) < k$ **do**
  if *shortest edge* $e \in E$ *connects two clusters* c_1 *and* c_2 *where* $c_1 \neq c_2$ **and** $(size(c_1) < k$ **or** $size(c_2) < k)$ **then**
    $C \leftarrow (C \setminus \{c_1, c_2\}) \cup \{c_1 \cup c_2\}$
  end
  $E \leftarrow E \setminus \{e\}$
end
return C

5.1.2. Selecting the protection sets and handling uncertainty

For each trajectory, we executed the truncation algorithm twice: once using the trajectory and its destination as input, and once using the reversed trajectory and its destination (i.e., the original origin). For simplicity, we refer to both the last and the first point of a trajectory (i.e., the endpoint of the reversed trajectory) as *endpoints* in the following.

The sites of origin and destination of the trajectories in our experiments were not explicitly given. Since the S-TT algorithm does not require $\tilde{s}$ but only $C(\tilde{s})$ to truncate a trajectory, $C(\tilde{s})$ could be initialized as the site cluster containing the site closest to the trajectory's endpoint. However, there would be no certainty of the true destination being among the sites in this site cluster. Positioning uncertainties may have occurred when the trajectory was recorded, or the recording could have been stopped before $\tilde{s}$ was reached. To account for these uncertainties, we drew a circular buffer of radius b around the endpoint. The sites contained in all cluster cells intersecting the buffer became part of the protection set used for truncation. We refer to the union of these cluster cells as *protection cell* (Figure 5).

5.1.3. Trajectory truncation

In each iteration of the S-TT algorithm, the algorithm decides whether to keep the current endpoint of the trajectory based on a two-part condition (Condition 9). The first part of the condition requires the site nearest to the endpoint to not be any site in the protection set $C(\tilde{s})$. To test this, our implementation utilizes the protection cell: the trajectory is truncated at least until its endpoint lies outside of the protection cell $C(\tilde{s})$, which is efficiently examined using a spatial index of the cluster cells. To test the second, direction-based part of the condition, the implementation calculates the

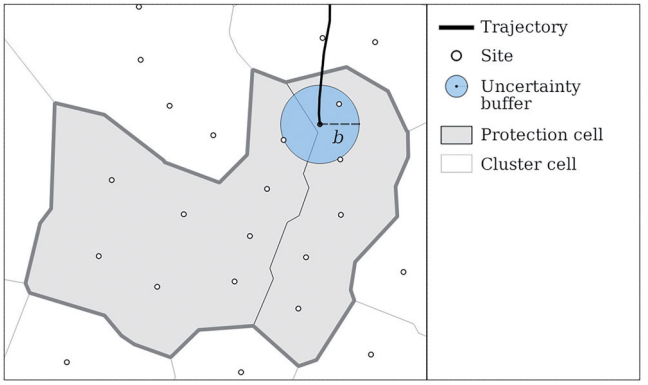

Figure 5. The uncertainty buffer with radius b of the trajectory's endpoint intersects two cluster cells. Together, they form the protection cell. The trajectory will be truncated with respect to all sites located in the protection cell.

Table 1. Trajectory sets were used in the experimental study.

	GL	H
City	Beijing	Helsinki
Trajectories	32,682	10,927
Type	Real-life	Synthetic
Contributors	182	0

bearing angles from the current endpoint to each site $C(\tilde{s})$ and compares them to the bearing angles of the V-shaped region's boundary lines.

5.2. Data

We used two trajectory sets to test our implementation; Table 1 provides an overview of the sets' characteristics. The trajectory sets were recorded in the structurally diverse cities of Beijing and Helsinki (Figure 6). We obtained the street network and the locations of buildings in both areas from OpenStreetMap (2021); the centroids of the building polygons were used as sites, while the street network was required for one of the utility tests. The OSM data was of sufficient quality for our purposes, although comparisons with current satellite imagery proved it to be missing some buildings, especially considering the City of Beijing. Our analysis required the street network to be segmented, which was accomplished by splitting the network edges at every intersection (*noding*).

5.2.1. Geolife trajectories

The *Geolife* dataset (identifier: GL), published by Microsoft Research Asia (Zheng *et al.* 2011), contained 17,621 trajectories, most of them located in the city of Beijing, China. The trajectories were collected by 182 users who recorded their movements, e.g., their commutes. The sampling rate of the trajectories varied, but the majority of the trajectories (91%) were densely logged (sampling interval $\leq$ 5 seconds).

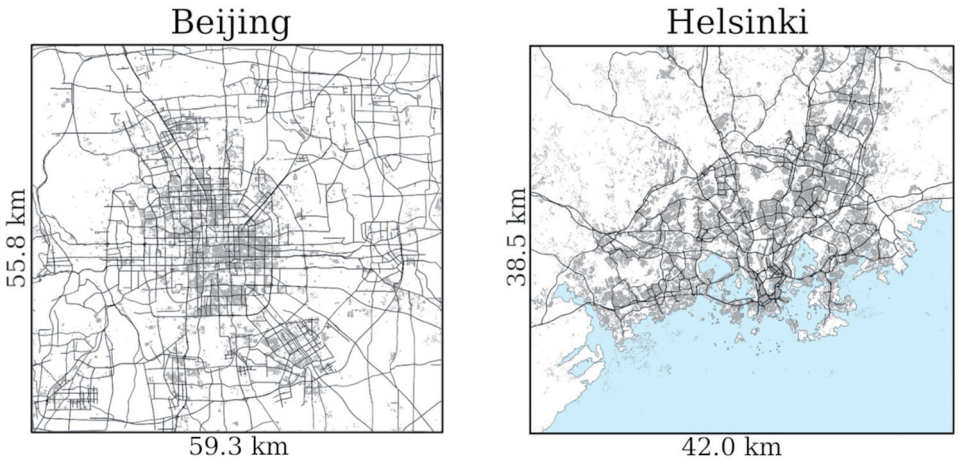

Figure 6. The extent of the study areas (a) Beijing and (b) Helsinki. OSM site polygons are depicted in gray, and the main transport routes of the road network in black.

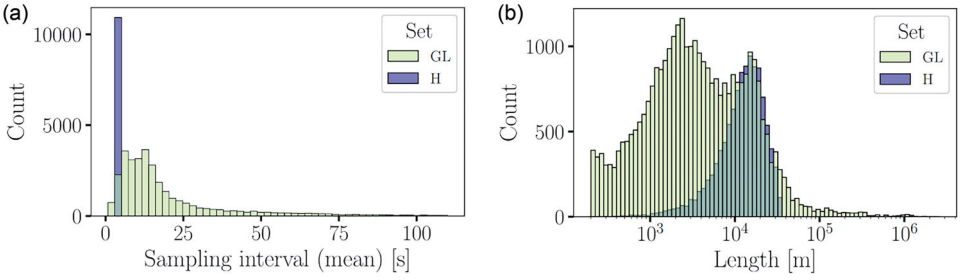

Figure 7. The sampling rate (a) and length of trajectories (b) differ considerably between the two datasets (GL and H).

We narrowed down the considered area with a bounding box (Figure 6). Trajectory endpoints outside of this area were not considered in the experiments. For trajectory preprocessing, we utilized the Python library scikit-mobility (Pappalardo *et al.* 2019). Severe outliers were removed, i.e., trajectory points where the speed from the previous point was greater than 150 km/h. The trajectories were split at stay points, i.e., sub-trajectories spending at least 15 min within a distance of 100 m from their first trajectory point. All trajectory points belonging to stay points were excluded. The surrounding sub-trajectories had to be at least 200 m long to be kept in the dataset.

After these processing steps, the dataset contained 32,682 trajectories of highly varying length and sampling rates (Figure 7). Parts of the analyses required the trajectories to be matched to the street network, which was accomplished with a hidden Markov model-based map matching procedure (Krumm 2007) using the open source engine Valhalla (2021).

5.2.2. Synthetic trajectories

We generated 10,927 synthetic trajectories (identifier: H) in the area of Greater Helsinki, Finland. First, our trajectory generator randomly sampled the position of the endpoints; the probability of a position being selected was determined by the population density. Then, the generator used a grid-based version of Dijkstra (1959)'s shortest path algorithm to find the shortest path around the sites to the road network. The two points on the road network were connected by calculating the shortest path on the road network, again using Dijkstra's algorithm. The generator assumed constant speed and a constant sampling rate so that a trajectory point was created every 5 m. This marks an important difference in the Geolife trajectories, along with the absence of measurement uncertainties and the higher median length (13037 m as opposed to 3379 m, see Figure 7).

5.3. Utility measures

Maximizing the utility of the sanitized trajectories means preserving as much of the original data and its features as possible. Thus, a straightforward utility measure is the length of the suppressed trajectory parts (Section 5.3.1). However, this does not fully capture the impact the S-TT algorithm has if it is applied to a large set of trajectories. Hence, we also consider measures that quantify the utility in common use cases of spatial analysis: the Jensen-Shannon distance of the distribution of endpoints (Section 5.3.2), the relocation distance of endpoints (Section 5.3.3), and the street-level suppression rate (Section 5.3.4).

5.3.1. Length of suppressed trajectory parts

For a trajectory T, the length of the suppressed trajectory parts $\delta_{\text{supp}}(T)$ is the difference between the length of T and the length of its truncated counterpart T'. For a set of trajectories D, we calculate the mean length of suppressed trajectory parts:

$$\bar{\delta}_{\text{supp}}(D) = \frac{1}{|D|} \sum_{T \in D} \delta_{\text{supp}}(T). \tag{10}$$

Note that $\delta_{\text{supp}}(T) = \mathit{length}(T)$ if the S-TT algorithm suppresses T completely.

5.3.2. Jensen-Shannon distance

Important features of a set of trajectories are the positions of the trajectories' endpoints. For example, origin-destination matrices help to estimate travel demands in transportation planning. Since the S-TT algorithm essentially shifts the position of the endpoints, we analyze the impact of this alteration on the distribution of the endpoints.

We compare the original distribution of endpoints and their distribution after truncation with the Jensen-Shannon distance (Endres and Schindelin 2003), i.e., the square root of the Jensen-Shannon divergence (Lin 1991), which measures the similarity of two probability distributions. To transform the spatial distribution of the endpoints into a normalized vector, the points are sorted into buckets of size r with respect to both the x and y dimensions. In other words, we create a grid of resolution r where

the value of each cell is the number of endpoints located in the cell. Finally, the grid is flattened and normalized to a vector v with $|v| = 1$. This way, we obtain two vectors v_0 and v_1 for a set of trajectories, where v_0 represents the distribution of the original trajectories' endpoints, and v_1 represents the endpoints' distribution after the trajectories have been truncated.

The Jensen-Shannon distance of v_0 and v_1, denoted by $\sqrt{JSD}(v_0, v_1)$, is calculated as follows:

$$\sqrt{JSD}(v_0, v_1) = \sqrt{\frac{KL(v_0||m) + KL(v_1||m)}{2}}, \tag{11}$$

where $m = \frac{1}{2}(v_0 + v_1)$ and $KL(a||b)$ is the Kullback-Leibler divergence of two vectors a and b (Kullback and Leibler 1951). In practice, $\sqrt{JSD}(v_0, v_1)$ is a value between 0 and 1. It is 0 if the distributions are identical, and 1, if they are so different than a random sample from the mixture of the two distributions, could be clearly assigned to either of them.

5.3.3. Relocation distance of endpoints

As the results of the S-TT algorithm are dependent on the spatial arrangement of the sites, the degree to which trajectories are truncated may vary for different regions. To investigate this, we consider the distance the endpoints are moved by the algorithm. We define this relocation distance for a region ρ and a trajectory set D. Since the areas covered by the trajectory sets we consider do not overlap, we omit D in the following definition. Hence, the mean endpoint relocation distance $\bar{\epsilon}(\rho)$ is calculated as

$$\bar{\epsilon}(\rho) = \frac{\sum_{(e_0, e_1) \in \Pi_\rho} d(e_0, e_1)}{|\Pi_\rho|}, \tag{12}$$

where Π_ρ is a set containing a tuple (e_0, e_1) for each endpoint in D that is located in region ρ and belongs to a trajectory that S-TT did not suppress completely. The endpoint of the original trajectory is denoted by e_0, its counterpart after truncation by e_1. The function d is the Euclidean distance.

5.3.4. Street-level suppression rate

Aggregated trajectory data is relevant in the context of traffic monitoring and urban planning. Since the utility of these aggregations is dependent on the magnitude and completeness of the trajectory data, the street-level suppression rate, i.e., the relative loss of data for street-level aggregations, serves as another measure of the algorithm's utility.

Consider the number of trajectories of a set D passing a street segment w before and after the S-TT algorithm is applied, denoted by $t_0(w)$ and $t_1(w)$, respectively. For a set of street segments W with $t_0(w) > 0$ for all $w \in W$, the street-level suppression rate $SR(W)$ is calculated as follows:

$$SR(W) = \frac{1}{|W|} \sum_{w \in W} \frac{t_0(w) - t_1(w)}{t_0(w)}. \tag{13}$$

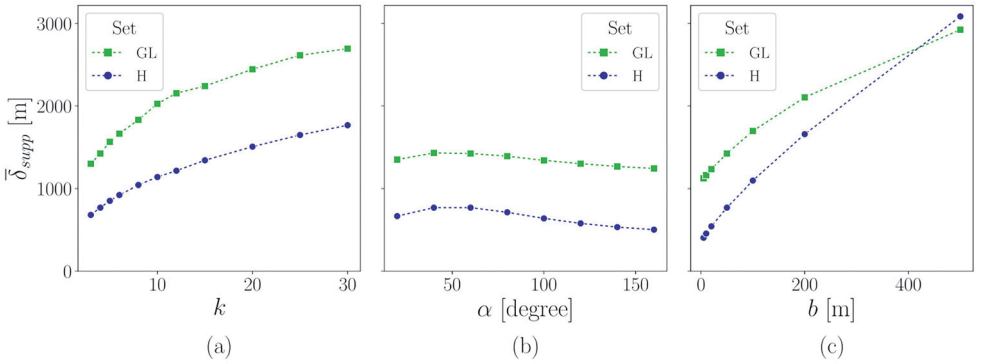

Figure 8. Mean length of suppressed trajectory parts over different parameterizations (a) of k, $\alpha = 60°$, $b = 50$m; (b) of α, $k = 4$, $b = 50$m; (c) of b, $k = 4$, $\alpha = 60°$.

5.4. Results

The outcome of the proximity- and direction-based S-TT algorithm depends on two factors: the configuration of the V-shaped region and the protection sets given as input. In our implementation, these are determined by the opening angle α, the minimum size of the protection set k, and the buffer radius b. The parameters influencing the size of the protection set, k and b, had a significant impact on the results (Figure 8). Raising k or b increased the mean length of suppressed trajectory parts $\bar{\delta}_{supp}$ almost at a linear rate; the increase was faster for small values but stagnated for larger ones. Differences between the datasets, notably the significantly higher base level of $\bar{\delta}_{supp}$(GL), are largely attributable to the much higher number of short trajectories suppressed completely even if k was small: for $\alpha = 60^{\circ}$, $b = 0$ m, $k = 4$, 7% of the trajectories in GL were suppressed completely, compared to none in H. Furthermore, trajectories in GL lingered longer in their protection cells which also tended to be larger. The median size of the protection cells of all endpoints in the Beijing area was 22.9 ha for $k = 4$, $b = 50$ m, but only 7.1 ha in the Helsinki area. The mean distance traveled from an endpoint until exiting the endpoint's protection cell was 646.8 m for the trajectories in GL, but only 195.1 m for the trajectories in H.

Regarding the angle α, the results of both trajectory sets suggest that the parameterizations leading to the highest data loss lie around 40° to 60° Thus, the intuition that raising α will reduce $\bar{\delta}_{supp}$ is only true for $\alpha \geq 80^{\circ}$. The narrower α, the higher the chance of the V-shaped region being too small to include any site (Figure 9). Furthermore, compared to k and b, the parameter α had a significantly smaller impact on $\bar{\delta}_{supp}$. Given the parameters $k = 4$, $\alpha = 60°$, and $b = 0m$, the direction-based truncation criterion, which is controlled by α, did not trigger the suppression of any trajectory points in 24% (H) and 40% (GL) of the cases. In other words, the algorithm stopped truncating the trajectories at the first trajectory point located outside their protection cell. Likewise, the share of trajectory points suppressed due to the direction-based criterion was 38% (H) and 11% (GL). This means that 62% (H) and 89% (GL) of the suppressed trajectory points were located in the protection cells.

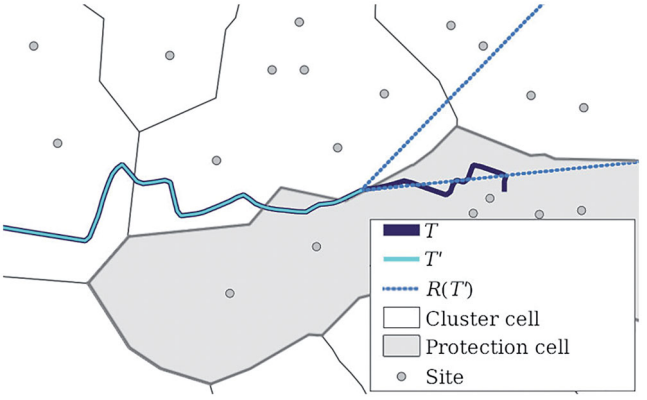

Figure 9. Trajectory T and its truncated counterpart T'. The angle of the V-shaped region $R(T')$ is too small to include any of the sites in the protection set, causing the algorithm to stop immediately upon reaching the first point outside of the protection cell.

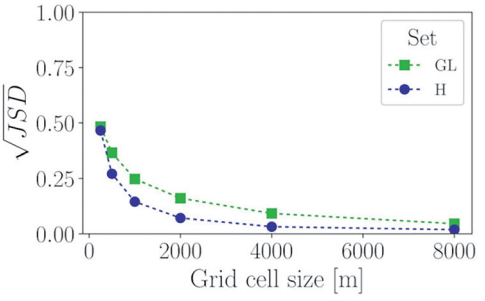

Figure 10. Jensen-Shannon distance $\sqrt{JSD}$ over grid cell size.

In all further analyses, we focus on the parameter values $k = 4$, $\alpha = 60°$, and $b = 50$ m. Under this parameterization, the site clusters are quite small, fulfilling the direction-based condition is comparably hard, and the buffer is large enough to mitigate GNSS errors and other uncertainties.

Moreover, we measured the runtime to provide its magnitude: on a single machine with a 2.4 GHz Intel i7 CPU, the average runtime of our implementation in Java was 0.003 seconds per trajectory.

5.4.1. Preservation of the distribution of endpoints

The Jensen-Shannon distance of the endpoint distribution before compared after the application of S-TT was approximately inversely proportional to the grid cell size r (Figure 10). The GL dataset, however, had a slightly higher base level of distortion. The S-TT algorithm distorted the endpoint distribution of both datasets significantly for high-resolution grids, but for grids with a resolution of $r \geq 2000$ m the Jensen-Shannon distance was smaller than 0.17 for both H and GL. This shows that when examined on a neighborhood or region level, the utility of the trajectory sets for analyses involving the endpoints is largely preserved.

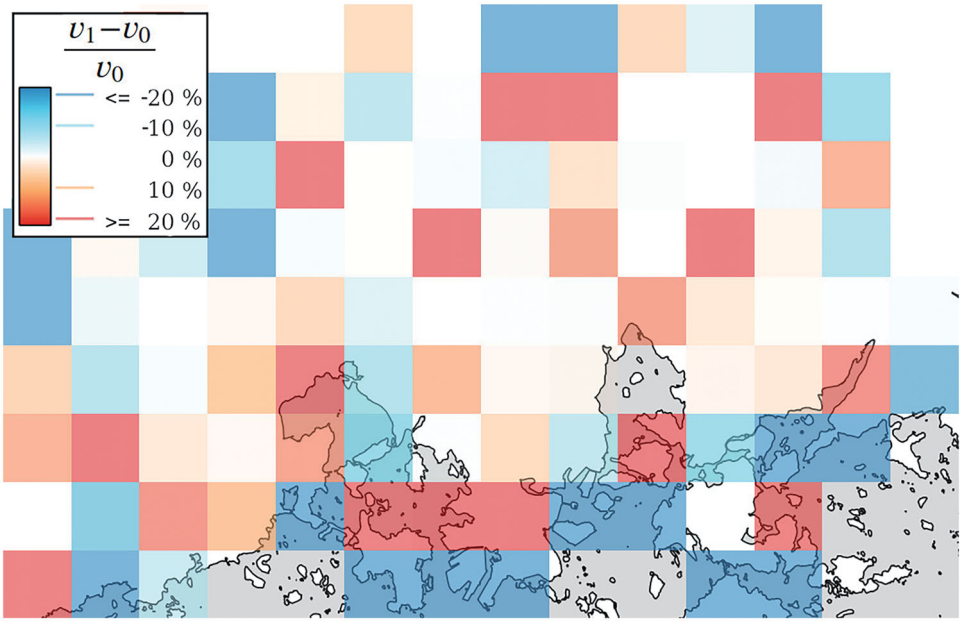

Figure 11. Normalized difference between endpoint count before (v_0) and after truncation (v_1) for the synthetic trajectories (H). Only cells containing more than 20 original endpoints are depicted. Cell size $r = 2000$ m. Negative percentages correspond to a reduced number of endpoints after truncation (blue), positive percentages correspond to an increased number (red). The map shows a coastal area, with the landmass depicted in white and the sea in gray.

Since the endpoints' distribution in H was not biased towards the behavior of a few individuals, we chose the dataset H to inspect the differences in the endpoint counts visually and analyze which areas were affected the most. Grid cells where a large number of the original trajectories' endpoints were located were naturally inclined to exhibit greater changes in the number of endpoints. On the other hand, not all grid cells with a large initial endpoint count lost an accordingly large number of endpoints due to the S-TT algorithm. Regions that lost a high number of endpoints could often be described as dead ends with respect to the road network. Correspondingly, areas that gained significantly were often directly neighboring the aforementioned regions. Since Helsinki is a coastal city, this effect was prominently visible for grid cells along the shoreline (Figure 11).

5.4.2 Utility with respect to urban structures

The results depended on the spatial distribution of geographical features, directly in the case of buildings or other POIs (i.e., the input sites), and indirectly in the case of features that restrict or direct movement, especially the road network. In the experiments, the relationship between the size of the protection cells and the mean relocation distance of the endpoints was close to linear (Figure 12a). This identifies the cell size as the most important determinant of the degree of truncation. The impact of other factors, such as the road network, is reflected in the deviations from linearity.

Large protection cells emerged where buildings were scattered so that trajectories with endpoints in less urbanized locations were truncated more. However, there was

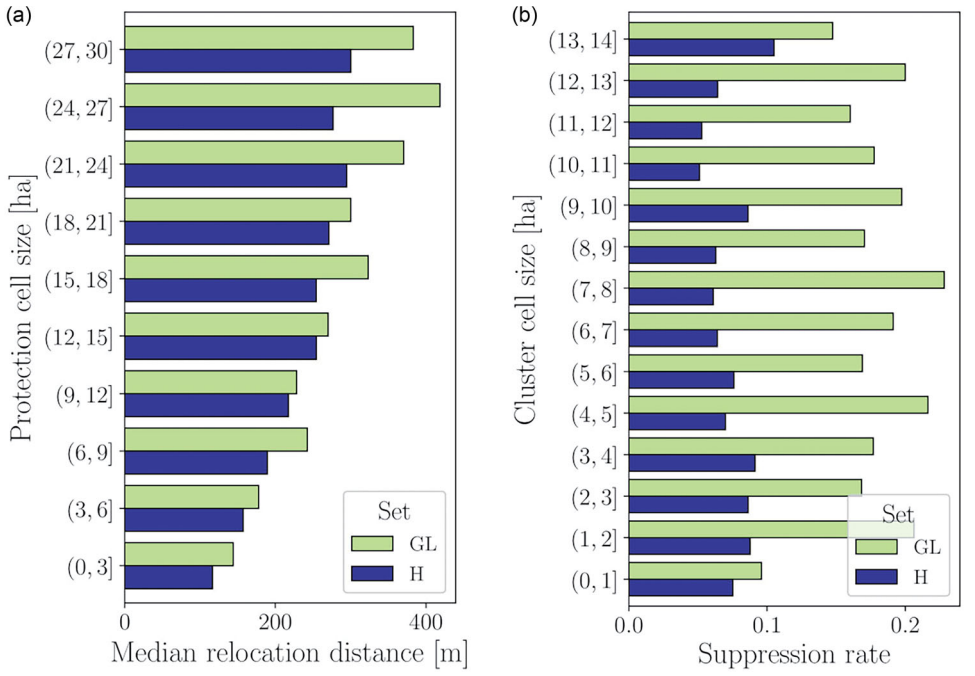

Figure 12. (a): Mean relocation distance of endpoints grouped by the size of their protection cell. (b): Suppression rate of street segments with $t_0(w) \geq 10$ grouped by the size of the cluster cell they fall into.

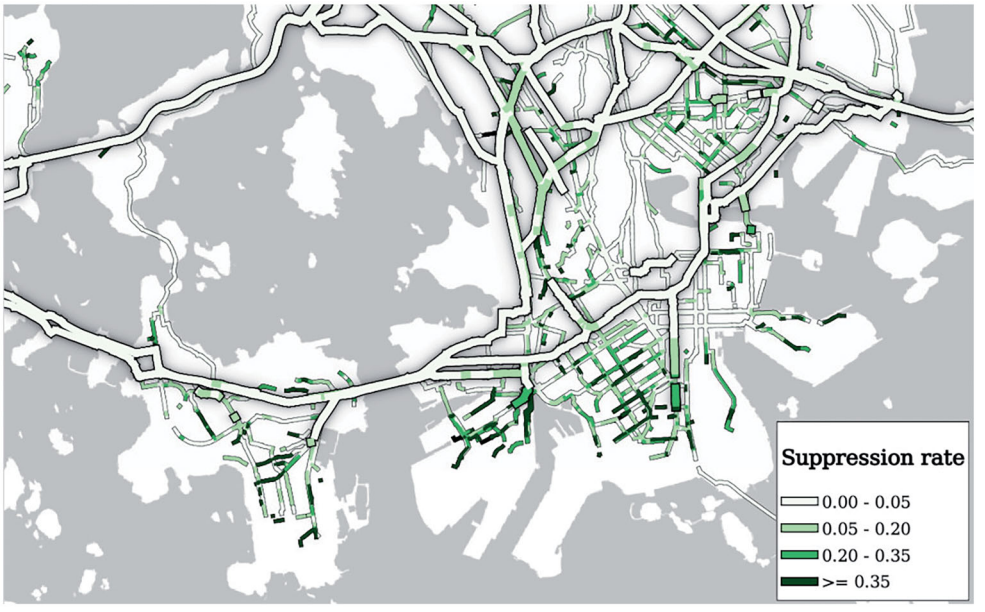

Figure 13. Suppression rate of street segments in central Helsinki. Segments where $t_0(w) \geq 60$ are broad and highlighted, and largely correspond to road links that are important for traffic. Segments where $t_0(w) \leq 5$ are not displayed.

no significant influence of a cluster cell's size on the suppression rate of the street segments contained in it (Figure 12b). Overall, the suppression rate of the GL dataset was consistently higher than the suppression rate of dataset H: the suppression rate of all street segments w with $t_0(w) \geq 10$ was 0.074 for H but 0.167 for GL.

The most significant data loss occurred in areas where many trajectories started or ended but only very few passed through, especially residential areas. The suppression rate for major traffic routes was comparably small (Figure 13).

6. Discussion

In the following, we discuss the results of the experimental study with respect to the utility, opportunities and challenges of using S-TT for privacy-preserving trajectory publication or in location-based services, and the conceptual strengths and weaknesses of the algorithm.

6.1. Utility of the S-TT algorithm

The results of the experimental study underline how S-TT preserves the utility of trajectory data, but they also point out aspects that need to be taken into account when publishing and analyzing trajectories protected with S-TT. Consider a scenario in which the proximity- and direction-based S-TT algorithm is used to protect all stay points in a set of high-sampling-rate trajectories. Based on the results, the data loss on streets that are most important for transit is expected to be negligible, yet areas that are not commonly passed through (e.g., residential areas, industrial parks, and shopping areas) may lose a significant share of trajectory data. Naturally, this should not only hold for S-TT, but all LPPMs that protect stay points by suppression or generalization. In S-TT, the degree of suppression is primarily controlled by the size of the protection set. To maximize utility, the protection sets should be as small as possible given the privacy requirements. The accuracy of detailed origin-destination matrices, for example, largely depends on the size of the protection sets as our results suggest, although neighborhood-level matrices or coarser are less affected.

The differences in the results between the two trajectory sets highlight the importance of testing with diverse trajectory sets and raise the question of generalizability. The Geolife dataset's number of participants is small compared to its size so the trajectories are not well distributed but heavily biased towards some locations and routes. The synthetic trajectory dataset, on the other hand, is biased towards longer trips, which results in a lower relative data loss. Additionally, it gravely underestimates the popularity of some routes since it was created considering only the shortest route between two locations. For comparison, we refer to Brauer *et al.* (2021), who analyze a real-world trajectory set in the same area. Being aware of these biases, we conclude that the observations of the experiments can be generalized regarding the tendencies of the utility measures. Their concrete magnitude, however, largely depends on the trajectories' properties, such as their distribution, cleanliness, sampling rate, and length.

6.2. S-TT in practical applications

The S-TT algorithm can be applied in several ways: to hide the origin and destination of a trajectory, to hide all stay points of a trajectory, or to hide a pre-defined set of sensitive locations. If the location that should be protected with S-TT is not an endpoint but an intermediate stop, the trajectory can be split at the stop, and S-TT can be applied to both trajectory halves before reassembling them.

Consider a scenario in which individuals collect trajectories that are then published by a service provider. Before publication, the trajectories are to be protected with S-TT. In this setting, S-TT could be executed by the provider or a trusted third party, but since S-TT processes trajectories independently of each other, there is also the possibility of executing it directly on the client side. For each trajectory, the client requires the protection sets to hide the sensitive locations. The client could provide coarse locations to obtain relevant protection sets from a server, or decide on the protection sets themselves. It should be noted that the algorithm, its parameters, and also the protection sets, given that they do not reveal anything about the sensitive locations, can be disclosed safely; *k*-site-unidentifiability would still be guaranteed.

Using a global site partition as the basis for the definition of all protection sets of a trajectory dataset is one way to prevent protection set-based reverse-engineering attacks. However, our experiments also introduced a buffer-based approach to handle uncertainty. This extension can theoretically allow an attacker in possession of the site partition to run successful protection set-based reverse-engineering attacks. For example, an attacker might be able to breach *k*-site-identifiability by inferring that a destination lies within a cluster cell but not close to its boundary, as otherwise, the buffer-based approach would have included adjacent cells in the protection cell as well. In practice, this risk is usually acceptable; if it cannot be tolerated, an alternative way to handle uncertainty is to define a trajectory's protection set using all sites within the cluster cell containing the endpoint and all cells surrounding it. This approach takes a higher toll on utility but guarantees *k*-site-unidentifiability in all cases and handles uncertainty just as effectively. Moreover, besides addressing uncertainty, these extensions mitigate deficits of the auxiliary data. Otherwise, unknown semantic connections between sites located in different cluster cells can cause insufficient truncation of trajectories leading to them, e.g., if the protection set contains some buildings of a hospital complex but the truncated trajectory still suggests a visit to the hospital.

Finally, another practical extension for handling noisy trajectories could improve the noise tolerance of testing the direction-based condition by using more than two trajectory points to calculate the direction.

6.3. Strengths and limitations of the S-TT algorithm

The S-TT algorithm was developed with the intention to create a mechanism for the protection of stay points that is simple, utility-preserving, and context-sensitive. The consideration of the spatial context, in this case, sites, is the main advantage of S-TT over otherwise similar simple suppression-based techniques. S-TT adapts the amount of suppressed data according to the presence of enough plausible destination sites.

Thus, it can protect destination sites in sparsely built areas as reliably as in densely built environments. This is in contrast to, e.g., the circle-based suppression method used by Krumm (2007), in which the circle size is determined by a fixed parameter. Furthermore, the experiments confirmed that our implementation of S-TT succeeds in limiting the possibilities of deducing the destination through visual reasoning. Humans excel in making deductions based on the spatial context, taking clues such as the proximity of nearby buildings and the direction of the trajectory into account. However, it was not in the scope of this study to investigate this finding beyond the members of our group.

Our proximity- and direction-based implementation of S-TT does not exhaust all possibilities of utilizing context information. For instance, it does not consider the semantics of the sites, e.g., whether a site is a public space. Such data could be used to define semantically diverse protection sets, and this diversity could be guaranteed by extending the *k*-site-unidentifiability criterion according to the principle of *ℓ-diversity* (Machanavajjhala *et al.* 2007). There is also a temporal aspect to sites that could be incorporated. Since the accessibility of some sites is temporally restricted (e.g., shopping malls), protection sets could be required to contain at least *k* sites that were accessible at the time of visit. Moreover, another way to utilize context information is to design a version of S-TT that takes the road network into account.

The S-TT algorithm controls the amount of suppressed trajectory points according to specific attack functions. Our implementation featured two, but the possibilities to explore others are endless. For example, an attack could consider the *n*-nearest sites to the endpoint. It should be noted that formally, S-TT cannot guarantee k-site-unidentifiability for any attack besides those that were used to control the suppression process. However, the success rate of any destination inference attack targeting a set of trajectories truncated with the proximity- and direction-based S-TT algorithm is expected to be reduced significantly, given that the attacker does not have any other additional knowledge.

It can be argued that the protection of certain sensitive locations, e.g., the home or workplace, also reduces the risk of re-identification. Nevertheless, S-TT does not necessarily prevent re-identification attacks, as the attacker could possess knowledge of the individual's whereabouts that were not considered sensitive. Additionally, home and work location pairs have been shown to be very unique, even on a coarse level (Golle and Partridge 2009).

7. Conclusion

In this paper, we present *S-TT*, an algorithm that protects sensitive locations in spatial trajectories. S-TT iteratively suppresses trajectory points that lead to the sensitive location until *k-site-unidentifiability* is guaranteed. *K*-site-unidentifiability is introduced as an adoption of the *k*-anonymity paradigm for sites, which requires that at least *k* sites could have been plausibly visited by the trajectory. We show how S-TT can ward off attacks aiming to identify sensitive locations based on geometric clues: the trajectory's

proximity to nearby sites, and its direction. S-TT is utility-preserving, as it maximizes the retained data.

Since S-TT can be executed on each trajectory separately, we consider it a helpful tool for individuals who would like to publish their trajectories online but are concerned about their privacy. Publishers of trajectory data could consider using S-TT to protect, for example, the home address of individuals, as reliably in sparsely as in densely built environments.

In future work, we plan to extend S-TT to take the road network into account as further context information. Moreover, k-site-unidentifiability may be expanded using the concepts of ℓ-diversity and t-closeness, so that S-TT could also prevent attribute linkage attacks.

Author Contributions

During a research visit of Jan-Henrik Haunert at FGI, he, Juha Oksanen, and Ville Mäkinen conceived the general idea and created a first design of the algorithm. Based on a prototypical implementation in Java by Jan-Henrik Haunert, Anna Brauer implemented the algorithm in Python and designed as well as conducted the experiments in consultation with Ville Mäkinen and Juha Oksanen. She also wrote the manuscript, incorporating short text passages as well as comments contributed by all authors.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Finnish Cultural Foundation under Grant [00200814] and the German Research Foundation under Grant [Ha 5451/7-1].

Notes on contributors

Anna Brauer's research is concerned with spatio-temporal analyses of human mobility data and focuses on privacy-preserving data publishing and data mining. In 2020, she received her diploma in computer science from the Dresden University of Technology, Germany. Currently, she is working towards her Ph.D. at the University of Helsinki, Finland, and carries out her research in the Department of Geoinformatics and Cartography at the Finnish Geospatial Research Institute, National Land Survey of Finland.

Dr. Ville Mäkinen received his Ph.D. degree from the University of Jyväskylä in theoretical physics in 2013. He is currently a senior researcher in the Department of Geoinformatics and Cartography at the Finnish Geospatial Research Institute FGI, which is part of the National Land Survey of Finland (NLS). His research interests include algorithm development, especially on parallel architectures, analyses related to digital elevation models, including hydrological analyses, and recently human mobility data and the various issues related to working with such data.

Axel Forsch finished his Master of Science in Geodesy and Geoinformation at the University of Bonn in 2019. Currently, he continues his studies at University of Bonn within the frame of a PhD program regarding Volunteered Geographic Information. His research is focused on the algorithmic analysis and visualization of movement patters. A central topic of Axel Forsch's research is to infer routing preferences from trajectories recorded by cyclists.

Professor Juha Oksanen is the Head of the Department of Geoinformatics and Cartography at the Finnish Geospatial Research Institute, National Land Survey of Finland. He received his Ph.D. degree in geography from the University of Helsinki, where he currently also acts as an Adjunct Professor of geoinformatics. His research interests include cartography, geovisualisation, location privacy and analysis of large spatio-temporal datasets.

Jan-Henrik Haunert holds a diploma and doctoral degree in geodesy and geoinformatics from the University of Hannover, Germany. He has been a postdoctoral researcher at the institute of computer science at the University of Würzburg, Germany, and a professor for geoinformatics at the University of Osnabrück, Germany. In 2016, he took up a full professorship for geoinformation at the University of Bonn, Germany. His research is concerned with the development of efficient algorithms for geovisualization and spatial analysis. In particular, he applies methods from combinatorial optimization and computational geometry to tasks in automated cartography, such as map generalization and cartographic label placement.

ORCID

Anna Brauer http://orcid.org/0000-0002-7092-1492
Ville Mäkinen http://orcid.org/0000-0002-7887-5646
Axel Forsch http://orcid.org/0000-0002-3849-4865
Juha Oksanen http://orcid.org/0000-0002-3381-9763
Jan-Henrik Haunert http://orcid.org/0000-0001-8005-943X

Data and codes availability statement

The data and code that support this research are available at https://doi.org/10.6084/m9.figshare.15156759.

References

Abul, O., Bonchi, F., and Nanni, M., 2008. Never walk alone: uncertainty for anonymity in moving objects databases. *In*: *2008 IEEE 24th International Conference on Data Engineering*. Cancun, Mexico, 376–385.

AllTrails. 2021. Available from: https://www.alltrails.com [Accessed: 13 August 2021].

Bike Data Project. 2021. Available from: https://www.bikedataproject.org. [Accessed 13 August 2021]

Bonomi, L., and Xiong, L., 2013. A two-phase algorithm for mining sequential patterns with differential privacy. In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management,* 269–278.

Brauer, A., Mäkinen, V., and Oksanen, J., 2021. Characterizing cycling traffic fluency using big mobile activity tracking data. *Computers, Environment and Urban Systems*, 85, 101553.

Chen, Q., and Poorthuis, A., 2021. Identifying home locations in human mobility data: an open-source R package for comparison and reproducibility. *International Journal of Geographical Information Science*, 35 (7), 1425–1448.

Chen, R., Acs, G., and Castelluccia, C., 2012. Differentially private sequential data publication via variable-length n-grams. In*: Proceedings of the 2012 ACM conference on Computer and communications security*. Raleigh, North Carolina, USA, 638–649.

Cicek, A.E., Nergiz, M.E., and Saygin, Y., 2014. Ensuring location diversity in privacy-preserving spatio-temporal data publishing. *The VLDB Journal*, 23 (4), 609–625.

Dai, Y., *et al.*, 2018. Personalized semantic trajectory privacy preservation through trajectory reconstruction. *World Wide Web*, 21 (4), 875–914.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 (1), 269–271.

Dong, Y., and Pi, D., 2018. Novel privacy-preserving algorithm based on frequent path for trajectory data publishing. *Knowledge-Based Systems*, 148, 55–65.

Dwork, C., 2008. Differential privacy: a survey of results. *In: 2008 International conference on theory and applications of models of computation*. Xi'an, China: Springer, 1–19.

Endres, D.M., and Schindelin, J.E., 2003. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49 (7), 1858–1860.

Fan, Z., *et al.*, 2015. Citymomentum: an online approach for crowd behavior prediction at a citywide level. *In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Osaka, Japan, 559–569.

Fiore, M., *et al.*, 2020. Privacy in trajectory micro-data publishing: a survey. *Transactions on Data Privacy*, 13, 91–149.

Ganta, S.R., Kasiviswanathan, S.P., and Smith, A., 2008. Composition attacks and auxiliary information in data privacy. *In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. Las Vegas, Nevada, USA, 265–273.

Golle, P., and Partridge, K., 2009. On the anonymity of home/work location pairs. *In: 2009 International conference on pervasive computing*. Nara, Japan: Springer, 390–397.

Goodchild, M.F., 2007. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69 (4), 211–221.

Gursoy, M.E., *et al.*, 2019. Differentially private and utility preserving publication of trajectory data. *IEEE Transactions on Mobile Computing*, 18 (10), 2315–2329.

Han, P.I., and Tsai, H.P., 2015. SST: Privacy preserving for semantic trajectories. *In: 2015 16th IEEE International Conference on Mobile Data Management*. Pittsburgh, Pennsylvania, USA: IEEE, vol. 2, 80–85.

Haunert, J.H., Schmidt, D., and Schmidt, M., 2021. Anonymization via clustering of locations in road networks. *In: 11th International Conference on Geographic Information Science (GIScience 2021) - Part II Short Paper Proceedings*. Poznan, Poland.

Hoh, B., *et al.*, 2006. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5 (4), 38–46.

Huo, Z., *et al.*, 2012. You can walk alone: trajectory privacy-preserving through significant stays protection. *In:* S. Lee, Z. Peng, X. Zhou, Y.S. Moon, R. Unland and J. Yoo, eds. *Database systems for advanced applications, lecture notes in computer science*. Berlin, Heidelberg: Springer, 351–366.

Imielińska, C., Kalantari, B., and Khachiyan, L., 1993. A greedy heuristic for a minimum-weight forest problem. *Operations Research Letters*, 14 (2), 65–71.

Keßler, C., and McKenzie, G., 2018. A geoprivacy manifesto. *Transactions in GIS*, 22 (1), 3–19.

Krumm, J., 2007. Inference attacks on location tracks. *In*: *2007 International Conference on Pervasive Computing*. Toronto, Canada: Springer, 127–143.

Kullback, S., and Leibler, R.A., 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22 (1), 79–86.

Kwan, M.P., Casas, I., and Schmitz, B., 2004. Protection of geoprivacy and accuracy of spatial information: How effective are geographical masks? *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39 (2), 15–28.

Li, N., Li, T., and Venkatasubramanian, S., 2007. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering. Istanbul, Turkey: IEEE, 106–115.

Lin, J., 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37 (1), 145–151.

Lu, W., Scott, D.M., and Dalumpines, R., 2018. Understanding bike share cyclist route choice using GPS data: comparing dominant routes and shortest paths. *Journal of Transport Geography*, 71, 172–181.

Machanavajjhala, A., *et al.*, 2007. ℓ-diversity: privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1 (1), 3.

Mazimpaka, J.D., and Timpf, S., 2016. Trajectory data mining: a review of methods and applications. *Journal of Spatial Information Science*, 13, 61–99.

Mir, D.J., *et al.*, 2013. Dp-where: differentially private modeling of human mobility. *In: 2013 IEEE international conference on big data*. Santa Clara, California, USA: IEEE, 580–588.

Monreale, A., *et al.*, 2010. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3 (2), 91–121.

Naghizade, E., Kulik, L., and Tanin, E., 2014. Protection of sensitive trajectory datasets through spatial and temporal exchange. In*: Proceedings of the 26th International Conference on Scientific and Statistical Database Management*. Aalborg, Denmark, 1–4.

Nergiz, M.E., Atzori, M., and Saygin, Y., 2008. Towards trajectory anonymization: a generalization-based approach. In: *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*. Irvine, California, USA, 52–61.

OpenStreetMap. 2021. Available from: https://www.openstreetmap.org [Accessed: 13 August 2021].

Pappalardo, L., *et al.*, 2019. Scikit-mobility: a python library for the analysis, generation and risk assessment of mobility data. Available from: https://arxiv.org/abs/1907.07062 [Accessed 30 May 2022].

Pensa, R.G., *et al.*, 2008. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. *In: International Workshop on Privacy in Location-Based Applications PiLBA'08*. Malaga, Spain: CEUR-WS.org, vol. 397, 44–60.

Primault, V., *et al.*, 2015. Time distortion anonymization for the publication of mobility data with high utility. *In: 2015 IEEE Trustcom/BigDataSE/ISPA*. Helsinki, Finland: IEEE, vol. 1, 539–546.

Primault, V., *et al.*, 2019. The long road to computational location privacy: A survey. *IEEE Communications Surveys & Tutorials*, 21 (3), 2772–2793.

Rossi, L., Walker, J., and Musolesi, M., 2015. Spatio-temporal techniques for user identification by means of GPS mobility data. *EPJ Data Science*, 4 (1), 11. https://doi.org/10.1140/epjds/s13688-015-0049-x

Salas, J., *et al.*, 2020. Swapping trajectories with a sufficient sanitizer. *Pattern Recognition Letters*, 131, 474–480.

Seidl, D.E., Jankowski, P., and Tsou, M.H., 2016. Privacy and spatial pattern preservation in masked GPS trajectory data. *International Journal of Geographical Information Science*, 30 (4), 785–800.

Shokri, R., *et al.*, 2011. Quantifying location privacy. *In: 2011 IEEE symposium on security and privacy*. Oakland, California, USA: IEEE, 247–262.

Song, Y., Dahlmeier, D., and Bressan, S., 2014. Not so unique in the crowd: A simple and effective algorithm for anonymizing location data. *CEUR Workshop Proceedings*, 1225, 19–24.

Strava, 2021. Available from: https://www.strava.com [Accessed: 13 August 2021].

Sweeney, L., 2002. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10 (05), 557–570.

Terrovitis, M., *et al.*, 2017. Local suppression and splitting techniques for privacy preserving publication of trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 29 (7), 1466–1479.

Tu, Z., *et al.*, 2019. Protecting trajectory from semantic attack considering k-anonymity, ℓ-diversity, and t-closeness. *IEEE Transactions on Network and Service Management*, 16 (1), 264–278.

Valhalla, 2021. Available from: https://github.com/valhalla [Accessed: 13 August 2021].

Wang, N., and Kankanhalli, M.S., 2020. Protecting sensitive place visits in privacy-preserving trajectory publishing. *Computers & Security*, 97, 101949.

Wong, R.C.W., *et al.*, 2011. Can the utility of anonymized data be used for privacy breaches? *ACM Transactions on Knowledge Discovery from Data*, 5 (3), 1–24.

Yarovoy, R., *et al.*, 2009. Anonymizing moving objects: how to hide a mob in a crowd? *In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. Saint Petersburg, Russia, 72–83.

Zhang, P., *et al.*, 2018. Trajectory-driven influential billboard placement. *In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. London, UK, 2748–2757.

Zheng, Y., *et al.*, 2009. Mining interesting locations and travel sequences from GPS trajectories. *In: Proceedings of the 18th international conference on World wide web*. Madrid, Spain, 791–800.

Zheng, Y., *et al.*, 2011. Geolife GPS trajectory dataset – user guide. Available from: https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide [Accessed 13 August 2021].