Securing Social Media User Data - An Adversarial Approach

Ghazaleh Beigi
Arizona State University
gbeigi@asu.edu

Kai Shu
Arizona State University
kai.shu@asu.edu

Yanchao Zhang
Arizona State University
yczhang@asu.edu

Huan Liu
Arizona State University
huan.liu@asu.edu

ABSTRACT

Social media users generate tremendous amounts of data. To better serve users, it is required to share the user-related data among researchers, advertisers and application developers. Publishing such data would raise more concerns on user privacy. To encourage data sharing and mitigate user privacy concerns, a number of anonymization and de-anonymization algorithms have been developed to help protect privacy of social media users. In this work, we propose a new adversarial attack specialized for social media data. We further provide a principled way to assess effectiveness of anonymizing different aspects of social media data. Our work sheds light on new privacy risks in social media data due to innate heterogeneity of user-generated data which require striking balance between sharing user data and protecting user privacy.

1 INTRODUCTION

Explosive growth of social media in the last decade has drastically changed the web and billions of people all around the globe can freely conduct numerous activities such as creating online profiles, interacting with other people, sharing posts, and various personal information in a rich *heterogeneous* environment [3, 4]. The resulted user-generated social media data consists of different aspects such as links, posts and profile information. This data provides opportunities for researchers and business partners to study and understand individuals at unprecedented scales [2, 6, 13].

However, publishing social media network data risks exposing people's privacy as the data is rich in content and relationship and contains individuals' sensitive and private information, resulting in privacy leakage [18, 24, 28]. For example, users' sensitive information such as vacation plans and medical conditions can be easily

inferred from their posts. Publishing complete and intact social media data could even result in inferring sensitive information the users do not explicitly disclose such as age and location [7].

Privacy issues of users mandate social media data publishers to protect users' privacy by anonymizing the data. One straightforward anonymization technique is to remove "Personally Identifiable Information" (a.k.a. PII) such as names, user ID, age and location information and keep the social graph structure as is. This solution has been shown to be far from sufficient to protects people's privacy [2, 24]. An example of this insufficient approach is the anonymized dataset published for the Netflix prize challenge. Later, the work of [23] showed that the structure of the data carried enough information for a potential breach of privacy to re-identify anonymized users. Consequently, various protection techniques have been proposed for anonymizing each aspect of the heterogeneous social media data. For example, some works perform anonymization on graph data structure [10, 20], and others anonymize users' location information [27]. In general, the ultimate goal of an anonymization approach is to preserve social network user privacy while ensuring the utility of published data.

Existing anonymization techniques often make a specific assumption regarding the way social media data is anonymized. In particular, these works assume that it's enough to anonymize each aspect of heterogeneous social media data (e.g., structure, textual, and location information) independently. At the first glance, this assumption makes sense as anonymization takes time and effort. Moreover, users privacy is protected while the data utility is preserved at the highest possible level. For example, lets consider the simplest case study in which published data includes only two aspects such as (i) structural (e.g., friendship, follower/followee links) and (ii) textual (e.g., posts) information. We will then have options as shown in Table 1 to anonymize the data: no anonymization for either aspect, anonymization for one aspect, and anonymization for both. To ensure anonymization efficiency, as each aspect can be of different data types, a common practice is to anonymize each aspect independently. With two aspects as shown in Table 1, case 4 is the backbone of the anonymization techniques for publishing data which is clearly the strongest protection of privacy.

Privacy advocates have argued that sensitive information could be still leaked from the dataset anonymized considering each of these cases, but we lack conclusive evidence. It is unclear how the latent relation between different aspects of the data could be captured, whether the sensitive information with the scale of millions of users could be still leaked and what the success rate of such an attack could be. In particular, in this research, we are interested to study these issues by answering the following research questions:

Table 1: Four different cases for social media data anonymization. Each check mark corresponds to the aspect of data being anonymized.

	Case 1	Case 2	Case 3	Case 4
Structural Anonymization	✗	✗	✓	✓
Textual Anonymization	✗	✓	✗	✓

- **(RQ1)**: Is the data private if just one of its two aspects is anonymized?
- **(RQ2)**: Is case 4, the strongest among four cases, sufficient for anonymizing social media data?

Following the work of [24], we seek to answer these questions by taking an adversary approach to assay the privacy level of anonymized social media data. However, existing de-anonymization attacks require a list of target users. A target user is an individual v with the known identity in social media network $\mathcal{T}$ which will be mapped to a user in the given anonymized dataset. These techniques also require background knowledge $\mathcal{B}_v$ for each targeted user v before initiating the attack. These methods require time and effort to find a proper set of target users and gather their knowledge which may not be realistic in practice. To address these challenges, we first introduce a new generation of adversarial attacks specialized for social media data which does not require collecting information before initiating the attack. Furthermore, to assess different ways of the social media dataset anonymization and answer the aforementioned questions, we propose a novel Adversarial Technique for Heterogeneous Data, namely, ATHD which utilizes the latent relationship between different aspects of data. This new approach particularly well suits for social media data in which it is concerned with assessing the strengths of anonymizing different aspects of data. Our contributions could be summarized as follows:

- We introduce a new generation of adversarial attacks applicable to social media network data.
- We propose a novel de-anonymization technique ATHD to assess the privacy level of anonymized heterogeneous social media data.
- We implement and evaluate ATHD on two real world datasets to study the strengths of anonymization techniques in context of heterogeneous social media data. Our results demonstrates hidden relations between different aspects of the heterogeneous data make data anonymization techniques inefficient.

2 BACKGROUND

In this section, we review the technical preliminaries of protecting user privacy in social media data, i.e. data anonymization, which is required for the rest of this discussion. Without loss of generality, in this paper, we assume that the published social media data consists of two aspects, namely, structure and textual information. More formally, we model the social network data as $\mathcal{D} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$ where $\mathcal{V} = \{i | i \text{ is a node}\}$ is the set of nodes or users, $\mathcal{E} = \{e_{i,j} | i, j \in \mathcal{V} \land \text{ there is a link from user } i \text{ to user } j\}$ is the set of links between any two nodes in $\mathcal{V}$ (e.g., friend and follower/followee relations), and $\mathcal{P} = \{\mathcal{P}_i | i \in \mathcal{V}\}$ is the set of all posts (textual information) associated with users in $\mathcal{V}$. $\mathcal{P}_i = \{p_1^i, p_2^i, ..., p_{m_i}^i\}$ denotes posts by user i where m_i is the number of posts for user i. Note that links in social networks could be either directed (e.g., follower/followee relation in Twitter) or indirected (e.g., friend relation in Facebook). We focus on directed graphs, although it is straightforward to apply

the settings on undirected graphs as well. In order to preserve users' privacy, data publisher should anonymize the social media data $\mathcal{D}$ using privacy preservation techniques. Next, we will discuss techniques deployed to secure structural and textual information.

2.1 Structural Information Anonymization

To anonymize structural information, we first remove users' personally identifiable information (PII) such as user's name and ID. Techniques such as k-degree anonymity [20], sparsification, perturbation and switching [16] are used for adding or removing nodes and links. The aim of k-anonymity methods is to anonymize each node so that it is indistinguishable from at least $k - 1$ other nodes [32]. Liu *et al.* proposed to achieve k-degree anonymization [20] through edge addition/deletion strategies [20]. Sparsification technique randomly removes a set of $p|\mathcal{E}|$ edges (p is the anonymiztion coefficient) while switching methods switches $\frac{p|\mathcal{E}|}{2}$ pairs of edges. Perturbation approach first removes a set of $p|\mathcal{E}|$ edges and then add same amount of edges randomly [16].

2.2 Textual Information Anonymization

In this work, we anonymize the textual information using ϵ-differential privacy [10] by first converting each user's post into a numerical vector using tokenizing and calculating Term Frequency Inverse Document Frequency (TF-IDF) scores and then adding Laplacian noise to the text vector. Details are discussed next.

2.2.1 Text Processing. To anonymize user i's posts, we first remove user's PII such as user ID (including mentioning and retweeting), name and link information from her texts. Then, we follow a standard process to convert each of user's posts to a numerical vector. To do so, we first consider posts by all users in the dataset and perform some pre-processing including stop word removal. The unigram model is then deployed to construct the word feature space $\mathcal{W}$. Finally, we use Term Frequency Inverse Document Frequency (TF-IDF) as a feature weight to derive the vector $\mathbf{x}_l^i$ for each post p_l^i of user i. TF-IDF score for each word t is calculated as:

$$\mathbf{x}_l^i(t) = f_l^i(t) * \log \frac{M}{n_t} \tag{1}$$

where, $f_l^i(t)$ is the number of times word t appeared in the post p_l^i, M is the total number of posts in the data and n_t is the number of posts that the word t was used in them. We can represent p_l^i with the corresponding vector $\mathbf{x}_l^i$. All users' posts can be then denoted by the post-word matrix $\mathbf{X} \in \mathbb{R}^{M \times |\mathcal{W}|}$ where $|\mathcal{W}|$ denotes the size of the word space. Relations between users and posts can be also represented via a user-post matrix $\mathbf{W} \in \mathbb{R}^{N \times M}$ where N is the number of users and $\mathbf{W}_{ij} = 1$ if post j was posted by user i and $\mathbf{W}_{ij} = 0$ otherwise. Next, we will discuss how we leverage differential privacy technique to anonymize the textual information.

2.2.2 Differential Privacy. We use differential privacy technique proposed in [10] to anonymize the textual information. Differential privacy aims at maximizing privacy of users when a statistical query is submitted over a database and an answer is retrieved. Formally, ϵ-differential privacy is defined as follows:

Definition 2.1. (ϵ-Differential Privacy [10]).Given an input dataset $\mathcal{H}$, a query $f(\mathcal{H})$ and a desirable output range, a mechanism

$K(.)$ with an output range $\mathcal{R}$ satisfies ϵ-differential privacy iff,

$$\frac{Pr[K(f(\mathcal{H}_1) = R \in \mathcal{R})]}{Pr[K(f(\mathcal{H}_2) = R \in \mathcal{R})]} \leq e^{\epsilon} \quad (2)$$

for any datasets $\mathcal{H}_1, \mathcal{H}_2$ that differ in only one row, where ϵ is the privacy budget.

Larger values of ϵ result in a larger privacy loss as the changes of the database can be inferred more easily, and, smaller values of ϵ lead to smaller privacy loss and a higher tolerance of database to privacy breach. Note that the differential privacy is just a condition on a mechanism which releases the dataset. The mechanism which achieves ϵ-differential privacy is called sanitization. Laplacian mechanism is one popular sanitization technique which gives differential privacy for real valued queries by adding a Laplacian noise [10]. Assume that $f(\mathcal{H})$ is the real value response to a certain query f. Then, a random noise $Y(\mathcal{H})$ is generated from Laplacian distribution and added to $f(\mathcal{H})$ as:

$$K(f(\mathcal{H})) = f(\mathcal{H}) + Y(\mathcal{H}) \quad (3)$$

The Laplacian distribution has zero mean and a scale parameter $\Delta(f)/\epsilon$ where $\Delta(f)$ is the sensitivity of f and defined as the maximum variation of the query function between datasets differing in at most one record:

$$\Delta(f) = \max\|f(\mathcal{H}_1) - f(\mathcal{H}_2)\|_1 \quad (4)$$

The density function of the Laplacian noise will be computed as:

$$p(x) = \frac{\epsilon}{2\Delta(f)} e^{-\frac{|x|\epsilon}{\Delta(f)}} \quad (5)$$

Note that higher sensitivity $\Delta(f)$ of the query function f with fixed ϵ, implies more Laplacian noise added to $f(\mathcal{H})$.

2.2.3 Anonymizing Textual Information with Differential Privacy. In order to anonymize the post-word matrix $\mathbf{X}$ in a way that ϵ-differential privacy is preserved, we need to apply the discussed mechanism $K(.)$ on the original matrix $\mathbf{X}$ and transform it into a new one $X' = K(X)$. Instead of transforming the entire matrix $\mathbf{X}$ at once, we can transform each individual row of the matrix by adding a Laplacian noise to $\mathbf{X}_i$ to create a new row $\mathbf{X}'_i$. Considering the identity query function $f_I(\cdot)$ where $f(D) = D$, the sensitivity of $f_I(\cdot)$ can be defined as follows:

$$\Delta(f_I) = \max\|\mathbf{X}_i - \mathbf{X}_j\|_1 \quad (6)$$

where $\mathbf{X}_i$ and $\mathbf{X}_j$ are any two random row vectors from $\mathbf{X}$. Following the equation 3, a Laplacian noise will be added to each vector $\mathbf{X}_i$:

$$K(f_I(\mathbf{X}_i)) = \mathbf{X}_i + \left[Y_{i1}, ... Y_{i|\mathcal{W}|}\right], i = 1, ..., n \quad (7)$$

Similarly, Y_{ij}'s are drawn i.i.d. from Laplacian distribution with zero mean and $\Delta(f_I)/\epsilon$ scale parameter. After anonymizing the textual information, the anonymized post-word $\mathbf{X}$ and user-post $\mathbf{W}$ matrices will be published. The information regrading the word feature space $\mathcal{W}$ will be released by the data publisher as well.

3 SOCIAL MEDIA ADVERSARIAL ATTACK

De-anonymization techniques have been proposed in the literature as a counterpart to data anonymization research direction [11, 18, 26, 28, 35]. De-anonymization works further help improve anonymization techniques and reduce privacy breach by probing the potential

drawbacks of anonymization techniques. Figure 1(a) depicts how these de-anonymization approaches work. These works assume that the adversary has been given a list of target users to de-anonymize requiring adversarial to collect background knowledge about target users before initiating the attack [1].
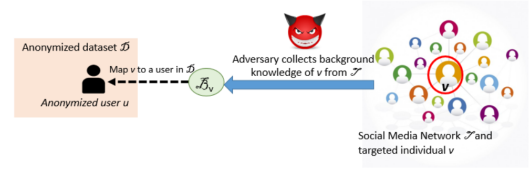
(a) Traditional de-anonymization.

(b) Proposed social media adversarial attack

Figure 1: Traditional de-anonymization vs. proposed social media adversarial attack.

Narayanan et.al. [24] discuss different ways of collecting background knowledge such as crawling data via social media networks API. Since these methods require time and effort to gather knowledge, it may not be realistic in practice for two reasons: (1) the number of target users can be very large, thinking about the number of users in Twitter; and (2) most of the online social media APIs have rate limits on the number of request a user can make through their APIs in a specific time window. Also, these APIs can only provide a random small portion of available data for each search query. This makes it infeasible to collect the background information for a significant number of users in $\mathcal{T}$ in order to find the one-to-one mapping between users in $\mathcal{D}$ and $\mathcal{T}$. Therefore, the above target-user-based approach cannot be applied to social media users when no list of target user is given.

To address these shortcomings, we introduce a new generation of adversarial attacks (Figure 1(b)) specialized for social media network data. This approach does not require the attacker to gather background knowledge $\mathcal{B}$ before starting the attack. In fact, users registered in social media which are available via online APIs are the adversaries' only source of information. The adversary can send queries to these APIs, anytime during the adversarial process. It is formally defined bellow:

Definition 3.1. (Social Media Adversarial Attack). Given an anonymized social media network dataset $\mathcal{D}$, the aim of adversarial attack is to find a one-to-one mapping between each user u in $\mathcal{D}$ and a real identity in targeted online social media network $\mathcal{T}$.

Next, we will accordingly discuss the details of our proposed de-anonymization approach, ATHD, which does not require collecting target users and their background information and is proposed to further evaluate heterogeneous social media anonymization.

4 ADVERSARIAL TECHNIQUE FOR HETEROGENEOUS DATA

Our proposed de-anonymization technique, adversarial technique for heterogeneous data (ATHD), uses different aspects of data, i.e., graph structure and users' textual information to identify the real identity of users in the anonymized dataset $\mathcal{D} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{W}, \mathcal{W})$. We posit that this attack could be applied on various aspects of data and is not limited to only two data aspects or structural and textual information.

The main idea behind de-anonymization is to find the most similar user in social media $\mathcal{T}$ to the user u in the anonymized dataset. Here, we follow the same approach as the existing works, meanwhile our goal is to design a new framework which exploits the hidden relations between different aspects of the data, to eventually map the users to their real profile in $\mathcal{T}$.

Our proposed de-anonymization consists of three main steps. Given the anonymized dataset $\mathcal{D}$, we first extract the most revealing information for u. Second, we search those information in search engine of the targeted social media $\mathcal{T}$. This search returns a list of people whose posts include the inquired query. We save all the returned candidates as a candidate set. Third, we identify the profile from the candidate set most similar to the user u . The details of each of three steps are discussed next.

4.1 Step 1: Extracting the Most Revealing Information

The first step includes extracting the most revealing information for user u via social media API. In this work, we rather use textual information since it is not straightforward to look up information related to links. We are thus interested in extracting the most revealing textual information of user u. We assign a score s_l to each post l of u, $\{l \in \{1, ..., M\} | \mathbf{W}_{ul} = 1\}$ to measure how unique each post l is. Each post l has been vectorized using tf-idf approach and is represented in l-th row of the post-word matrix $\mathbf{X}$. Given the vector representation $\mathbf{X}_l$ of post l, the score s_l is calculated as,

$$s_l = \frac{\sum_{t=1}^{\mathcal{W}} \mathbf{X}_l(t)}{|\mathcal{W}|} \quad (8)$$

The higher this score is, the more unique and thus the more revealing post l would be. Based on this, we rank user u's posts and select the top-k posts as the most revealing information.

4.2 Step 2: Finding a Set of Candidates

The goal of this step is to find a set of candidates for each user u, given the top-k most revealing posts. To do so, for each nominated post l from step 1, we select set of words $\mathcal{S}$ whose tf-idf scores are greater than the average of the tf-idf scores for the words in the post l, $\mathcal{S}^l = \{t | \mathbf{X}_l(t) > s_l\}$. This approach helps to not to select useless words which have non-zero tf-idf values only due to data distortion during the anonymization process. Therefore, the words with higher chances of being posted in a real text are selected. This step results in a set of queries $Q_u = \{q_u^{(1)}, q_u^{(2)}, ..., q_u^{(k)}\}$. We construct the query $q_u^{(i)}$ from set $\mathcal{S}^i$, $i \in \{1, ..., k\}$ as $q_u^{(i)} = \{word \in \mathcal{S}^i\}$. Each of $q_u^{(i)} \in Q_u$ is queried through the $\mathcal{T}$'s search engine. Result includes a set of users who have published posts including keywords in $q_u^{(i)}$.

Integrating results from all queries in Q_u, we have a set of candidate users for user u which is denoted by $C = \{c_1, c_2, ..., c_{|C|}\}$. Combining steps 1 and 2, we first find posts which are the most revealing for user u and then for each selected post we select the words that are more likely to be used by the same user. The result will be a set of candidates for u.

4.3 Step 3: Matching-Up Candidates to Target

In the last step, we find the most similar candidate to user u. We shall define a metric which measures the similarity between each user u and ith candidate $c_i \in C$. Previous works [18, 19, 24, 25, 28] have solely leveraged the structural properties to find the similarity between a target user v and users in an anonymized dataset. However, given the properly anonymized network, the attacker is not be able to accurately find the similarity between users by just incorporating structural properties. We use other aspects of the data (even if they are anonymized) along with structural properties to reveal interesting information that could be leveraged for inferring the similarity. Location, textual and profile information are good examples of such social media data aspects. We consider textual information as the second aspect of the data. We stress that out proposed approach is not limited to textual and structural information and could be generalized to any data type. We also assume that the adversary is not aware of details of deployed anonymization techniques. Next, we define two sets of features to calculate the similarity between u and her i-th candidate c_i.

4.3.1 Structural Features. It has been also shown that users can be uniquely identified using their neighbors degree distributions [15]. Following previous works [30], we thus leverage degree distribitons of u's neighbors $\mathcal{N}(u)$ (i.e. all followers and followees of u) in order to represent her structural features. Note that properties such as betweenness, closeness, and eigenvector centrality cannot be considered as u's structural features since it requires having access to the complete network of users in $\mathcal{T}$ which is not feasible in practice. We quantify degree distributions by categorizing them into b bins with size of δ in a way that each bin contains the number of neighbors that have the degree in assigned range of that bin. For directed graphs, neighbors of each user can be divided into two groups of follower and followee and final feature set is computed by concatenating result of each group.

4.3.2 Textual Features. Remind that for user u, the attacker is given a set of m_u textual vectors as well as word space features $\mathcal{W}$. For each candidate c_i, we collect a set of θ recent posts by sending requests to the $\mathcal{T}$ API. The collected posts are then concatenated in one unified document. Next, c_i's PII will be removed from the document and the corresponding text vector is then created given the word space $\mathcal{W}$ following the similar approach of the Section 2.2.1. Textual features for users u and c_i are thus represented by a set $m_u = \{t_1, t_2, ...t_{m_u}\}$ and a textual vector t_{c_i}, respectively.

4.3.3 Calculating Users Similarity. Given two groups of structural and textual features, similarity between u and c_i is computed as the linear combination of their textual and structural similarities,

$$Sim(u, c_i) = \alpha Sim_{struct}(u, c_i) + (1 - \alpha) Sim_{text}(u, c_i) \quad (9)$$

where α controls the contribution of structural similarity. We further define $Sim_{struct}(u, c_i)$ as the cosine similarity between the two structural vectors computed as $Sim_{struct}(u, c_i) = \cos(s_u, s_{c_i})$. Textual similarity between u and c_i is also computed as the average of cosine similarity between $\forall t_j \in m_u$ and t_{c_i},

$$Sim_{text}(u, c_i) = \frac{\sum_{j=1}^{|m_u|} \cos(t_j, t_{c_i})}{m_u} \qquad (10)$$

4.3.4 Improving Similarity Measure. Merely checking the structural and textual similarity between the two users' features may lead to biased and not accurate results. Moreover, the attacker needs a more powerful similarity metric which could reduce the effect of anonymization. To handle this issue, we follow a fundamental well-defined problem in the field of image processing [8], *image denoising*. Non-local mean filters are a traditional way to remove noise from image data [8]. This approach replaces a pixel's value with the weighted average of all other pixels around it. The amount of weighting for neighboring pixels is based on the degree of similarity between a small patch centered on that pixel and a small patch centered on the pixel being denoised [8]. Inspired by the idea behind non-local mean filters [8], we use the feature values of other users similar to user u in order to reduce effect of anonymization. To apply this idea, we first need to find similar users to u– here is where the concept of *homophily* comes in handy. Homophily is one of the most important social correlation theories which is also observed in social media and explains the tendency of individuals to associate and create relationship with similar ones [9, 21].

Following the similar idea to non-local means filtering, we leverage homophily and consider user u's neighbor set $\mathcal{N}(u)$ as set of similar users to her. Utilizing homophily also helps in capturing the hidden relations between different aspects of the data. We thus calculate the similarity between $\mathcal{N}(u)$ and neighbors set $\mathcal{N}(c_i)$ for candidate c_i. We first quantify the degree distributions for *all* users in both neighbors set $\mathcal{N}(u)$ and $\mathcal{N}(c_i)$ as discussed earlier in Section 4.3.1. The structural similarity of neighbors are then calculated based on the cosine similarity between $s_{\mathcal{N}(u)}$ and $s_{\mathcal{N}(c_i)}$.

Following the procedure introduced in Section 4.3.2, we collect and concatenate θ recent posts for all neighbors in $\mathcal{N}(c_i)$. Textual similarity between $\mathcal{N}(u)$ and $\mathcal{N}(c_i)$ is then computed by taking average over the cosine similarities between textual vector of each user in $\mathcal{N}(u)$ and the textual vector of $t_{\mathcal{N}(c_i)}$. The total similarity between neighbors will be then calculated as follows,

$$Sim(\mathcal{N}(u), \mathcal{N}(c_i)) = \alpha Sim_{struct}(\mathcal{N}(u), \mathcal{N}(c_i))$$
$$+ (1 - \alpha) Sim_{text}(\mathcal{N}(u), \mathcal{N}(c_i)) \qquad (11)$$

This metric quantifies the fitness of $\mathcal{N}(u)$ and $\mathcal{N}(c_i)$ as the similarity scores of their structural and textual properties. It reduces the effect of data anonymization and also aligns well with the assumption that if u and c_i correspond to the same identity, their neighbors $\mathcal{N}(u)$ and $\mathcal{N}(c_i)$ should also match [11]. Finally, the total similarity between u and c_i can be computed as the combination of their individual similarity and the fitness of their neighbors:

$$Sim_{total}(u, c_i) = \beta Sim(u, c_i) + (1 - \beta) Sim(\mathcal{N}(u), \mathcal{N}(c_i)) \qquad (12)$$

We empirically find that random selection of c_i's neighbors with the size λ works well in our problem and we are not required to collect all neighbors information from $\mathcal{T}$'s API. This will make the

---

Algorithm 1 Adversarial Technique for Heterogeneous Data

Input: user u, Anonymized Data $\mathcal{D} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}, \mathbf{X}, \mathcal{W}\}, k, \lambda, \theta, h,$
  α, β
Output: Top-h mapped accounts in targeted social network $\mathcal{T}$
 1: Initialize the candidate set $C = \phi$.
 2: **for** each anonymized text vector of post l for u **do**
 3:     Calculate score s_l according to Eq.8.
 4: **end for**
 5: Select top-k posts with the highest score s_l as the most revealing information.
 6: **for** For each text vector l in top-d posts **do**
 7:     Select words with tf-idf scores $\mathbf{X}_l(t) > s_l$ to create search query $q_u^{(l)}$.
 8:     Search query $q_u^{(l)}$ in $\mathcal{T}$ search engine and add results to C.
 9: **end for**
10: **for** each candidate c_i in C **do**
11:     Calculate similarity between u and c_i according to Eq.12.
12: **end for**
13: Return the top-h candidates with maximum similarity

---

de-anonymization approach more efficient. Note that many noise removal approaches have been designed for specific kinds of noise (e.g., Guassian noise) which could be used to remove the noise from the data and particularly the vector of textual information. However, using certain noise removal approaches may not always have positive effects. In fact, it can lead to a wrong estimation of users' properties when the attacker does not have any prior knowledge of the deployed anonymized technique.

The proposed ATHD approach is shown in Algorithm 1. The input to the algorithm is the anonymized dataset and the output is the top-h mapped profile accounts in T. Lines 2–5, correspond to the first step of ATHD. The set of candidate set (step 2) is then found through lines 6–9. The similarity between u and each of the selected candidates is calculated in lines 10–12. Finally, top-h candidates with the maximum similarity to u will be returned. This re-identification procedure is then run over all users in the anonymized dataset. Note ATHD is independent of deployed anonymization techniques either for the textual or structural information. In the next section we will discuss how our proposed de-anonymization could be generalized to the social media data with any type of components.

4.4 Generalizability of ATHD

Our framework can be generalized through abstraction to different social media data, assuming that our anonymized data consists of two different aspects, $\mathcal{A}_1$ and $\mathcal{A}_2$ and the attacker is willing to initiate an attack by mapping u to a real profile in the targeted social media $\mathcal{T}$. As discussed before, the first step is to extract the most revealing information from $\mathcal{A}_1$ for user u by using the same concept as tf-idf scores. The second step includes selecting a set of candidate profiles for u by searching for the extracted information from the previous step through $\mathcal{T}$'s search engine. Finally, the similarity between u and her candidates are calculated using the combination of features of existing data components, $\mathcal{A}_1$ and $\mathcal{A}_2$. Features of the most similar users to u (e.g., neighbors) are also incorporated as

well to reduce the anonymization effect while capturing the hidden relation between different aspects of the data.

5 EXPERIMENTS

In this section, we seek to answer the introduced research questions, but we first need to evaluate the efficiency of proposed adversarial technique ATHD. We begin this section by introducing the dataset and anonymization techniques we used. Then, we compare the results of ATHD against the state-of-the-art de-anonymization benchmarks to evaluate its effectiveness. Next, we use ATHD to assess the anonymization power of each of the four cases to answer the research questions:

- **(RQ1)**: Is the data private if just one of its two aspects is anonymized?
- **(RQ2)**: Is case 4, the strongest among four cases, sufficient for anonymizing social media data?

5.1 Datasets

We use two different datasets from two large social media websites, Twitter and Foursquare. Twitter is a prevalent and well-known microblogging social media allows millions of active users interacting with each other via short posts, called tweet. Foursquare is a location based social media in which users share their location with friends. Users can also leave tips about different places. We collect the Twitter dataset using Twitter API using the snowball sampling technique as follows. We begin with a random initial seed of users and for each user u in the seed, we obtain a random subset of size 100 of her posted tweets as well as a subset of size 500 of her follower/followee information. We repeat the same process for each u's followers/followees. This way we build our final dataset which consists of the users in the initial seed and their 2-hops connections. We follow the same procedure to collect the data from Foursquare API by considering a random initial seed of users. We collect each user friends as well as her tips on different locations. We build the final dataset by repeating this process for 2-hops connections. Note that in both datasets, we only keep the information of users who have posted at least one tweet or tip.

Next, we will apply various anonymization techniques on the obtained dataset– this is described in the next section. Also, we utilize the Twitter's advanced search engine[1] and Foursquare search [2] during the de-anonymization process for Twitter and Foursquare data, respectively. It would be also worthwhile to add that we already have the ground truth for the re-identification, since the real profiles of the crawled users are known to us beforehand. Table 2 summarizes the statistics of our datasets.

5.2 Anonymization Approaches

We use different anonymization techniques to evaluate the introduced different anonymization cases in Table1. Following previous work [11], we choose different algorithms for *structural information* anonymization as follows:

- **Naive Anonymization**. This approach only masks users' identifiers (PII), and does not change the graph structure. This is the simplest approach and thus we would expect the highest vulnerability and hence best de-anonymization result.

Table 2: Statistics of the crawled datasets.

(a) Twitter

# of Users	# of Edges	Avg. Clustering Coefficient
6,789	244,480	0.219
Density	# of Tweets	# of Unigrams
0.005	478,129	208,483

(b) Foursquare

# of Users	# of Edges	Avg. Clustering Coefficient
22,332	229,234	0.295
Density	# of Tips	# of Unigrams
0.0005	124,744	103,264

- **Sparsification**. This work randomly eliminates $p|\mathcal{E}|$ edges where p is the anonymiztion coefficient.
- **k-deg(add)** [20]. This anonymization method ensures that k-degree anonymity is preserved by only adding edges.
- **k-degree(add & del)** [20]. This method ensures that k-degree anonymity is preserved by performing simultaneous add/removal of the edges.
- **Switching**. This method selects two random edges (i_1, j_1) and (i_2, j_2) from the original graph such that $\{(i_1, j_2) \notin \mathcal{E} \wedge (i_2, j_1) \notin \mathcal{E}\}$. Then, it switches pairs of edges, i.e. remove edges (i_1, j_1) and (i_2, j_2) and add new edges (i_1, j_2) and (i_2, j_1) instead. This step is repeated $\frac{p|\mathcal{E}|}{2}$ times which results in $p|\mathcal{E}|$ edge removals/additions.
- **Perturbation**. This method is also known as *unintended* anonymziation and has two main steps. It first removes $p|\mathcal{E}|$ edges in a same way as sparsification method does. Then, it adds random false edges until the number of edges in the anonymized graph is the same as the original one.

Furthermore, the *Textual information* is anonymized using the techniques discussed earlier in Section 2.2 as follows:

- **Naive Anonymization**. This approach first removes users' identifiers and links from the tweets and then vectorize it.
- **Diff Privacy**. This method takes the output of the naive anonymization technique and then ensures differential privacy by adding Laplacian noise to the generated text vector.

5.3 Results and Discussion

5.3.1 Experimental Settings. We evaluate de-anonymization approaches by a metric called *success rate* $X = \frac{n_c}{N}$, where n_c is the total number of users that have been successfully re-identified and N is the total number of users in the anonymized dataset![24]. Larger values of this measure correspond to higher privacy breach.

Following the previous works [11, 28], we set $k = 10$ for k-degree anonymity and $p = 0.1$ for sparsification, purturbation and switching methods. The ϵ for differential privacy technique is set as $\epsilon = 0.01$. We also set the parameters of ATHD as follows: $\{k = 10, \alpha = 0.5, \beta = 0.7, \lambda = 20, \theta = 50, b = 7, \delta = 50\}$. The values of δ and b for quantifying degree distributions are chosen such that it can accommodate higher degrees variation. Empirical results showed that the choice of δ and b does not have a huge impact on the final results. We also set the number of returned profiles as $h = 1$. Clearly, increasing the value of h will increase the de-anonymization success rate. To answer the research questions,

(a) Twitter

	ATHD-Improved		ATHD-Simple		ADA		Narayanan et. al.	
	Naive	**Diff Privacy**	**Naive**	**Diff Privacy**	**Naive**	**Diff Privacy**	**Naive**	**Diff Privacy**
Naive	0.9435 (1)	0.8020 (2)	0.8200 (1)	0.6951 (2)	0.6729 (1)	0.5513 (2)	0.5073 (1)	0.4100 (2)
Sparsification	0.8087 (3)	0.6998 (4)	0.7327 (3)	0.6213 (4)	0.6099(3)	0.5114 (4)	0.4316 (3)	0.3437 (4)
k-deg(add)	0.7894 (3)	0.6814 (4)	0.6900 (3)	0.6125 (4)	0.5898 (3)	0.4982 (4)	0.3979 (3)	0.3139 (4)
k-deg(add & del)	0.7580 (3)	0.6533 (4)	0.6891 (3)	0.5821 (4)	0.5800 (3)	0.4727 (4)	0.3815 (3)	0.2997 (4)
Switching	0.6911 (3)	0.5812 (4)	0.6013 (3)	0.5186 (4)	0.4971 (3)	0.4014 (4)	0.3520 (3)	0.2618 (4)
Perturbation	0.6500 (3)	0.5685 (4)	0.5367 (3)	0.4249 (4)	0.4322 (3)	0.3618 (4)	0.2987 (3)	0.2018 (4)

(b) Foursquare

	ATHD-Improved		ATHD-Simple		ADA		Narayanan et. al.	
	Naive	**Diff Privacy**	**Naive**	**Diff Privacy**	**Naive**	**Diff Privacy**	**Naive**	**Diff Privacy**
Naive	0.8004 (1)	0.6799 (2)	0.7107 (1)	0.5989 (2)	0.5699 (1)	0.4821 (2)	0.4400 (1)	0.3754 (2)
Sparsification	0.7238 (3)	0.6299 (4)	0.6400 (3)	0.5499 (4)	0.5118(3)	0.4532 (4)	0.3968 (3)	0.3028 (4)
k-deg(add)	0.6947 (3)	0.5999 (4)	0.6112 (3)	0.5288 (4)	0.5138 (3)	0.4157 (4)	0.3487 (3)	0.2748 (4)
k-deg(add & del)	0.6612 (3)	0.5739 (4)	0.5918 (3)	0.4989 (4)	0.4867 (3)	0.3947 (4)	0.3025 (3)	0.2639 (4)
Switching	0.6134 (3)	0.5431 (4)	0.5517 (3)	0.4614 (4)	0.4300 (3)	0.3521 (4)	0.2987 (3)	0.2120 (4)
Perturbation	0.5642 (3)	0.4930 (4)	0.4518 (3)	0.3670 (4)	0.3402 (3)	0.2836 (4)	0.2300 (3)	0.1876 (4)

Table 3: Comparison of the de-anonymization success rates for various anonymization techniques. Higher values imply higher privacy breach. Numbers in parentheses demonstrate the corresponding case number in Table 1.

we make 12 copies of the original data and sanitize each copy with a different combination of structural and textual anonymization techniques discussed earlier. For evaluation, we define two different variants of our proposed approach, ATHD, as follows:

- **ATHD-Simple:** This uses Eq.9 and Eq.10 to calculate similarity.
- **ATHD-Improved:** This variant uses Eq.12 to improve similarity measure by incorporating features from neighbors to reduce the anonymization effect.

5.3.2 Perfomrnace Comparison. To evaluate the effectiveness of ATHD, we benchmark its two variants, ATHD-Simple and ATHD-Improved, against the following two baselines.

- **Narayanan et. al.** [24]: It computes the similarity between an unmapped user u and a candidate c_i, by using the number of neighbors of u that have been mapped to neighbors of c_i.
- **ADA** [18]: This method considers a combination of structural, relative distance and inheritance similarity. We only use degree centrality for measuring structural similarity as we do not have access to the global structure of c_i in $\mathcal{T}$.

In general, these baselines are seed-based approaches, meaning that they map a known target user v in $\mathcal{T}$ to a user in the anonymized data by utilizing a small set of initially mapped seed users and then propagating the mappings through the whole data. These works also need a previously collected background knowledge $\mathcal{B}$. We need to use same settings to make a fair comparison between the baselines and our proposed framework. To do so, we first make an initial seed set of the size $v = 20$, by mapping a set of random users in the anonymized dataset to their real identities for each of Twitter and Foursquare data. Then, we repeat the same 3-step procedure as in the ATHD for the baselines, except that we the similarity metric in the last step is replaced with those of the baselines. Performance comparison results for both datasets are demonstrated in Table 3 with the following observations:

- Narayanan et. al. is the least effective de-anonymization on both datasets. The reason is because its utilized similarity metric relies on the set of previously mapped neighbors and ignores the available structural and textual information provided in the data.
- ADA approach is more powerful than Narayanan et. al. since it incorporates structural properties of the data.
- Anonymized data is more vulnerable to ATHD-Simple compared to ADA and Narayanan et. al. This is because both structural and textual information are incorporated in the similarity metric used in ATHD-Simple. This confirms that integrating different components of data plays an important role in de-anonymization for heterogeneous social media data.
- ATHD-Improved technique achieves the best results for both Twitter and Foursquare datasets. This demonstrates the effectiveness of utilizing homophily and the features of neighbors for more effective de-anonymization.

To recap, the above observations confirm the efficiency of our proposed approach ATHD.

5.3.3 Assessing Effectiveness of Anonymization. Having discussed the efficiency of the proposed ATHD de-anonymization approach, we now seek the answer to the last two questions. The performance results w.r.t. the four anonymization cases are demonstrated in Table 3. The numbers in parentheses demonstrate the corresponding case number defined earlier in the introduction. We make the following observations for both datasets:

- Publishing the data with no anonymization for either aspect (i.e., case 1) resulted in a large information breach in both ATHD-Simple and ATHD-Improved approaches which suggests the least amount of protection as expected.
- In general, anonymizing either aspect of the data (i.e., cases 2 and 3) protects users privacy more than case 1.

- Case 4 is the strongest protection among the four cases. Accordingly, the answer to the second question is no.
- Although case 4 provides the strongest protection, ATHD-Improved was able to re-identify at least 56% of the users in the anonymized dataset, which is a significant number in the field of privacy. This shows that case 4 is far from sufficient for data anonymization.
- Sparsification is the most vulnerable anonymization approach against both ATHD-Simple and ATHD-Improved techniques as it makes the least amount of changes to the link information.
- Although the switching and perturbation methods both add and deletes the same number of edges, switching is more vulnerable to the de-anonymization since it preserves the node degrees.
- Despite the fact that k-degree anonymity based approaches guarantee the user re-identification probability to be at most $\frac{1}{k}$, but they fail because of using extra textual information.

According to these observations, the answers to the introduced research questions are no. These results further indicate that despite anonymization of all aspects of data is essential, but it is not sufficient to anonymize each aspect independently from others. This is because an adversary could easily breach privacy no matter what anonymization algorithm has been used. Consequently, serious privacy breach could happen when the published data is heterogeneous. This necessitates taking into account the latent relations in different portions of the social media data for anonymization.

6 RELATED WORK

Social Network Anonymization. Social networks contain private profile information and sensitive social relationships which provide opportunities for researchers to study and understand individuals at unprecedented scales [3, 5, 12, 14]. However, this information may leak users' privacy [2]. Anonymization methods serve as an important role to maintain data utility as well as protecting privacy [34]. Existing social network anonymization methods can be categorized mainly into three categories: *k-anonymity*, *edge randomization*, *clustering-based generalization* and *differential privacy*. The aim of k-anonymity methods is to anonymize each node so that it is indistinguishable from at least $k-1$ other nodes [32]. Liu *et al.* proposed to achieve k-degree anonymization [20] through edge addition/deletion strategies [20]. Zhou *et al.* further considered the assumption that the adversary knows subgraph constructed by the immediate neighbors of a target node, and aims to achieve k-neighborhood anonymity [37]. Edge randomization algorithms for social networks usually utilize edge-based randomization strategies to anonymize data, such as random adding/deleting and random switching [36]. Clustering-based anonymization methods group nodes and edges, and only reveal the density and size so that individual attributes are protected [33]. Another work seeks to generate an anonymized graph which guarantees differential privacy [29]. **Social Network De-anonymization.** De-anonymization approaches on social networks aim to re-identify the anonymous user data by using previously collected background information. Existing de-anonymization methods can be categorized into i) *seed-based* and ii) *seed-free*, according to whether pre-annotated seed users exist or not. Seed-based de-anonymization attack on social network was proposed to use only structural information and propagates node mappings based on seed user pairs [24]. Later, Narayanan *et al.* [22] employed a simplified attack using less heuristics rules for link

prediction problem. Nilizadeh *et al.* further proposed a community-enhanced de-anonymization scheme, which first de-anonymizes data in community-level and then de-anonymizes the users within the communities [25]. Yartseva *et al.* proposed a percolation-based de-anonymization method using neighborhood overlap information [35]. Seed-free approaches assume there is no seed users available. Pedarsani *et al.* presented a Bayesian model to iteratively perform a maximum weighted bipartite graph matching starting from the nodes with the highest degree [26]. Moreover, Ji *et al.* proposed to use optimization based methods to minimize the edge difference between anonymized network and background information [17]. Recently, another group of works have focused on exploiting additional sources of information such as profile information [11] and users attributes [28] for social graph de-anonymization. Fu *et al.* proposed to use structural and descriptive information to de-anonymize users without seed nodes [11]. A thorough survey on graph data anonymization and de-anonymization is presented in [19]. Note that de-anonymization methods are similar to those of user identity linkage across social network when only network information is available [31]. In addition to the different goals of these two research direction, the main difference is that the given graph structured is not anonymized in case of user identity linkage problem. This makes the de-anonymization much more challenging.There are two main differences between our work and the above works. First, we introduce a new adversarial attack specialized for social media data and second, we assess the efficiency of existing anonymization techniques for heterogeneous social media data.

7 CONCLUSION

In this work, we study a new problem of user data privacy for social media via an adversarial approach. Our work differs from the existing works due to unique properties of social media data: a social media site has an inordinate number of users and the site only allows for a limited number of data queries. Since anonymization takes time and requires dedicated efforts, anonymization efficiency should be maximized. Thus, we evaluate the strengths of anonymization techniques in the context of social media data and verify if it is sufficient. We propose ATHD, a novel adversarial technique by exploiting heterogeneous characteristics of social media data. Our results illustrate that anonymizing even all aspects of data is not sufficient for protecting user privacy due to hidden relations between different aspects of the heterogeneous data. One future research direction for our work is to examine how different combinations of heterogeneous data (e.g., a combination of location and textual information) are vulnerable to the de-anonymization attack, though the work reported in this paper is sufficient to show the need for better anonymization with resource constraints. Another potential direction is to improve anonymization techniques to preserve the privacy of users in social media data by considering hidden relations between different components of the data due to the innate heterogeneity of user-generated data.

REFERENCES

[1] Jemal H Abawajy, Mohd Izuan Hafez Ninggal, and Tutut Herawan. [n. d.]. Privacy preserving social network data publication. *IEEE communications surveys & tutorials* 18, 3 ([n. d.]), 1974–1997.

[2] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. 2007. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of international conference on World Wide Web*. ACM.

[3] Ghazaleh Beigi, Mahdi Jalili, Hamidreza Alvari, and Gita Sukthankar. 2014. Leveraging community detection for accurate trust prediction. In *ASE International Conference on Social Computing, Palo Alto, CA, May 2014*.

[4] Ghazaleh Beigi and Huan Liu. 2018. Similar but Different: Exploiting Users' Congruity for Recommendation Systems. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer.

[5] Ghazaleh Beigi, Jiliang Tang, and Huan Liu. 2016. Signed link analysis in social media networks. In *10th International Conference on Web and Social Media, ICWSM 2016*. AAAI Press.

[6] Ghazaleh Beigi, Jiliang Tang, Suhang Wang, and Huan Liu. 2016. Exploiting emotional information for trust/distrust prediction. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 81–89.

[7] Valentina Beretta, Daniele Maccagnola, Timothy Cribbin, and Enza Messina. 2015. An interactive method for inferring demographic attributes in Twitter. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*. ACM.

[8] Antoni Buades, Bartomeu Coll, and J-M Morel. 2005. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 2. IEEE, 60–65.

[9] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. 2008. Feedback effects between similarity and social influence in online communities. In *Proceedings of ACM SIGKDD*. 160–168.

[10] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*. Springer, 1–19.

[11] Hao Fu, Aston Zhang, and Xing Xie. 2015. Effective social graph deanonymization based on graph structure and descriptive information. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 4 (2015), 49.

[12] Alireza Hajibagheri. 2017. Learning Dynamic Network Models for Complex Social Systems. (2017).

[13] Alireza Hajibagheri, Gita Sukthankar, and Kiran Lakkaraju. 2017. Extracting Information from Negative Interactions in Multiplex Networks Using Mutual Information. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 322–328.

[14] Alireza Hajibagheri, Gita Sukthankar, Kiran Lakkaraju, Hamidreza Alvari, Rolf T Wigand, and Nitin Agarwal. 2018. Using Massively Multiplayer Online Game Data to Analyze the Dynamics of Social Interactions. In *Social Interactions in Virtual Worlds An Interdisciplinary Perspective*. Cambridge University Press.

[15] Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. 2007. Anonymizing social networks. *Computer science department faculty publication series* (2007), 180.

[16] Shouling Ji, Weiqing Li, and Prateek Mittal. 2015. SecGraph: A Uniform and Open-source Evaluation System for Graph Data Anonymization and De-anonymization.. In *24th USENIX Security Symposium (USENIX Security 15)*.

[17] Shouling Ji, Weiqing Li, Mudhakar Srivatsa, and Raheem Beyah. [n. d.]. Structural data de-anonymization: Quantification, practice, and implications. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*.

[18] Shouling Ji, Weiqing Li, Mudhakar Srivatsa, Jing Selena He, and Raheem Beyah. 2016. General graph data de-anonymization: From mobility traces to social networks. *ACM Transactions on Information and System Security (TISSEC)* (2016).

[19] Shouling Ji, Prateek Mittal, and Raheem Beyah. 2016. Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey. *IEEE Communications Surveys & Tutorials* (2016).

[20] Kun Liu and Evimaria Terzi. 2008. Towards identity anonymization on graphs. In *Proceedings of international conference on Management of data*. ACM.

[21] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[22] Arvind Narayanan, Elaine Shi, and Benjamin IP Rubinstein. 2011. Link prediction by de-anonymization: How we won the kaggle social network challenge. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, 1825–1834.

[23] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*. IEEE.

[24] Arvind Narayanan and Vitaly Shmatikov. 2009. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE, 173–187.

[25] Shirin Nilizadeh, Apu Kapadia, and Yong-Yeol Ahn. 2014. Community-enhanced de-anonymization of online social networks. In *Proceedings of the 2014 acm sigsac conference on computer and communications security*. ACM, 537–548.

[26] Pedram Pedarsani, Daniel R Figueiredo, and Matthias Grossglauser. 2013. A bayesian method for matching two similar graphs without seeds. In *Communication, Control, and Computing (Allerton)*. IEEE.

[27] Krishna PN Puttaswamy, Shiyuan Wang, Troy Steinbauer, Divyakant Agrawal, Amr El Abbadi, Christopher Kruegel, and Ben Y Zhao. 2014. Preserving location privacy in geosocial applications. *IEEE Transactions on Mobile Computing*.

[28] Jianwei Qian, Xiang-Yang Li, Chunhong Zhang, and Linlin Chen. 2016. De-anonymizing social networks and inferring private attributes using knowledge graphs. In *INFOCOM International Conference on Computer Communications*.

[29] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y Zhao. 2011. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*.

[30] Kumar Sharad. 2016. True friends let you down: Benchmarking social graph anonymization schemes. In *Proceedings of Workshop on Artificial Intelligence and Security*. ACM.

[31] Kai Shu, Suhang Wang, Jiliang Tang, Reza Zafarani, and Huan Liu. 2017. User Identity Linkage across Online Social Networks: A Review. *ACM SIGKDD Explorations Newsletter* 18, 2 (2017), 5–17.

[32] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* (2002).

[33] Tamir Tassa and Dror J Cohen. 2013. Anonymization of centralized and distributed social networks by sequential clustering. *IEEE Transactions on Knowledge and Data Engineering* 25, 2 (2013), 311–324.

[34] Xintao Wu, Xiaowei Ying, Kun Liu, and Lei Chen. 2010. A survey of privacy-preservation of graphs and social networks. *Managing and mining graph data* (2010), 421–453.

[35] Lyudmila Yartseva and Matthias Grossglauser. 2013. On the performance of percolation graph matching. In *Proceedings of the first ACM conference on Online social networks*. ACM, 119–130.

[36] Xiaowei Ying and Xintao Wu. 2009. Graph generation with prescribed feature constraints. In *Proceedings of SDM*. SIAM.

[37] Bin Zhou and Jian Pei. 2008. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of International Conference on Data Engineering*.