walk2friends: Inferring Social Links from Mobility Profiles

Michael Backes
CISPA, Saarland University
Saarland Informatics Campus

Mathias Humbert
Swiss Data Science Center
ETH Zurich and EPFL

Jun Pang
FSTC and SnT
University of Luxembourg

Yang Zhang
CISPA, Saarland University
Saarland Informatics Campus

ABSTRACT

The development of positioning technologies has resulted in an increasing amount of mobility data being available. While bringing a lot of convenience to people's life, such availability also raises serious concerns about privacy. In this paper, we concentrate on one of the most sensitive information that can be inferred from mobility data, namely social relationships. We propose a novel social relation inference attack that relies on an advanced feature learning technique to automatically summarize users' mobility features. Compared to existing approaches, our attack is able to predict any two individuals' social relation, and it does not require the adversary to have any prior knowledge on existing social relations. These advantages significantly increase the applicability of our attack and the scope of the privacy assessment. Extensive experiments conducted on a large dataset demonstrate that our inference attack is effective, and achieves between 13% to 20% improvement over the best state-of-the-art scheme. We propose three defense mechanisms – hiding, replacement and generalization – and evaluate their effectiveness for mitigating the social link privacy risks stemming from mobility data sharing. Our experimental results show that both hiding and replacement mechanisms outperform generalization. Moreover, hiding and replacement achieve a comparable trade-off between utility and privacy, the former preserving better utility and the latter providing better privacy.

KEYWORDS

Social relationship privacy, location sharing, link prediction

1 INTRODUCTION

With the widespread usage of portable devices, mobility data has become available to a plethora of service providers, such as telecommunication operators, credit card companies, location-based services and online social networks (OSNs). While substantially improving mobile users' experience and providing them with convenient services, e.g., location recommendation, such availability also raises serious concerns about privacy. Previous studies have shown that a user's mobility trace is higly unique [28] and can be effectively deanonymized [39] with side channel information, and that a user's location data can unveil his personal attributes [32] and identity [16].

Social relationships represent highly privacy-sensitive information that is deeply connected with our social identity [10, 27]. In practice, online social network users have realized the extent of this threat and increasingly concealed their social relationships. For instance, the percentage of Facebook users in New York hiding their friend lists increased from 17.2% in 2010 to 56.2% in 2011 [11]. However, many individuals do not yet realize that their mobility data can also unveil their social relationships. Using location data to infer the underlying social relations between mobile users is of particular interest to various adversarial parties getting access to mobility data but not to social relations. For example, it is now well known that the NSA collects location and travel habit data to find unknown associates of targets it already knows about [1].

Previous works [5, 10, 35, 37, 42, 44] have demonstrated that mobility data can indeed serve as a strong predictor for inferring social relationships. However, these studies are all conducted with a data-mining perspective, e.g., for recommending friends to users in OSNs. They notably impose several requirements on the mobility data needed to infer social links, which dramatically reduces the scope of their applicability. For instance, almost all existing effective methods can only be applied if two individuals share locations in common. However, from a privacy point of view, in order to fully assess the extent to which location data can reveal the social relationships of any possible user, no such requirement should be a priori imposed. Moreover, no mitigation techniques have been proposed and evaluated so far for countering potential adversarial social link inference. This paper aims at filling these two essential gaps. First, the link prediction system must be as generic as possible to be able to evaluate, for *any* possible mobile user, the extent of the privacy risk towards his social links. Second, it is of utmost importance to design effective defense mechanisms for reducing the inherent risk towards social link privacy in location-data sharing.

Inference attack. Our link inference attack aims at predicting whether any pair of individuals are socially related, regardless of whether they have shared any common locations before. The attack relies on constructing an informative mobility profile/features for each user, and comparing two users' profiles, with the assumption that the mobility profiles of friends should be more similar than the profiles of strangers. However, manually constructing mobility features normally involves tedious efforts and domain experts' knowledge. Instead, we rely on an advanced feature learning model

CCS '17, October 30-November 3, 2017, Dallas, TX, USA

ACM ISBN 978-1-4503-4946-8/17/10…$15.00
https://doi.org/10.1145/3133956.3133972

(based on neural networks) to automatically learn each user's mobility features. The feature learning method we adopt [25, 26] is able to preserve a user's *mobility neighbors*, containing the locations he has visited, and the other users who have visited these locations. This method assumes that a user's mobility neighbors represent his mobility features to a large extent. After each user's mobility features are learned, we utilize pairwise similarity measures to compare two users' features and infer if these users are socially related. As our inference technique is unsupervised, the adversary does not need any prior knowledge on existing social relationships, which broadens the range of scenarios our attack can cover.

We empirically evaluate our inference attack on a large-scale dataset containing millions of location data points, i.e., check-ins, shared by Instagram users. Compared to well-known mobility datasets containing social relationships [8, 9], our dataset notably includes detailed information about each location, such as the location category/semantics. Extensive experimental evaluation shows that our attack is effective (with an area under the ROC curve equal to 0.8), and achieves between 13% to 20% improvement over the previous methods. We also empirically study the impact on the performance of various parameters involved in our machine-learning model. Then, we demonstrate that our attack is robust when a user only shares a small number of locations (down to 5 check-ins), and can even identify relationships between pairs of users that have shared no common location. Finally, we show that our attack is also effective when the adversary only has access to coarser-grained mobility data.

Countermeasures. In order to mitigate the aforementioned privacy risks, we extend and evaluate three defense mechanisms initially proposed by the location privacy community [6, 38]: hiding, replacement and generalization. In particular, for the replacement mechanism, we rely on the random walk approach proposed in [27] to find socially close locations to be replaced with. For generalization, we use two levels of generalization for both the semantic and geographical dimensions [6]. For the inference attack carried out with this countermeasure, we consider an enhanced adversary who is equipped with background knowledge on each location's popularity. This allows us to evaluate the generalization mechanism under a realistic setting, and thus have a more meaningful privacy assessment. We evaluate the effectiveness of the three defense mechanisms on our inference attack as well as on the previously proposed inference methods.

To quantify the utility degradation resulting from our mitigation techniques, we adopt an information-theoretic metric, the Jensen-Shannon divergence, which measures the difference between each user's location distribution in the original and in the obfuscated dataset. This utility measurement is meaningful since a user's location distribution is an essential element for building useful applications from mobility data, such as location recommendation systems.

Our experimental results show that hiding and replacement achieve equivalent privacy-utility trade-off: the former preserves better utility but the latter can reduce the attack's performance to a larger extent. Furthermore, both hiding and replacement significantly outperform the generalization mechanism.

Contributions. In summary, we make the following contributions.

- We propose a new social relation inference attack based on mobility data. The attack relies on a feature learning method and is able to predict any two users' social relationship regardless of whether they have visited common locations. This allows us to comprehensively evaluate the social link privacy risks stemming from location sharing.
- Extensive experiments demonstrate that our attack significantly outperforms state-of-the-art methods, and that it is robust to different real-world conditions, including a small number of available location data points.
- We propose the first defense mechanisms for protecting social link privacy from mobility-based attacks, and experimentally demonstrate their effectiveness.

Organization. Section 2 presents the notations and the adversary model considered in this paper. Our inference attack and its evaluation are presented in Sections 3 and 4, respectively. In Section 5, we introduce the defense mechanisms and their evaluation. Section 6 presents related work. We conclude the paper in Section 7.

2 MODEL

In this section, we introduce the notations used throughout the paper, as well as the adversary model.

2.1 User Model

We typically denote a user by $u \in \mathcal{U}$ and a location by $\ell \in \mathcal{L}$ with $\mathcal{U}$ and $\mathcal{L}$ representing the sets of users and locations, respectively. Note that each location considered in this paper is mapped to a fine-grained point of interest (POI), such as MoMA in New York. A user u visiting a location ℓ is referred to as a check-in denoted by a tuple $\langle u, t, \ell \rangle$, where t is the time when the check-in happens. We define $\tau(u, \ell)$ as the set of all the check-ins of u at ℓ, and $\tau(u)$ as the set of u's check-ins in the dataset. Moreover, $\omega(u)$ is used to denote all the locations u has been to.

2.2 Adversary model

The adversary's objective is to infer the social links, or relationships, between users by merely observing their mobility data. More precisely, he wants to infer whether two individuals are socially related or not, that is, make a binary prediction on the existence of a social link between two users. Such adversary can typically represent some location-based services, such as telecommunication operators, credit card companies and mobile apps on smart phones, that collect users' data without having access to their social graph.

It can also model an OSN user who has access to someone's location check-ins but not his social link information. This is possible on Facebook where a user can choose to hide his friends list, but keep other information, such as location check-ins, public. Our attack could be used by attackers to learn social links in order to further deanonymize users of the social network(s) [29]. Finally, it can also represent a global intelligence agency that gets access to mobility patterns of citizens through their mobile phones' metadata [1].

Figure 1: Social link inference attack based on location data: a schematic overview.

3 SOCIAL LINK INFERENCE ATTACK

To infer two users' social relationship with mobility data, one approach would be to design informative features based on the common locations they have visited,[1] as proposed in the state-of-the-art works [35, 37, 42]. However, as shown in Section 4, more than 50% user pairs do not share any common locations, meaning that such approaches cannot be applied to infer their social relationships. Alternatively, we can summarize each user's mobility features (or profile), then compare two users' features to predict their social link, with the assumption that friends have more similar mobility profiles than strangers. This approach enables the adversary to predict any pair of users' social link. However, defining informative mobility features is a non-trivial task because it falls into the domain of feature engineering in machine learning, which normally involves tedious efforts and domain experts' knowledge. For instance, features such as users' home locations, as proposed in [37], have led to poor inference performance (see Section 4).

The recent advancement of representation/feature learning (deep learning) provides us with an alternative approach. In this setting, features are automatically learned following an objective function that is independent from the downstream prediction task, in our case, social link inference. Promising works in this field include [15, 34, 40], whose objective functions preserve each user's neighbor information in the social network. The assumption of these works is that a user's social neighbors can reflect who he is. Similarly, we believe that a user's mobility neighbors can summarize his mobility profile to a large extent. Therefore, we utilize feature learning to automatically learn each user's mobility features, and apply the learned features for social relation inference.

Our attack can be decomposed into three stages, as depicted schematically in Figure 1. In the first stage, we adopt a random walk approach on the user-location bipartite graph to obtain random walk traces, which represent each user's neighbors in the mobility context. In the second stage, we feed the obtained random walk traces to a state-of-the-art feature learning model, namely skip-gram [25, 26], to obtain each user's mobility features in a continuous vector space. In the third stage, we measure the pairwise similarity between two users' vectors to predict whether there exists a social link between them in an unsupervised setting.

[1]Two users sharing a common location indicates that they have both visited the location, regardless of time.

3.1 Mobility Neighbors with Random Walk

We organize users and locations into a weighted bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{L}, \mathcal{E})$ where $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{L}$ contains all the edges between $\mathcal{U}$ and $\mathcal{L}$. For an edge $(u, \ell) \in \mathcal{E}$ between u and ℓ, we define its edge weight $w_{u,\ell}$ as the number of check-ins of u at ℓ, i.e., $w_{u,\ell} = |\tau(u, \ell)|$. A user's graph neighbors in the mobility context should contain locations he has been to, especially those locations he frequently visits, but also indirect neighbors such as other users who have visited the same locations, locations these users have visited, and so on. It is worth noting that this representation has demonstrated its effectiveness in numerous real-world applications, such as recommendation systems.

To define a user's mobility neighbors, we could rely on breadth-first sampling (BFS) or depth-first sampling (DFS) [15]. However, the neighbors resulting from BFS and DFS cannot reflect properly the user's top visited locations and other users that are similar to him, as the number of times a user visited a location is not taken into account. The random walk method fits our problem better, as it considers edge weights and is computationally more efficient than the aforementioned approaches [15]. Previously, the random walk approach has been demonstrated to be effective on homogeneous networks, such as social networks, to define a node's neighbors for feature learning [15, 34]. We generalize it to bipartite graphs in this work.

We denote a random walk trace by ϕ, which is composed of users and locations and a set Φ contains all the random walk traces. The procedure for generating random walk traces from a user-location bipartite graph is listed in Algorithm 1. For each user, the algorithm generates t_w random walk traces (Line 3), and each trace is l_w steps long (Line 6). Here, t_w and l_w, referred as *walk times* and *walk length*, are two hyperparameters and their values are set experimentally. For each current node *curr_v* in a random walk trace, we extract its neighbors, i.e., *curr_v_nb*, from $\mathcal{G}$ and the corresponding edge weights from *curr_v* to *curr_v_nb*, i.e., *curr_v_w* (Line 7). Then, the

Algorithm 1: Generating random walk traces

Data: A user-location bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{L}, \mathcal{E})$
Result: Random walk traces Φ

$\Phi \leftarrow [\,]$;
for $u \in \mathcal{U}$ **do**
  for $i = 1$ **to** t_w **do**
    $\phi \leftarrow [u]$;
    $curr\_v \leftarrow u$;
    for $j = 2$ **to** l_w **do**
      $curr\_v\_nb, curr\_v\_w \leftarrow$ GetNb($curr\_v, \mathcal{G}$);
      # extract $curr\_v$'s neighbors ($curr\_v\_nb$)
      and the corresponding weights ($curr\_v\_w$);
      $next\_v \leftarrow$ Sampling($curr\_v\_nb, curr\_v\_w$);
      append $next\_v$ to ϕ;
      $curr\_v \leftarrow next\_v$;
    end
    append ϕ to Φ;
  end
end

next node $next\_v$ in the random walk given the current node $curr\_v$ is chosen with the alias method [41] according to the following transition probability:

$$P(next\_v = y | curr\_v = x) = \begin{cases} \frac{w_{x,y}}{Z} & \text{if } x \in \mathcal{U} \wedge (x,y) \in \mathcal{E}, \\ \frac{w_{y,x}}{Z} & \text{if } x \in \mathcal{L} \wedge (y,x) \in \mathcal{E}, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where Z is the normalizing constant equal to the sum of the edge weights connected to x (Line 9). In the end, we obtain Φ which contains $|\mathcal{U}| \times t_w$ random walk traces and each trace is l_w steps long. The mobility neighbors of a user u, denoted by $N(u)$, are the nodes precedent and after u in all the random walk traces Φ.

3.2 Skip-Gram Model

In the second stage of our inference attack, we feed the random walk traces Φ into the skip-gram model to map each user's mobility information into a continuous vector. The model outputs one vector per user, which represents his mobility features. Skip-gram is a (shallow) neural network with one hidden layer that preserves a user's graph neighborhood information. Two users sharing similar mobility neighbors will be closer in the vector space (have similar mobility features) under skip-gram, which makes this model suitable for our prediction task.

The objective function of skip-gram is formalized as follows:

$$\arg\max_{\theta \in \mathbb{R}^{|\mathcal{U} \cup \mathcal{L}| \times d}} \prod_{v \in \mathcal{U} \cup \mathcal{L}} p(N(v) | v; \theta) \quad (2)$$

where θ represents the parameters of the model, i.e., the vectors (features) of all nodes in $\mathcal{G}$, and d is the dimension of the learned vectors. Similar to the walk times t_w and walk length l_w in the first stage, d is also a hyperparameter that we will study in Section 4. As we can see from objective function 2, skip-gram uses each node to predict its neighbor nodes in Φ. Next, by assuming that predicting neighbor nodes are independent of each other, objective 2 can be factorized into:

$$\arg\max_{\theta \in \mathbb{R}^{|\mathcal{U} \cup \mathcal{L}| \times d}} \prod_{v \in \mathcal{U} \cup \mathcal{L}} \prod_{n \in N(v)} p(n | v; \theta). \quad (3)$$

The conditional probability $p(n|v;\theta)$ is modeled with a softmax function:

$$p(n|v;\theta) = \frac{e^{\theta(n) \cdot \theta(v)}}{\sum_{m \in \mathcal{U} \cup \mathcal{L}} e^{\theta(m) \cdot \theta(v)}} \quad (4)$$

where $\theta(v) \in \mathbb{R}^d$ is the vector we aim to obtain for v and $\theta(n) \cdot \theta(v)$ is the dot product of the two vectors.

By plugging softmax into objective function 3 and applying log-likelihood transformation, skip-gram is turned into:

$$\arg\max_{\theta \in \mathbb{R}^{|\mathcal{U} \cup \mathcal{L}| \times d}} \sum_{v \in \mathcal{U} \cup \mathcal{L}} \sum_{n \in N(v)} \left(\theta(n) \cdot \theta(v) - \log \sum_{m \in \mathcal{U} \cup \mathcal{L}} e^{\theta(m) \cdot \theta(v)} \right). \quad (5)$$

From objective function 5, we can observe that if two nodes share similar neighbors, then their vectors will be similar. However, due to the term $\log \sum_{m \in \mathcal{U} \cup \mathcal{L}} e^{\theta(m) \cdot \theta(v)}$, solving objective function 5 is computationally expensive since it requires summation over all nodes in $\mathcal{G}$. In order to speed up the learning process, we adopt the negative sampling approach [26].

The negative sampling approach targets a different objective than the original skip-gram model, which is whether two nodes n and v appear together in a random walk trace or not: $n \in N(v)$ or $n \notin N(v)$. It is easy to see that this objective can be interpreted as a binary classification, and we use a random variable Δ to describe the binary choice: $\Delta = 1$ if two nodes appear together in any trace in Φ, and $\Delta = 0$ otherwise. Then, the new objective function of skip-gram is:

$$\begin{aligned} \operatorname*{argmax}_{\theta \in \mathbb{R}^{|\mathcal{U} \cup \mathcal{L}| \times d}} & \prod_{v \in \mathcal{U} \cup \mathcal{L}} \prod_{n \in N(v)} p(\Delta = 1 \mid n, v; \theta) \cdot \\ & \prod_{v \in \mathcal{U} \cup \mathcal{L}} \prod_{n \in N(v)'} p(\Delta = 0 \mid n, v; \theta), \end{aligned} \quad (6)$$

where $N(v)'$ is a sampled set that contains nodes which are not the neighbors of v in Φ.[2] The conditional probability $p(\Delta \mid n, v; \theta)$ now is modeled as the binary version of softmax, i.e., logistic regression, which is denoted by:

$$p(\Delta \mid n, v; \theta) = \begin{cases} \frac{1}{1 + e^{-\theta(n) \cdot \theta(v)}} & \text{if } \Delta = 1, \\ \frac{1}{1 + e^{\theta(n) \cdot \theta(v)}} & \text{if } \Delta = 0. \end{cases} \quad (7)$$

By adding all the pieces together, we have the following objective function for skip-gram:

$$\begin{aligned} \operatorname*{argmax}_{\theta \in \mathbb{R}^{|\mathcal{U} \cup \mathcal{L}| \times d}} & \sum_{v \in \mathcal{U} \cup \mathcal{L}} \sum_{n \in N(v)} \log \frac{1}{1 + e^{-\theta(n) \cdot \theta(v)}} + \\ & \sum_{v \in \mathcal{U} \cup \mathcal{L}} \sum_{n \in N(v)'} \log \frac{1}{1 + e^{\theta(n) \cdot \theta(v)}}. \end{aligned} \quad (8)$$

Compared to objective function 5, which is a multi-label classification, objective function 8 is more efficient to compute. We apply stochastic gradient descend (SGD) in our experiments to solve it,

[2] We adopt the same method as in [26] to sample non-neighbors.

which eventually outputs the feature vectors of all the users in the dataset.[3]

3.3 Social Link Prediction

In the last stage, for each pair of users u and v whose social link we aim to predict, we adopt a pairwise similarity measurement s to compare their feature vectors learned through skip-gram. We decide that u and v are socially related if their similarity $s(\theta(u), \theta(v))$ is above a given threshold. We experimentally compare the effectiveness of various similarity measurements in Section 4.

To the best of our knowledge, our attack is the first to utilize pairwise similarity metrics to infer two users' social relation based on skip-gram learned vectors. It is also worth noting that the existing feature learning methods [15, 34, 40] focus on user-specific prediction tasks, such as user attribute inference, and rely on supervised learning algorithms.

3.4 Advantages of Our Approach

There are three main advantages of our link inference attack. First, our attack is performed in an unsupervised setting, i.e., the adversary does not need any prior knowledge about any existing social relationships among the users. Second, our method can be applied to predict a social link between any pair of users without requiring them to share common locations. Both of these advantages result in our attack being more generic and applicable to large-scale privacy assessment than previous works. Third, our attack outperforms state-of-the-art attacks significantly, as shown in the next section.

4 ATTACK EVALUATION

We evaluate our proposed social link inference attack in this section. We first describe our experimental setup, including dataset, evaluation metric, baseline models and parameter setting. Then, we present the general results for the inference, and experimentally study the sensitivity of the hyperparameters involved in our inference attack. Next, we evaluate the robustness of our attack with respect to the number of check-ins a user shares, and the number of common locations between two users. Finally, we assess the performance of our attack when the adversary only has access to coarse-grained location information.

4.1 Experimental Setup

Dataset. Since we need social relationships to be explicitly disclosed to construct our ground truth, we rely on OSN data to conduct our evaluation. Among all the OSNs, we chose Instagram for two reasons. First, Instagram is the second largest social network with a fast growing number of users, and its users are more likely to share check-ins than other OSNs'. For instance, Instagram users share 31 times more their locations than Twitter users [22]. Second, Instagram's location service is linked with Foursquare, a popular location-based social network, which allows us to collect detailed information about each location such as its name and category. In particular, the location category information serves as the basis for one of the defense mechanisms, namely generalization, which will be presented in Section 5.

[3]Besides users' vectors, we also obtain locations' vectors. As we want to predict users' social links, the location vectors are simply dropped.

Table 1: Statistics of the pre-processed dataset.

	New York	Los Angeles	London
No. check-ins	1,843,187	1,301,991	500,776
No. locations	25,868	22,260	10,693
No. users	44,371	30,679	13,187
No. social links	193,995	129,004	25,413

The data collection was conducted in January 2016. We concentrate on three major English-speaking cities worldwide: New York, Los Angeles and London. In the first step, we use Foursquare's API to collect all the Foursquare's location IDs in these cities, together with these locations' category information. Then, we use Instagram's API to transform Foursquare's location IDs to the corresponding Instagram's location IDs.[4] In the end, we use Instagram's API to extract all the users' check-ins at each location in 2015. In total, 6.3 million check-ins are collected in New York, 4.6 million check-ins in Los Angeles and 2.9 million check-ins in London. Furthermore, the dataset includes 35,389 different locations in New York, 31,991 locations in Los Angeles, and 16,802 locations in London. Each check-in is organized in the following form:

$$\langle userID, time, latitude, longitude, locationID, category \rangle.$$

To collect the ground truth, i.e., the social network data, we utilize Instagram's API to collect all the IDs for the followees of the users in the check-in dataset.[5] As in many previous works [9, 12, 19], we consider two users to have a social relation if they mutually follow each other.

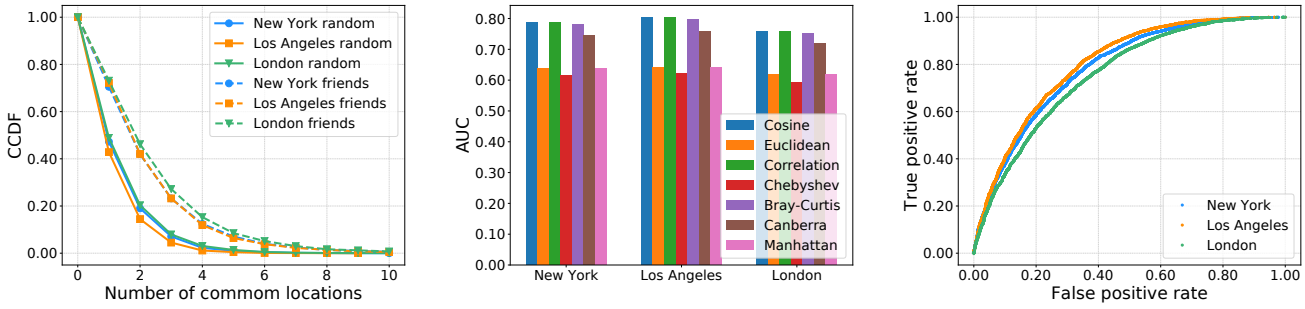
Compared to the well-known mobility datasets containing explicit social relation information collected from Gowalla [9] and Twitter [8], our dataset has two advantages. First, our dataset has a denser volume. We collected more than 13 million check-ins in only three cities, while the Gowalla dataset contains 6 million and the Twitter dataset contains 22 million check-ins in the whole world. Second, as mentioned above, our dataset contains detailed information about each location, which both Gowalla and Twitter datasets do not. For reproducibility purposes, the dataset will be made available upon request.

In order to get a representative yet usable dataset, we perform some pre-processing on the collected data. First, since accounts that share many check-ins at one location are generally local businesses, such as restaurants, we filter out users who have not visited at least two *different* locations. Second, some accounts in Instagram are celebrities or bots who are not the targets of our inference attack, therefore, we filter out those whose numbers of followers[6] are above the 90th percentile (celebrities) or below the 10th percentile (bots). Third, to resolve data sparseness issues, we run most of our experiments on users with at least 20 check-ins, whom we consider to be *active users*. This is in line with existing works such as [7, 9, 42, 43]. However, as there is no standard rule for defining

[4]The connection between Instagram's API and Foursquare's API was aborted in April 2016 (https://www.instagram.com/developer/changelog/).

[5]We only collect each user's followees not followers for efficiency reasons: some users in Instagram have millions of followers, such as celebrities, and Instagram's API only returns 50 followers per request.

[6] We use Instagram's API to collect each user's number of followers without collecting the detailed follower list.



(a) (b) (c)

Figure 2: (a) Distribution of the number of common locations between any two randomly chosen users and two socially related users (friends); (b) Area under the ROC curve (AUC) with respect to various pairwise similarity measures; (c) ROC curves with cosine similarity.

active users ($\geq$ 20 check-ins in our case), we also study how filtering based on a smaller number of check-ins (down to 5) influences the inference attack's performance in Subsection 4.4. The statistics of our pre-processed dataset is listed in Table 1.

Metric. We adopt AUC (area under the ROC curve) as our attack evaluation metric for two reasons. First, due to the nature of social networks, link inference has a huge prediction space and the labels are highly imbalanced, e.g., there are more than 9.8 billion pairs of active users in New York and less than 0.02% of them are friends (Table 1). To tackle this problem, we adopt the down-sampling strategy used in [15, 21], that is, we randomly sample the same number of stranger pairs as the number of friend pairs. To properly evaluate the inference in the down-sampled prediction space, a metric that is not sensitive to the label distribution is needed. As pointed out in [21, 24], AUC satisfies this requirement, and previous inference algorithms [15, 37, 42] have adopted it for evaluation too. Second, there exists a conventional standard for interpreting AUC (whose range is [0.5, 1]): 0.5 is equivalent to random guessing, 1 is perfect guessing (100% true positives and no false positives), and 0.8 represents already a good prediction.[7] This allows us to intuitively get a sense of the attack's performance, even without comparing against baseline models. Finally, note that privacy is defined as the opposite of the attack success. This means that privacy is minimal when AUC equals 1, and maximal when AUC equals 0.5.

Baseline models. We consider 14 baseline models proposed in three state-of-the-art papers inferring social relationships with mobility data [35, 37, 42]. They are denoted by `common_p` [35], `overlap_p` [35], `w_common_p` [35], `w_overlap_p` [35], `aa_ent` [35], `min_ent` [35], `aa_p` [35], `min_p` [35], `geodist` [35], `w_geodist` [35], `pp` [35], `diversity` [37], `w_frequency` [37, 42] and `personal` [42]. The formal definitions of these baseline models can be found in their original papers. Each of the baseline models rely on manually-designed features, thus can be evaluated in an unsupervised setting as well.

Among all the baseline models, 7 of them (`aa_ent`, `min_ent`, `aa_p`, `min_p`, `diversity`, `w_frequency` and `personal`) require that two users share at least one common location, in order to infer whether there is a social link between them or not. However, Figure 2a shows that more than half of the active user pairs and around 30% of friends' pairs do not share any common locations in each city. Therefore, to evaluate these 7 baselines, we first apply them on pairs of users who share at least one location, then randomly guess the rest of the pairs' social relationships.[8]

Parameter settings. As presented in Section 3, our model mainly involves three hyperparameters: walk length l_w, walk times t_w and feature vectors' dimension d. We set their default values to $l_w = 100$, $t_w = 20$ and $d = 128$. and evaluate how different values affect the attack performance in Section 4.3. Another parameter is the size of the neighbor nodes in the random walk traces, i.e., $|N(v)|$. Following [15, 34], we set it to 20, considering 10 nodes preceding and 10 nodes after v in Φ. Finally, the learning rate for SGD is set to 0.025. The source code of our implementation is available at https://github.com/yangzhangalmo/walk2friends.

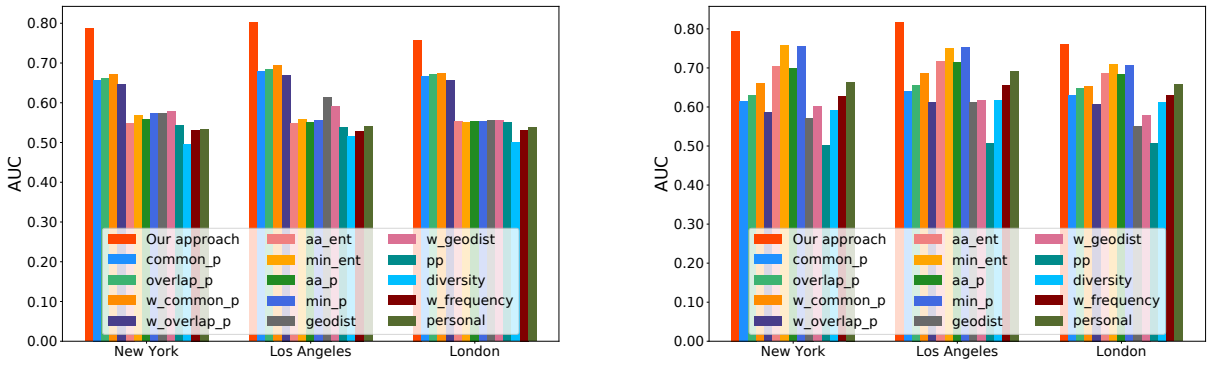
4.2 Social Link Inference

Our social link inference attack relies on the pairwise similarity between two users' mobility features learned by the skip-gram model. We have evaluated 7 common distance or similarity measures:[9] cosine similarity, Euclidean distance, correlation coefficient, Chebyshev distance, Bray-Curtis distance, Canberra distance, and Manhattan distance. The corresponding AUC values are depicted in Figure 2b. Among these measures, cosine similarity, correlation coefficient and Bray-Curtis distance achieve the best performance with AUC near 0.8, which represents a good prediction result. On the other hand, Chebyshev distance performs the worst with AUC around 0.6. By looking into all the similarity measures' definition, we notice that the best performing ones are those whose values are bounded. For instance, correlation coefficient lies within the range [-1, 1]. This indicates that bounded similarity measures provide better results for link prediction based on mobility data.

[7] http://gim.unmc.edu/dxtests/roc3.htm

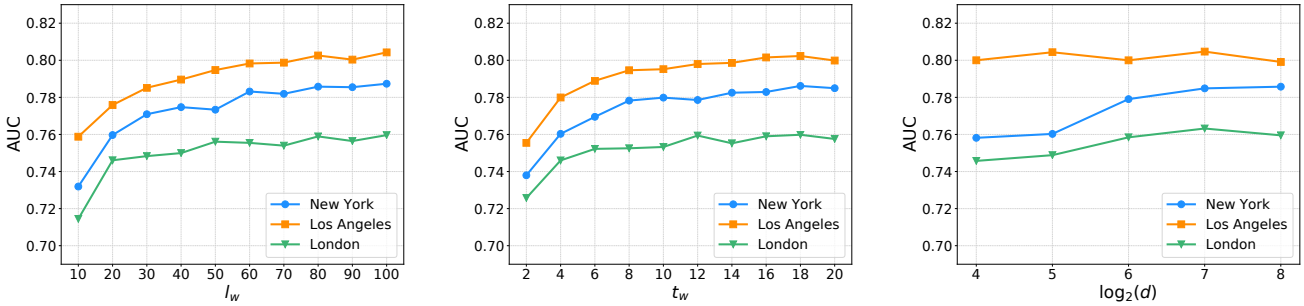
[8] The use of random guessing is due to the fact that our prediction is conducted in the down-sampled space.

[9] The formal definitions of these distances are in Appendix A.1.



(a) (b)

Figure 3: Comparison of our attack against baseline models: (a) using all users, (b) using only pairs of users who share at least one common location.



(a) (b) (c)

Figure 4: Influence of parameters (a) walk length, (b) walk times and (c) dimension of feature vectors on the inference performance.

As cosine similarity maximizes the attack success, we use it for our inference attack in the rest of this work. Figure 2c shows the ROC curves corresponding to the cosine similarity's AUCs in Figure 2b. The inference performs slightly better for Los Angeles than for New York or London with the true positive rate being 0.8 while the false positive rate staying at 0.34. The threshold at this point is equal to 0.86, i.e., inferring user pairs whose features' cosine similarity is above 0.86 as friends leads to a good prediction.

We then compare our inference attack against all the baseline models, Figure 3a shows that our attack outperforms all the baseline models significantly. For the best performing baseline model, i.e., `w_common_p`, we achieve a 20% performance gain in Los Angeles, and a 17% gain in New York. In the worst case, i.e., London, the performance gain is still 13%. This shows that our attack is much more effective than the existing state-of-the-art attacks.

As discussed before, 7 baseline models can only be applied to pairs of users who share common locations. We further compare our attack against them (as well as the other baselines) on pairs of users with at least one common location. Figure 3b shows that these baselines' performances indeed increase as reported in the original papers, but our prediction still outperforms the best baseline model, in this case `min_ent`, by 9% in Los Angeles, 5% in New York and 7% in London. By taking into account the fact that our attack can predict any pair of users' social link, this further demonstrates the effectiveness of our attack.

4.3 Parameter Sensitivity

Next, we examine how the different choices of the three hyperparameters walk length (l_w), walk times (t_w) and dimension of feature vectors (d) affect our attack performance. When testing each parameter, the two remaining ones are kept to their default settings, i.e., $l_w = 100$, $t_w = 20$ and $d = 128$.

Among all the three hyperparameters, l_w and t_w are directly linked with the size of the random walk traces, i.e., the amount of data being fed into skip-gram. Intuitively, larger values of l_w and t_w should lead to better inference performance. This is indeed the case

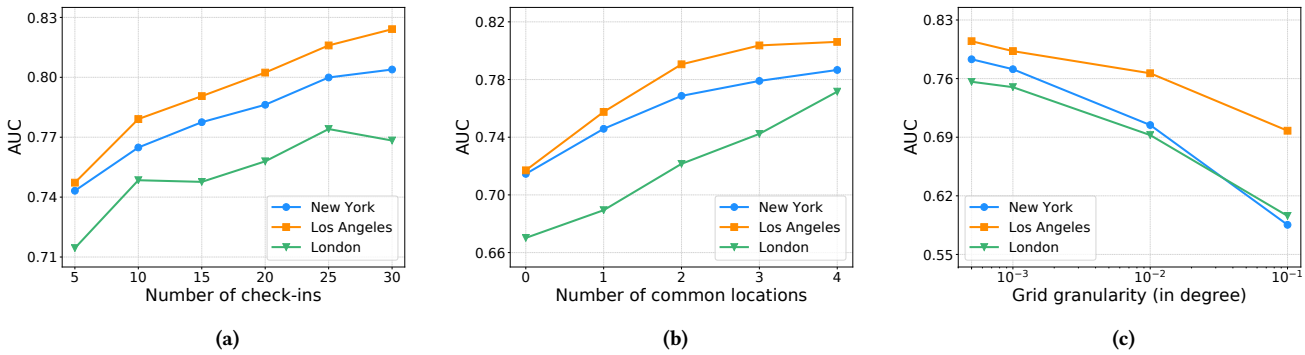


Figure 5: Evolution of the attack performance with respect to (a) the minimal number of check-ins shared by every user, (b) the number of common locations two users share and (c) different granularities of the geographic grids.

as shown in Figures 4a and 4b. The AUC values in all three cities increase sharply when l_w increases from 10 to 50, and saturates afterwards. Similarly, increasing t_w from 2 to 10 leads to around 5% performance gain in all the cities.

The effect of the vector dimension (d), on the other hand, is more subtle. Previous studies [15, 26, 34] have shown that larger d results in better performance on node-level prediction. However, the last stage of our inference attack relies on measuring two vectors' pairwise similarity, in this case longer vectors do not always yield better performance, as observed in other data domains such as biomedical data [3]. As depicted in Figure 4c, AUC is rather stable when increasing d compared to l_w and t_w, especially for Los Angeles: regardless of the choice of d, AUC stays around 0.80. In conclusion, our default hyperparameter settings are suitable for our inference attack.

4.4 Attack Robustness

Number of check-ins. As discussed above, our inference attacks are performed on active users, i.e., users with at least 20 check-ins in each city. This is in line with the existing works on social relation inference attacks and mining user check-ins in general. However, the optimal definition for active users, i.e., how many check-ins a user should at least share, is not clear. The authors of [9] use 10, [43] uses 40, and [42] uses the top 5,000 users with the most check-ins. To demonstrate that our attack's performance is robust under all circumstances, we further study the different choices for defining active users with respect to AUC.

Figure 5a shows that, as we increase the minimum number of check-ins, AUC increases almost linearly, especially for New York and Los Angeles. This is expected since the more check-ins a user shares, the more accurately the adversary can profile his mobility, which in turn leads to a better social relation inference. More importantly, even when concentrating on users with at least 5 check-ins, our inference attack still achieves a strong performance, e.g., AUC is near 0.75 in Los Angeles. This indicates that our model can effectively infer a large number of individuals' social links, and it shows the extent of the privacy threat carried by mobility data at a large scale. We also discover that the performance differences between our attack and the best baseline models are consistent under different active user definition. These results demonstrate the robustness of our inference attack.

Number of common locations. One of the major advantages of our inference attack is that it can predict any two individuals' social relationship regardless of whether they share common locations. Nevertheless, we expect that two users sharing many locations in common will be more likely to be socially related than two users sharing none. Therefore, we evaluate here how our inference attack performs with respect to the number of locations users have in common. Recall that, by common location, we mean any location where two users have checked in, not necessarily at the same time.

We select a subset of pairs of users who share between 0 and 4 locations in common, which includes the vast majority of pairs of users (see Figure 2a), and show the results by number of common locations, in Figure 5b . We observe that the inference performance increases monotically with the number of common locations between two users. However, even when two users share no common locations, our inference attack can still predict social links with fair performance (AUC around 0.7), especially for New York and Los Angeles (AUC equal to 0.72). This is essentially due to the fact that our inference attack takes mobility neighborhood into account. With a random walk method, a user's mobility neighbors not only consist of locations he visits but also of users who visit the same locations as him and the locations these users visit. This enables to establish a connection between users sharing no common locations. It is worth noting that our inference attack performs much better than baselines when there is no common location available between users. Indeed, the most effective baselines have AUCs close to 0.5 (equivalent to random guessing) when two users share no location in common.

4.5 Attack with Geographical Grids

So far, our attack has been performed on fine-grained mobility data, i.e., check-ins at POIs. However, in some cases, the adversary may not have access to mobility data with such fine-grained location information, but only geo-coordinates (latitude and longitude). In this subsection, we investigate whether our inference attack is still effective in this situation.

To proceed, we partition the region covered by each city into geo-grids, and assign a check-in into a grid if its geo-coordinates lie in the grid. In our experiments, we have tried multiple granularity for partitioning, including 0.0005°, 0.001°, 0.01°and 0.1° (similarly to the partitioning used in [10]). Results are presented in Figure 5c. At the finest granularity, i.e., 0.0005° (around 50m by 50m), our inference attack achieves similar results as the case of POIs.[10] We have AUC equal to 0.80 in Los Angeles, 0.79 in New York and 0.75 in London. With geo-grids being coarser-grained, the AUC values decrease monotonically. However, even when the adversary only has the geo-coordinates at the granularity of 0.01° (around 1km by 1km), our inference algorithm still performs quite well. More interestingly, at the coarsest granularity, the AUC value is around 0.7 in Los Angeles, while the results are much worse in New York and London. This can be explained by different location densities in different cities. Locations in Los Angeles are more uniformly distributed in the geo-space and distant from each other than those in the other two cities. In conclusion, our attack is also effective when fine-grained location information is not available, which further demonstrates the generality of our approach.

5 COUNTERMEASURES

In this section, we present three obfuscation mechanisms for enhancing users' social relationship privacy while preserving the check-in dataset's utility as much as possible. The three mechanisms, namely hiding, replacement and generalization, are based on well-founded obfuscation schemes proposed by the research community [6, 38] for protecting users' location privacy. We extend them here to protect users' social link privacy. Since these defense mechanisms are not specific to a certain inference attack, we evaluate them not only on our attack but also on baseline models introduced in Section 4.

We first describe the utility metric considered in our defense, then present the obfuscation mechanisms in detail and, finally, we experimentally study the performance of our defense.

5.1 Utility Metric

One approach to quantify utility is to consider the global properties of the obfuscated dataset, such as the check-in distribution over all locations in each city. However, metrics of this kind neglect the individual check-in behavior, and could lead to obfuscated datasets becoming useless for a handful of applications, such as location recommendation [14, 33]. For keeping as much user utility as possible, we design a metric which aims at measuring to what extent each user's check-in distribution is preserved.

We first denote a user u's check-in distribution in the original dataset as $P_u^o(A)$ where A is the random variable to represent locations a user has visited. Formally,

$$P_u^o(A=\ell) = \begin{cases} \frac{|\tau(u,\ell)|}{|\tau(u)|} & \text{if } \ell \in \omega(u), \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Accordingly, u's check-in distribution in the dataset obfuscated by a certain defense mechanism b is defined as $P_u^b(A)$ and $P_u^b(A=\ell) = \frac{|\tau^b(u,\ell)|}{|\tau^b(u)|}$ for $\ell \in \omega^b(u)$. Here, $\tau^b(u,\ell)$, $\tau^b(u)$ and $\omega^b(u)$ denote u's check-ins at ℓ, u's check-ins and the set of unique locations he has visited in the obfuscated dataset, respectively. Then, u's utility loss is defined as the statistical distance between $P_u^o(A)$ and $P_u^b(A)$. In this work, we adopt Jensen-Shannon divergence as the statistical distance. Formally, u's *utility loss* is defined as

$$\begin{aligned} \phi^b(u) = \sum_{\ell \in \mathcal{L}} & P_u^o(A=\ell) \log_2 \frac{P_u^o(A=\ell)}{M_u(A=\ell)} + \\ & P_u^b(A=\ell) \log_2 \frac{P_u^b(A=\ell)}{M_u(A=\ell)}, \end{aligned} \tag{10}$$

where $M_u(A=\ell) = \frac{P_u^o(A=\ell)+P_u^b(A=\ell)}{2}$. We use Jensen-Shannon divergence since it satisfies the symmetry property of a distance metric (contrary to the Kullback-Leibler divergence), and has been used in previous works such as [27]. Moreover, Jensen-Shannon divergence lies in the range between 0 and 1 which allows us to easily define utility from the Jensen-Shannon divergence as follows:

$$\psi^b(u) = 1 - \phi^b(u). \tag{11}$$

In the end, the utility of the whole dataset after applying b is defined as the average utility loss over all users

$$\Psi^b = \sum_{u \in \mathcal{U}} \frac{\psi^b(u)}{|\mathcal{U}|}. \tag{12}$$

5.2 Obfuscation Mechanisms

We now introduce the three obfuscation mechanisms for protecting social link privacy.

Hiding. This mechanism simply removes a certain proportion of check-ins in the original dataset. The check-ins to be removed are randomly sampled and the remaining check-ins are used to calculate the utility following the previous definition.

Replacement. This mechanism replaces a certain proportion of check-ins' locations with other locations to mislead the adversary. A location in a certain check-in can be replaced by any location in the dataset. In order to retain as much utility as possible, we adopt the random walk approach proposed by Mittal et al. [27] to find locations close to the original ones from a social mobility point of view. For each check-in $\langle u, t, \ell \rangle$ chosen to be replaced, we perform a random walk from u on the bipartite graph $\mathcal{G}$ and replace the location of the check-in with the last node in the random walk trace. Since $\mathcal{G}$ is bipartite, the length of the random walk trace, another hyperparameter, needs to be odd such that the random walk stops at a location (not at a user). We empirically study how its length affects the performance of replacement with respect to inference performance and utility in the evaluation subsection.

It is worth noting that random walk used here has a different purpose from the random walk used in the first stage of our inference attack (Section 3). The latter aims to reorganize $\mathcal{G}$ into random walk traces for skip-gram to learn each user's mobility features, while the former utilizes the graph structure to find close locations in order to keep the utility of the obfuscated dataset.

[10] Note that there are already multiple POIs mapped to one single grid with 0.0005°, including POIs at the same latitude-longitude position but different height (e.g., in a building).

Generalization. As presented in Section 4, for each location, we have its category information (collected from Foursquare) and geo-coordinates, i.e., latitude and longitude. Our third defense mechanism aims at generalizing both the semantic and geographical dimensions.

Foursquare organizes its location categories[11] into a two-level tree structure: 9 high-level categories and 427 low-level categories.[12] Therefore, for semantic generalization, we logically rely on the two-category levels provided by Foursquare. For geographical generalization, we partition check-ins into geographic grids of different granularity (as in Section 4.5). Here, we also consider two-level generalization: 0.01° (around 1 km by 1 km) grids for low-level generalization, and 0.1° (around 10 km by 10 km) grids for high-level generalization. We consider 0.01° as low-level generalization and not 0.001° since, as shown in Figure 5c, the inference performance with 0.001° grids is almost as good as for the original attack. As in [6], geographic and semantic generalizations are considered jointly, which gives us four different combinations of generalization, denoted by *lg-ls* (low-level geo-grid, low-level semantics), *lg-hs* (low-level geo-grid, high-level semantics), *hg-ls* (high-level geo-grid, low-level semantics) and *hg-hs* (high-level geo-grid, high-level semantics).

Different from hiding and replacement, the generalization mechanism will modify the original set of locations (IDs) in the dataset by merging multiple locations belonging to the same generalized location together. However, when the adversary obtains the generalized dataset, he can use external knowledge to map the generalized locations back to the original ones, and thereby increase the inference performance or utility provided to the user, respectively. For instance, MoMA and Bernarducci Meisel Gallery in New York are generalized into the same location under *lg-hs*, i.e., art and entertainment place at geographic coordinates (40.76° N, -73.97° W). When a user shares a check-in at this generalized location, the attacker or service provider is more confident that the check-in is at MoMA than at Bernarducci Meisel Gallery, since the former is much more popular than the latter.

In order to get conservative privacy guarantees for the generalization mechanism, we assume the adversary and service provider to be equipped with such external knowledge. Practically, we construct the adversary's background knowledge by collecting each location's total number of check-ins from Foursquare's API (independently from the Instagram data). For each check-in shared at a generalized location, we sample a location that is included in this generalized location as the check-in's original location with a sampling rate equal to the proportion of check-ins at this original location in the generalized location area. [13]

5.3 Defense Evaluation

We evaluate all the three obfuscation mechanisms against our inference attack as well as baseline models. Both hiding and replacement mechanisms involve randomly obfuscating a certain proportion of check-ins in the original dataset. In our experiments, we choose to hide or replace from 10% to 90% check-ins in incremental steps of 10%. For presentation purposes, we only depict the results for New York, results for Los Angeles and London following a similar trend and being presented in Appendix A.2.

Figure 6a presents our inference attack's performance against hiding and replacement. We observe that replacement is more effective than hiding on decreasing our inference attack's performance when the proportion of obfuscated check-ins is fixed. For instance, when obfuscating 30% of check-ins, replacement decreases our attack's AUC by 7% while hiding only decreases it by 3%. Moreover, in order to degrade the inference performance sufficiently to make a poor prediction (AUC < 0.7), we need to hide 80% of the check-ins or replace 50% of them. This is due to the fact that the replacement mechanism introduces more noise to the original dataset than randomly hiding check-ins, which will result in skip-gram learning less informative features for each user. However, as hiding does not cause significant changes to a user's mobility distribution, it preserves more utility than replacement for a fixel level of obfuscation (Figure 6b). This demonstrates that there exists a tension between privacy and utility in social link privacy protection, and that there is no free lunch in such a setting.

We empirically evaluate the impact of the number of steps considered in the random walk for the replacement mechanism. Our experiments show that increasing the steps from 5 to 15 decreases attack performance quite significantly (Figure 6a), but that further step increase does not provide much more privacy to the users (as the AUC value then saturates for all obfuscation proportions). The same decreasing behavior holds for utility, but the difference is much smaller between 5 steps and 15, 25 and 35 steps than for the AUC value decrease. By further taking into account the computational time (bigger walk steps leads to longer execution time), we believe that 15 provides the best trade-off between privacy, utility, and efficiency for the replacement mechanism. Figure 6c further shows AUC for hiding and replacement against the three best performing baseline models, i.e., `w_common_p`, `common_p` and `overlap_p`. As for our attack, replacement is more effective than hiding on decreasing the AUC of the baselines for all proportions of obfuscation except 90%.

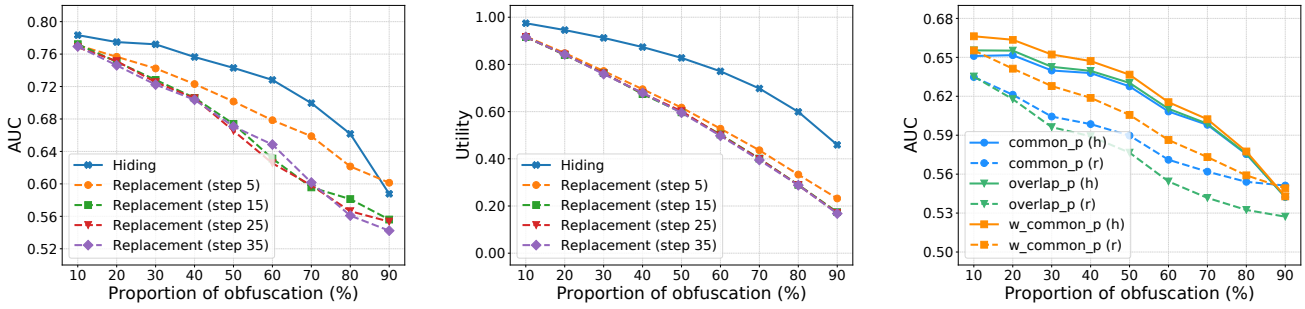
Table 2 presents the AUC values and utility of the generalization mechanism (for our attack and the three best baselines). First, we observe that higher-level generalization leads to the worst inference performance, thus best privacy provision, as expected. However, we also notice that utility is decreased a lot with this countermeasure, down to 0.06 for maximal generalization. Interestingly, the lowest-level generalization, i.e., *lg-ls*, is not very helpful for social link privacy (AUC = 0.77 compared to AUC = 0.79 without countermeasure) for a utility decrease that is still substantial. This indicates that generalization does not provide an optimal balance between utility and privacy. This is essentially due to the fact that the external knowledge (about location popularity) helps the adversary improve his inference attack in presence of this countermeasure.

Second, *lg-hs* provides a better inference performance and utility than *hg-ls*, which means that getting more precise geo-coordinates is more informative about social relationships than having more precise semantic information. Nevertheless, by comparing results from Figure 5c in Section 4.5 to those reported here, we clearly observe that semantic information brings a lot of information to the

[11] https://developer.foursquare.com/categorytree

[12] This number is based on the result given by Foursquare's API in January 2016

[13] We do not consider external knowledge in Section 4.5 since we want to evaluate the performance of our attack. In that case, a simple adversary is a reasonable choice. On the other hand, for evaluating the generalization mechanism and get safe privacy guarantees, it is necessary to consider a stronger adversary with external knowledge.



(a) (b) (c)

Figure 6: Hiding vs. replacement with respect to the proportion of obfuscation: (a) inference performance (AUC) of our attack, (b) utility, and (c) inference performance (AUC) of baseline models in New York. The length of random walk steps in replacement is set to 15 for baseline models, h represents hiding and r stands for replacement.

adversary (as shown in [2] for location inference). Indeed, we notice that the AUC here with *hg-hs* is equal to 0.67 whereas it is equal to around 0.6 in Figure 5c with similiar geographic information but no semantics. Hence, we see that even high-level semantic information brings sufficient knowledge to increase the attack's AUC by 12%. Lower-level semantic data increases it by 22% to 0.73.

We further calculate the adversary's recovery rate, i.e., the proportion of original check-ins that are recovered. The results are presented in Table 2 too. As we can see, when the generalization level is *lg-ls*, the adversary is able to recover 52% of the original location IDs. Given that we only use a very simple recovery algorithm based on the global locations' distribution, this confirms that generalization is not enough to protect location and social link privacy against adversaries with external knowledge. Moreover, *lg-hs* has a higher location recovery rate than *hg-ls* (23% vs. 14%), which also explains why the attacker achieves a higher AUC in *lg-hs* than in *hg-ls*.

Table 2: Inference performance and utility for generalization in New York.

	AUC		Utility		Recovery rate	
	ls	*hs*	*ls*	*hs*	*ls*	*hs*
lg	0.77	0.75	0.57	0.30	52%	23%
hg	0.73	0.67	0.20	0.06	14%	2%
	`w_common_p`		`overlap_p`		`common_p`	
	ls	*hs*	*ls*	*hs*	*ls*	*hs*
lg	0.65	0.63	0.65	0.63	0.65	0.64
hg	0.61	0.58	0.60	0.57	0.62	0.58

When comparing the three obfuscation mechanisms by fixing the AUC value (with our inference attack), hiding and replacement achieve a comparable performance in general, and they both outperform generalization (Figure 7). For instance, if we want to achieve a utility of at least 0.6, then the AUC values of hiding and replacement are very close to each other, of 0.66 and 0.67, respectively. However, we observe that, for a similar AUC value, utility drops to 0.06 with the generalization mechanism. From Figure 7, it seems that hiding

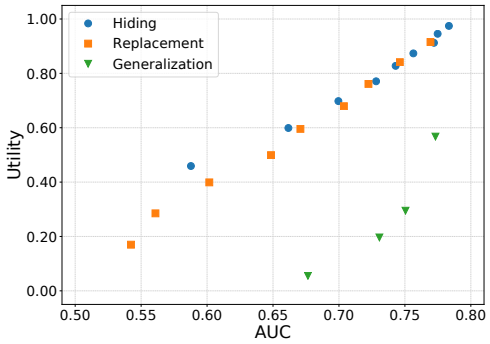


Figure 7: AUC vs. utility for three obfuscation mechanisms.

performs better than replacement. But we should also notice that replacement can decrease inference attack's performance, thus improve privacy, to a larger extent than hiding: when obfuscating 90% check-ins, replacement decreases our attack's AUC to 0.54, while hiding only leads to a minimal AUC of 0.59.

6 RELATED WORK

With the increasing usage of portable devices, a large amount of mobility data has become available. On the one hand, this represents an unprecedented chance to study the interaction between human mobility and social networks. On the other hand, it raises new concerns towards privacy. In the following, we separate the most related literature into two main research topics. The first line of research concentrates on inferring hidden location information from social data while the second line focuses on leveraging mobility data to infer social relationships.

Backstrom et al. [4] develop a maximal likelihood estimator to predict a user's undisclosed home location with his friends' data. Experiments on a large Facebook dataset show that their model outperforms traditional IP-based approaches significantly. Following this work, the authors of [23] have incorporated fine-grained social

relation information into their home location prediction model. Evaluation on a Twitter dataset has demonstrated that social features such as number of followers indeed increase the prediction performance. Cho et al. [9] have observed on a Gowalla dataset that a user's mobility is centered around two states: home and work. They develop a Gaussian mixture model to learn the two hidden states and further incorporate friendship influence. Extensive experiments demonstrate the effectiveness of their approach. Recently, Olteanu et al. [30] have shown how co-location information about OSN users (e.g., via location check-ins with two or more users) can be used by an attacker to degrade these users' location privacy. They provide an analytical framework based on Bayesian networks to formally quantify the effects of co-location data on location privacy and also consider the impact of some obfuscation mechanisms. Other interesting works in this direction include [17, 18, 31, 36, 43].

The second line of research tackles the dual problem, i.e., using mobility data to infer the underlying social relations. Our inference attack and the baseline models [35, 37, 42] we compare it to fall into this topic. Eagle et al. [13] have first shown that there exist correlations between people's co-occurrences and their social connections by conducting a study based on mobile phone records. Crandall et al. [10] go one step beyond by relying on a Bayesian model to show that the friendship probability of two users with joint mobility behavior is 5,000 times higher than those without joint behavior. These results shed light on the social relation privacy threat carried by mobility data. However, the model they use makes an over-simplified assumption that each user only has one friend.

Scellato et al. [37] tried to get closer to a realisitic setting by proposing 15 novel machine learning features. Among the 15 features, 4 of them follow a classical link prediction setting [20] by relying on some existing social network structure. In our work, we assume that our adversary has no knowledge of any existing social links. Besides, we evaluate all the other 11 features as part of the baseline models. Moreover, their evaluation is conducted on some predefined inference spaces such as two users need to share common friends or common locations. In our experiments, we do not impose any constraint on the mobility profiles of users, and thus make a more realistic evaluation of these baseline models and our inference attack.

Pham et al. [35] propose two features for social link inference, i.e., `diversity` and `w_frequency`. The former concentrates on the diversity of two users' joint check-in behaviors and the latter reflects the popularity of two users' common locations. Both `diversity` and `w_frequency` are based on entropy measures. The authors of [42] propose three mobility factors, namely personal, global and temporal, Among them, the global factor is the same as `w_frequency` in [35], while the personal factor (`personal`) follows the intuition that two users are more likely to know each other if they meet at locations they do not visit frequently.

Different from [37], both [35] and [42] consider two users' meeting events (visiting the same location at roughly the same time) instead of common locations. However, meeting events are really rare even in our large dataset, meaning that the methods in [35] and [42] can only apply to a small set of users. Even when we concentrate on users with meeting events, features in [35] and [42] do not achieve any performance gain compared to the case of common locations, especially for `personal` in [42], where the performance even worsens. Therefore, we decide to use common locations as in [37] instead of meeting events to evaluate the baselines in [35] and [42].[14] As shown in Section 4, our inference attack significantly outperforms these baselines, which demonstrates the effectiveness and relevance of our approach.

7 CONCLUSION

Mobility data are nowadays largely available to a wide range of service providers. This raises many privacy issues, especially when such providers' data ends up into the hands of intelligence agencies. This paper aims at evaluating, with a principled approach, the impact on social link privacy of this wide availability of location data. To this endeavor, we propose a new generic method for inferring social links without imposing any prior condition on users' mobility patterns. Furthermore, we design countermeasures for mitigating the extent of the privacy threat towards social relationships.

The empirical evaluation of our inference attack demonstrates that our principled approach outperforms previously proposed inference algorithms by up to 20% on a large-scale dataset, with an area under the ROC curve of around 0.8. Our results further show that our attack provides fair prediction results (AUC equal to 0.71 or 0.75 depending on the targeted city) even when the number of available location points per user is small (down to 5). Moreover, our attack is quite robust to a low number of common locations between two users. For two cities, it even provides fair prediction performance (AUC around 0.72) when two users share no location at all in common. Finally, we observe that our attack performs also well with geographic grids of size up to 1-by-1 km instead of exact semantic and geographic location data.

In order to counter the presented attack against social link privacy, we propose and evaluate three well-established privacy-preserving techniques: hiding, replacement and generalization. Our empirical results demonstrate that, in order to degrade the inference performance sufficiently to make a poor prediction (AUC smaller than 0.7), we need to hide 80% of the location points or replace 50% of them. However, we notice also that replacement decreases utility more than hiding, which shows that there is no free lunch in such a privacy setting. Furthermore, we notice that the generalization mechanism provides a much poorer privacy-utility trade-off than the hiding and replacement techniques. Finally, by comparing our defense and attack results, we observe that the semantic dimension of locations can have substantial positive effect on the social link inference when geographic information is obfuscated with generalization.

ACKNOWLEDGMENTS

This work was partially supported by the German Federal Ministry of Education and Research (BMBF) through funding for the Center for IT-Security, Privacy and Accountability (CISPA) (FKZ: 16KIS0656). Part of this work was carried out while Mathias Humbert was with CISPA, Saarland University. The authors would like to thank Rose Hoberman, Jonas Schneider and Kathrin Grosse for their valuable comments on the submitted manuscript.

[14] Following the same reason, we do not implement the temporal factor in [42] as one baseline model.

REFERENCES

[1] 2013. How the NSA is tracking people right now. https://www.washingtonpost.com/apps/g/page/world/how-the-nsa-is-tracking-people-right-now/634/. (2013).

[2] Berker Ağır, Kévin Huguenin, Urs Hengartner, and Jean-Pierre Hubaux. 2016. On the Privacy Implications of Location Semantics. *Proceedings on Privacy Enhancing Technologies* 2016, 4 (2016), 165–183.

[3] Michael Backes, Pascal Berrang, Anne Hecksteden, Mathias Humbert, Andreas Keller, and Tim Meyer. 2016. Privacy in Epigenetics: Temporal Linkability of MicroRNA Expression Profiles. In *Proceedings of the 25th USENIX Security Symposium (Security)*. USENIX, 1223–1240.

[4] Lars Backstrom, Eric Sun, and Cameron Marlow. 2010. Find Me If You Can: Improving Geographical Prediction with Social and Spatial Proximity. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*. ACM, 61–70.

[5] Igor Bilogrevic, Kévin Huguenin, Murtuza Jadliwala, Florent Lopez, Jean-Pierre Hubaux, Philip Ginzboorg, and Valtteri Niemi. 2013. Inferring Social Ties in Academic Networks Using Short-Range Wireless Communications. In *Proceedings of the 12th Workshop on Privacy in the Electronic Society (WPES)*. ACM, 1–10.

[6] Igor Bilogrevic, Kévin Huguenin, Stefan Mihaila, Reza Shokri, and Jean-Pierre Hubaux. 2015. Predicting Users' Motivations behind Location Check-ins and Utility Implications of Privacy Protection Mechanisms. In *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS)*.

[7] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You Are Where You Tweet: A Content-Based Approach to Geo-locating Twitter Users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 759–768.

[8] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z. Sui. 2011. Exploring Millions of Footprints in Location Sharing Services. In *Proceedings of the 5th International Conference on Weblogs and Social Media (ICWSM)*. The AAAI Press, 81–88.

[9] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-based Social Networks. In *Proceedings of the 17th ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 1082–1090.

[10] David J. Crandall, Lars Backstrom, Dan Cosley, Siddharth Suri, Daniel Huttenlocher, and Jon Kleinberg. 2010. Inferring Social Ties from Geographic Coincidences. *Proceedings of the National Academy of Sciences* 107, 52 (2010), 22436–22441.

[11] Ratan Dey, Zubin Jelveh, and Keith Ross. 2012. Facebook Users Have Become Much More Private: A Large-Scale Study. In *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 346–352.

[12] Yuxiao Dong, Jie Tang, Sen Wu, Jilei Tian, Nittiest V Chawla, Jinghai Rao, and Huanhuan Cao. 2012. Link Prediction and Recommendation across Heterogeneous Social Networks. In *Proceedings of the 12th International Conference on Data Mining (ICDM)*. IEEE, 181–190.

[13] Nathan Eagle, Alex Sandy Pentland, and David Lazer. 2009. Inferring Friendship Network Structure by Using Mobile Phone Data. *Proceedings of the National Academy of Sciences* 106, 36 (2009), 15274–15278.

[14] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring Temporal Effects for Location Recommendation on Location-Based Social Networks. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*. ACM, 93–100.

[15] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 855–864.

[16] Mathias Humbert, Théophile Studer, Matthias Grossglauser, and Jean-Pierre Hubaux. 2013. Nowhere to Hide: Navigating around Privacy in Online Social Networks. In *Proceedings of the 18th European Symposium on Research in Computer Security (ESORICS)*. Springer, 682–699.

[17] David Jurgens. 2013. ThatâĂŹs What Friends Are For: Inferring Location in Online Social Media Platforms Based on Social Relationships. In *Proceedings of the 7th International Conference on Weblogs and Social Media (ICWSM)*. The AAAI Press, 273–282.

[18] David Jurgens, Tyler Finethy, James McCorriston, Yi Tian Xu, and Derek Ruths. 2015. Geolocation Prediction in Twitter Using Social Networks: A Critical Analysis and Review of Current Practice. In *Proceedings of the 9th International Conference on Weblogs and Social Media (ICWSM)*. The AAAI Press, 188–197.

[19] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a Social Network or a News Media?. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*. ACM, 591–600.

[20] David Liben-Nowell and Jon Kleinberg. 2007. The Link-Prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology* 58, 7 (2007), 1019–1031.

[21] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. 2010. New Perspectives and Methods in Link Prediction. In *Proceedings of the 16th ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 243–252.

[22] Lydia Manikonda, Yuheng Hu, and Subbarao Kambhampati. 2014. Analyzing User Activities, Demographics, Social Network Structure and User-Generated Content on Instagram. *CoRR* abs/1410.8099 (2014).

[23] Jeffrey McGee, James Caverlee, and Zhiyuan Cheng. 2013. Location Prediction in Social Media Based on Tie Strength. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 459–468.

[24] Aditya Krishna Menon and Charles Elkan. 2011. Link Prediction via Matrix Factorization. In *Proceedings of the 2011 Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*. Springer, 437–452.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR)*.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionally. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 3111–3119.

[27] Prateek Mittal, Charalampos Papamanthou, and Dawn Song. 2013. Preserving Link Privacy in Social Network Based Systems. In *Proceedings of the 20th Network and Distributed System Security Symposium (NDSS)*.

[28] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the Crowd: The Privacy Bounds of Human Mobility. *Scientific Reports* 3 (2013), 1376.

[29] Arvind Narayanan and Vitaly Shmatikov. 2009. De-anonymizing Social Networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy (S&P)*. IEEE, 173–187.

[30] Alexandra-Mihaela Olteanu, Kévin Huguenin, Reza Shokri, Mathias Humbert, and Jean-Pierre Hubaux. 2017. Quantifying Interdependent Privacy Risks with Location Data. *IEEE Transactions on Mobile Computing* 16, 3 (2017), 829–842.

[31] Jun Pang and Yang Zhang. 2015. Location Prediction: Communities Speak Louder than Friends. In *Proceedings of the 3rd ACM on Conference on Online Social Networks (COSN)*. ACM, 161–171.

[32] Jun Pang and Yang Zhang. 2017. DeepCity: A Feature Learning Framework for Mining Location Check-Ins. In *Proceedings of the 11th International Conference on Web and Social Media (ICWSM)*. The AAAI Press, 652–655.

[33] Jun Pang and Yang Zhang. 2017. Quantifying Location Sociality. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media (HT)*. ACM, 145–154.

[34] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 701–710.

[35] Huy Pham, Cyrus Shahabi, and Yan Liu. 2013. EBM: An Entropy-Based Model to Infer Social Strength from Spatiotemporal Data. In *Proceedings of the 2013 ACM Conference on Management of Data (SIGMOD)*. ACM, 265–276.

[36] Adam Sadilek, Henry Kautz, and Jeffrey P. Bigham. 2012. Finding Your Friends and Following Them to Where You Are. In *Proceedings of the 5th ACM Conference on Web Search and Data Mining (WSDM)*. ACM, 459–468.

[37] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. 2011. Exploiting Place Features in Link Prediction on Location-based Social Networks. In *Proceedings of the 17th ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 1046–1054.

[38] Reza Shokri, Georgios Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. 2011. Quantifying Location Privacy. In *Proceedings of the 32nd IEEE Symposium on Security and Privacy (S&P)*. IEEE, 247–262.

[39] Mudhakar Srivatsa and Mike Hicks. 2012. Deanonymizing Mobility Traces: Using Social Network as a Side-channel. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS)*. ACM, 628–637.

[40] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*. ACM, 1067–1077.

[41] Alastair J Walker. 1977. An Efficient Method for Generating Discrete Random Variables with General Distributions. *ACM Transactions on Mathematical Software* 3, 3 (1977), 253–256.

[42] Hongjian Wang, Zhenhui Li, and Wang-Chien Lee. 2014. PGT: Measuring Mobility Relationship using Personal, Global and Temporal Factors. In *Proceedings of the 14th IEEE Conference on Data Mining (ICDM)*. IEEE, 570–579.

[43] Fei Wu and Zhenhui Li. 2016. Where Did You Go: Personalized Annotation of Mobility Records. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 589–598.

[44] Yang Zhang and Jun Pang. 2015. Distance and Friendship: A Distance-based Model for Link Prediction in Social Networks. In *Proceedings of the 17th Asia-Pacific Web Conference (APWeb)*. Springer, 55–66.

A APPENDIX

A.1 Pairwise Similarity Measurements

We present the formal definitions of the 7 pairwise similarity measurements used in our evaluation.

Cosine similarity.

$$s(\theta(u), \theta(u')) = \frac{\theta(u) \cdot \theta(u')}{||\theta(u)||_2 \; ||\theta(u')||_2}$$

Euclidean distance.

$$s(\theta(u), \theta(u')) = ||\theta(u) - \theta(u')||_2$$

Correlation coefficient.

$$s(\theta(u), \theta(u')) = \frac{(\theta(u) - \overline{\theta(u)}) \cdot (\theta(u') - \overline{\theta(u')})}{||\theta(u) - \overline{\theta(u)}||_2 \; ||\theta(u') - \overline{\theta(u')}||_2}$$

Here, $\overline{\theta(u)}$ represents the mean value of $\theta(u)$.

Chebyshev distance.

$$s(\theta(u), \theta(u')) = \max_{i=1}^{d} |\theta(u)_i - \theta(u')_i|$$

Here, $\theta(u)_i$ represents the ith element in $\theta(u)$.

Bray-Curtis distance.

$$s(\theta(u), \theta(u')) = \frac{\sum_{i=1}^{d} |\theta(u)_i - \theta(u')_i|}{\sum_{i=1}^{d} |\theta(u)_i + \theta(u')_i|}$$

Canberra distance.

$$s(\theta(u), \theta(u')) = \sum_{i=1}^{d} \frac{|\theta(u)_i - \theta(u')_i|}{|\theta(u)_i| + |\theta(u')_i|}$$

Manhattan distance.

$$s(\theta(u), \theta(u')) = \sum_{i=1}^{d} |\theta(u)_i - \theta(u')_i|$$

A.2 Defense Evaluation for Los Angeles and London

The defense evaluation results for Los Angeles and London are presented as the following.

Table 3: Inference performance and utility for generalization in Los Angeles.

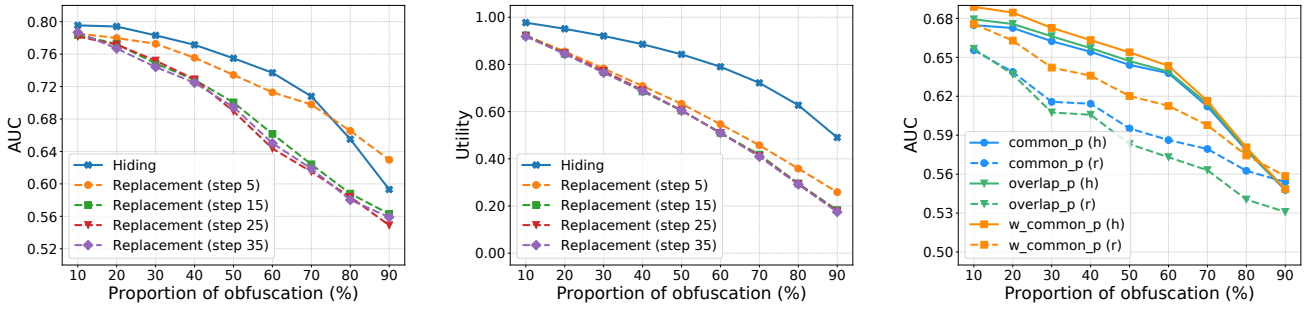
	AUC		Utility		Recovery rate	
	ls	*hs*	*ls*	*hs*	*ls*	*hs*
lg	0.79	0.78	0.79	0.48	74%	40%
hg	0.77	0.74	0.37	0.13	29%	7%

	w_common_p		overlap_p		common_p	
	ls	*hs*	*ls*	*hs*	*ls*	*hs*
lg	0.68	0.67	0.68	0.66	0.68	0.67
hg	0.66	0.63	0.66	0.63	0.66	0.64

Table 4: Inference performance and utility for generalization in London.

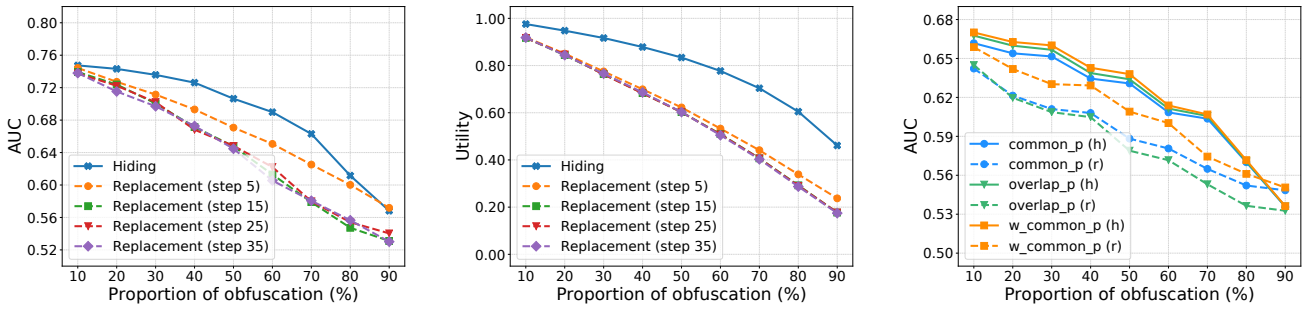
	AUC		Utility		Recovery rate	
	ls	*hs*	*ls*	*hs*	*ls*	*hs*
lg	0.74	0.72	0.72	0.43	68%	36%
hg	0.71	0.66	0.28	0.08	21%	4%

	w_common_p		overlap_p		common_p	
	ls	*hs*	*ls*	*hs*	*ls*	*hs*
lg	0.65	0.63	0.66	0.63	0.65	0.63
hg	0.63	0.59	0.62	0.58	0.62	0.59



(a) (b) (c)

Figure 8: Hiding vs. replacement with respect to (a) inference performance on our attack, (b) utility and (c) inference performance on baseline models in Los Angeles. The length of random walk steps in replacement is 15 for baseline models, h represents hiding and r represents replacement.



(a) (b) (c)

Figure 9: Hiding vs. replacement with respect to (a) inference performance on our attack, (b) utility and (c) inference performance on baseline models in London. The length of random walk steps in replacement is 15 for baseline models, h represents hiding and r represents replacement.