On the Inference of User Paths from Anonymized Mobility Data

On the Inference of User Paths from Anonymized Mobility Data

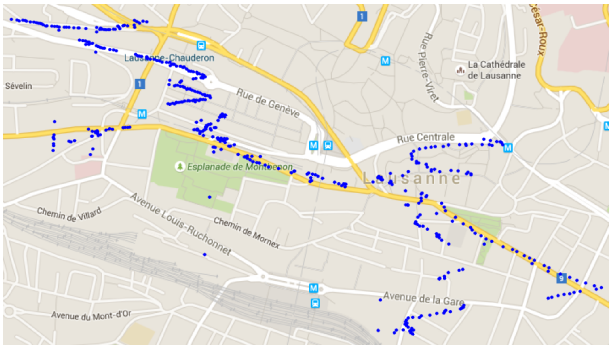
Galini Tsoukaneri*, George Theodorakopoulos†, Hugh Leather*, Mahesh K. Marina*
*School of Informatics, The University of Edinburgh
†School of Computer Science and Informatics, Cardiff University

***Abstract*—Using the plethora of apps on smartphones and tablets entails giving them access to different types of privacy sensitive information, including the device's location. This can potentially compromise user privacy when app providers share user data with third parties (e.g., advertisers) for monetization purposes. In this paper, we focus on the interface for data sharing between app providers and third parties, and devise an attack that can break the strongest form of the commonly used anonymization method for protecting the privacy of users. More specifically, we develop a mechanism called Comber that given completely anonymized mobility data (without any pseudonyms) as input is able to identify different users and their respective paths in the data. Comber exploits the observation that the distribution of speeds is typically similar among different users and incorporates a generic, empirically derived histogram of user speeds to identify the users and disentangle their paths. Comber also benefits from two optimizations that allow it to reduce the path inference time for large datasets. We use two real datasets with mobile user location traces (Mobile Data Challenge and GeoLife) for evaluating the effectiveness of Comber and show that it can infer paths with greater than 90% accuracy with both these datasets.**

1. Introduction

Besides the sophisticated sensing capabilities (location etc.) and the ubiquitous connectivity they enable, one of the most appealing aspects of smartphones and tablets is the incredibly large number of apps they make available to the users. As of February 2015 there are 1.4 million apps on the Google Play Store for Android based mobile devices [1]. On the flip side, device owners are requested to give consent to the apps for accessing certain phone features (e.g., network, location, device sensors), potentially risking their privacy. Some apps such as Brightest Flashlight [2] seek access to information they do not need, purely for collecting and monetizing user data. Of relevance to this paper is the device location which implicitly holds information about user context and identification. The authors of [3] find that a third of the most popular free Android apps request access to location.

Sharing user location and other data to advertisers is a popular form of app monetization [4]. While such data collection and sharing is predominantly associated with free apps, the practice is also widespread in paid apps [5].



(a) *Completely anonymized mobility data*

(b) *Inferred paths*

Figure 1: *Illustration of user path inference with completely anonymized mobility data. Input is shown in sub-figure (a) in which paths of all users are jumbled up and indistinguishable. Result of path inference is shown in sub-figure (b) in which each color and shape represents the path of a different user.*

A well-cited research study [6] showed that half of the tested apps reported user location to advertisers. Especially if we consider that most mobile apps come without any privacy policies, such unscrupulous data sharing can have serious privacy implications. Recently, there has been pressure on app developers to publish their privacy policies [5]. While privacy policies lay out an explicit framework for the use of collected data, they do not preclude data sharing. *Anonymization* is a widely used method for preventing identification of users and protecting their privacy when app providers share data with third parties (including advertisers) [7], [8], [9], [10].

In this paper, we investigate the robustness of anonymization for protecting mobile user identification and mobility patterns. As simple anonymization is proven to be vulnerable [11], we consider a stronger form of anonymization that we refer to as *complete anonymization*. In simple anonymization user-identiyfing information is replaced by random values (usually numerical) called pseudonyms. In complete anonymization any user-identifying information (including pseudonyms) are removed from the data. The aim of this research is to infer the user paths from a completely anonymized dataset consisting of location-time pairs. Fig. 1 illustrates our problem on a map with mobility data belonging to multiple users. Our input is a completely anonymized dataset shown in Fig. 1 (a) while our objective is to recover the user paths as shown in Fig. 1 (b). Fig. 2 shows a different illustration of our problem, treating the mobility data of users as a database table.

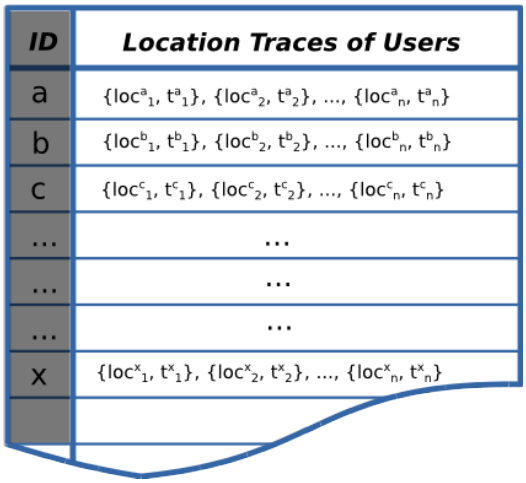


Figure 2: *Path inference problem illustrated from a database viewpoint. Completely anonymized input data would essentially be stored in a database without IDs. Location traces (paths) in different rows belonging to different users would then be indistinguishable. Our aim is to disentangle these paths and, in effect, de-anonymize the data. In other words, solving our problem entails separating the paths from the given data and associating a different pseudonym for each of them.*

Towards this end, we develop a novel mechanism called Comber. Comber exploits the observation that mobility patterns of users are similar, in terms of distribution of speeds, which allows for the use of a generic and empirically derived mobility model. It then uses the mobility model to associate a probability of moving from one location to another in a time instant. To limit the computational overhead and make the inference practical, Comber features two optimizations: (i) the minimum probability threshold and (ii) the early rejection. Using two real datasets (MDC [12] and GeoLife [13]) with around 180 users in each, we evaluate Comber focusing on its accuracy and computation time. Our results show that Comber can infer paths with greater than 90% accuracy and under 5 minutes (on a typical workstation) for the datasets used.

To put Comber in the context of the wider location privacy literature, it is an *attack* to infer different users and their paths from completely anonymized mobility datasets. Broadly speaking, location privacy attacks and defenses can be divided into two categories: (i) user-side and (ii) provider-side. The schematic in Fig. 3 helps clarify the difference between these two categories. The majority of the location privacy literature is focused on the user side, whereas *our focus is on the provider side*. As mentioned above, anonymization is a commonly used privacy protection method employed on the provider side for data sharing with third parties. Through Comber, we show the limitations of this method in the context of location privacy, considering complete anonymization (the strongest form of anonymization).

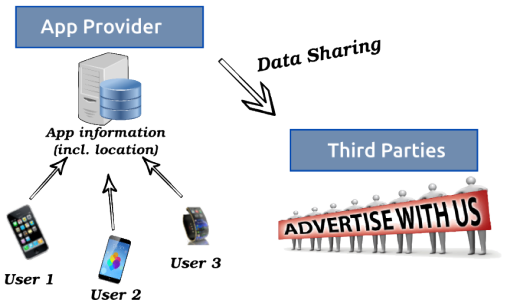


Figure 3: *Schematic showing different interacting entities from the perspective of privacy of a mobile user running an app: users running the app on their devices, the app provider, and the third parties (including advertisers). Our focus is on the interface between the app provider and third parties.*

The rest of the paper is organized as follows. Section 2 outlines the two different datasets used in this research and their characteristics. In Section 3 we provide a detailed description of the Comber algorithm. Section 4 presents our evaluation results along with their discussion in Section 5. In Section 6, we review the related work. Section 7 concludes.

2. Datasets

For this research we use two well-known mobility datasets with GPS traces of mobile phone users; Mobile Data Challenge (MDC) [12] and GeoLife [13]. Both these datasets provide pseudonymized location traces of people. Please note that we require datasets that include pseudonyms purely for evaluation purposes. Datasets that are shared in practice are likely to be completely anonymized without any pseudonyms included.

2.1. Mobile Data Challenge (MDC) Dataset

MDC dataset [12] includes movement and behavioral information for 185 individuals in the area of Lake Geneva

in Switzerland. In this research we only use a part of the MDC dataset. Specifically, we are interested in the location information, provided by GPS. The GPS information is in the form of <*User_ID, timestamp, latitude, longitude*>, and spans a period of a year and a half, from the 1st September 2009 until the 31st of March 2011. The total number of location points is almost 12 million, however, not all users are present on all days. The maximum number of users present on the same day is 101. Moreover, the rate at which users provide a new location point is not constant. The shortest interval between two consecutive location points of the same user is 10 seconds, and the longest is as large as several days. The original dataset includes unique user IDs in the form of pseudonyms, meaning that the identifiable information of the users, such as the name, have been replaced by a random numeric value. All pseudonyms were removed during testing so that there is no information related to the points as to which user they belong. The pseudonyms were only used during evaluation. Further details about the dataset and its collection process can be found in [14].

2.2. GeoLife Dataset

GeoLife dataset [13] is provided by Microsoft Research Asia and includes GPS traces of 182 users. It spans a period of over five years and the participants are mainly located in Beijing. The GPS information includes timestamps and location coordinates in the form latitude and longitude. The spatial and temporal resolution is six decimal digits and one second respectively. Although the data in the GeoLife dataset are recorded over a longer period compared to the MDC dataset, the total number of location points is smaller than that of MDC.

While MDC dataset includes only walking traces of people who contributed to the dataset, GeoLife includes traces of people using additional transporation modes. For consistency and simplicity, we focus on the walking traces of the GeoLife dataset.

For our evaluations, we use a portion of MDC as training data to derive user mobility model (discussed later), and we evaluate Comber on the remainder of the MDC dataset as well as the whole of GeoLife dataset.

3. Comber Path Inference Algorithm

In this section, we describe Comber, our proposed path inference algorithm. The goal of Comber is to *comb through* a completely anonymized mobility dataset in order to identify and distinguish between underlying user paths and give each of them a pseudonym. In other words, we aim to convert a completely anonymized dataset into a simply anonymized one with pseudonyms. Although our algorithm does not attempt to reveal the users' real identities, there exists prior work [15], [16], [17], [18], [19] that can determine the real user identities from simply anonymized location traces.

Before we go into the details of Comber, we start with a few useful definitions.

3.1. Definitions

Anonymized location point: We define an *anonymized location point* as a triplet $\mathcal{C}_i = \{t_i, lat_i, lon_i\}$, where t_i, lat_i, lon_i are the timestamp, latitude and longitude of that point respectively.

Users: We consider that n users exist in the dataset, where $n \geq 1$. We define the set of users as $\mathcal{U} = \{u_1, ..., u_n\}$. The number n is unknown.

Inferred Path: An *inferred path* $\mathcal{P}$ of a user is the sequence of anonymized location points $\mathcal{C}_1$, $\mathcal{C}_2$, ..., $\mathcal{C}_m$ such that $t_1 < t_2 < ... < t_m$, all of which have been inferred to belong to the same user.

3.2. Overview

In terms of the above definitions, our problem is as follows: *Given a set of anonymized location points, we want to identify the users and infer their respective paths.*

Comber addresses this problem in two steps: (i) generating a user mobility model and (ii) reconstructing the paths of the users based on the mobility model.

1) **User Mobility Model:** The purpose of the user mobility model is to describe the *typical* user movement. In our work, this model is generic that applies to all users and is based only on the speeds; it gives the probability of a user moving with some speed *v*. We describe our approach to model user mobility in section 3.3.
2) **Path Reconstruction:** Reconstructing the paths of the users is the process of associating anonymized location points with users by creating/extending paths for each user. The sequence of points increasing in time and associated with a given user make up the path of that user. We use the mobility model generated in the previous step to find the most likely/probable association of location points to users.

Our two-step approach stems from the observation that movement of users is similar in terms of distribution of speeds. To demonstrate that this is true, we consider the MDC dataset and first generate a histogram of speeds for each of the 185 users separately. We then perform a histogram intersection for each pair of users. The histogram intersection [20] quantifies the similarity between two histograms and can take values ranging from 0 to 1; value 0 indicates that the intersected histograms are totally different while value 1 indicates that the histograms are identical. Fig. 4 shows the CDF of the histogram intersection values for each pair of users in the MDC dataset. We see that in more than 90% of the cases, the histogram intersection value is greater than 0.75, indicating a high degree of similarity. The minimum value itself is 0.5419 which suggests that speeds with which the users move are more similar than different. This observation allows us to employ a common mobility model to guide the path inference process.

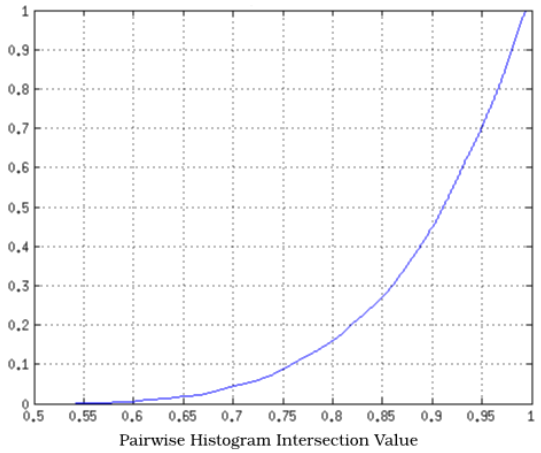


Figure 4: *CDF of histogram intersection values for each pair of users in the MDC dataset. For each user, a histogram of his speeds is used. Higher values of histogram intersection indicate high degree of similarity. We can see that in more than 90% of the cases, the histogram intersection value is greater than 0.75.*

3.3. User Mobility Model

We take an empirical approach to model user mobility based on a histogram of user speeds. More specifically, we assume the availability of a training set with a small number of users, and use it to generate a histogram of speeds. To obtain different speeds for each user in the training set, we calculate the Haversine distance of the latitude and longitude between two consecutive points of the same user. The Haversine distance is the distance between the latitude and longitude of two points on the Earth, and it is given by:

$$d_{ij} = 2r \arcsin\left(\left(\sin^2\left(\frac{lat_j - lat_i}{2}\right) + \cos(lat_i)\cos(lat_j)\sin^2\left(\frac{lon_j - lon_i}{2}\right)\right)^{\frac{1}{2}}\right) \quad (1)$$

where lat_i, lat_j and lon_i, lon_j are the latitude and longitude for two points C_i and C_j respectively and r is the radius of the Earth.

The speed needed to move from point C_i to point C_j is defined as:

$$v_{ij} = \frac{d_{ij}}{t_j - t_i} \quad (2)$$

Therefore, the empirical histogram of speeds gives the probability of a speed v_{ij} defined as $p(v_{ij})$.

The mobility model based on an empirical histogram is shown in Fig. 5, using a training set of 20 users, randomly chosen from the MDC dataset. In section 4, we show that a training set with a limited number of users is sufficient, because of the similarities between users in terms of their distribution of speeds (subsection 3.2).

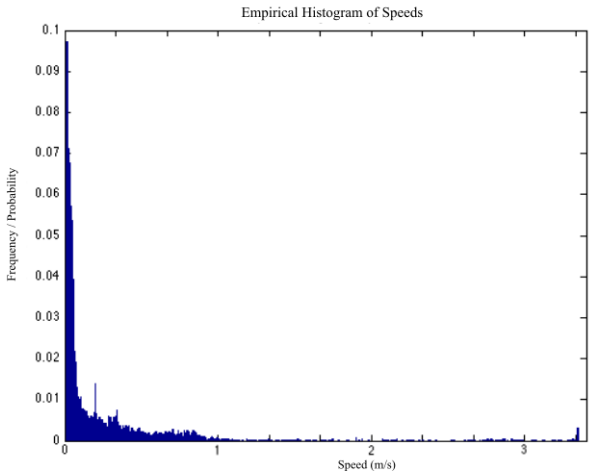


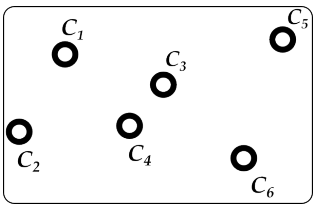
Figure 5: *Empirical histogram based mobility model. The x-axis shows the different speeds in m/s and the y-axis indicates the corresponding frequencies of occurence, or alternatively their probabilities.*

3.4. Path Reconstruction

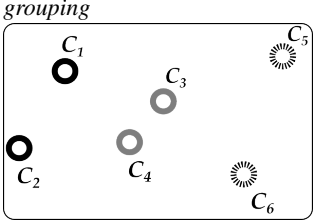
3.4.1. Time-based Grouping. Since the input dataset may have a very large number of anonymized location points, it may not be computationally feasible to naively associate every pair of points to determine if they belong to the same user. For example, the MDC dataset consists of almost 12 million points. Therefore, we first round the timestamps down to the closest multiple of 10 and group together all location points with the same timestamp: for example, timestamps that range from second 0 until second 9 have been rounded down to the timestamp value of 0, timestamps that range from second 10 until second 19 have been rounded down to the timestamp value of 10, and so on. We call these groups as into *time-groups*. Since the shortest time interval that a user provided a trace is 10 seconds, the time-groups of 10 seconds ensure that there will be no more than one point of the same user in a time-group. The timestamps in our datasets are in epoch time format, meaning that we have accurate time information at the granularity of a second. T

Fig. 6 illustrates the time-based grouping. We assume that a user's state (position and velocity) obeys the Markov property, i.e., the user's next state depends only on his current state. We use the time-groups as our states and consequently, our inference algorithm attempts to find the globally optimal path assignment by sequentially finding the most probable assignment between consecutive time-groups; the assignment that maximizes the speed probabilities between points, based on the mobility model.

The time-groups do not necessarily contain the same number of points. Users may start or stop providing location



(a) *Location points before grouping*



(b) *Location points after grouping*

Figure 6: *Illustration of time-based grouping of location points. Temporal dimension is shown with different types of circles and the spatial relationship of the points is shown in 2D space. Sub-figure 6a shows six location points prior to time-grouping. Sub-figure 6b shows the same location points after the time-grouping. Three different groups are created, denoted by a continuous black circle (points $\mathcal{C}_1$ and $\mathcal{C}_2$), a continuous grey circle (points $\mathcal{C}_3$ and $\mathcal{C}_4$), and a black dashed circle (points $\mathcal{C}_5$ and $\mathcal{C}_6$)*

points at any time; also a user device may not obtain a GPS fix at some locations. A larger number of location points in a time-group compared to its previous one indicates that a new user has appeared. Similarly, a smaller number indicates that a user stopped providing location information.

Moreover, even if two consecutive time-groups contain the same number of location points does not necessarily mean that the same users appear in these groups. It is perfectly possible that a number of users stopped providing location information in the first group and the same number of new users appeared in the next group. Our path reconstruction mechanism, takes all of these cases into account by including an extra location point X in each time group. This new location point X denotes the beginning of a new path. Associating a point in the current time-group with the point X of the next time-group indicates that a path terminates in the current time-group and a new path starts in the next time-group.

3.4.2. Association of Points to Users. At the start of the path reconstruction process, after the time-based grouping, we randomly assign IDs to the location points of the first time-group. These user IDs are pseudonyms created by the algorithm and do not have any explicit relationship with the actual (true) user IDs. However, mapping to a different set of IDs is straightforward after the execution of the algorithm. A path can now be created by sequentially associating points of consecutive time-groups. In Figure 7 we show how the association between points of two consecutive time-groups is done. The location points are associated based on the mobility model, in such a way that the point associations maximize the probability of the speed between those points.

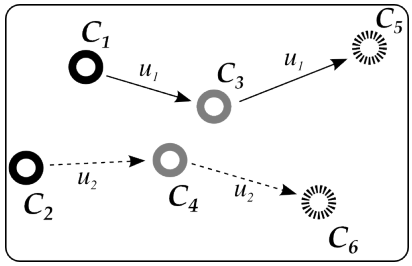


Figure 7: *Associations of points between time-groups. By associating points $\mathcal{C}_1, \mathcal{C}_3$ and $\mathcal{C}_5$ together we denote that they belong to the same user u_1.*

To find the most likely association between two consecutive time-groups, we first need to compute the individual probabilities of all the possible associations between each point of the first time-group to each point of the immediately following one. The probability of an individual association between two points C_i and C_j in two consecutive time-groups t_1 and t_2 is obtained from the user mobility model, i.e., the probability of the speed to move from C_i to C_j.

The individual associations and their probabilities are represented as a probability matrix, an $n \times m$ matrix where n is the number of anonymized location points in the current time-group and m is the number of anonymized location points in the next time-group. An example probability matrix is shown in Table 1. We define the *joint association probability* at time-group t as the product of the individual probabilities.

	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_6$	$\mathcal{C}_7$
$\mathcal{C}_1$	0.9870	$8.75 * 10^{-7}$	0.0031	0.0016
$\mathcal{C}_2$	0.0104	$1.18 * 10^{-5}$	0.9713	$1.83 * 10^{-5}$
$\mathcal{C}_3$	$4.13 * 10^{-4}$	$2.62 * 10^{-6}$	$8.48 * 10^{-5}$	0.6336

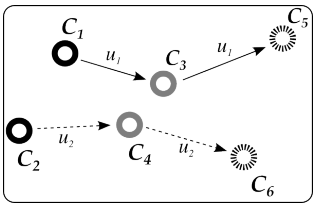
TABLE 1: *Example probability matrix. Points $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ belong to the current time-group. Points $\mathcal{C}_4$, $\mathcal{C}_5$, $\mathcal{C}_6$ and $\mathcal{C}_7$ belong to the next time-group. The probability of associating $\mathcal{C}_1$ with $\mathcal{C}_4$ is 0.9870. In contrast, the probability of associating $\mathcal{C}_1$ with $\mathcal{C}_5$ is $8.7530 * 10^{-7}$. We can see that the association with the maximum joint probability is $\mathcal{C}_1 \rightarrow \mathcal{C}_4$, $\mathcal{C}_2 \rightarrow \mathcal{C}_6$ and $\mathcal{C}_3 \rightarrow \mathcal{C}_7$. Point $\mathcal{C}_5$ cannot be associated with any point in the current time-group, indicating the start of a new path.*

On each step we choose the association that results in the maximum joint association probability (Fig. 8). The selected joint association needs to satisfy certain criteria:

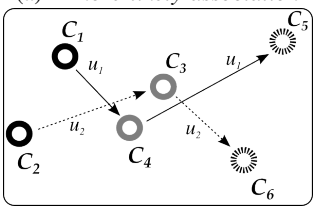
1) Each point in either group may only be associated with at most one point in the other time-group.
2) There is a minimum probability threshold for the individual associations. The probability of each of the individual associations must be above that

threshold. The process of defining the minimum probability threshold is described in section 3.5.1.

3) If a point in a given time-group cannot be associated with any points in the next time-group with a probability greater than the minimum probability threshold, we consider the path to which it belongs as completed.
4) If the next time-group has more points than the current one, then we consider the unassociated points in the next time-group as the starting points of new paths and assign a new user ID to each of them.



(a) *A more likely association*



(b) *A less likely association*

Figure 8: *Association of location points across three different time-groups: Two users (u_1 denoted by continuous line, u_2 denoted by dashed line) move from points of the first time-group (continuous black circle) to points of the second time-group (continuous grey circle), and then to points of the third time-group (dashed black circle). Sub-figure 8a shows a more likely association while sub-figure 8b shows an association that is less likely.*

3.5. Optimizations

In this subsection, we present two optimizations, the minimum probability threshold and the early rejection, that allow for faster evaluation of joint association probabilities while ensuring the calculation of the maximum joint association probability at each time step.

3.5.1. Minimum Probability Threshold (Θ). The minimum probability threshold (Θ) indicates the minimum probability that a valid association may have. If an association has a probability less than Θ, then it is unlikely that the two location points belong to the same user. In other words, we use the value of Θ as our "decision boundary" to classify whether or not an association is valid. In machine learning and data mining, the *decision boundary* is the plane that separates the training samples belonging to two different classes [21]. In our context, the decision boundary is the Θ value that separates valid from invalid associations.

To decide on a suitable value of Θ, we first calculate probabilities of valid associations (positive examples) as well as invalid associations (negative examples)[1]. We obtain both positive and negative examples from the training set. We generate the positive examples by calculating the probabilities of associations between points of the *same* user. The negative examples are the probabilities of associating location points of *different* users. The value of Θ is the one that satisfies the equation (3) and separates the instances of the two classes as clearly as possible.

$$\begin{aligned} &\arg\max_{\Theta} |P_\Theta| + |N_\Theta| \\ s.t. : &P_\Theta : assoc.\ probabilty > \Theta \\ &N_\Theta : assoc.\ probabilty \leq \Theta \end{aligned} \tag{3}$$

where P_Θ is the set of positive examples whose association probability is greater than Θ, and N_Θ is the set of negative examples whose association probability is less than or equal to Θ.

Figure 9 shows an example of positive and negative examples and the probability threshold selected as the decision boundary. Positive examples are denoted with 1 while negative examples are represented with -1. The selected minimum probability threshold is denoted by the dashed line. Please note that the presence of positive examples on the left side of the decision boundary does not mean that the probabilities of these examples are less than the probability associated with the decision boundary (value Θ). The dashed line simply indicates the point of separation between the examples of the two classes.

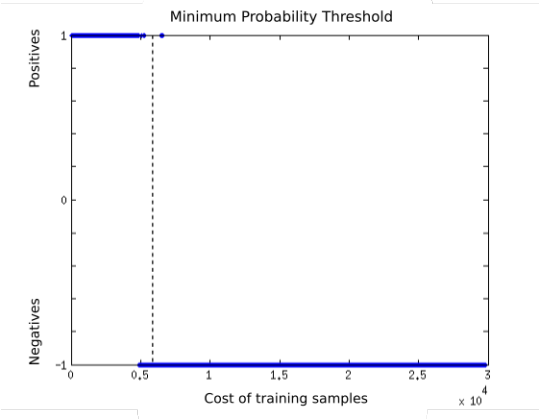


Figure 9: *Example of the minimum probability threshold. The threshold is denoted by the dashed line. Valid associations are represented by the value 1, while invalid associations are denoted by the value -1.*

The minimum probability threshold is applied in the probability matrix before evaluating possible joint associations and helps us reduce the number of required calcula-

1. By negative examples, we do not mean that the association probability is a negative value, this is not possible. Instead we represent valid associations as instances of the positive class and invalid associations as instances of the negative class.

	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_6$	$\mathcal{C}_7$
$\mathcal{C}_1$	0.9870	0.0	0.0031	0.0016
$\mathcal{C}_2$	0.0104	0.0	0.9713	0.0
$\mathcal{C}_3$	0.0	0.0	0.0	0.6336

TABLE 2: *Probability matrix after applying the minimum probability threshold Θ. Associating any current point with $\mathcal{C}_5$ results in a probability below that threshold, meaning that this association is infeasible. Therefore, a new path is created starting from point $\mathcal{C}_5$.*

$\mathcal{C}_1$	0.9870 ($\mathcal{C}_4$)	0.0031 ($\mathcal{C}_6$)	0.0016 ($\mathcal{C}_7$)	0.0 ($\mathcal{C}_5$)
$\mathcal{C}_2$	0.9713 ($\mathcal{C}_6$)	0.0104 ($\mathcal{C}_4$)	0.0 ($\mathcal{C}_5$)	0.0 ($\mathcal{C}_7$)
$\mathcal{C}_3$	0.6336 ($\mathcal{C}_7$)	0.0 ($\mathcal{C}_4$)	0.0 ($\mathcal{C}_5$)	0.0 ($\mathcal{C}_6$)

TABLE 3: *Probability matrix after sorting each row. The parentheses indicate the point in the next time-group that the probability corresponds to.*

tions by treating some less likely associations as infeasible. Any associations with probability equal to or below the threshold are excluded from the path inference algorithm and are not evaluated at all. Table 2 shows the resulting probability matrix of Table 1 after applying the minimum probability threshold. Any probabilities below the value Θ are set to zero.

3.5.2. Early Rejection. In order to calculate possible associations between N points of a given time-group and N points of the next time-group (supposing that both time-groups have the same number of points), $N!$ calculations are required. This is very computationally expensive. However, we observe that some associations are more likely than others while a lot of the associations are highly unlikely (e.g., they have a probability lower than the minimum probability threshold). Therefore, if we could find a joint association with a relatively high probability *early* on, we could then discard associations with lower probabilities altogether and thus save on the number of calculations made. The early rejection feature is based on this observation and can quickly find the set of associations with the maximum joint association probability.

The idea behind this optimization is as follows. We first sort each row of the probability matrix in descending order, keeping an index of the location points of the next time-group. The sorting ensures that for each location point in the current time-group the available associations with the higher probabilities will be evaluated first. The minimum probability thresholds (if any) appear at the end of each row. Table 3 shows the resulting probability matrix after the sorting.

We then find the first possible joint association for all the points in the current time-group and the equivalent joint probability. We consider that as our maximum probability found so far. Referring to Table 3, the first possible joint association is $C_1 \rightarrow C_4$, $C_2 \rightarrow C_6$ and $C_3 \rightarrow C_7$, with an overall probability of 0.6074. We then start evaluating the next possible association for all the points in the current time-group. On each step we update the joint probability accordingly and compare it with the maximum probability found so far. The moment the joint probability of the association that we currently evaluate falls below the maximum probability, we can discard the association and move to the next one.

For example, referring again to Table 3, a possible association (apart from the one discussed before) starts by associating point $C_1 \rightarrow C_6$. The probability of this association is 0.0031, which is already lower than the maximum probability found so far. At this point there is no reason to attempt and associate any points to C_2 as the probability of this association cannot increase.

This optimization allows us to evaluate all possible associations and quickly find the one with the highest probability. The initial sorting of the matrix increases the chances that the joint association with the maximum probability overall will be found early in the process and that all joint associations with lower probabilities will be discarded quickly. Depending on the probability matrix, the early rejection implementation combined with the initial sorting can reduce the required number of calculations by up to 90%. This allows the path inference algorithm to scale to large numbers of concurrent users.

4. Evaluation

To assess the effectiveness of Comber and whether it is possible to reconstruct the original paths of the users from a completely anonymized dataset, we use three different evaluation metrics:

1) accuracy of inferred paths
2) purity of inferred paths
3) time needed for the path inference algorithm to complete

Accuracy of inferred paths: The *accuracy* of the inferred paths shows how accurate the associations of the location points are compared to the ground-truth. To evaluate the accuracy of our path inference algorithm we need to assess whether the anonymized location points were associated correctly. A correct association is one that the triplets $\mathcal{C} = \{t, lat, lon\}$ of the anonymized location points that have been associated with the same inferred user ID also belong to the same user in the ground-truth dataset.

To assess the path accuracy we first split the paths into edges. An *edge* is the association of two consecutive anonymized location points. We start by evaluating whether each of the inferred edges has been accurately associated. Fig. 10 shows how we calculate the edge accuracy. Missed edges, i.e., the edges that were not created at all by Comber, are also counted as wrong edges. We can then combine the correctly inferred edges to form longer paths.

Purity of inferred paths: Although the edge accuracy shows whether the location points have been associated correctly, it has a limitation; if a wrong edge is found along a ground-truth path then that path is split in two (Fig. 11).

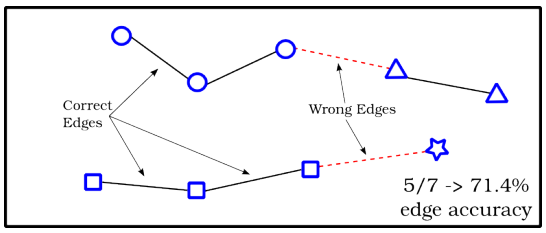
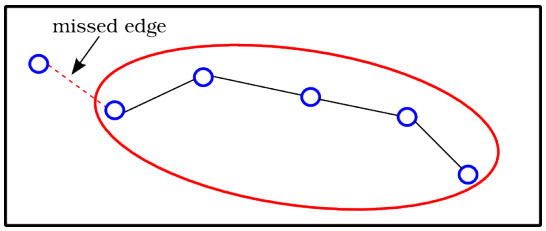
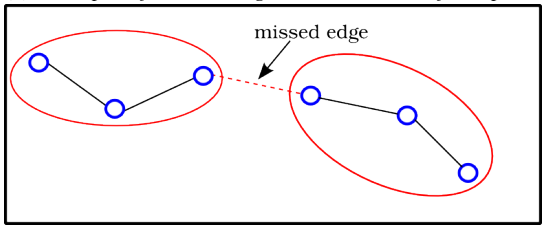


Figure 10: *Illustration of edge accuracy measure calculation. The different shapes indicate different users in the ground-truth. The lines indicate the results with the inference algorithm. We consider an edge as correct if it originates and terminates from location points that belong to the same user in the ground truth.*

The position of the missed edge plays an important role. If the missed edge is one of the initial or last edges of the path, then the result is an almost totally accurate path (Fig. 11a). However, if the missed edge is found in the middle of the path, then the original path is split into two sub-paths (Fig. 11b). Therefore, we also need to evaluate the purity of the inferred paths compared to the ground-truth.



(a) *Example of missed edge near the start of the path.*



(b) *Example of missed edge in the middle of the path.*

Figure 11: *Illustration if the limitation of the edge accuracy measure. The red dashed line indicates a missed edge. The edge accuracy in both sub-figures is 80%. However, in sub-figure (a) the original path is almost accurately inferred, whereas in sub-figure (b) it is split into two sub-paths of equal length.*

If we consider each of our inferred paths as a different cluster then the *purity* of a cluster [22] is the percentage of correctly assigned points to a class (user) in the ground-truth, and is given by:

$$purity(\Omega, \mathcal{C}) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j| \qquad (4)$$

where Ω is the set of clusters (inferred paths) with $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$, $\mathcal{C}$ is the set of classes (user paths in the ground-truth) with $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ and N is the total number of points across all clusters.

We first find the most frequent class in each cluster and we then calculate the purity by counting the number of correctly assigned points (number of points of the most frequent class) divided by the total number of points across all clusters. We assume that each inferred path is mapped to the user ID in the ground-truth with the greater number of points in the inferred path (the user ID with the majority). The purity of the inferred path then is calculated by taking the ratio of the sum of points of a ground-truth path with the user ID with the majority in each inferred path to the total number of points in the inferred paths. Fig. 12 illustrates the purity metric. Note that path purity is always less than or equal to edge accuracy. This is because only the part of the path that is mapped to the ground-truth user ID with the majority is considered as pure , whereas each corrected inferred edge is independently counted towards edge accuracy.

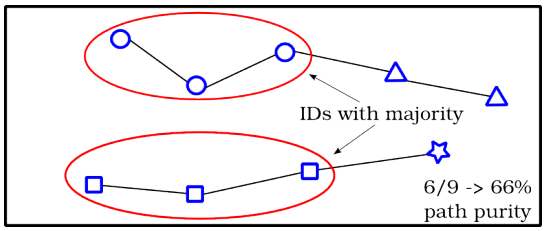


Figure 12: *Illustration of the path purity measure calculation. The different shapes indicate different users in the ground-truth. The lines indicate the results of the inference algorithm.*

4.1. Edge Accuracy and Path Purity with MDC Dataset

We first consider the MDC dataset to evaluate both the accuracy and purity of the inferred paths. As mentioned before, we selected a number of users to be used as our training set. However, the more users we select for training, the more we reduce the testing set as we cannot use the same users both for training and testing. We do not want to significantly decrease the size of the testing set since the smaller it is the easier the problem becomes, as the chances of having intersections between user paths is decreased. Therefore, we use only 20 users for training and the remaining 165 are used for testing.

Having decided on the number of users in the training set, we perform a variation of a commonly used method of k-fold cross validation with k set to 10, in order to assess the accuracy and purity of the inferred paths. Although in a normal k-fold cross validation we would train on $k-1$ parts of the dataset and test on the remaining part, this is not appropriate in our case as we would significantly decrease the size of the testing dataset. Therefore, we use a variation of the standard k-fold cross validation by training on one part and testing on the other $k-1$ parts. With this variation

of the k-fold cross validation, not only we assess whether Comber is able to produce results of significant accuracy, but we also assess whether it can be generalized by using different training and testing sets.

With our variation of the 10-fold cross validation, we perform 10 different iterations of Comber and in each of them we train and test on different sets. Therefore, we limit the chances of getting higher or lower accuracy due to capturing specific moving patterns in our training set. Consistent results across all iterations would suggest that Comber is not biased by distinct moving patterns and provides an accurate estimation of the expected accuracy and purity of Comber.

The training users of each fold were selected randomly. Fig. 13 shows the edge accuracy and path purity of each of the 10 different runs with the MDC dataset. The mean edge accuracy is 98.7%, while the mean path purity is 91.6%. The results present significant consistency across all iterations, with very small variations. This behavior indicates that the effectiveness of Comber is not affected by moving patterns of specific users and can be generalized to different training and testing sets.

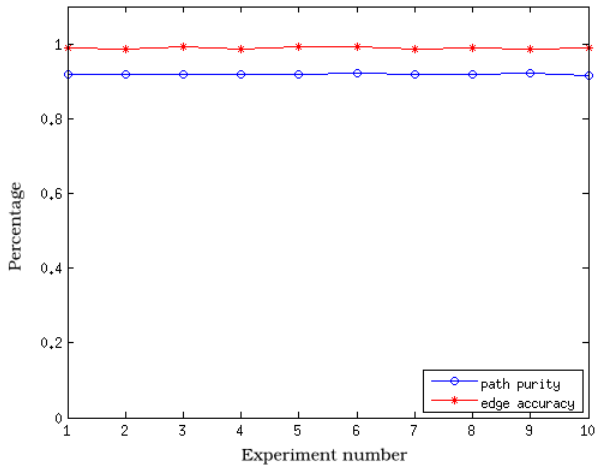


Figure 13: *Results of edge accuracy and path purity with the variation of the 10-fold cross validation, using the MDC dataset. The blue line with circles denotes the path purity and the red line with stars shows the edge accuracy. We see that results for both these measures remain almost the same for all the runs, with very slight variations.*

4.2. Path Inference Computation Time

The computation time of Comber is also an important metric that needs to be evaluated. The time that Comber needs to complete heavily depends on the number of concurrent users in a given time-group. The larger the number of concurrent users the greater the number of possible associations that need to be evaluated. To evaluate the elapsed time of our algorithm, we consider the time needed for the inference of the whole MDC dataset (almost 12 million location points). Fig. 14 shows the minimum, maximum and mean time needed to complete the path inference for different numbers of concurrent users. We can see that the computation time increases with the number of concurrent users as expected, but even in the worst case Comber needs less than 5 minutes to complete. With 22 concurrent users, a naive implementation would require 22! calculations. These results stress the benefits of using the minimum probability threshold and early rejection features. Note that all our path inference computations were carried out using a commodity workstation with an Intel Core i5 3.20GHz processor and 8GB memory.

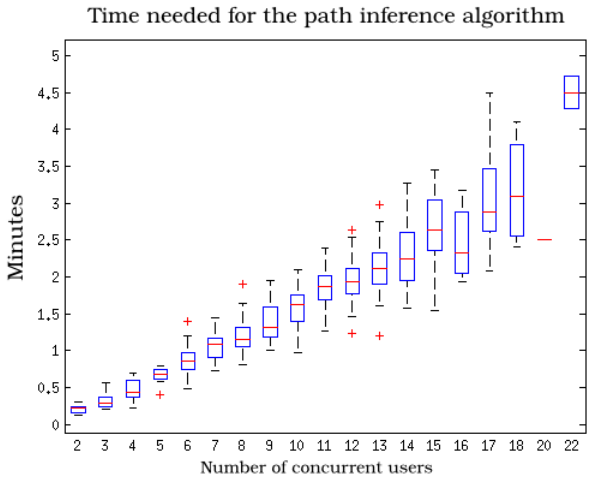


Figure 14: *Time needed for the path inference to complete for different numbers of concurrent users. In the MDC dataset, there are 2 to 22 concurrent users. We can see that the total time never exceeds 5 minutes. We only had one case of 20 concurrent users, therefore only the exact value of the time needed is shown in the plot. In addition, we did not have any cases of 19 or 21 concurrent users, thus they are missing from the plot.*

4.3. Training Set Size Required

So far we considered a training set of 20 users. We decided on this number somewhat arbitrarily in order not to significantly decrease the size of the testing set and at the same time gain as much information as possible from the training set. However, it is perfectly possible that fewer users would still be sufficient to build a user mobility model with similar effectiveness. This is of great importance as in a real-life scenario the number of users that can be used for training may be limited. Therefore, we want to evaluate the minimum number of users that are required to get the desired accuracy and path purity.

For this evaluation, we use a slightly different methodology. We evaluate Comber using different training set sizes ranging from 2 and up to 20 users. We do not use a training set of 1 user as we cannot generate negative examples

to determine the minimum probability threshold. For each different size of the training set, we perform 10 experiments. The training set on each run is randomly selected from the whole dataset, however we ensured that the training sets of the same size differed in at least one user. Fig. 15 shows how the edge accuracy and path purity vary with the different training set sizes. The edge accuracy results are shown in Fig. 15 (a) whereas the path purity results are shown in Fig. 15 (b). In each of the sub-figures, the cases with few number of users in the training set are magnified for better clarity.

We can see from the figures that both the accuracy and purity do not present significant differences regardless of the number of users in the training set. This means that we can achieve the same level of edge accuracy and path purity using a smaller training set with fewer users. The variation in edge accuracy with different training set sizes is quite small, with a minimum accuracy of 98.63% and a maximum 99.23%. Concerning path purity, the minimum can be seen to be 90.11% whereas the maximum is 92.30%.

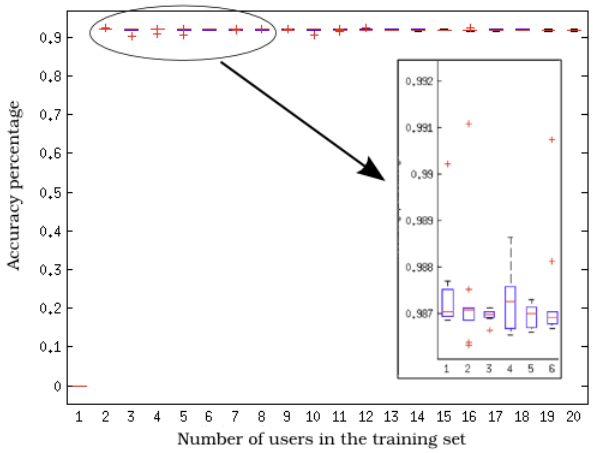
4.4. Generality of Comber

The results presented so far were obtained using the MDC dataset. To assess the generality of the Comber approach, we now consider the GeoLife dataset [13], for which we are using the same user mobility model as before, trained with up to 20 users from the MDC dataset. Recall that GeoLife dataset also consists of location traces of mobile users but in the Beijing area. The total number of users in the GeoLife dataset is 182 (about the same as MDC dataset). Similarly to MDC, we also removed the pseudonyms from GeoLife and performed the same variation of 10-fold cross validation described earlier. The total time needed for Comber to complete all of the 10 experiments was just around 2 minutes, which can be explained by the relatively smaller number of location points.

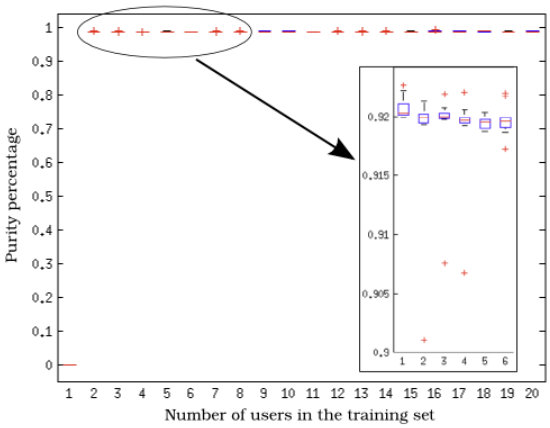
We can see from Fig. 16 that both the accuracy and the purity obtained with GeoLife dataset remain significantly high. The minimum accuracy is 97.2% and the maximum is 98%. The purity follows a similar pattern.

The above results show that although the mobility model was created using a training set from a completely different dataset, we can still infer the paths of different datasets with high accuracy. The main reason behind these results is the fact that human mobility patterns present significant similarities which can be exploited to create a generic mobility model as we do in this paper. According to [23], these similarities can be used not only for path inference attacks but also for other purposes including epidemic prevention, emergency responses, urban planning and agent-based modelling.

Given the very good accuracy and purity results, we investigated further to see if the user paths in the GeoLife intersected at all. If the paths of the users were isolated, very few associations would be possible, which would significantly decrease the complexity of the problem. We can see from Fig. 17 that for those users in the area of Beijing,



(a) *Edge accuracy of inferred paths.*



(b) *Path purity of inferred paths.*

Figure 15: *Edge accuracy and path purity of Comber using different training set sizes, from 2 to 20. For this evaluation we used the MDC dataset. Both the accuracy and the purity do not present significant changes, indicating that fewer users in the training set would be sufficient for path inference to be reliable.*

the path intersections are quite common so the excellent results are not due to the lack of path intersections. This behavior is similar to the users in MDC (Fig. 18), where paths of users in the centre of Lausanne also intersect for long periods.

4.5. Comber with Denser Datasets

We now study the impact of denser datasets on effectiveness of Comber by merging data from different days. More specifically, we randomly selected up to 5 different days from the MDC dataset and altered the timestamps so that they all belong to the same day (the hour, minute and

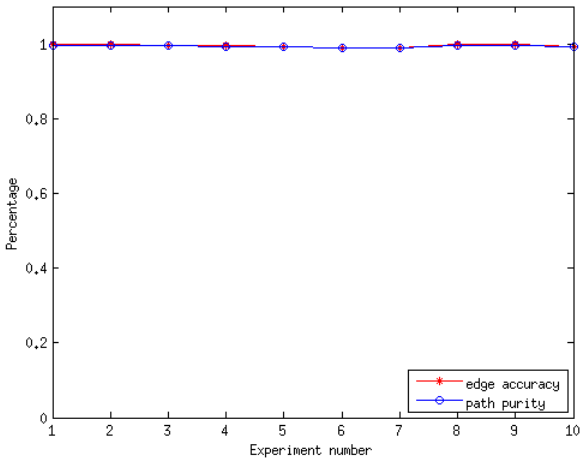


Figure 16: *Results of edge accuracy and path purity with the variation of the 10-fold cross validation, using the GeoLife dataset. The blue line with circles denotes the path purity and the red line with stars shows the edge accuracy. We can see that both the accuracy and the purity remain very high and almost identical.*

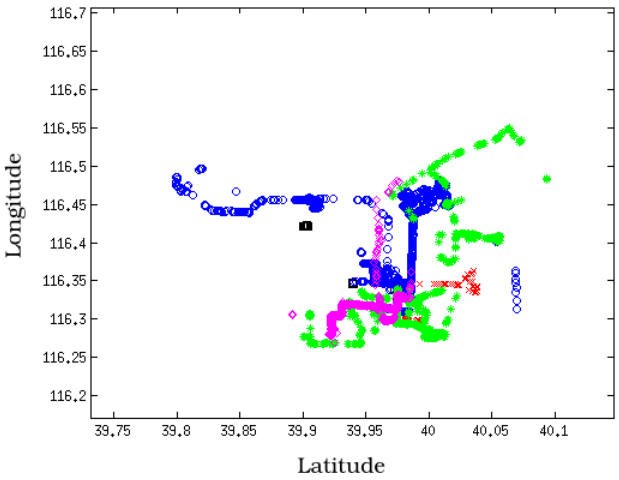


Figure 17: *Path intersections of users in the GeoLife dataset. For clarity reason we present 6 users that are only a subset og the spatially and temporally overlapping users in the dataset. They have been chosen purely for visualisation purposes. Each user is denoted by a different color and shape. We can see that the paths intersect significantly.*

second of each timestamp were left unchanged). Although the days were selected randomly, we did make sure that the number of users present in each day was above the average number of users present in a day. We run Comber on each of the generated datasets. Figure 19 shows the results of these experiments.

We can see from the figure that both the edge accuracy and the path purity drop to around 0.85, as expected with

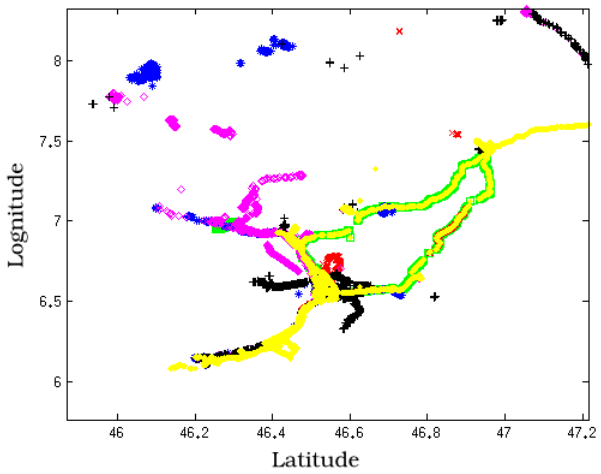


Figure 18: *Path intersections of users in the MDC dataset. For clarity reasons we present 6 users, denoted with different colors and shapes. We can see that despite some isolated parts the paths mostly intersect.*

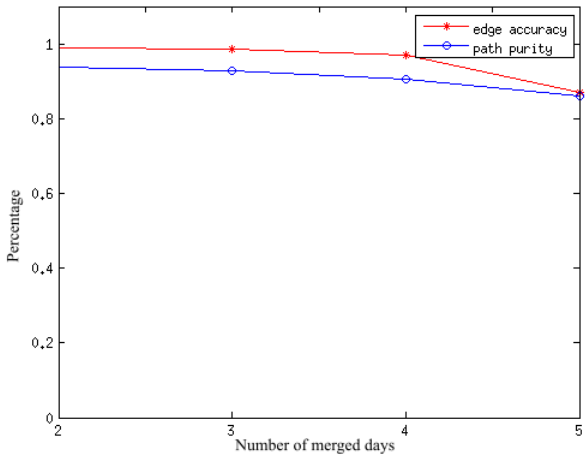


Figure 19: *Results of edge accuracy and path purity with increasingly dense datasets. We can see that both the edge accuracy and path purity drop to around 0.85, however the results indicate that path inference is possible even in denser datasets.*

a denser dataset. However, this number is still significantly high, indicating that path inference with Comber is possible even with datasets denser than MDC and Geolife.

5. Discussion

In this work we focused on the strongest form of anonymization, in which any user-identifying information is completely removed from the dataset, and demonstrated that complete anonymization alone is insufficient to preserve location privacy of users. A natural next step is to improve

the robustness of complete anonymization via some form of perturbation, a commonly used privacy preservation technique. Impact of different types of perturbation on path inference with Comber and the utility of the perturbed dataset is a key issue for future work.

In this work we have only considered the speeds of the users, which we used to build the mobility model. Although such information is shown to be sufficient, the mobility model can be further improved by adding directionality information of the inferred paths. Our intuition is that people tend to move in the same direction with few abrupt changes in their directionality (e.g. city blocks). Such information can be exploited so that drastic changes in the directionality of an inferred path are considered less likely than a smoother change. We expect that combining such information with the current mobility model can result in greater accuracy of the inferred paths, even in very dense datasets.

We have also tested Comber using more challenging datasets in which the density of the location points is increased, by merging several, randomly chosen days. As expected, the results of Comber decrease in these cases however, both the edge accuracy and path purity still remain fairly high. We do expect that the accuracy will drop as the datasets become even denser. In such cases considering further information in the mobility model, such as the directionality discussed above and user movement patterns, can be useful.

Apart from the aforementioned, there are also other ways in which Comber can be improved. For example the use of real maps could provide better accuracy and faster path inference algorithm execution. People in cities can follow existing roads and paths. By incorporating real maps we can exclude possible associations that are impossible (e.g. travelling through buildings).

Finally, another interesting point is that our algorithm uses a single mobility model to infer the paths of different datasets. This is justified in view of previous work on human mobility patterns [23] which shows that they are very similar and can thus be predictable. However the mobility model in Comber can be further extended to account for different transportation modes beyond walking.

6. Related Work

Location privacy research is broadly divided into attacks and defenses, with a large majority of papers focusing on the latter, in contrast to this paper. We now briefly summarize this research, and discuss its relation to ours.

6.1. Location Privacy Preservation Mechanisms (LPPMs)

One way to categorize privacy mechanisms is based on whether they reside at the device that produces the data (user-centric or user-side mechanisms) or at an intermediate or final collection point that acts as a trusted third party, e.g. just before the data is made public (provider-side mechanisms). See Fig. 3. The method of defense that we consider in this paper, complete anonymization, makes sense when many users are considered simultaneously, hence it applies to the case of provider-side mechanisms.

Provider-side privacy research typically originates in the database research community, with a popular example being k-anonymity [24] and its descendants (l-diversity [25], t-closeness [26]), while a more recent and even more popular example is differential privacy [27]. Both approaches perturb/distort user data to protect privacy, hence they are examples of obfuscation-based methods. As such, they are orthogonal to the anonymization-based defense that we consider in this paper.

As a brief summary of the state of the art in user-side defenses, Shokri et al. [28] design an obfuscation mechanism that is provably optimal in the context of sporadic Location-Based Services (LBSs), i.e. apps that only send queries infrequently. The optimality arises naturally as the authors design the mechanism via an optimization problem whose variables are the parameters of the privacy mechanism. Later, Bordenabe et al. [29] combine the optimization approach with a novel privacy definition that is an extension of differential privacy to location privacy, thus producing an optimal mechanism that provides *geo-indistinguishability*. Fawaz and Shin [30] describe a working prototype of a location privacy mechanism, which they implement on Android. Their prototype defends against a selection of threats by using different mechanisms in each case, the most sophisticated of which resemble either differential privacy or the aforementioned geo-indistinguishability.

6.2. Location and Path Inference Attacks

To the best of our knowledge there has been very limited research on privacy attacks on completely anonymized mobility datasets. Gruteser and Hoh [31] use a multi-target tracking technique to classify GPS points from three anonymized users. They further extend the dataset to include two more users. However, the size of the dataset is significantly limited. In this research we use much larger datasets of around 180 users. Moreover, built into their model is the assumption that the difference in position and speed between two successive location samples of a user follows a zero-mean Gaussian distribution. This and other similar Gaussian assumptions necessarily follow from the Kalman filtering approach that they take, which is rather limiting. Instead, we make no such assumptions, but rather estimate user mobility empirically, directly from a training set involving small number of users.

6.2.1. Uniqueness of Human Mobility Traces. In a recent study, de Montjoye et al. [32] performed a statistical analysis on the uniqueness of human mobility traces. The authors showed that any four points from a user's path are enough to unique distinguish it from any other path in the dataset, with 95% accuracy. The experiments were conducted on a dataset consisting of cellular tower associations and the user ID of each association event was known.

[16] come up with a similar observation about the uniqueness of mobility traces of individuals but using GPS traces. More specifically, the authors of [16] present two techniques that can be used to re-identify a user from his/her mobility data. In that research the original user paths were known from the start and the techniques focus on classifying distinct location points to those paths. The goal of these techniques is to reveal the real identity of the individuals by exploiting the inherent uniqueness of the mobility trace of person. This serves as the basis of our research. We exploit this characteristic and attempt to reconstruct user paths with very limited information.

Song et al. [33] perform experiments on human mobility traces to assess their uniqueness. The dataset used includes information about half a million users over a period of a week and the mobility information is pseudonymized. The research assesses the uniqueness of the trajectories of different users assuming that an adversary has partial knowledge of the path of a given user. The goal is to measure the total number of trajectories that an adversary can find with this partial knowledge. Their experiments are in line with previously reported attacks, and show that user traces are highly unique and that the locations of the users can be retrieved using only a few points.

6.2.2. Inferring User Identities from Pseudonmyized Mobility Data. In one of the first papers on location privacy [34], it is already emphasized that just assigning pseudonyms to the users of an LBS is problematic. In the context of users in a workplace, the office of a given pseudonymous user can be easily identified by keeping track of the frequency of visits of the given pseudonym to each office. Another location inference attack [19] is on GPS traces provided by volunteers and tests four different algorithms to infer the users' home address. It also uses a web search engine in order to reveal the real identities of the subjects. The results show that it is possible to reveal a small fraction of the real identities of the users and a large fraction of the users' home addresses. However, the authors did not focus on reconstructing the paths. Each volunteer was pseudonymized and the end-to-end path corresponding to a given pseudonym was known from the start.

A similar attack was performed in [17] in which the authors used GPS-based mobility traces to infer the real identities of the users. In this work, the adversary is assumed to have prior knowledge of an individuals' paths and attempts to reveal the identities by matching the new, unknown paths to those he already knows. In a similar vein, the attack presented in [18] uses side information to infer the whereabouts of a victim. The attacker is assumed to have access to a small amount of information (side information) regarding a user that appears in an anonymous trace. Using this information the authors are able to reveal the whereabouts of the victim. Although this work is somewhat similar to our attack, the authors assume prior knowledge of a victim's path, which we do not do.

Another attack [15] showed that an attacker can almost always infer a user's real identity (in 87% of cases studied) with knowledge of the user's home and work location to a granularity of one city block. However, the authors assume knowledge of the work and home places of the users. Shokri et al. [35] provide a novel framework for analyzing various LPPMs and perform a location attack that is based on user-specific mobility models and the use of Markov chains to infer a user's path. The attack is performed on location traces that are again assumed to be pseudonymized, rather than completely anonymized as in our work.

In [36] the authors devised an attack that is based on information collected from users' personal devices to identify train trips. The authors apply machine learning techniques on the collected data to identify the activities of users (e.g. walking, moving in a vehicle) and correlate the results with information from different railway companies to infer possible train routes of that user can take.

In the context of mobile telephony (3GPP), Arapinis et al. [37] show that the periodic anonymization done with the TMSI reallocation procedure in a mobile network can be exploited to link back the TMSIs allocated to a given user, thus successfully tracking the user. Periodic anonymization is a middle ground between simple anonymization (assigning a single pseudonym that does not change) and complete anonymization (removing the username and assigning no pseudonym). Complete anonymization is equivalent to assigning a fresh pseudonym to every location sample, independent of all other pseudonyms.

The above set of attacks highlight the sensitivity of a dataset in which each user's visited locations are linked to each other through a pseudonym. In this light, one can better appreciate our proposed attack, which shows the feasibility of linking together a given set of independent visited locations into a set of user paths.

Due to the high popularity of social networks, recent research has also focused on the privacy risks of check-ins in those social networks as well as on the importance of peer location information to a user's privacy. For example, a user's privacy is dramatically decreased even in the case of an adversary considering only one peer's location information. A decrease in user privacy is also observed even when the given user does not share any information about his location [38].

Rossi et al. [39] assess the importance of social network check-ins, such as Facebook and Foursquare, for user identification. They show that specific places that are present create re-identification opportunities due to the frequency that users check-in at these places (such as homes and work places). Thus, an adversary that monitors such places significantly facilitates the identification process. Also, due to the nature of these places, it is easy to identify a user even if his check-in frequency is relatively small.

7. Conclusions

In this paper we challenge the privacy protection that complete anonymization provides for user location and mobility data. We have introduced Comber, a novel attack mechanism that exploits the similarity of human mobility

patterns in terms of distribution of speeds. Comber uses a generic and empirically derived histogram of speeds to reconstruct the original paths of the users. We have evaluated Comber with two different datasets, MDC [12] and GeoLife [13], which consist of GPS based human mobility traces collected in Lausanne and Beijing, respectively. Both datasets span more than a year and include location information of around 180 users each.

We have evaluated the accuracy and purity of the inferred paths and our results show that Comber is able to infer the original paths of the users with more than 90% accuracy. These results raise serious security concerns about the privacy of mobile users using apps that access their location. Due to the inherent complexity of the problem, Comber includes two different optimization features which decrease its run-time by up to 90%, making path inference feasible even for large datasets.

Issues for future work include enhancing Comber to consider multiple transportation modes and directionality information, and studying its effectiveness with other/hybrid location privacy protection mechanisms that involve some form of perturbation.

References

[1] "Number of available applications in the Google Play Store from December 2009 to February 2015," http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/, 2015.

[2] M. Elgan, "Are your smartphone apps selling you out?" http://www.computerworld.com/article/2486596/mobile-apps/are-your-smartphone-apps-selling-you-out-.html, December 2013.

[3] K. Micinski, P. Phelps, and J. S. Foster, "An empirical study of location truncation on Android," in *Proceedings of Mobile Security Technologies (MOST'13)*, 2013.

[4] "The data brokers: Selling your personal information," http://www.cbsnews.com/news/the-data-brokers-selling-your-personal-information/, March 2009.

[5] K. J. O'Brien, "Data-gathering via apps presents a gray legal area," http://www.nytimes.com/2012/10/29/technology/mobile-apps-have-a-ravenous-ability-to-collect-personal-data.html?_r=1, March 2013.

[6] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 2, p. 5, 2014.

[7] J. Jonas, "To anonymize or not to anonymize, that is the question," http://jeffjonas.typepad.com/jeff_jonas/2007/02/to_anonymize_or.html, February 2007.

[8] A. Solanas, J. Domingo-Ferrer, and A. Martínez-Ballesté, "Location privacy in location-based services: Beyond TTP-based schemes," in *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications (PILBA)*, 2008, pp. 12–23.

[9] C. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *Dependable and Secure Computing, IEEE Transactions on*, vol. 8, no. 1, pp. 13–27, Jan 2011.

[10] "Opensignal privacy policy," March 2013, http://opensignal.com/privacypolicy/.

[11] A. Narayanan, J. Huey, and E. W. Felten, "A precautionary approach to big data privacy," in *Proceedings of Computers, Privacy & Data Protection*, March 2015.

[12] IDIAP and NRC-Lausanne, "Mobile Data Challenge (MDC) Dataset," Downloaded from https://www.idiap.ch/dataset/mdc, 2011.

[13] M. Research, "Geolife trajectories (v. 1.3)," Downloaded from http://research.microsoft.com/jump/131675, August 2012.

[14] N. Kiukkonen, B. J., O. Dousse, D. Gatica-Perez, and L. J., "Towards rich mobile phone datasets: Lausanne data collection campaign," in *Proc. ACM Int. Conf. on Pervasive Services (ICPS,',',')), Berlin.*, 7 2010.

[15] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *Proceedings of the 7th International Conference on Pervasive Computing*, ser. Pervasive '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 390–397. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01516-8_26

[16] R. Rossi, J. Walker, and M. Musolesi, "Spatio-temporal techniques for user identification by means of GPS mobility data," *CoRR*, vol. abs/1501.06814, 2015. [Online]. Available: http://arxiv.org/abs/1501.06814

[17] S. Gambs, M. O. Killijian, and M. Nunez del Prado Cortez, "De-anonymization attack on geolocated data," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, July 2013, pp. 789–797.

[18] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao, "Privacy vulnerability of published anonymous mobility traces," in *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '10. New York, NY, USA: ACM, 2010, pp. 185–196. [Online]. Available: http://doi.acm.org/10.1145/1859995.1860017

[19] J. Krumm, "Inference attacks on location tracks," in *Proceedings of the 5th International Conference on Pervasive Computing*, ser. PERVASIVE'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 127–143. [Online]. Available: http://dl.acm.org/citation.cfm?id=1758156.1758167

[20] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vision*, vol. 7, no. 1, pp. 11–32, Nov. 1991. [Online]. Available: http://dx.doi.org/10.1007/BF00130487

[21] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[22] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[23] M. C. Gonzalez, C. A. Hidalgo, and A. L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, June 2008.

[24] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[25] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, Mar. 2007. [Online]. Available: http://doi.acm.org/10.1145/1217299.1217302

[26] L. Ninghui, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, April 2007, pp. 106–115.

[27] C. Dwork, "Differential privacy," in *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 338–340.

[28] R. Shokri, G. Theodorakopoulos, C. Troncoso, J. Hubaux, and L. B. J.Y., "Protecting location privacy: Optimal strategy against localization attacks," in *19th ACM Conference on Computer and Communications Security (ACM CCS 2012)*, Raleigh, NC, USA, October 2012.

[29] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 251–262.

[30] K. Fawaz and K. G. Shin, "Location privacy protection for smartphone users," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 239–250.

[31] M. Gruteser and B. Hoh, "On the anonymity of periodic location samples," in *Proceedings of the Second International Conference on Security in Pervasive Computing*, ser. SPC'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 179–192. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-32004-3_19

[32] Y. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Sci. Rep.*, vol. 3, 03 2013. [Online]. Available: http://dx.doi.org/10.1038/srep01376

[33] Y. Song, D. Dahlmeier, and S. Bressan, "Not so unique in the crowd: a simple and effective algorithm for anonymizing location data," *ACM PIR*, 2014.

[34] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 46–55, Jan 2003.

[35] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Security and Privacy (SP), 2011 IEEE Symposium on*, May 2011, pp. 247–262.

[36] W. T., A. M., and M. T., "Routedetector: Sensor-based positioning system that exploits spatio-temporal regularity of human mobility," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. Washington, D.C.: USENIX Association, Aug. 2015. [Online]. Available: https://www.usenix.org/conference/woot15/workshop-program/presentation/watanabe

[37] M. Arapinis, L. I. Mancini, E. Ritter, and M. Ryan, "Privacy through pseudonymity in mobile telephony systems," in *Network and Distributed System Security Symposium, NDSS*, 2014.

[38] A. Olteanu, K. Huguenin, R. Shokri, and J. Hubaux, "Quantifying the effect of co-location information on location privacy," in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, E. De Cristofaro and S. Murdoch, Eds. Springer International Publishing, 2014, vol. 8555, pp. 184–203. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-08506-7_10

[39] R. Rossi, M. Williams, C. Stich, and M. Musolesi, "Privacy and the city: User identification and location semantics in location-based social networks," *CoRR*, vol. abs/1503.06499, 2015. [Online]. Available: http://arxiv.org/abs/1503.06499